Towards an Infrastructure Description Language for Modeling Computing Infrastructures

Mattijs Ghijsen, Jeroen van der Ham, Paola Grosso and Cees de Laat
System and Network Engineering research group
Informatics Institute, University of Amsterdam
Email: {m.ghijsen, vdham, p.grosso, delaat}@uva.nl

***Abstract*—This paper describes the Infrastructure and Network Description Language (INDL). The aim of INDL is to provide technology independent descriptions of computing infrastructures. These descriptions include the physical resources and the network infrastructure that connects these resources. The description language also provides the necessary vocabulary to describe virtualization of resources and the services offered by these resources. Furthermore, the language can be easily extended to describe federation of different existing computing infrastructures, specific types of (optical) equipment and also behavioral aspects of resources, for example, their energy consumption.**

Before we introduce INDL we first discuss a number of modeling efforts that have lead to the development of INDL, namely the Network Description Language, the Network Markup Language and the CineGrid Description Language. We also show current applications of INDL in two EU-FP7 projects: NOVI and GEYSERS. We demonstrate the flexibility and extensibility of INDL to cater the specific needs of these two projects.

***Keywords*-computing infrastructures; knowledge representation; semantic web;**

I. Introduction

One of the main ingredients in the design, implementation and operation of computing infrastructures is the information model. This information model must describe both the physical infrastructure and its virtualization aspects. In this paper we describe the fundamentals of such an information model: the Infrastructure and Network Description Language (INDL). INDL draws on our earlier work on modeling computer networks using a semantic approach [1] and on the application of these models in the context of digital cinema [2]. Furthermore, INDL relates to ongoing efforts in the OGF Network Mark-up Language Working Group (NML-WG), and two European projects: GEYSERS [3] and NOVI [4].

As argued in [5], an infrastructure modeling framework provides the basis for virtualization and management of infrastructure resources. This framework should include description, discovery, modeling, composition, and monitoring of those resources and is therefore one of the key components of computing and cloud infrastructures. In this paper we focus mainly on how to describe computing and cloud infrastructures in such a way that the resulting model is technology independent, reusable, easily extensible and linkable to other existing models. To meet these demands we base our modeling approach and the models itself on Semantic Web technologies [6]. In Section II we explain the rationale for adopting this approach.

The structure of this paper is as follows: first we describe existing work on using Semantic Web technology for modeling computer networks and their infrastructure. Next we introduce the Infrastructure and Network Description Language (INDL) and present its main concepts in section III. In section IV we demonstrate the reusability and extensibility of INDL by discussing its application in two projects GEYSERS and NOVI, each with their specific requirements. For NOVI we show how INDL is extended and used to federate existing computing platforms. For GEYSERS we focus on how INDL is extended and used to model optical network devices and virtualization of network and IT resources. We conclude in section V with a comparison of INDL with two other similar efforts on modeling computing infrastructure.

II. Information Modeling

There are many ways to model network and computing infrastructures. The Open Cloud Computing Interface (OCCI) [7] is an API developed by the OCCI working group within the OGF. OCCI provides a number of UML diagrams to model computing infrastructures including network, storage and computing resources but explicit models for virtualization are lacking. The Common Information Model (CIM) [8] was developed within the DMTF and it provides detailed XML schema for describing infrastructure resources and virtualization of those resources. Network topology descriptions however are not very well supported by CIM. The Virtual resources and interconnection networks description language (VXDL) [9] also uses XML schema to describe (requests for) resources and network topology of virtual infrastructures. The Resource Specification, RSpec [10] is used in the GENI project and it provides XML schemas for infrastructure and request descriptions in the form of advertisements, requests and manifests schemas. NDL-OWL is another model developed in the ORCA-BEN [11] project, also within the GENI initiative. Their Semantic Web-based model includes network topologies, layers, utilities and technologies (PC, Ethernet, DTN, fiber switch) as well as cloud computing, and in particular software and virtual machine, substrate measurement capabilities and service procedures and protocols.

In this paper, we argue that the Semantic Web paradigm, and RDF - Resource Description Language - and OWL - Web Ontology Language - are the best solution compared to other options such as XML schema or UML. In this section we provide the reader with the motivation for this choice and we show how the philosophy guided us first in the development of the Network Description Language - NDL - and later on of the CineGrid Description Language - CDL.

The Semantic Web was first proposed as a way for machines to comprehend web pages and data. It uses the Web Ontology Language (OWL) [12], which is a knowledge representation language used to describe ontologies. In OWL data is represented using triples of the form *subject, predicate, object*, meaning you provide some information about a certain *subject*. An *object* can then be used as the *subject* of another triple, which results in a graph structure. Ontologies in OWL are used as vocabularies for these triplets, defining what kind of *predicates* there are, which standard *types* are available, and so on.

