RAINS Resource Modeling
(Description and Abstraction)

1. Introduction

An increasingly important requirement across a wide range of science applications is the need for flexible and seamless integration across multiple resources to support workflows. In this context, we focus on the key resource categories of networks, storage, instruments, and compute. While this need for flexible resource integration is not new, the technology and capability advances in each of these resource realms represents a paradigm shift where this lack of integration is now a limiting factor. The higher-level vision is the development of a collaborative, federated, multi-resource end-to-end service instantiation system that provides real benefits to science applications and workflows.

A key, and some may argue "the key", to realizing this vision is technologies that enable flexible, resource owner controlled resource descriptions. This is often referred to as **Resource Modeling (Description and Abstraction)**. This is important because collaborative federated distributed ecosystems will need to use a compatible method, semantics, and syntax for resource description to allow the higher-level orchestration functions to reason about end-to-end services. This is what will ultimately provide the value added services for applications workflows. In addition, these future collaborative, federated systems will require that the resource owners be firmly in control of the resource descriptions and how they may be provided to external systems. We identify the following high-level requirements for future resource description, modeling, and abstraction systems:

- Resource Owners must be in control of the generation of their resource descriptions
- Resource Owners must be in control of the export of their resource descriptions
- There must be mechanism that allows Resource Owners to create and tailor abstracted resource views based on local polices
- A specific collaborative/federated service region must use compatible methods, semantics, and syntax for resource description to allow the higher-level orchestration functions to reason about end-to-end services
- The higher-level systems will need to include "resource computation" functions that can take multiple resource descriptions as inputs and compute across them in an integrated fashion.

The Resource Aware Intelligent Network Services (RAINS) project is conducting research and developing technologies to provide these types of multi-resource description, modeling, and abstraction systems and the associated computation functions. The architecture and design for the RAINS Resource Description, Modeling, and Abstraction is presented in the remainder of this document.

2. RAINS Resource Description, Modeling, and Abstraction

The RAINS project believes that developing methods to automatically generate, maintain, and securely distribute resource descriptions is the most critical item needed to enable multi-resource intelligent services. In order to decide what services are needed, the resources over which the services may be provisioned must be described and understood. In order to actually provision services, the resource capabilities and provisioning mechanisms must be described. In order to monitor end-to-end services, instrument placement and data retrieval must be reasoned about in the context of the service resources. That is, "Resource Description is Starting Point for Everything".

It should be noted that these resource descriptions are not necessarily detailed descriptions of the actual physical hardware and software systems. Especially for the higher-level orchestration processes, the resources descriptions made externally available will likely be abstracted views that include services capability descriptions. The individual resource owners will likely maintain detailed resource models that do track actual resource topologies and real time states. These detailed resource models will be utilized by the resource owners to manage and monitor their internal systems.

A key goal of the RAINS Resource Description, Modeling, and Abstraction system is to develop one system which can be used for the detailed descriptions needed by resource owners, and allow them to flexibly create synchronized abstracted views which can be used for export to the higher-level orchestration processes.

Fortunately, much work has already been accomplished for description of network resources. The ESnet OSCARS [oscars] system includes network resources description capability that allows for programmatic retrieval. OSCARS enables dynamic network provisioning and is an established production system deployed across many networks in the R&E community. The OSCARS system recently transitioned to a network description method based on the Network Markup Language (NML) [nml] that was defined by the Open Grid Forum (OGF) [ogf]. The NML defines a schema for description of network topologies and resources. Extensibility and tailoring for specific network deployments are strengths of the NML framework.

The RAINS [rains] project has defined extensions to the NML framework to allow other resource types in addition to network elements/topologies to be described and modeled. This is referred to as the Multi-Resource Markup Language (MRML) [mrml]. In this context, other resource types may include systems that are connected to the network such as Data Transfer Nodes (DTNs), storage systems, instruments, and compute nodes. The MRML framework provides the mechanisms to construct integrated multi-resource models.

