Network Markup Language Base Schema

Jeroen van der Ham Martin Swany Freek Dijkstra
Lars Fischer

August 14, 2009

Abstract

A recommendation document describing a normative schema which allow the description of a basic network topology. This schema does not include any layer or technology specific information.

1 Introduction

2 NML Topology Schema

The NML Topology schema describes an information model with elements and their relations that describe computer networks. This schema is kept intentionally general, with provisions to extend the base schema to describe layer-specific information.

TODO: Determine URI.

2.1 Network Object

The basic abstract element of the schema is the *Network Object*. Other basic elements inherit from it. The *Network Object* can have a *Location*, can be related to other instances via a *Relation* and can be described by a *Lifetime*. Every Network Object MUST have an *id* attribute, which MUST be a unique URI. These characteristics are inherited by the subclasses of the *Network Object* class.

The base *Network Object* has three elements that describe it and its relationships:

- Location
- Lifetime
- Relation

The base *Network Object* is subclassed into the top-level topology components, that are sufficient to cover the description of networks. The top-level network elements in this schema are:

- Node
- Port
- Link
- Group
- Service

2.2 Node

A *Node* is generally a device connected to, or part of, the network. A Node does not necessarily correspond to a physical machine. It may be a virtual device or a group of devices.

A Node is connected to the network by its *Port*s. A Node provides *Services*.

The location of a node in the physical world can be described using the *Location* object. The actual location is then described using properties of the *Location* object.

The Relations of Node:

- A *Node* MAY share a *Relation* with one or more *Ports*.
- A *Node* MAY share a *Relation* with one *Location*
- A *Node* MAY share a *Relation* with one or more *Nodes*.

2.3 Port

A *Port*, or interface, connects a *Node* to the rest of the network.

A *Port* is related to zero or one *Nodes*, and also has a relation with zero, one or two (uni-directional) *Links*.

Relations of Port:

- A *Port* MAY have a *hasPort* relation with one *Node*.
- A *Port* MAY have up to two *server* and *client* relations with *Adaptations*.
- A *Port* MAY have a *client* relation with up to two *Adaptations*.
- A *Port* MAY have a *source* relation with up to two *Unidirectional Links*.
- A *Port* MAY have a *sink* relation with up to two *Unidirectional Links*.

2.4 Link

As the name states, a *Link* is a unidirectional link, it provides connectivity from one *Port* to another. These ports are identified using the *source* and *sink* relationships.

A Unidirectional Link MUST have an attribute *type* which is either *Link* or *Crossconnect*. When the type is Crossconnect, the source and sink MUST be of the same device.

Relations of Unidirectional Link:

- A *Unidirectional Link* MAY have a *source* relation with one *Port*.
- A *Unidirectional Link* MAY have a *sink* relation with one *Port*.
- A *Unidirectional Link* SHOULD have a *capacity* attribute which describes the capacity of the link in *bytes per second*.

2.5 Group

To describe collections of network elements, there is a group element. Any element defined above can be part of a group, including another group.

We also define a set of special groups:

- Link
- Path
- Topology
- Domain

2.5.1 Link

A *Link* is a special group, which is a group of two *Unidirectional Links* together forming a bidirectional link between two ports.

2.5.2 Topology

A *Topology* is a group of network elements with the restriction that this group must be connected.

2.5.3 Path

A *Path* is an ordered collection of network elements.

A Path describes a path taken through the network from the source, the first element, to the destination, the last element.

2.5.4 Domain

A *Domain* is an unordered collection of network elements. The *type* attribute can be used to define what kind of Domain is meant.

The value of the type attribute is unrestricted, but several predefined values and their meaning are given below:

- *User*
- *Linking*
- *...*

2.6 Lifetime

All network elements can potentially have a *lifetime*, that consists of vector of *time* elements, which contain a start time and an end time.

2.7 Summary

Figure 1 shows an overview of all the objects in the NML schema in a UML class diagram. The figure also shows the relations between the objects, and their cardinalities.

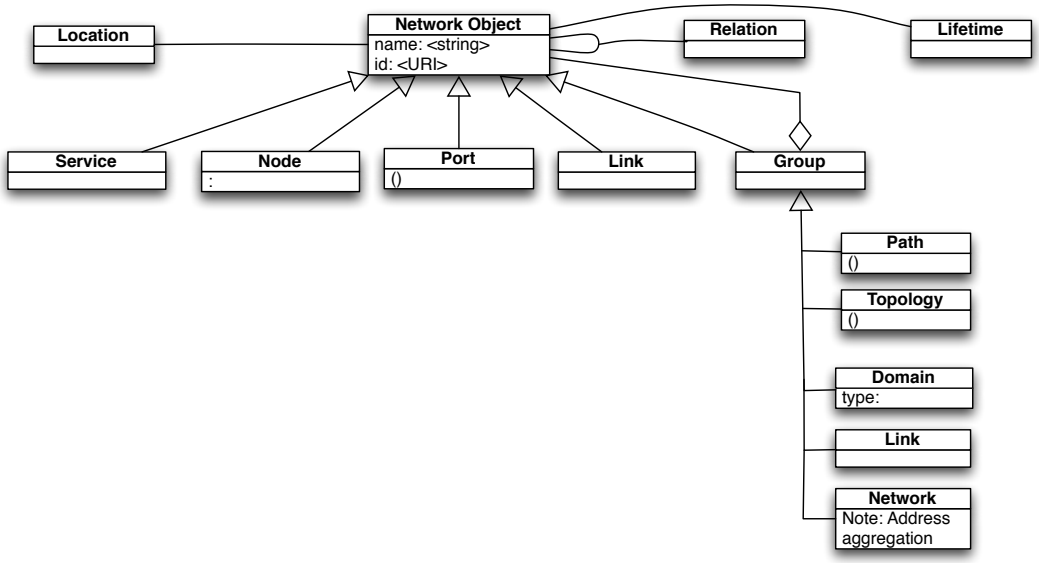

Figure 1: A UML class diagram of the objects in the NML schema and their relations

3 Identifiers

4 Examples