GWD-R-P
NSI-WG
nsi-wg@ogf.org

Jeroen van der Ham, UvA

January 2013

Network Service Interface Topology Representation

Status of This Document

Group Working Draft (GWD), candidate Recommendations Proposed (R-P).

Copyright Notice

Abstract

This document describes a normative extension to the Network Markup Language base schema version 1 which allows the description of service plane objects required for the Network Service Interface Connection Service.

Contents

1 Introduction

The NSI Connection Service requires topology descriptions to do pathfinding. In order to do that some representation of the topology is required. Once represented, some form of topology distribution is also needed. This document describes an extension of the Network Markup Language[**?**] to support the NSI Connection Service[**?**] and NSI Topology Service[**?**].

Section 2 describes the NSI topology representation extension of the Network Markup Language base schema version 1. Only section 2 and appendices **??** and **??** are normative and considered part of the recommendation.

1.1 Scope

The NSI topology representation is an extension of the Network Markup Language version 1. The NSI topology covers concepts relevant for supporting the Connection Service, which are outside of the scope of NML.

The scope of this topology representation extension is limited to what the Connection Service requires.

1.2 Context

The NSI topology representation is defined based on the concepts defined in the NSI Connection Service version 2[**?**], the NSI Topology Service[**?**] and the Network Markup Language base schema version 1[**?**].

1.3 Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC 2119].

Names of classes are capitalised and written in italics (e.g. the *Node* class). Names of relations are written in camel case and in italics (e.g. the *hasNode* relation).

2 NSI Topology Schema

The NSI topology schema describes the components required for the NSI Connection Service which are not part of the NML Base schema.

2.1 Service Termination Point Identifiers

A source or destination of a connection request in the NSI Connection Service is identified by the Topology identifier and two unidirectional *Ports* or *PortGroups*. Each of these must be globally unique identifiers.

A recommended way of constructing such an identifier is by using the `urn:ogf:network` namespace, for example `urn:ogf:network:example.net:2013:A1`.

This identifier has three components: the prefix, `urn:ogf:network` which describes that it is a network identifier, the authoring namespace, `example.net:2013` which is the DNS name and a (at least) year to make a globally unique prefix[1], and the local component, `A1` defined by the originating network.

2.2 Service Termination Point Groups

Endpoints in a network often have a technology label associated with them, for example VLANs or wavelengths. Rather than describing each of these available labels as individual STPs, we introduce the *STP Group*, equivalent to an NML *PortGroup*.

An STP with a specific label can then be selected using the query component syntax as specified in [**?**], so for example:

`urn:ogf:network:example.net:2013:A2?vlan=1781` is a way to phrase a request to an STP with VLAN 1781 part of the STP Group identified by `urn:ogf:network:example.net:2013:A2`.

If no specific label or attribute is given to select an STP from an STP group, the Network Service Agent for that network will select one from that STP group. The confirmation back to the requester will contain the fully specified STP selected for the request. An example for this kind of request is by specifying an STP which has VLAN labels, but not requesting a specific VLAN label. Continuing the example above, the STP `urn:ogf:network:example.net:2013:A2` has been specified to have a specific VLAN range available. A request with just that identifier as the destination will allow the pathfinder to select a VLAN on that specific endpoint, and return it to the user, using the query component.

[1] The date component in the identifier is optional but recommended. The DNS name is a temporary lease, which can change hands, so in order to guarantee uniqueness, the year component can be added.

NSI Concept	*Representation*
STP	2x `nml:Port` / `nml:BidirectionalPort`
Connected To	`nml:isAlias`
NSNetwork	`nml:Topology`
Has STP	`nml:hasPort`
Located at	`nml:locatedAt`
Location	`nml:Location`
GPS coords	`nml:lat`, `nml:long`
NSA	`nsi:NSA`
Network managed by NSA	`nsi:managedBy`
Admin Contact	`nsi:adminContact`
Provider endpoint URL	`nsi:csProviderEndpoint`
Control-plane connections	`nsi:peersWith`

Table 1: Relation of NSI and NML terminology

3 NSI Topology Representation

The NSI topology representation is defined as an extension to the NML topology representation. It builds as much as possible to build on standardized work in the NML group. An overview of the NSI concepts and the related representations are shown in table 1. The definitions of the new concepts are defined below.

3.1 Classes

3.1.1 NSA

An NSA represents a Network Service Agent which can accept Connection Service requests and manages a network.

NSA inherits from the NML *Network Object*

An *NSA* may have the following relations:

- *adminContact* to a vCard object describing contact details for its administrator

An *NSA* may have the following attributes:

- *csProviderEndPoint* to an NSI CS webservice URL endpoint
- *id* to assign a persistent globally unique URI

- *name* to assign a human readable string

3.2 Relations

3.2.1 adminContact

adminContact is used to provide contact information about a *Network Object*. It relates a *Network Object* to a *vCard*.