The NML and the MRML extensions support various representation methods, including the Extensible Markup Language (XML) [xml] and the Resource Description Framework (RDF) [rdf]. The RAINS MRML implementation leverages industry standards such as the RDF specifications from the World Wide Web Consortium (W3C) [w3c]. RDF is a data model similar to classic modeling techniques such as entity-relationship or class diagrams, which has been tailored for use with distributed network and web based applications. Similar to the classic data modeling methods, the RDF model is based on concept of making statements about resources, in the form of subject-predicate-object expressions. In this context, RDF is a directed, labeled graph data format for representing information and/or describing resources. This framework is utilized in many contexts throughout a variety of internet technologies. As a result there are many open source tools available which facilitate i) development of specific resource models, ii) model resource queries and automated element relationship establishment, iii) automatic generation of code libraries, iv) and model visualization techniques. Examples of these readily available tools include Jena [jena] and SPARQL [sparql].

As with any data model, an RDF/OWL representation can readily be stored into a SQL or noSQL databases for persistent storage. Due to the dynamic properties of the RDF/OWS data model, the RAINS project will also focus on utilizing these models in the context of in-memory data structures to support the real-time resource computation and orchestration processes.

This will provide the basis for the Multi-Resource Service Plane (MRSP) to compute topologies and develop intelligent services and workflows.

These resource description methods form the basis for the RAINS Resource Computation Engine (RCE) functions. The RCE will have the algorithms and computation processes needed to provide the mappings between abstract requests and the actual resources. Additional details on the RAINS RCE will be provided in a separate document.

The following sections provide additional details and example on the RAINS Resource Description, Modeling, and Abstraction technologies.

3. RAINS Resource Description, Modeling, and Abstraction – Technical Details

This section provides an overview of the standard and schemas utilized for the NML and RAINS MRML modeling and resource description methods.

3.1 Network Markup Language (NML)

The ESnet OSCARS system includes mechanisms to automatically generate network descriptions based a standard known as Network Markup Language (NML) as defined by an OGF standard. The NML defines a schema for description of network topologies, individual elements, and user tailored abstraction levels. Extensibility and tailoring for specific network deployments are strengths of the NML framework. The key standards for the OSCARS network resource description and provisioning are the following:

- Network Markup Language (NML)
    - OGF GFD-R-P.206, NML Base Schema version 1 (https://www.ogf.org/documents/GFD.206.pdf)

The NML supports various representation methods, including the Extensible Markup Language (XML) [xml] and the Resource Description Framework (RDF)/ Web Ontology Language (OWL) [owl]. The formal NML schema is available here in these two different formats:

- https://redmine.ogf.org/projects/nml-wg/repository/revisions/master/entry/schemas/nmlbase.xsd
- https://redmine.ogf.org/projects/nml-wg/repository/revisions/master/entry/schemas/nml-base.owl

These schemas define the specific standards and requirements for network resource description. These schema formats are designed for machine interpretation, and can be difficult for humans to read and conceptualize. For this reason, graphical descriptions are often generated to facilitate understanding of the hierarchies and relationships. A graphical representation of the NML schema is included in Appendix A of this document.

3.2 Multi-Resource Markup Language (MRML)

The OSCARS and NML system described above is focused solely on network resources. That is, the end-systems connected to the networks such as DTNs, storage systems, instruments, and compute nodes are not included in the above standards, schemas, and implementation. As noted earlier, the RAINS project believes the including these systems into the end-to-end service orchestration is critical for current and future science applications and workflows. The RAINS project decided to build on what the ESnet OSCARS systems developed with respect to network resource descriptions and NML as described above. One reason for this is that network attachment is a common and unifying feature around which subsequent resource integration and coordination activities can be organized. Another reason is that within the DOE and R&E community we expect to that OSCARS based network services will continue to be a critical mechanism for future end-to-end network service instantiation.