There are two main advantages of using OWL to describe computing infrastructures. First, the triples, the main data-format of OWL, form a semantic graph structure describing information about the elements. Such semantic graphs are a good match to computing infrastructures that can also be seen as large graphs of connected resources.

Second, OWL provides explicit separation between semantics and syntax. An OWL Schema defines the ontology, *i.e.* the set of classes and the relations that can exist between those classes. Instances of these classes and their properties are then defined using an OWL syntax. One of the most popular syntaxes for OWL is XML/RDF which uses an XML notation to describe RDF triples, but other terser notations also exist. The clear separation of ontology and syntax also allow users to mix different ontologies. Below we will show an example where network descriptions are combined with descriptions of cinematic content and display capabilities.

A. Network Description Language

In the past years network architectures, especially in research and educational networks, have seen a gradual shift in the type of services offered to end users and applications. They have moved from pure packet-switched data delivery services to a mixture of packet-switched and circuit-switched services. These *hybrid networks* [13], [14] use optical and photonic devices to create circuits in a natural manner, e.g. DWDM devices. These circuits are nowadays an essential component in providing integrated network-computing services in cloud infrastructures. The extensive use of circuits in hybrid networks showed the need for interchangeable network models that could support the operation of control planes protocols, e.g. GMLPS.

The Network Description Language - NDL [1], [15] was developed in the mid 2000s at the University of Amsterdam to fill this gap. NDL adopted the Semantic Web for its schemas, and in particular RDF. NDL is in fact a series of RDF schemas that model network infrastructures in a technology independent manner. NDL identifies the basic elements present in networks, e.g. **`Device`**s, **`Interface`**s, and **`Link`**s. It describes how communication flows between network layers by introducing the concept of adaptation. It defines **`SwitchMatrix`**es as the elements that allow data to move, i.e. switched, within devices. It models the existence of **`Network domains`**, with different administrators and policies. One of the strengths of NDL is that it provides natural support for distribution of information. Independent network operators can create topology descriptions for their infrastructure based on NDL; they can publish them (on the Web) and control plane software can independently gather the information needed to create circuits across domains.

B. Network Markup Language

The NDL schemas have been used, and are used, extensively in the management of the SURFnet6 network in the Netherlands. Still it has been clear to us from the start that a broader standardization effort was necessary to form consensus in the community and to enlarge the adoption base of Semantic Web-based models.

The Network Markup Language Working Group - NML-WG - within the Open Grid Forum has gathered together experts in the area and is working towards a first standard. The modeling effort done in NDL has largely been incorporated in the NML schemas, but is has been enlarged and expanded by adopting concepts and models from the PerfSONAR community [16].

C. Cinegrid Description Language

NDL and NML are network-centric models, and as such, they do not provide ways to describe computing infrastructure at large. While as we argued before one of the novel use-cases for circuits and advanced hybrid networks is exactly in providing the data fabric for applications.

The idea to extend the Semantic Web-based approach used in NDL for networks to generic ICT infrastructures came within the Cinegrid project [17], [18]. Also in CineGrid exchangeable descriptions for the supporting devices are the basis to run applications across domains. Visualization devices, streaming nodes, storage nodes, rendering clusters need to be modeled, in particular their capabilities. The result of this modeling effort is the Cinegrid Description Language (CDL).

The CDL ontology has two distinct parts:

- an architecture ontology, which describes all the **`Element`**s that can be part of CineGrid. These can be either individual components described through the class **Device** or **Group**s of elements.

- a service ontology, which describes the tasks a device can perform for the users of CineGrid. Cinegrid devices can potentially perform multiple types of tasks, possibly at the same time. These tasks maps into services; and the user of the ontology deals directly with services.

The infrastructure ontology is clearly separated from NDL, but there are mappings between the two using the `owl:sameAs` property; this allows that a certain object in the CDL namespace is equivalent to an object in the NDL namespace. This has a very essential practical implication: network administrators describe the network portion in NDL and CineGrid administrators can link their device objects to the NDL objects, and do reasoning on both the CineGrid infrastructure elements and the supporting network topology.

III. The Infrastructure and Network Description Language

The integration of NDL and CDL has been our first step towards an Infrastructure and Network Description Language, INDL. The next major push in this direction has come from two FP7 projects we are involved in: GEYSERS and NOVI. Working at these two information model has helped us to more clearly identify some of the features needed in INDL and to make a first attempt in defining it.