Allowed relations are:

- *Network Object* —(adminContact ▸, * : *)— *vCard*

Defined relations are:

- *NSA* —(adminContact ▸, * : *)— *vCard*

3.2.2 managedBy

managedBy relates an *NSA* to a *Topology* to define that the *NSA* can take requests for that *Topology*.

Allowed relations are:

- *Network Object* —(managedBy ▸, * : *)— *NSA*

Defined relations are:

- *Topology* —(managedBy ▸, 1 : 1)— *NSA*

3.2.3 peersWith

The *peersWith* relation defines the control-plane connections between different *NSA*s.

Allowed relations are:

- *NSA* —(peersWith ▸, * : *)— *NSA*

Defined relations are:

- *NSA* —(peersWith ▸, * : *)— *NSA*

3.3 Attributes

3.3.1 csProviderEndpoint

csProviderEndpoint defines the URL at which the webservice for the NSI Connection Service can be reached.

4 NSI Topology Description Example

A simple example NSI Network topology description is provided below.

```
@prefix nml: <http://schemas.ogf.org/nml/2013/10/base#> .
@prefix nmleth: <http://schemas.ogf.org/nml/ethernet/2013/10#> .
@prefix nsi: <http://schemas.ogf.org/nsi/topology/2013/10#> .
@prefix exa: <urn:ogf:network:example.com:2013:> .
@prefix exb: <urn:ogf:network:example.org:2013:> .

exa:ExampleCom a nml:Topology ;
        nml:version "2013012301" ;
        nml:name "exa" ;
        nml:locatedAt exa:location ;
        nml:hasOutboundPort exa:eth0-out ;
        nml:hasInboundPort exa:eth0-in ;
        nml:hasOutboundPort exa:eth1-out ;
        nml:hasInboundPort exa:eth1-in ;
        nsi:managedBy exa:nsa .
exa:location a nml:Location ;
        nml:lat "55.637"^^<http://www.w3.org/2001/XMLSchema#float> ;
        nml:long "12.641"^^<http://www.w3.org/2001/XMLSchema#float> .
exa:nsa a nsi:NSA ;
        nsi:csProviderEndpoint "http://nsa.example.com/" ;
        nsi:peersWith sara:nsa .
exa:eth0-out a nml:PortGroup ;
        nmleth:vlans "1780-1783" ;
        nml:isAlias exb:if0-in .
exa:eth0-in a nml:PortGroup ;
        nmleth:vlans "1780-1783" ;
        nml:isAlias exb:if0-out .
exa:eth0-out a nml:PortGroup ;
        nmleth:vlans "1780-1783" .
exa:eth0-in a nml:PortGroup ;
        nmleth:vlans "1780-1783" .

exb:nsa a nsi:NSA ;
      nsi:csProviderEndpoint "http://nsa.example.org"
```

The above example provides a minimal description to expose, a *Topology*, a *Location*, an *NSA*, two *PortGroups* for the connection with Example.org, and finally two *PortGroups* describing the Storage endpoint in the network, all with VLAN ranges.

The *Topology* element is used to hide internal connectivity, and a full-mesh is assumed. By adding more NML topology information, it is possible to include more detailed descriptions of the internal network.

The *NSA* provides the management information for networks, how the NSI interface can be reached, and who actually maintain the NSA.

The *Location* element has also proven to be quite useful in allowing us to quickly create stunning visualizations using Google Earth.

5 Security Considerations

There are important security concerns associated with the generation and distribution of network topology information. For example, ISPs frequently consider network topologies to be proprietary. We do not address these concerns in this document, but implementers are encouraged to consider the security implications of generating and distributing network topology information.

Implementers should be aware that the NML descriptions do not provide any guarantee regarding their integrity nor their authenticity. The NML documents also can not provide this for the identifiers contained in the documents. Implementers should use external means of verifying the authenticity of identifiers contained in the documents.

6 Contributors

Jeroen J. van der Ham (Editor)
Faculty of Science, Informatics Institute, University of Amsterdam
Science Park 904, 1098 XH Amsterdam
The Netherlands
Email: vdham@uva.nl

7 Acknowledgments

The author would like to thank Roman Łapacz for writing the XML Schema document, and also the OGF NSI and GLIF DTOX groups for their input.

8 Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

9 Disclaimer

This document and the information contained herein is provided on an "As Is" basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

10 Full Copyright Notice

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assignees.