The RAINS approach was to develop a set of extensions to the NML to allow modeling of the network-attached resources such as DTNs, storage systems, instruments, and compute nodes. The NML with the RAINS extensions are referred to as the Multi-Resource Markup Language (MRML). These extensions have been constructed in a "backward compatible manner". This simply means that an existing tool which can understands parse a NML description, should be able to parse and MRML description, simply by ignoring the extended namespace and associated new elements. That is, the MRML uses the network resource descriptions as they are defined in the NML schemas and standards.

As for NML, MRML supports various representation methods, including the Extensible Markup Language (XML) [xml] and the Resource Description Framework (RDF)/ Web Ontology Language (OWL) [owl]. The formal MRML schema is available here in these two formats:

- https://github.com/RAINS-Project/nml-mrs-model/blob/master/schema/nml-mrs-ext-v1.xsd
- https://github.com/RAINS-Project/nml-mrs-model/blob/master/schema/rdf-owl/nml-mrs-ext.owl

These schemas define the specific standards and requirements for network resource description. These schema formats are designed for machine interpretation, and can be difficult for humans to read and conceptualize. For this reason, graphical descriptions are often generated to facilitate understanding of the hierarchies and relationships. A graphical representation of the MRML schema in the form of resource specific examples is included in Appendix B of this document.

For the RAINS project we have decided to focus on the RDF/OWL mechanism for expression of the resource schemas. This is due to the powerful data model underpinnings of RDF/OWL in comparison to the XML representation format. The following sections discuss this data model paradigm, standards, methods, and tools. An overview of how this modeling method compares to similar classic modeling techniques such as entity-relationship or class diagrams is also presented.

3.3 MRML RDF/OWL Data Model Discussion

The NML and MRML schemas described above represent the first step toward developing models for the various resources of interest. These schemas define multiple resource elements and provide a basic structure for how the elements are related. There are multiple technology options for the representation of these schemas. These options revolve around a decision to use XML technologies vs. RDF/OWL (i.e., Semantic Web) technologies.

XML is a serialization format that provides a method to express the schemas in a manner that can be exchanged between machines and parsed. The widespread use of XML was motivated by the problem of too many different data formats. That is, XML provided a common framework for multiple document types to be encoded.

In contrast RDF is a data model similar to classic modeling techniques such as entity-relationship or class diagrams, which has been tailored for use with distributed network and web based applications. Similar to the classic data modeling methods, the RDF model is based on concept of making statements about resources, in the form of subject-predicate-object expressions. In this context, RDF is a directed, labeled graph data format for representing information and/or describing resources.

This allows the definition of an abstract set of rules for representing information and element relationships. These expressions are known as triples in RDF terminology. RDF provides a general, flexible method to decompose any knowledge into these small pieces, i.e., triples, with some rules about the meaning or semantics of those pieces. Each edge in the graph represents a fact, or a relation between two elements. The RDF model can be summarized by the following simple rules:

- A fact is expressed as a Subject-Predicate-Object triple, also known as a statement. It is similar to an English sentence.
- Subjects, predicates, and objects are given as names for entities, also called resources or nodes (from graph terminology).
- Names are URIs, which are global in scope, always referring to the same entity in any RDF document in which they appear.

In summary, the purpose of RDF is to provide a structure (framework) for describing identified things (resources). RDF accomplished this via the definition of a directed labeled graph. RDF is composed of three basic elements:

- Resources – the things to be described
- Properties – the relationships between things
- Classes – the buckets to group the things

The element are combined to make simple statements in the form of Triples:

- <Subject> <Predicate> <Object>

RDF has the following properties:

- type
- subClassOf
- subPropertyOf
- range
- domain
- label
- comment

RDF is uniquely suited to defining complex data models and relationships:

- Flexibility: element relationships can be created in context specific ways
- Efficiency: large-scale data can be processed faster. Not linear like traditional databases, not hierarchical like XML.

OWL is a companion technology to RDF and greatly expands the ability to express relationships between resources and associated properties. This allows inferred and associative relationships to be communicated in the resource data model.