The information model that is being developed in the GEYSERS project [3] focusses on one specific component in the GEYSERS architecture, the Logical Infrastructure Composition Layer (LICL) [19]. The LICL uses virtualization to decouple the physical infrastructure from its associated control-plane and enables on-demand provisioning of virtual infrastructures. A key aspect of the LICL is that it enables composing resources that belong to different infrastructure providers. This aspect was a main motivation for adopting the semantic approach also in this project. The LICL information modeling framework enables the different LICL components – possibly residing at different infrastructure providers – to interact using a common vocabulary. The initial required ingredients for this information model are: physical resources (IT and network resources), virtual infrastructures and virtual infrastructure requests, energy related aspects, quality of service, and security aspects.

NOVI (Networking innovations Over Virtualized Infrastructures) [4] researches methods, algorithms and information systems that can enable composition and management of isolated (virtual) resources provided by different federated Future Internet (FI) platforms. The NOVI Information Model provides abstractions and semantics of federated virtualized resources, enabling ontology-based tools and algorithms used by all the various services operating in the architecture. The information model developed in NOVI has two main objectives. First, the support of the modeling abstractions to cater for the federation of the FEDERICA and PlanetLab Europe platforms and second, to define the necessary modeling concepts needed by other Future Internet platforms to join the NOVI innovation cloud at a later stage.

Concretely the project set out to reach the above objectives by defining an information which supports virtualization concepts, monitoring and measurement concepts, and management policies. NOVI has also decided to base this model on Semantic Web.

A. The INDL Ontology

Figure 1 describes the main concept hierarchy of INDL with its two main classes **Resource** and **Service**. The three main subclasses of **Resource** are: **Node**, **Network Element** and **Node Component**. As shown in Figure 2, every resource is identified by a unique URI and a name.

Services are provided by a node as shown in Figure 3. The service concept allows us to define specific services depending on the domain in which INDL is applied. In the Cinegrid domain, **DisplayService**, **StorageService** and **StreamService** are defined as subclasses of **Service** while in the NOVI domain, other types of services are defined (see Figure 8).

Virtualization is modeled using the **VirtualNode** concept, which is modeled as shown in Figure 1 a subclass of **Node** (i.e. virtual node inherits all properties of node). A virtual node is also implemented on a node (see Figure 4). The implementing node itself can be either a physical node or another virtual node. This allows us to create layers of virtualization stacked on top of each other.

Figure 5 shows how the internal components of a node are modeled by defining **Node** to consist of a number of **Node Component**. The **Node Component** is an abstract class which describes the following essential components of machines which are of interest to the user:

- **Memory Component** shows how much memory is available at a node.
- **Processor Component** to describe how many cores a node has, their speed, et cetera.
- **Storage Component** defines the space available for local storage. Additional work is needed to investigate the modeling of disk partitioning, different file formats and the use of disk-images.
- **Switching Component** is a component that performs the handling of network traffic from **Interface** to **Interface**. This concept is derived from NDL and it describes the switching and/or swapping capabilities of a node.

The **Network Element** is an abstract class with two specific subclasses which are used to model network connectivity.

- **Interface** is the point at which a **Node** is connected to the network. As shown in Figure 6, each node can have multiple inbound and/or outbound interfaces.

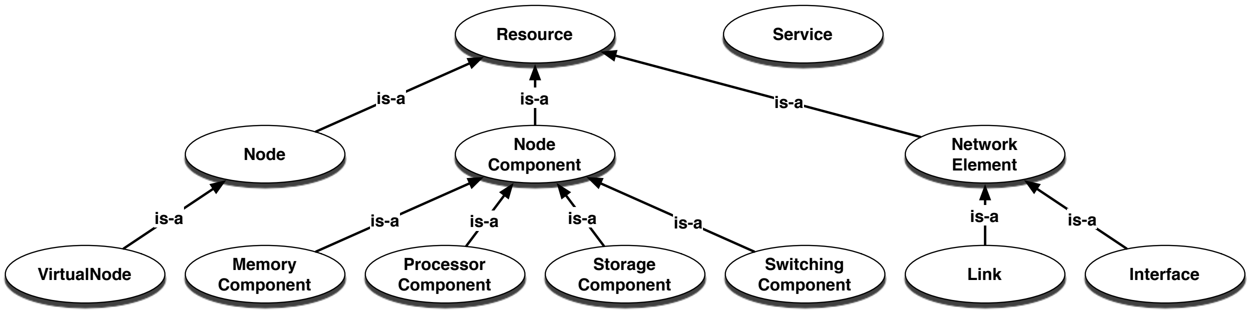

Figure 1. INDL: Main Concept Hierarchy.

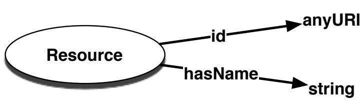

Figure 2. INDL: Resource identification.

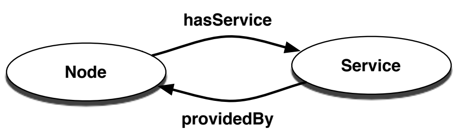

Figure 3. INDL: Modeling services provided by nodes.