References

Normative References

[URN-OGF-NETWORK] Freek Dijkstra, and Jeroen van der Ham. A URN Namespace for Network Resources. GWD-I *draft-gwdi-urn-ogf-network* (Work in Progress), September 2012. URL `https://forge.ogf.org/sf/go/doc16260`.

[ISO 8601] Data elements and interchange formats – Information interchange – Representation of dates and times. ISO 8601:2004 (Third edition), December 2004. Section 4.3.2 (a), Complete representations of a date and time. Calendar date in basic format. URL `http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=40874`.

[G.800] Unified functional architecture of transport networks. ITU-T Recommendation G.800, February 2012. URL `http://www.itu.int/rec/T-REC-G.800/`.

[RDF-XML] Dave Beckett (editor) RDF/XML Syntax Specification (Revised) W3C Recommendation 10 February 2004. URL `http://www.w3.org/TR/rdf-syntax-grammar/`.

[RFC 2119] Scott Bradner. Key words for use in RFCs to Indicate Requirement Levels. RFC 2119 (Best Current Practice), March 1997. URL `http://tools.ietf.org/html/rfc2119`.

[RFC 3492] A. Costello Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA) RFC 3492 (Standards Track), March 2003 URL `http://tools.ietf.org/html/rfc3492`.

[RFC 3986] Tim Berners-Lee, Roy T. Fielding, and Larry Masinter. Uniform Resource Identifier (URI): Generic Syntax RFC 3986 (Standards Track), January 2005. URL `http://tools.ietf.org/html/rfc3986`.

[Unicode] The Unicode Consortium. The Unicode Standard, Version 6.2.0. Mountain View, CA, USA. November 2012. ISBN 978-1-936213-07-8 Section 5.18, Case Mappings. Paragraph about Caseless Matching. Normative URL `http://www.unicode.org/versions/Unicode6.2.0/` Informative URL `ftp://ftp.unicode.org/Public/UNIDATA/CaseFolding.txt`

[UNLOCODE] United Nations Code for Trade and Transport Locations UN/LOCODE, revision 2012-01, September 2012. URL `http://www.unece.org/cefact/locode/welcome.html`.

[WGS84] Department of Defense World Geodetic System 1984, Its Definition and Relationships With Local Geodetic Systems NIMA Technical Report TR8350.2, Third Edition, June 2004 URL http://earth-info.nga.mil/GandG/publications/tr8350.2/tr8350_2.html.

[XML] Henry S. Thompson, David Beech, Murray Maloney and Noah Mendelsohn XML Schema Part 1: Structures Second Edition W3C Recommendation 28 October 2004. URL http://www.w3.org/TR/xmlschema-1/.

Informative References

[Dijkstra13] Freek Dijkstra, et al. Experimental Features for NML 1. Work in Progress.

[GFD.165] Paola Grosso, Aaron Brown, Aurélien Cedeyn, Freek Dijkstra, Jeroen van der Ham, Anand Patil, Pascale Primet, Martin Swany, and Jason Zurawski. Network Topology Descriptions in Hybrid Networks GFD 165 (Informational), March 2010. URL http://www.ogf.org/documents/GFD.165.pdf.

[RFC 6350] Simon Perreault. vCard Format Specification RFC 6350 (Standards Track), August 2011. URL http://tools.ietf.org/html/rfc6350.

[RFC 6351] S. Perreault. xCard: vCard XML Representation RFC 6351 (Standards Track), August 2011. URL http://tools.ietf.org/html/rfc6351.

[RDFVCARD] Harry Halpin, Renato Iannella, Brian Suda, Norman Walsh Representing vCard Objects in RDF W3C Member Submission 20 January 2010. URL http://www.w3.org/TR/vcard-rdf/.

