

FLEXPLANET, a Flexible Multi-Layer Network DesignTool

Zoltán Zsóka,

László Jereb,

Tamás Izsó,

Ferenc Unghváry



Outline

- Introduction
- Modeling approaches
- Structure and main components of the tool
- Network model and design algorithms
- Flexible network model
- Components of the model
- Generic hierarchy concept
- Summary



Introduction

- Motivations
 - A continuous evolution of the area has to be followed:
 - network technologies are changing,
 - new technologies come up,
 - existing technologies live together,
 - the provided services are changing,
 - new services come up.
 - The above factors imply significant changes in the network architecture.
- Main objectives
 - The applied network model has to be flexible enough to adopt later introduced technologies and modifications.
 - The tool has to be flexible enough to adopt later developed methods and algorithms.
 - More directions of the application has to be addressed:
 - design of large scale, realistic, operating networks,
 - support of the education,
 - support of research.

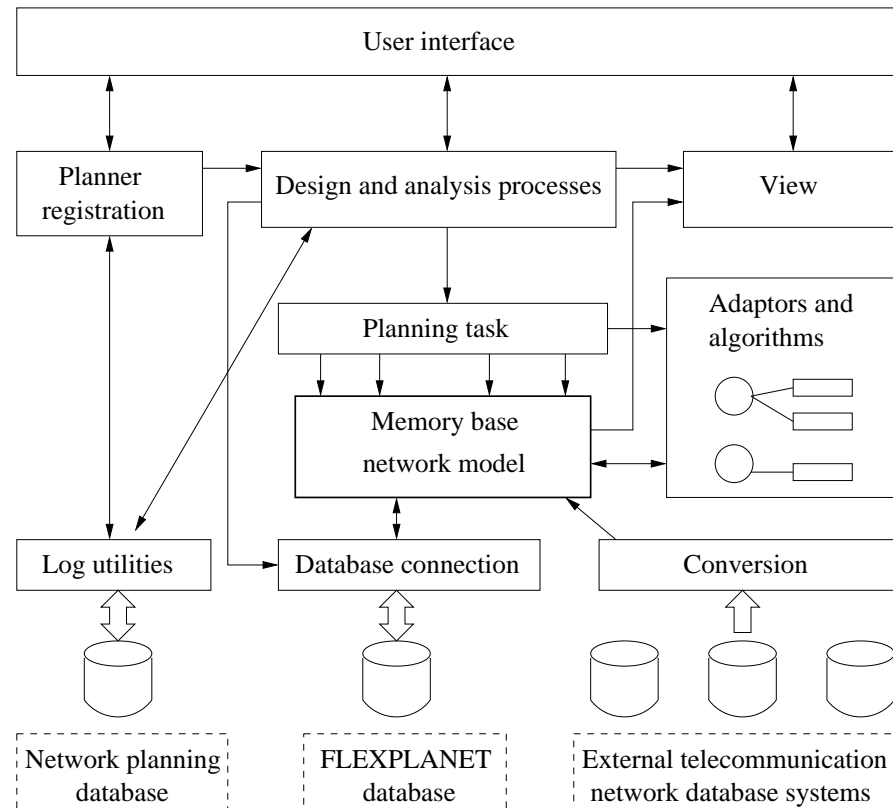


Modeling approaches

- The networks to be designed are multi-layer networks:
 - the layers are formed on the basis of client-service connections, that exist between the network entities,
 - the layers do not always represent the technological layers, but also logical levels can be introduced in modeling.
- Tools are using different approaches:
 - Appropriate models:
 - the model is created for the support of one special method or algorithm,
 - Frameworks:
 - more methods are supported by the same model,
 - case 1: the network model is used directly by the algorithms,
 - case 2: the algorithms use their own model, that was created according to the network model,
 - case 3: the network model is used by the algorithms, but its inside structure is hidden behind well defined interfaces.



Structure and main components of the tool



- Database connection maintains the synchrony between the inner FLEXPLANET network model (*netmodel*) loaded in the memory and the physical storage of the netmodel's data.
- Conversion realizes the connection between the netmodel and the data extracted from external databases that describe operating networks.
- Planning tasks are the implemented methods that a planner can apply to modify the netmodel. The task controls the algorithms. Processes are built up from tasks.
- User interface is not a universal program interface, but it is realized in more functional programs, e.g. administrative frame, graphical presentation of the network, etc.



Network model and design algorithms

- FLEXPLANET is a framework case 2, i.e., it separates the network model from the algorithms.
- The netmodel is constructed for storing the information about the network. It considers the design aspects instead of the record approach of an operator.
- The algorithms
 - work on their own data structures,
 - run effectively,
 - need an adaptor to use the information stored in the netmodel.
- Generic models can be applied by more than one algorithms.
- The methods can be implemented independently from the tool and can be of any concept, e.g. graph based, LP, combinatoric etc.
- Commercial or free tools and frameworks can be embedded too, e.g. Lemon, CPLEX, Gecode.