Note that the network connectivity is modeled unidirectionally. Furthermore, to model internal switching of interfaces inside a node, the ***switchedTo*** property connects the inbound interface of a node to the outbound interfaces of that node.

- **`Link`** describes the (unidirectional) connection between two **`Node`**s. At each end of each link there is an **`Interface`** which acts as source and sink respectively, and thus indicates the directionality of the link.

Note that all relations between **`Node`**, **`Interface`** and

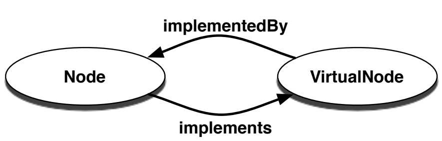

Figure 4. INDL: Modeling virtualization of nodes.

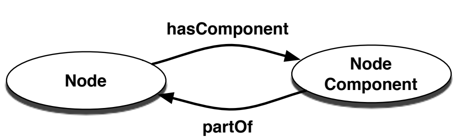

Figure 5. INDL: Modeling internal node components.

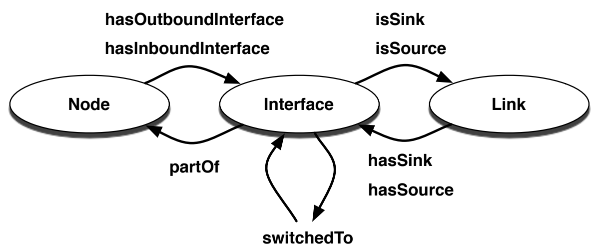

Figure 6. INDL: Modeling connectivity between nodes.

`Link` are symmetric (e.g. ***hasSink/isSink***). This allows search algorithms to travel easily through the connectivity graph of any network topology.

The key feature of INDL which makes it reusable and easy to extend is that we have decoupled virtualization, functionality and connectivity. This allows us to add new functionality (e.g. adding a new type of **`NodeComponent`**) without impacting how we model its connectivity with other devices or how we model virtualization of the new resource. Furthermore, connectivity and functionality is modeled the same for physical nodes and virtual nodes which allows INDL to describe physical computing infrastructures as well as virtual infrastructures.

IV. Applications

To demonstrate the extensibility and flexibility of INDL, we will discuss a number of examples on how INDL has been used to meet the specific challenges in the GEYSERS and NOVI projects. The development of INDL and the information models in NOVI and GEYSERS is still ongoing; for this reason, the model fragments we present here are neither complete nor final.

A. The NOVI Information Model

The NOVI project aims to federate Future Internet platforms and one of the challenges of the NOVI Information Model is to provide interaction with these platforms [20]. Using INDL in the information model provides the basis for interaction between NOVI and the FEDERICA and

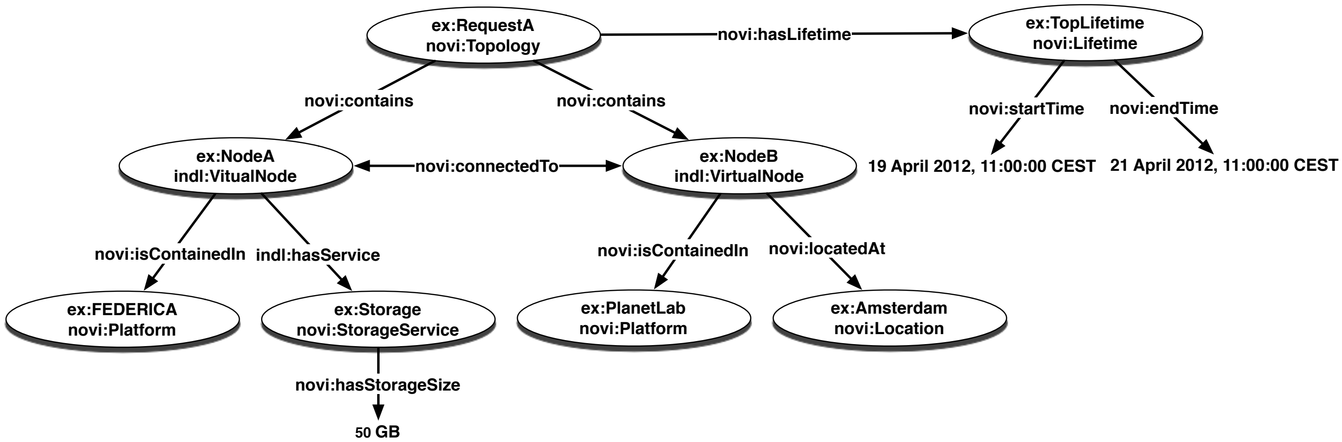

Figure 7. Example request showing two nodes on federated platforms in NOVI

PlanetLab platforms. Not only does the information model have to map to concepts used in these platforms, it also needs to be able to accommodate interaction with other platforms that may be added to the federation in the future.

