Service Modeling Language, Version 1.1

2. Notations and Terminology

2.2 Terminology

Target-complete Identifier

> A **target-complete Identifier** is a URI or IRI that contains all the information required to locate a target of an SML reference. It is a consequence of this definition that a target-complete identifier cannot be a relative reference.

4. SML References

4.2 Reference Semantics

4.2.3 Identical Targets

To determine if two targets are the same or different, a model validator MUST obey the following rules.

1. If both of the following are true, then a model validator MUST consider both targets to be the same.
   1. The definition of the scheme(s) used to locate the targets specifies that the scheme uses target-complete identifiers. [**4.3 Reference Schemes**]
   2. The two target-complete identifiers used to locate the targets are identical using a case-sensitive, codepoint-by-codepoint comparison.
2. Otherwise a model validator MUST consider both targets to be different when there is something available in the element information items for the targets that tells them apart. For example, if there is an infoset property for which the 2 targets have different values, they are different. This applies recursively for complex-valued properties.
3. For all other cases, it is implementation-defined whether to treat the targets as the same or not.

4.3 Reference Schemes

An SML reference MAY be represented by using a variety of schemes, including those defined in this specification. SML does not mandate the use of any specific schemes. A reference scheme MAY use child elements, attributes, both, or neither to capture the information necessary to identify the reference target. It is not required that all elements in an SML model be reachable via an SML reference. This will depend on the support defined by the chosen reference scheme.

Although SML does not require the use of any specific scheme, it does specify how a reference MUST be represented when using SML-defined reference schemes. This specification defines the **4.3.1 SML URI Scheme** for identifying reference targets.

Comment: Changing "URI" to "identifier", because this term is used to cover both URI and IRI.

Deleted: URI

Deleted: URI

Comment: Target-complete identifiers are the ones actually used to locate targets, also the ones used to compare for target identity. Relative references (typically) can't be used for either purpose.

Comment: This change is to specify the order in which conditions in bullet 1 and 2 are checked, to avoid the question "what if both 1 and 2 are true".

Deleted: A

Deleted: when both of the following are true

Deleted: target-complete URIs

Deleted: URIs or IRIs

Deleted: A

All of the following MUST be defined for each SML reference scheme,

1. The set of rules that, when satisfied, identify a reference element as an instance of the scheme.
2. The set of rules that, when evaluated, resolve the reference to a set of target element nodes.
3. An assertion that states whether or not the scheme uses target-complete identifiers. Using target-complete identifiers requires all the following:
   1. URI-references or IRI-references are specified in instances of the scheme. These references may or may not be target-complete identifiers.
   2. If these references are allowed to be relative references, i.e. they are not already target-complete, then some implementation-dependent base URI/IRI is used to resolve them to URIs or IRIs. [See section 5 of RFC 3986 and section 6.5 of RFC 3987.]
   3. The resulting references are target-complete identifiers.

A scheme definition MAY impose additional requirements on references recognized as instances of that scheme. Such requirements, if any, MUST NOT be applied by scheme implementations to references that are not instances of the corresponding scheme.

4.3.1 SML URI Scheme

The SML URI Scheme is defined as follows:

1. An SML reference element is identified as using the SML URI scheme if and only if exactly one element information item whose [local name] is uri and whose [namespace name] is http://www.w3.org/2008/01/sml is present as a child of that reference element.
2. The SML URI scheme is resolved using the following steps:
   1. If the URI is a relative reference, then use an implementation dependent base URI to resolve it to an URI.
   2. A document is obtained by dereferencing the URI reference, ignoring its fragment component, using the appropriate operation defined for the scheme used in that URI reference. If there is no document retrieved, the scheme instance is unresolved.
   3. If no fragment component is present in the URI reference, the scheme instance resolves to the root element of the retrieved document.
   4. If a fragment component is present in the URI reference, the scheme instance resolves to the set of elements obtained by applying the fragment component to the root element of the retrieved document.
3. The SML URI Scheme uses target-complete identifiers.
4. Scheme syntax:

```
SMLURI ::= URI ('#' SMLXPath1_Fragment_ID)?
```

where ,

1. URI is a URI reference without a fragment component.

2. `SMLXPath1_Fragment_ID` is a production that defines the syntax of the fragment portion of the SML URI scheme. This is defined in section **4.3.1.1 The smlxpath1() scheme**.

Service Modeling Language Interchange Format Version 1.1

5.3.3 URI Reference Processing

When processing an SML-IF document, there are 3 categories of URI references that may need to be resolved:

1. `schemaLocation` attributes on `xs:include` and `xs:redefine` in schema documents, when they are model definition documents.
2. URI references specified in instances of SML reference schemes that use target-complete URIs [*SML 1.1*].
3. URI references specified in instances of SML reference schemes that do not use target-complete URIs.