The RDF/OWL framework is utilized in many contexts throughout a variety of internet technologies. As a result there are many open source tools available which facilitate i) development of specific resource models, ii) model resource queries and automated element relationship establishment, iii) automatic generation of code libraries, iv) and model visualization techniques.

Examples of these readily available tools include the following:

- SPARQL [sparql]
  RDF is a directed, labeled graph data format for representing information in the Web. This specification defines the syntax and semantics of the SPARQL query language for RDF. SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graphs.

- Apache Jena [jena]
  Jena, a Java RDF API and toolkit is a Java framework to construct Semantic Web Applications. It provides a programmatic environment for RDF, OWL, and SPARQL, and includes a rule-based inference engine.

3.4 Relationship of RDF/OWL to Classic Modeling Techniques Such as Entity-Relationship or Class Diagrams

This section will discuss the relationship/comparison between RDF/OWL and the following technologies:

- Unified Modeling Language (UML)
  - http://www.uml.org
  - OMG Unified Modeling Language (OMG UML), Infrastructure, V2.1.2. Object Management Group, OMG Available Specification, November 2007, http://www.omg.org/spec/UML/2.1.2/Infrastructure/PDF/

- Entity Relationship Model
    - https://en.wikipedia.org/wiki/Entity–relationship_model
    - http://fadyart.com/en/index.php?option=com_content&view=article&id=135:why-not-to-use-entity-relationship-diagrams-to-describe-semantic-assets-if-owl-ontologies-are-considered&catid=46:ontology-general
- YANG
    - https://tools.ietf.org/html/rfc6020
    - http://en.wikipedia.org/wiki/YANG

3.5 Resource Model Examples

This section will contain examples of RAINS Resource Models for the following resources:

- ANL DTNs
- ANL Shock Storage System
- ANL Science DMZ
- ESnet
- WIX
- MAX
- UMD ScienceDMZ
- UMD DTNs

These resources represent an end-to-end path for a UMD to ANL data flow.

References

[w3c] World Wide Web Consortium (W3C), http://www.w3.org

[rdf] Resource Description Framework (RDF) W3C Specifications, http://www.w3.org/RDF/

[owl] OWL 2 Web Ontology Language Document Overview, W3C. 2009-10-27, http://www.w3.org/TR/owl2-overview/

[uml] OMG Unified Modeling Language (OMG UML), Infrastructure, V2.1.2. Object Management Group, OMG Available Specification, November 2007, http://www.omg.org/spec/UML/2.1.2/Infrastructure/PDF/.

[RDF Semantics] RDF Semantics. Patrick Hayes, ed., W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-rdf-mt-20040210/. Latest version available as http://www.w3.org/TR/rdf-mt/.

[RDF Syntax] RDF/XML Syntax Specification (Revised). Dave Beckett, ed. W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/. Latest version available as http://www.w3.org/TR/rdf-syntax-grammar/.

[Turtle] Turtle: Terse RDF Triple Language. Eric Prud'hommeaux and Gavin Carothers. W3C Last Call Working Draft, 10 July 2012, http://www.w3.org/TR/2012/WD-turtle-20120710/. Latest version available at http://www.w3.org/TR/turtle/.