The federation of different platforms goes further than just providing access to resources. Several other services from the different platforms also have to be combined, such as authentication and policy, but also monitoring so as to provide a single access point to the user.

Both the requesting as well as the monitoring services will require an information model describing resources from the different platforms. In NOVI we use a general ontology based on INDL to describe resources and infrastructure. Two other ontologies to describe monitoring data and policies respectively are left outside the scope of this paper.

Figure 8 shows the classes that are currently defined in the NOVI Information Model (IM).

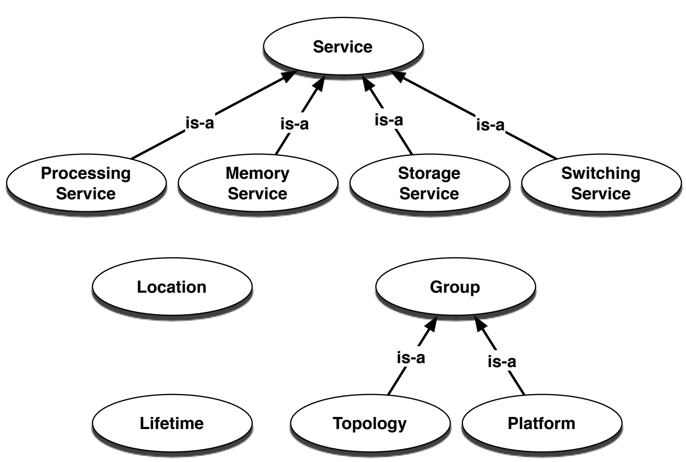

Figure 8. The classes defined in the NOVI resource ontology

- **`Group`** is an abstract concept to describe a grouping of resources, examples are Platform and Topology;
- **`Platform`** describes a particular platform in the NOVI federation. Resources can be linked to the class to denote membership of that platform, and it can provide pointers to other information such as the management service of that platform.
- **`Topology`** can define a group of resources that a user requests, or the implementation of a users' request.
- the **`Service`** concept is extended with four additional subclasses to describe the services which can be implemented by the NOVI resources. For example requesting a total of X GB of storage. We have different services to describe CPU *Processing*, *Memory* size, *Storage* size, and network *Switching* capabilities.

We also have some helper classes to describe some generic properties of resources:

- **`Location`** is used to describe the physical location of objects.
- **`Lifetime`** describes the start and end-time of reservations and their associated resources.

The NOVI IM also defines several properties specific to the NOVI federation and architecture. See [21] for more detailed descriptions of the IM and its properties.

Figure 7 shows a possible request for NOVI. The request contains a very simple topology with two nodes that are connected to each other, starting April 19th 2012 at 11:00:00 CEST and lasts 48 hours from that time. An added constraint here is that the nodes must be on different platforms, in this case PlanetLab and FEDERICA respectively. Furthermore, the node in PlanetLab is requested to be in Amsterdam, and the node in FEDERICA should have a storage service of at least 50 GB. This request can of course be extended with more resources, nodes in other platforms, et cetera.

The construction in the above example allows users to formulate requests for resources on different platforms, which together form a federation. In the architecture this request is split and sent to the different platforms, after which a connection is created between them.

B. Modeling Optical Switches in GEYSERS

One of the key innovations of GEYSERS is to enable virtualization of optical infrastructures. The GEYSERS Information Modeling Framework (IMF), based on INDL, is currently under development to provide an information model for the Logical Infrastructure Composition Layer (LICL) [19]. The LICL is the element responsible of managing physical resource virtualization and composing Virtual Infrastructures (VI). The VIs are offered as a service within the GEYSERS architecture.

One of the requirements for the IMF is to describe optical switches. Because INDL itself does not contain any concepts specifically for describing optical switches, the INDL **`SwitchingComponent`** is extended with a new subclass **`OpticalSwitchComponent`** as shown in Figure 9. A more elaborate description of these concepts and their properties can be found in [22].

