To a Proposal towards Standardization of Network Design Markup Language

Andriy Luntovskyy, *Dresden University of Technology,*
Nöthnitzer Str. 46, 01062 Dresden, Germany, Andriy.Luntovskyy@tu-dresden.de,
Taissia Trofimova, *Dresden University of Technology,*
Natalia Trofimova, *Dresden University of Technology,*
Dietbert Gütter, *Dresden University of Technology,*
Alexander Schill, *Dresden University of Technology*

Keywords: CANDY (Computer-Aided Network Design UtilitY), NDML/NDML+ (Network Design Markup Language), ERM (Entity- Relationship-Model)

1. Introduction

The presented work introduces into network description language NDML+. This language has been developed in frame of CANDY project and is aimed to support computer-aided network design via CANDY Framework [i] [ii]. The language represents an advancement of NDML (Network Design Markup Language) as one of integration component for CANDY tools.

Capacity and quality are essential characteristics of a network and simultaneously its important competition factors. Therefore accurate planning is the precondition to get a decisive advantage in the increasing pressure of competition. In view of the complexity of networks this task can only be solved by using efficient software tools.

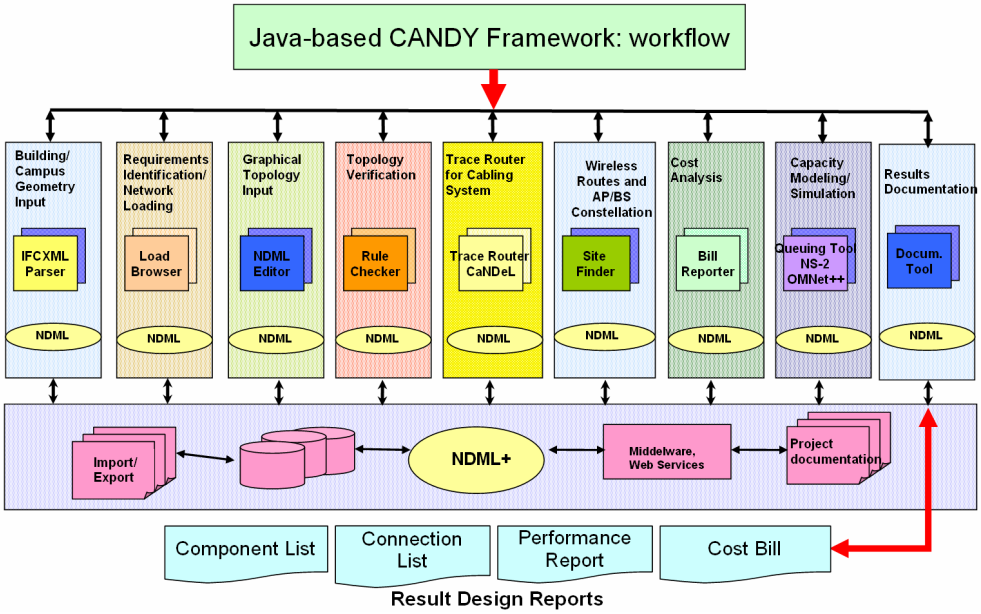

Fig.1: CANDY Framework

Engineers have to evaluate, optimize and verify large scaled objectives characterized by complex coherences with immense amount of data. In order of creation of a suitable support, the CANDY project has been started at the TUD Dresden University of Technology. The main aim of this project is the development of an integrated planning tool for IEEE802.3 Ethernet, IEEE802.11 WLAN and IEEE802.16 WiMAX networks. The CANDY Framework (Fig.1) represents an uniform platform, consists of several tools. Each tool within the platform is responsible for a certain task (e.g. CANDY Bill

Reporter creates a cost analysis bill). Thereby the network description language NDML is an important integration component, serving data exchange between the tools.

2. NDML versions and dialects: analysis and modeling

The predecessor language NDML [iii] suffers from some lacks. First of all, not all aspects of the network project planning were considered in the language (e.g. Traffic mapping, Costs and Geometry analysis etc.). Secondly, it was not defined uniformly (coexistence of XML Schema, or XSD, and Document Type, or DTD, definitions) and last but not least, the version assignment did not correspond to regular software conventions. Therefore it was an imperative to revise NDML and eliminate remaining disadvantages. To the improved and standardization- ready language has been given a new name: Network Design Markup Language Plus [iv].