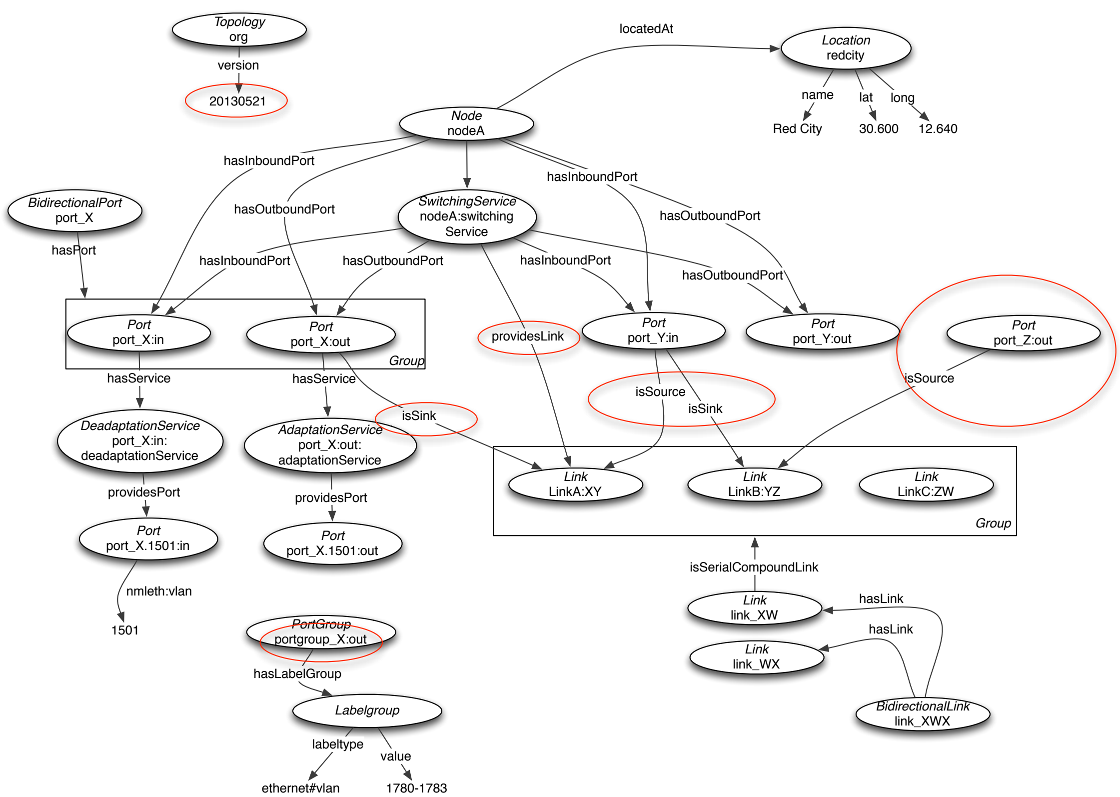
[sparql] SPARQL query language for RDF, https://www.w3.org/2001/sw/wiki/SPARQL

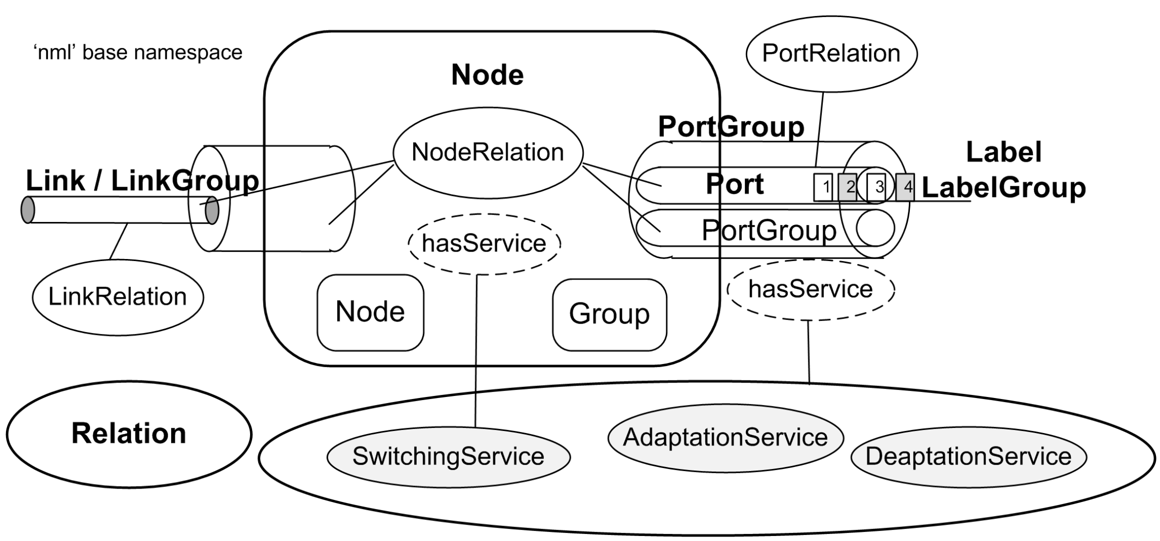
[jena] Apache Jena, https://www.w3.org/2001/sw/wiki/Apache_Jena