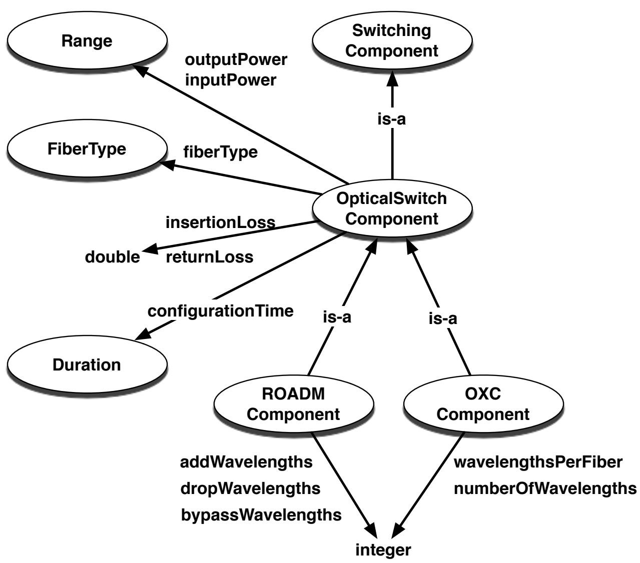

Figure 9. Concepts for modeling Optical Switches in the GEYSERS Information Modeling Framework.

The **`OpticalSwitchComponent`** has the following properties:

- ***inputPower*** and ***outputPower*** describe the input and output range of the power that an optical switch is able to handle and provide.
- ***fiberType*** describes whether the type of fiber supported by the optical switch is single-mode or multi-mode.
- ***insertionLoss*** and ***returnLoss*** describe the optical loss in power of the optical switch.
- ***configurationTime*** is the time needed to configure the switch by adding or removing internal ***switchedTo*** links between the **`Interface`**s.
- ***impairment*** describes the linear and nonlinear effects such as polarization-dependent loss (PDL), polarization-mode dispersion (PMD), chromatic dispersion, crosstalk, etc.

To model a ROADM and OXC, specific components for these types of devices have been added as subclasses of **`OpticalSwitchComponent`**.

- ***addWavelengths*** describes the number of wavelengths added by a ROADM.
- ***dropWavelengths*** describes the number of wavelengths dropped by a ROADM.
- ***bypassWavelengths*** describes the number of wavelengths that are bypassed by a ROADM.
- ***wavelengthsPerFiber*** to describe the number of wavelengths an OXC is able to put on a single fiber.
- ***numberOfFibers*** the total amount of (input plus output) fibers an OXC has.

C. Modeling Virtualization in GEYSERS

The GEYSERS LICL acts as a middleware for the decoupling of the physical substrate and the provisioning of a virtual infrastructure as a service. In order to accomplish this, a complex layer of virtualization needs to be modeled in which we need to distinguish between different types of virtual nodes. Therefore, the **`VirtualNode`** concept has been extended with three new subclasses as shown in Figure 10.

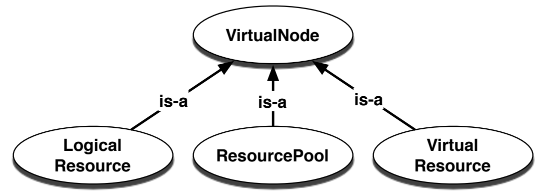

Figure 10. GEYSERS Extension of the VirtualNode concept.

- **`LogicalResource`** represents the aggregation of a number of physical IT resources (i.e. **`Node`**s) into a single resource. Systems like OpenNebula or OpenStack can be used to manage such a cluster of IT nodes and offer its capacity to the upper layer in the LICL system. The **`LogicalResource`** also describes the hypervisor that is being used.
- **`ResourcePool`** is used to indicate a reservation of (part of) an **`LogicalResource`**s capacity for future use.
- **`VirtualResource`** represents an instantiated virtual machine or virtual switch. A **`VirtualResource`** can support different types of disk-images and also point to a URI where the VM image is located.

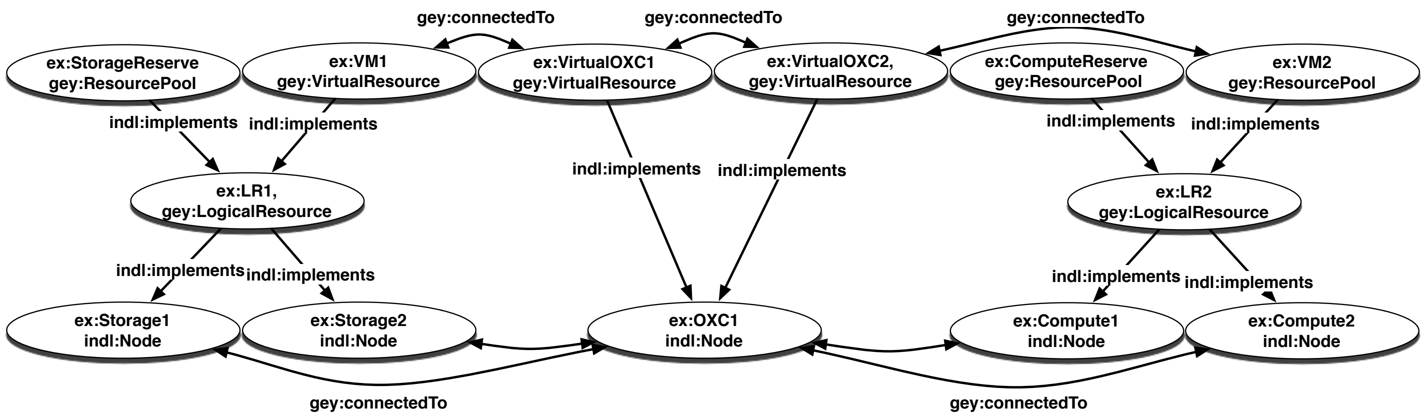

Figure 11. GEYSERS Virtualization Model.