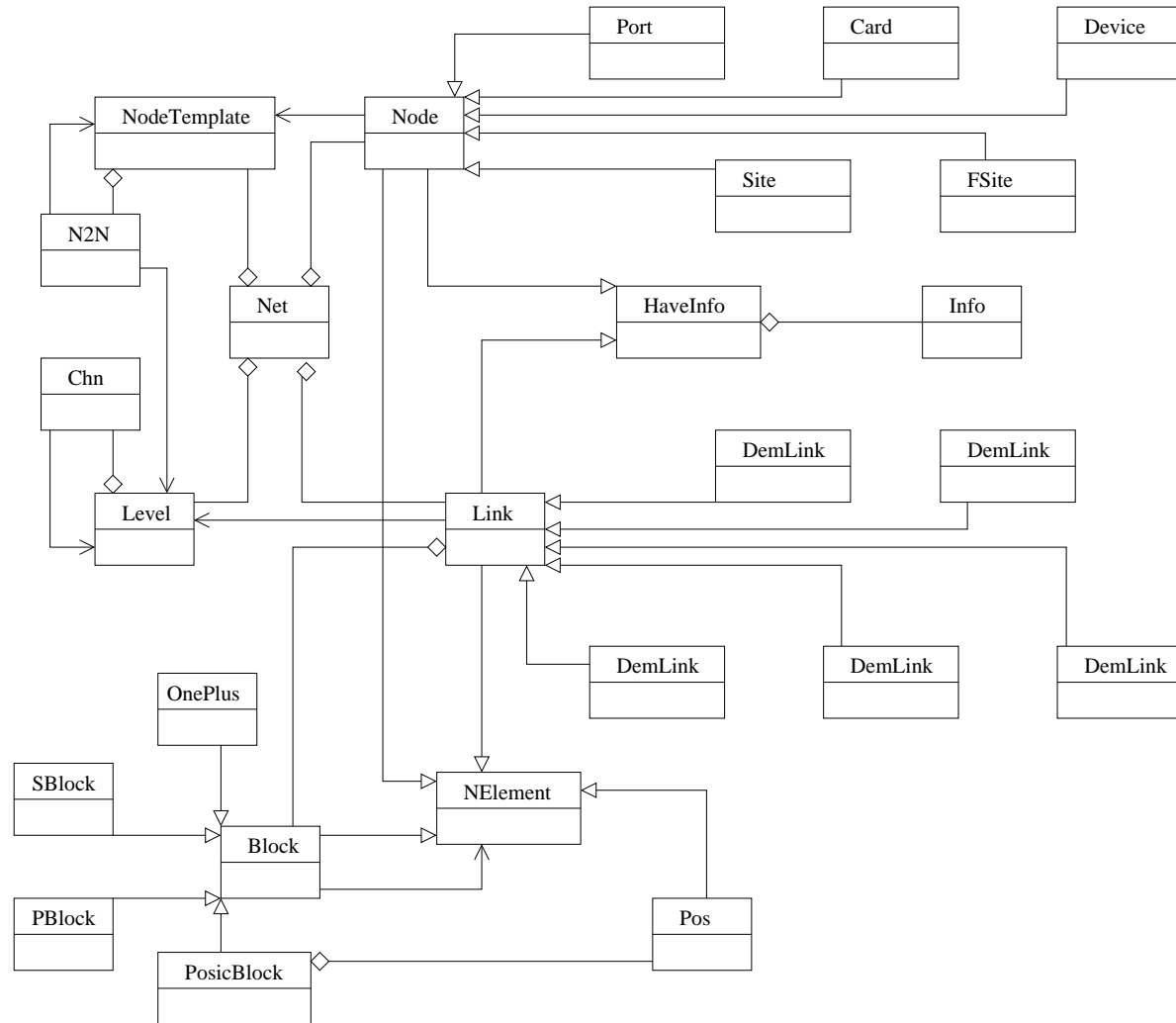


Flexible network model

- Main requirements for the netmodel:
 - multi-layer network architectures have to be supported,
 - flexible extension of the network description with new layers and technologies has to be provided,
 - network information has to be stored in a design oriented manner,
 - beside the network links, modeling node equipments and their internal configuration is required.
- The basic concept is a layered graph.
- Beside the node and link type network entities the model defines also rules for storing structural connection options:
 - link rules that define the layers and their connectivity opportunities,
 - node rules that identify the equipment, cards, ports etc. and their connectivity opportunities,
 - node-link rules that describe the possible node-link connectivity opportunities.
- Flexibility is provided by
 - Object oriented structure, where new entity types can be inserted in the model easily due to the inheritance options.
 - Generic hierarchy concept.
 - Using the additional information set that can be added to the entities.

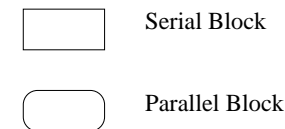
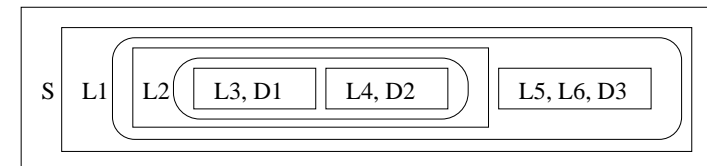
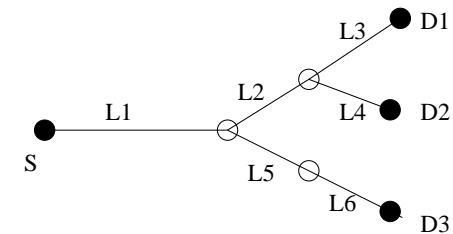


Components of the model



Generic hierarchy concept

- The direct server elements realizing a network link are listed in a *Block* object called the **Extension** of the link.
- The main idea of the generic structure is to store beside Links also Nodes and other embedded Blocks in the Extension. This is implemented by using the NElement type as basis of inheritance.
- New types of blocks can be introduced for general and special networking schemes.
- An example: multicast demand



Summary

- There is a permanent need for planning and analysis of existing networks, provisioning of new demands, installing new components in the network, and study possible networking options.
- To fulfill all these requirements, a flexible design tool is needed, that is based on a flexible network model.
- FLEXPLANET, the presented modeling, design and analysis tool makes it possible to consider real size operating networks using existing technologies, while supports the introduction of future networking options.
- The main properties that provide the flexibility are:
 - the separation of the network model and the data structures, that the algorithms work on,
 - the object oriented design of the model,
 - the generic server-client hierarchy description realized with the concept of Extension.