A XML Schema

This section describes the normative schema of XML documents using the XML Schema language.

```
<?xml version="1.0" encoding="UTF-8"?>


<!--

 File: nsi-ext.xsd - The NSI extension of NML
 Version: $Id$
 Purpose:

-->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace="http://schemas.ogf.org/nsi/topology/2013/10#"
           xmlns:nsi="http://schemas.ogf.org/nsi/topology/2013/10#"
           xmlns:nml="http://schemas.ogf.org/nml/2012/10/base#"
           elementFormDefault="qualified">

  <xs:import schemaLocation="nmlbase.xsd"
             namespace="http://schemas.ogf.org/nml/2012/10/base#"/>


  <xs:complexType name="NsaRelationType">
    <xs:choice>
      <xs:element ref="nsi:Nsa" minOccurs="1" maxOccurs="unbounded"/>
    </xs:choice>
    <xs:attribute name="type" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="http://schemas.ogf.org/nsi/topology/2013/10#managedBy"/>
          <xs:enumeration value="http://schemas.ogf.org/nsi/topology/2013/10#peersWith"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>


  <xs:element name="Relation" type="nsi:NsaRelationType"/>


  <!-- address: rfc6351 xCard: vCard XML Representation -->
  <xs:element name="adminContact">
    <xs:complexType>
      <xs:sequence>
        <xs:any namespace="##other" processContents="lax" minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>


  <xs:group name="BaseNsaContent">
    <xs:sequence>
      <xs:element ref="nsi:adminContact" minOccurs="0" maxOccurs="unbounded"/>
```

```
53        <xs:element name="csProviderEndPoint" type="xs:anyURI" minOccurs="0" maxOccurs="unbounded"/>
54      </xs:sequence>
55    </xs:group>
56
57
58    <xs:complexType name="NsaType">
59      <xs:complexContent>
60        <xs:extension base="nml:NetworkObject">
61          <xs:sequence>
62            <xs:group ref="nsi:BaseNsaContent"/>
63            <xs:element ref="nsi:Relation" minOccurs="0" maxOccurs="unbounded"/>
64          </xs:sequence>
65        </xs:extension>
66      </xs:complexContent>
67    </xs:complexType>
68
69
70    <xs:element name="Nsa" type="nsi:NsaType"/>
71
72
73
74  </xs:schema>
```

B OWL Schema

This section describes the normative schema of the OWL syntax using the OWL ontology definition below.

```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://schemas.ogf.org/nsi/2013/03/topology#"
     xml:base="http://schemas.ogf.org/nsi/2013/03/topology"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:nml="http://schemas.ogf.org/nml/2012/10/base#"
     xmlns:nsi="http://schemas.ogf.org/nsi/2013/03/topology#">
    <owl:Ontology rdf:about="http://schemas.ogf.org/nsi/2013/03/topology#">
        <owl:imports rdf:resource="http://schemas.ogf.org/nml/2012/10/base#"/>
    </owl:Ontology>



    <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // Datatypes
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->



    <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // Object Properties
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->



    <!-- http://schemas.ogf.org/nsi/2013/03/topology#adminContact -->

    <owl:ObjectProperty rdf:about="http://schemas.ogf.org/nsi/2013/03/topology#adminContact">
        <rdfs:domain rdf:resource="http://schemas.ogf.org/nml/2012/10/base#NetworkObject"/>
    </owl:ObjectProperty>



    <!-- http://schemas.ogf.org/nsi/2013/03/topology#managedBy -->

    <owl:ObjectProperty rdf:about="http://schemas.ogf.org/nsi/2013/03/topology#managedBy">
        <rdfs:domain rdf:resource="http://schemas.ogf.org/nml/2012/10/base#NetworkObject"/>
        <rdfs:range rdf:resource="http://schemas.ogf.org/nsi/2013/03/topology#NSA"/>
    </owl:ObjectProperty>

```

```
53
54
55      <!-- http://schemas.ogf.org/nsi/2013/03/topology#peersWith -->
56
57      <owl:ObjectProperty rdf:about="http://schemas.ogf.org/nsi/2013/03/topology#peersWith">
58          <rdfs:domain rdf:resource="http://schemas.ogf.org/nsi/2013/03/topology#NSA"/>
59          <rdfs:range rdf:resource="http://schemas.ogf.org/nsi/2013/03/topology#NSA"/>
60      </owl:ObjectProperty>
61
62
63
64      <!--
65      ///////////////////////////////////////////////////////////////////////////////////
66      //
67      // Data properties
68      //
69      ///////////////////////////////////////////////////////////////////////////////////
70       -->
71
72
73
74
75      <!-- http://schemas.ogf.org/nsi/2013/03/topology#csProviderEndpoint -->
76
77      <owl:DatatypeProperty rdf:about="http://schemas.ogf.org/nsi/2013/03/topology#csProviderEndpoint">
78          <rdfs:domain rdf:resource="http://schemas.ogf.org/nsi/2013/03/topology#NSA"/>
79          <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#anyURI"/>
80      </owl:DatatypeProperty>
81
82
83
84      <!--
85      ///////////////////////////////////////////////////////////////////////////////////
86      //
87      // Classes
88      //
89      ///////////////////////////////////////////////////////////////////////////////////
90       -->
91
92
93
94
95      <!-- http://schemas.ogf.org/nsi/2013/03/topology#NSA -->
96
97      <owl:Class rdf:about="http://schemas.ogf.org/nsi/2013/03/topology#NSA">
98          <rdfs:subClassOf rdf:resource="http://schemas.ogf.org/nml/2012/10/base#NetworkObject"/>
99      </owl:Class>
100 </rdf:RDF>
101
102
103
104 <!-- Generated by the OWL API (version 3.2.3.22702) http://owlapi.sourceforge.net -->
```