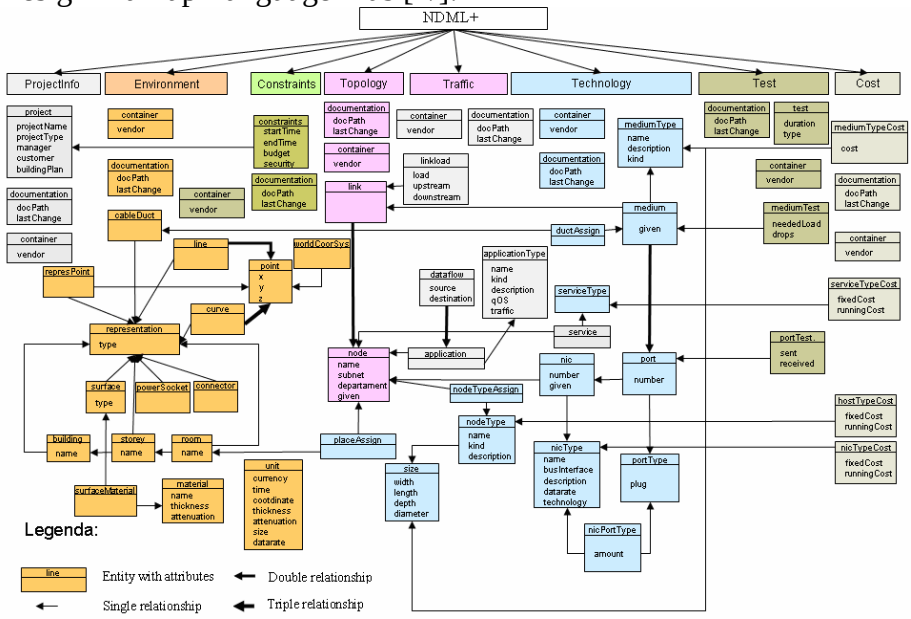

Fig.2: ER-model for NDML+

This language maps the input data and network design results in better way. The data is partitioned inside a single file into topic groups (viewpoints). Thus each group contains associated data. Beyond, data of different viewpoints can also be related to each other, what carried to certain redundancies in older versions. Further improvement of the new language specification is use of just one unified XSD. Hereby the definition of just a single instance for network description is required, leading to a simplified data management (e.g. consistency can be ensured). Another advantage of NDML+ is also that all necessary information for data exchanges, network design and important updates became available on a common basis. The Fig.2 represents the elaborated ERM-based meta-model under use of standard “primary/foreign key”-constraints [v]. The meta-model for NDML+ has been semantically split up into the following viewpoints: ProjectInfo, Environment, Constraints, Topology, Traffic, Technology, Test and Cost, which are marked accordingly colorful in Fig.1. The color of the entities shows, to which viewpoints they are belong to. The arrows represent the single, double and triple relations between the entities. It shall be noticed, that the existing language dialect RadioNDML [vi] for wireless network routes was also integrated into the new version of NDML+. Table 1 represents a comparison between existing languages and NDML+, which

reserves thereby its own place in this referent group of the network description languages. It has to be emphasized that NDML+ is a mostly suitable language for the CANDY Framework, as well as for other network design suits: its primary objective is network design routines; XML-based (ideal for data exchange) and covers all necessary aspects of network engineering; relevant to describe the mostly used local-area networks standards like IEEE 802.3, IEEE 802.11, IEEE 802.16 and the networks solutions on this basis:

Features	Languages			
	Netml[vii]	NetML[viii]	TOOL[ix],[x]	NDML+
XML-based	+	+	-	+
Declarative	+	+	-	+
Procedural	-	-	+	o
Object-oriented	-	-	+	-
Network description	+	+	+	+
Network design	+	-	+	+
Network configuration	-	+	-	o
Network optimization	-	-	+	o
Acceptability to a special network type	o	o	Abstract telco networks	Ethernet, WLAN, WiMAX, TCP/IP
Building geometry	-	-	-	+
Topology specification	+	+	+	+
Traffic mapping	+	-	+	+
Network costs	-	-	+	+
Network components	+	+	+	+
Network test	-	-	-	+
Network cable trace routing	-	-	-	+
NetML – Network Markup Language; Netml – Network Modeling Language **Legend:** "+" means "available"; "o" – "limited available"; "-"– "not available"				

Table 1: Qualitative comparison of NDML/NDML+ and other languages for computer networks description

Naturally, the mentioned referent group can be extended via well-known tools and languages. In this regard the CAD-tools OPNET Gurus, NS-2, AWE Comm. WinProp as well as the definition languages of the Netopeer and the Verinec projects for network configuration must be mentioned. The CANDY tools based on the unified NDML+ version solve firstly the problems of automated design of combined wired and wireless local-area networks. The XML-based grammar facilitated data exchange between various design tools. The offered freeware tools can be encapsulated and the formats trans-coded.