Figure 11 gives a simplified example on how these three concepts are used to represent a virtual infrastructure that is embedded on the physical substrate. It shows two storage nodes and two compute nodes that are both aggregated into a **LogicalResource**. For the sake of the example, the OXC resource is partitioned into two virtual OXCs, both represented as a **VirtualResource**. On both of the logical resources, a **ResourcePool** is implemented to reserve some capacity for future use. Two VMs, each represented as **VirtualResource**, are also instantiated and connected via the two virtual OXCs.

Using the basic virtualization mechanism provided by INDL and the extensions on **VirtualNode** we are able to create the complex virtualization models that are needed by the LICLs architecture.

V. Discussion and Conclusions

As discussed earlier in this paper, INDL is not the only model available to model computing infrastructures. Many of the existing models are based on XML schemata. One notable example is the Resource Specification, RSpec [10] used in the GENI project. RSpec provides infrastructure and request descriptions in the form of advertisements, requests and manifests schemas. Advertisements are used to describe the resources available; requests specify the resources a client is selecting and they may contain a (perhaps incomplete) mapping between physical components and abstract nodes and links; manifests provide useful information about the set of virtual resources (*slivers*) actually allocated to a client. This involves information that may not be known until the sliver is actually created (i.e. dynamically assigned IP addresses, hostnames), or additional configuration options provided to a client. The advantages of the INDL is that it provides a richer semantic about resource and infrastructure components and that it can support service requests in a more user friendly way.

A more interesting comparison can be made with another model also developed within the GENI initiative: the NDL-OWL model [11] in the ORCA-BEN project. NDL-OWL extended NDL and chose the Web Ontology Language (OWL) in place of RDF. Their ontology models networks topology, layers, utilities and technologies (PC, Ethernet, DTN, fiber switch) and it is based on NDL. This is also the main difference between NDL-OWL and INDL. The approach for modeling network topologies in NDL-OWL is based on NDL while INDL uses the latest developments in the OGF NML-WG. Furthermore, NDL-OWL covers cloud computing, and in particular software and virtual machine, substrate measurement capabilities and service procedures and protocols. In this respect it is a necessary next step to try to align INDL and NDL-OWL as much as possible, and we see this as an opportunity to be pursue within a standardization working group.

Because of the semantic approach, INDL provides interoperability with other models. Already INDL can be easily coupled to network centric models such as NDL and/or NML. Also, the CineGrid Description Language, which is more service oriented and does not provide a detailed model for resources, can be added on top of INDL. This results in a stack of ontologies with NDL or NML at the bottom to model the network infrastructure, INDL in the middle model the resources in the network and virtualization of those resources, and finally CDL on top to provide specific services that are offered by the (virtual) infrastructure for the CineGrid domain.

By decoupling connectivity, functionality and virtualization of resources, new types of resources can be easily added without influencing connections with these new resources with other existing resource types. Also the mechanisms for modeling virtualization will not be affected by new resource types. Furthermore, this enables INDL to describe functionality and connectivity of physical resources and virtual resources in a similar manner. Thus INDL enables the description of both physical and virtual infrastructures and how these two infrastructures are coupled.

Ongoing development in the GEYSERS and NOVI projects and possible alignment with other infrastructure models will further improve and refine the current version of INDL to form a basis for describing any computing infrastructure in future modeling efforts as well as providing a foundation for other service oriented models such as CDL.

Acknowledgment

This work was partially supported by the European Commission, 7^{th} Framework Programme for Research and Technological Development, Capacities, Grant No. 257867 – NOVI and Grant No. -248657 - GEYSERS. Furthermore, this publication was supported by the Dutch national program COMMIT and the GigaPort 2012 Research on Networks project. The authors would like to thank Ralph Koning for his effort in the development of CDL.

References

[1] J. J. van der Ham, F. Dijkstra, F. Travostino, H. M. A. Andree, and C. T. A. M. de Laat, "Using RDF to describe networks," *Future Generation Computer Systems - IGrid 2005: The global lambda integrated facility*, vol. 22, no. 8, pp. 862–867, 2006.