[oscars] ESnet On-Demand Secure Circuits and Advance Reservation System (OSCARS), www.es.net/oscars

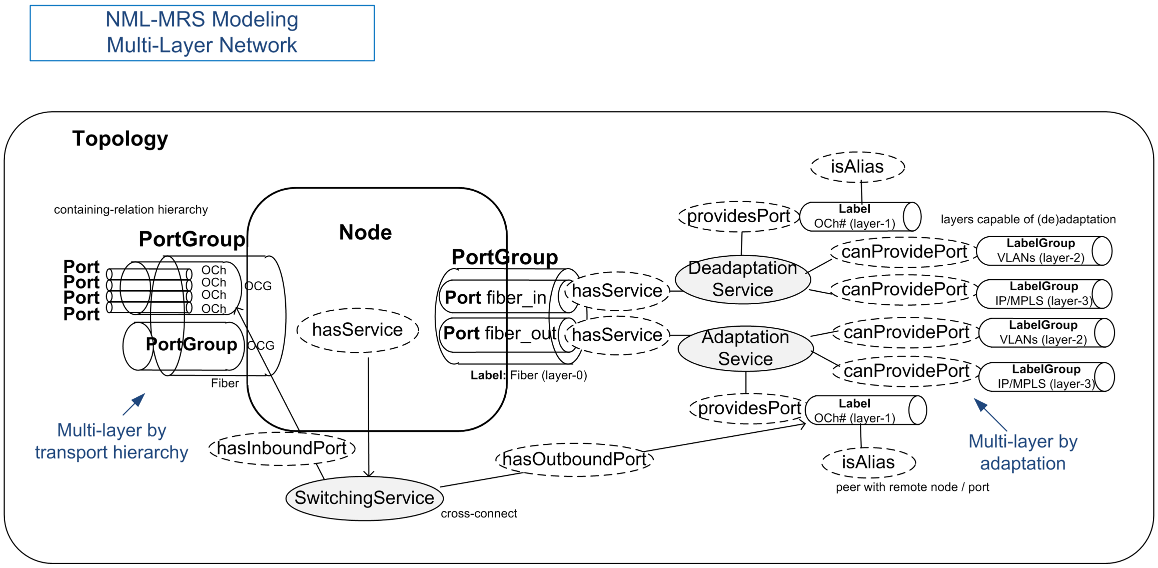
[xml] Extensible Markup Language (XML) 1.0, http://www.w3.org/TR/REC-xml/