3. Formal grammar of NDML+ and Design Patterns

The language can be described by a formal grammar. For these purposes the well-known EBNF-Grammar (Extended Backus-Naur-Form) has been used. It can be represented by the following quadruple:

$$G\,(\text{NDML+}) = (T, V, P, S) \qquad (1)$$

where **G** – grammar, **S** – start symbol; **V** – set of all variables (elements) of NDML+; **T** – set of all terminal symbols; **P** – set of all production rules P= {V x T}; **x** – Cartesian

symbol. The start symbol **S** represents the root element of the language NDML+. Let us to give an example. The language is defined as follows in EBNF terms (e.g. for the Viewpoint ProjectInfo):

S := [ProjectInfo][Environment][Constraints][Traffic][Topology][Technology][Test][Cost].
ProjectInfo := Projects [ProjectDocumentation][ProjectContainers].
Projects := Project.
Project := id ProjectName projectType manager customer buildingPlan (2)
ProjectDocumentation := {ProjectDocumentation}.
ProjectDocumentation := id docPath lastChange.
ProjectContainers := {ProjectContainer}.
ProjectDocumentation := vendor {any}.

NDML+ reflects the design process of local-area networks. All data sets are summarized under certain abstract views (viewpoints). Table 2 elucidates their contents:

Viewpoint	Provides data
ProjectInfo	Project name, customer, manager
Environment	Building plan, environment properties
Constraints	Project start time and end time, budget
Topology	Structure of networks
Traffic	Traffic analysis and description, traffic matrix
Technology	Components and technology
Cost	Cost of Network
Test	Test data, test documentation

Table 2: The overview of all NDML+-viewpoints

The viewpoints were combined within a hierarchy (Fig.3). This refers to the dependence of the data on each other and does not only force a certain order when working on viewpoints, but is also important for data consistency. If changes at a specified viewpoint were accomplished, dependent viewpoints also must be modified accordingly.
As schema definition language was used XSD, because: XSD is more powerful than other schema languages (e.g. DTD); multiple XSD libraries and frameworks are available. For designing of the XSD the Russian Doll Design (RDD) has been deployed due to the following advantages: design routine became more comfortable parallel to better schema instantiating possibility (general advantage for CANDY tools today!).
Fig.4 illustrates a network topology containing traffic descriptions, topology specifications and components predefinitions. In order to designate the affiliation to a certain viewpoint, there are used different colors (“blue” – for traffic descriptions, “black” – topology definition, and “green” – selected networking technology).

4. Web Portal for NDML+ and Tools

Publication of the NDML-specifications follows via a Web Portal. It contains the language definitions via XML-schema and the detailed description of NDML+. A special validator for the instances written in NDML+ is integrated. The Web Portal has been built in standard 3-tier-architecture: the presentation layer (via client browser), the application layer (via application server) and the data layer (via relational data base). Two databases are therefore in use: one to serve the Web Forum and one to support the information about grammatical structures of NDML+. The XML-schema of NDML+ is stored in a relational database; thus the defined ERM can be seen. Due to the XML-schema was extracted from the ERM by suitable transformations, the XSD can be

transferred backward into an ERM without any further problems. In addition the ERM was normalized accordingly (up to the BCNF, Boyse-Codd-Normal-Form). In Fig.5 the regularity of transformations is represented; firstly from the meta-model ERM into XSD and afterwards into concrete tables of a relational database. Furthermore the Web Portal offers information about CANDY Framework tools and related works in the surrounding field of computer network design. In future inline with the portal the design tools will be available for interested partners for free use (download), in particular the tools, which can process geometry data, presented in the formats: IFCXML, JPG, Python CAD and NDML (for CANDY Wireless Site Finder, CANDY Trace Router, CANDY Bill Reporter [xi]).

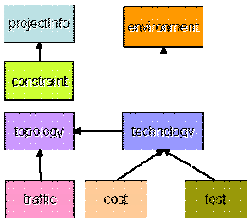

Fig.3: Viewpoint Hierarchy in NDML+

```
<topology>
  <nodes>
    <node_id="1"  name="pc_001" subnet="1"
department="marketing given="yes" />
    <node_id="2" name="pc_002" subnet="1"
department="marketing" given="yes"/>
    <node_id="3" name="sw_001" subnet="1"
department="marketing" given="no"/>
  </nodes>
  <links>
    <link_id="1" node1_id="1" node2_id="3"/>
    <link_id="2" node1_id="2" node2_id="3"/>
 </links>
</topology>
<traffic>
<linkLoads>
   <linkLoad id="1" link_id="1" upstream="50Mbps"
downstream="50Mbps"/>
   <linkLoad id="2" link_id="2" upstream="50Mbps"
downstream="50Mbps"/>
</linkLoads>
<dataflows>
<dataflow id="1" source_id="1"
destination_id="2"/>
 <dataflow id="2" source_id="2"   destination_id="1"/>
</dataflows>
</traffic>
```

```
<technology>
 <mediums>
   <medium id="1" link_id="1" type_id="1"
given="no"/>
   <medium id="2" link_id="2" type_id="1"
given="no"/>
</mediums>
<nodeTypes>
 <nodeType id="1" kind="workstation"/>
 <nodeType id="2" kind="switch"/>
</nodeTypes>
<mediumTypes>
   <mediumType id="1" size_id="1" kind="utp-cat5e"/>
 </mediumTypes>
</technology>
```