[2] R. Koning, P. Grosso, and C. T. A. M. de Laat, "Using ontologies for resource description in the CineGrid Exchange," *Future Generation Computer Systems*, vol. 27, no. 7, pp. 960–965, 2011.

[3] E. Escalona, S. Peng, R. Nejabati, D. Simeonidou, J. A. Garcia-Espin, J. Ferrer Riera, S. Figuerola, Y. Demchenko, C. de Laat, and E. Al., "GEYSERS : A Novel Architecture for Virtualization and Co-Provisioning of Dynamic Optical Networks and IT Services," in *Proceedings Future Network and Mobile Summit*, 2011.

[4] NOVI - Networking innovations Over Virtualized Infrastructures. [Online]. Available: http://www.fp7-novi.eu/

[5] Y. Demchenko, J. J. van der Ham, V. Yakovenko, C. T. A. M. de Laat, M. Ghijsen, and M. Cristea, "On-Demand Provisioning of Cloud and Grid based Infrastructure Services for Collaborative Projects and Groups," in *The 2011 International Conference on Collaboration Technologies and Systems (CTS 2011)*, Philadelphia, Pennsylvania, USA, May 2011.

[6] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, pp. 29–37., May 2001.

[7] "Open Cloud Computing Interface (OCCI)." [Online]. Available: http://occi-wg.org/

[8] "Common Information Model." [Online]. Available: http://dmtf.org/standards/cim

[9] G. P. Koslovski, P. Vicat-Blanc Primet, and A. Schwertner Charão, "VXDL: Virtual resources and interconnection networks description language," *Networks for Grid Applications, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering,*, vol. 2, p. 138, 2009.

[10] ProtoGENI RSpec. [Online]. Available: http://www.protogeni.net/trac/protogeni/wiki/RSpec

[11] I. Baldine, Y. Xin, A. Mandal, C. Heermann, J. Chase, V. Marupadi, A. Yumerefendi, and D. Irwin, "Networked cloud orchestration: A geni perspective," in *Workshop on Management of Emerging Networks and Services*, 2010.

[12] "Web Ontology Language (OWL)." [Online]. Available: http://www.w3.org/2004/OWL/

[13] C. T. A. M. de Laat, E. Radius, and S. Wallace, "The Rationale of the Current Optical Networking Initiatives," *Future Generation Computer Systems*, vol. 19, no. 6, 2003.

[14] C. T. A. M. de Laat and P. Grosso, "Lambda Grid developments, History - Present - Future," in *Closing Conference of ESLEA Project, Proceeding of Science*, Jun. 2007.

[15] The Network Description Language. [Online]. Available: http://www.science.uva.nl/research/sne/ndl

[16] A. Hanemann, J. W. Boote, E. L. Boyd, J. Durand, L. Kudarimoti, R. apacz, D. M. Swany, S. Trocha, and J. Zurawski, "PerfSONAR: A Service Oriented Architecture for MultiDomain Network Monitoring," in *Proceedings of the Third International Conference on Service Oriented Computing - LNCS 3826*. Springer Verlag, Dec. 2005, pp. 241–254.

[17] P. Grosso, L. Herr, N. Ohta, P. Hearty, and C. T. A. M. de Laat, "Editorial: CineGrid: Super high definition media over optical networks," *Future Gener. Comput. Syst.*, vol. 27, no. 7, pp. 881–885, Jul. 2011.

[18] "The CineGrid collaboration." [Online]. Available: http://www.cinegrid.org/

[19] J. A. Garcia-Espin, J. Ferrer Riera, S. Figuerola, M. Ghijsen, Y. Demchenko, J. Buysse, M. de Leenheer, C. Develder, F. Anhalt, and S. Soudan, "Logical Infrastructure Composition Layer, the GEYSERS Holistic Approach for Infrastructure Virtualisation," in *Terena Networking Conference (TNC2012)*, 2012.

[20] J. J. van der Ham, P. Chrysa, S. Jozsef, M. Peter, K. Yiannos, P. Grosso, and L. Lymberopoulos, "Challenges of an information model for federating virtualized infrastructures," in *5th International DMTF Academic Alliance Workshop on Systems and Virtualization Management: Standards and the Cloud*, 2011.

[21] J. J. van der Ham, M. Campanella, A. Chuang, F. Farina, P. Grosso, Y. Kryftis, M. Peter, A. Monje, C. Papagianni, C. Pittaras, J. Steger, C. Velayos, A. Wibisono, and K. Wieringa, "NOVI Deliverable D2.2 - First Information and Data Models," Tech. Rep., 2011.

[22] R. Nejabati, E. Escalona, S. Peng, and D. Simeonidou, "Optical Network Virtualization," in *Optical Network Design and Modeling (ONDM), 2011 15th International Conference on*, 2011, pp. 1–5.