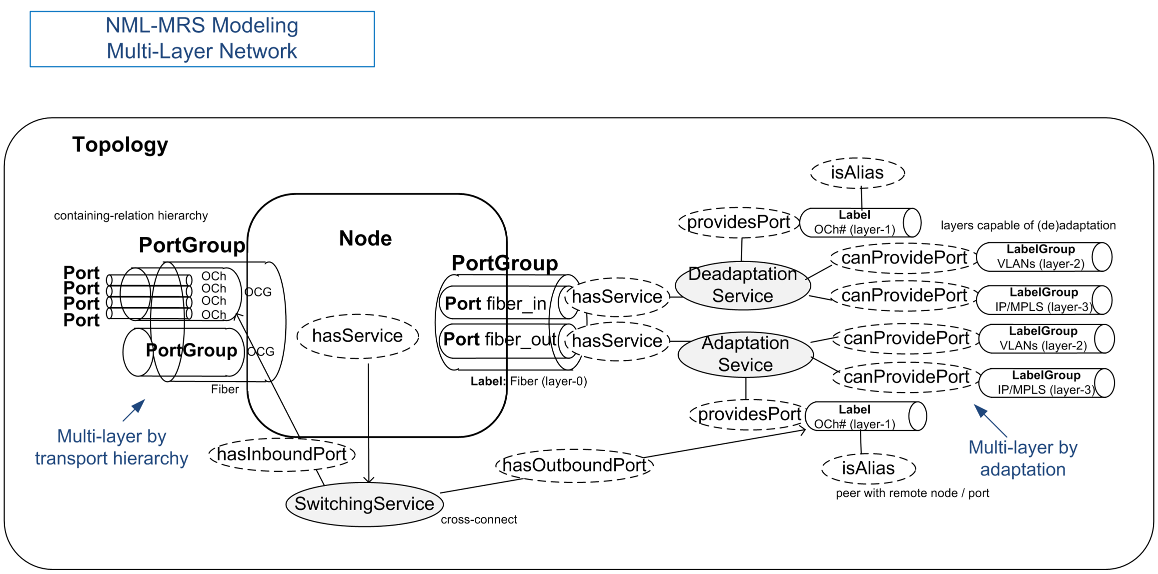
Appendix A
NML Base Schema





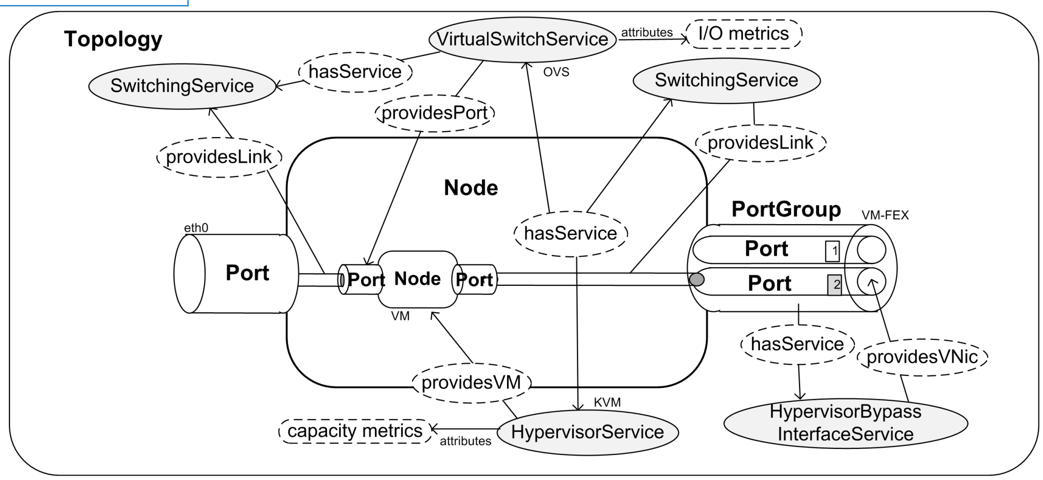
Appendix B
MRML Schema



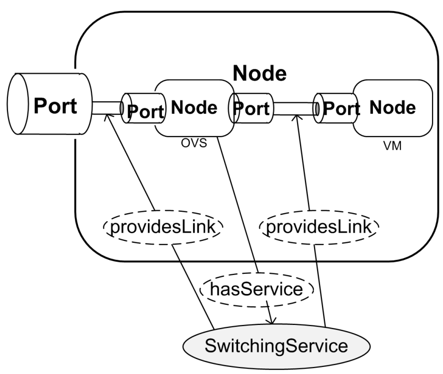


Appendix B
MRML Schema

NML-MRS Modeling
VM Host



Alternative Modeling for OVS



Appendix B
MRML Schema