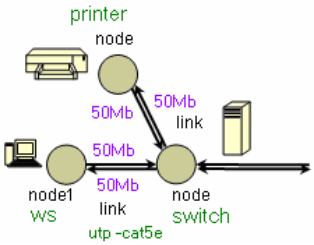

Fig.4: Simple network example with accompanied instance written in NDML+

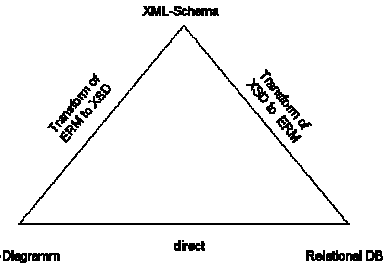

Fig.5: Transformation paradigm for NDML+

5.Conclusions

This new language version NDML+ considers the current state-of-art in the development of CANDY Tools and provides a good framework for the integration of further

applications (NS-2 Simulator, Queuing Tool for WLAN/Ethernet etc.). The next development goals of CANDY Framework and NDML+ are as it follows:

1. Use of the only standardized NDML+ for mapping on the existed network design document model in the CANDY Framework: it is only necessary to have an interface from each tool to NDML+.
2. Development of a theoretically founded methodology, which can be used to create XML-based target description languages. It can be expected that the number of XML-based markup languages aimed to the network design will proliferate in the future. Sooner or later a lot of developers will have to create one.
3. Creation of a laborious Web-Forum to support NDML+ development via 3rd parties is necessary. Deployment examples for the standardized language via the services of Virtual CANDY Lab must be available in Web. Thereby in the future demonstration of characterized features of the language and certain basic CANDY tools will be possible.
4. Improvement of CANDY Framework functionality and supported computer networks models for wired and wireless local networks as well as WiMAX-nets. The newly elaborated tools must provide full support of the standardized language.

References

[i] CANDY@TUD Project: http://www.rn.inf.tu-dresden.de

[ii] Schill, A., Gütter, D., Luntovskyy, A., Pfeifer, G., Panchenko, A.: CANDY: Integrated Environment for Network Design; *8th WMSCI-04, Orlando, Florida, USA, pp.252-257, ISBN 980-6560-13-2*

[iii] Schill, A., Luntovskyy, A., Gütter, D., Vasyutynskyy, V., Pfeifer, G.: Approaches to the Integrated Network Design for Modern Health Care Institutions *(in Scientific Magazine "Electronics and Communication" at NTUU KPI accredited by Superior Attestation Commission of Ukraine "VAK Ukraine", pp.114-121, Vol.1, ISSN 1811-4512)*

[iv] Trofimova, T.: Entwicklung eines Vorschlages zum Standardentwurf der Fachsprache NDML, *Bachelorarbeit, 2006, TU Dresden, Fak. Informatik, Professur RN (in German)*

[v] Meyer,H., Klettke,M.: *XML und Datenbanken.Konzepte,Sprachen und Systeme,2005*

[vi] Luntovskyy, A., Gütter, D., Schill, A.: Models and Methods for WLAN/WIMAX-Network Design; *IEEE CriMiCo-06, Sevastopol,Vol.1: pp.391-393; ISBN: 966-322-006-6, IEEE catalog number: 06EX1376*

[vii] Netml: Network Modelling Language, *Univ. of Southern Queensland*: http://www.sci.usq.edu.au/research/NRG/

[viii] NetML: Network Markup Language, *Univ. of Roma*: http://www.dia.uniroma3.it/

[ix] Speetzen, A.: Methodische Beiträge zur redundanten und dynamischen Planung von Kommunikationsnetzen, *2003, Dissertation, RWTH Aachen*

[x] AIXCOM GmbH & COMPOSIS: http://www.aixcom.com/Produkte/Composis/Manual/userman_ger.html

[xi] Luntovskyy, A., Gütter, D., Schill, A.: Network Design and Optimization under Use of CANDY Framework; *IEEE CriMiCo-06, Sevastopol, Vol.2: pp.394-397, ISBN: 966-322-006-6, IEEE catalog number: 06EX1376*