PSE Verkehrssimulation

Traffic Networks in XML

Michael Moltenbrey, Dirk Pflüger

16. November 2007

Content

- Current State and Drawbacks
- Traffic Networks
  - Nodes
  - Links
  - Turns
- Visualisation

Current State

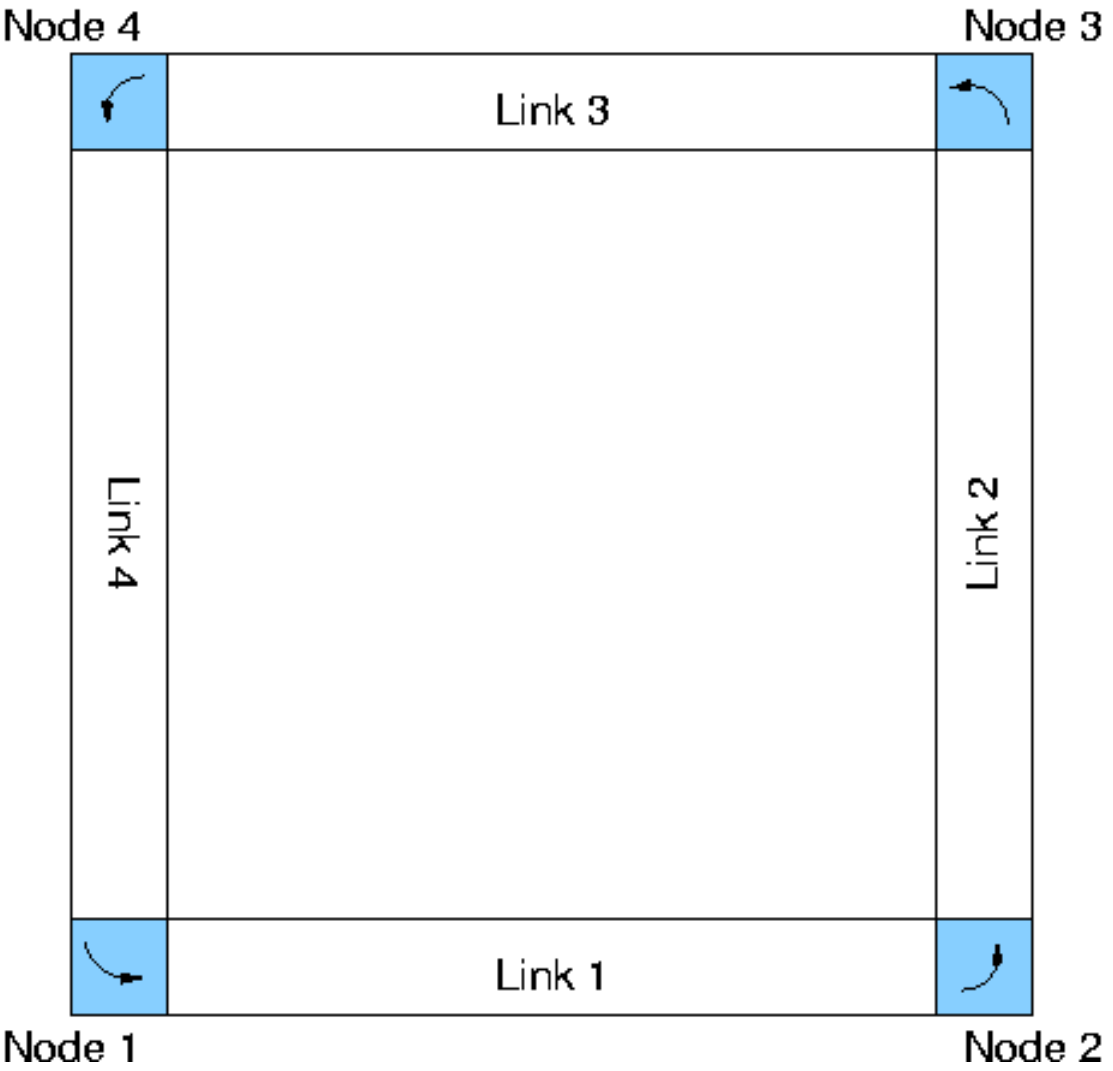
- Your traffic simulator is capable of simulating
  - Simple, hard-wired traffic netowrks
  - Textual and visual output of simulation results

- Drawbacks
  - Traffic networks are not flexible
    - No arbitrary networks
    - Changes in traffic network require re-compile
    - ...

Traffic Networks

- Traffic networks describe structure and interplay of street networks
- Several network components are used:
  - Nodes (Knoten)
  - Links (Strecken)
  - Turns (Abbiegebeziehungen)

Traffic Networks (2)



Traffic Networks – Nodes

- Nodes used to connect Links
- Nodes can represent intersections
- Each node has several parameters:
  - Unique identifier
  - Name
  - Coordinates (x,y) on simulation plane
- Parameters

```
<Node>
   <id>    </id>
   <name>  </name>
   <x>     </x>
   <y>     </y>
</Node>
```

Traffic Networks – Links

- Links represent streets or street segments
- Each Links starts and ends with a node
- Link parameters:
  - Unqiue identifier
  - Source Node
  - Destination Node
  - Length (in kilometres)
  - Number of lanes (fixed to 1 in our case)
  - Maximum velocity allowed on link (in km/h)

Traffic Networks – Links (2)

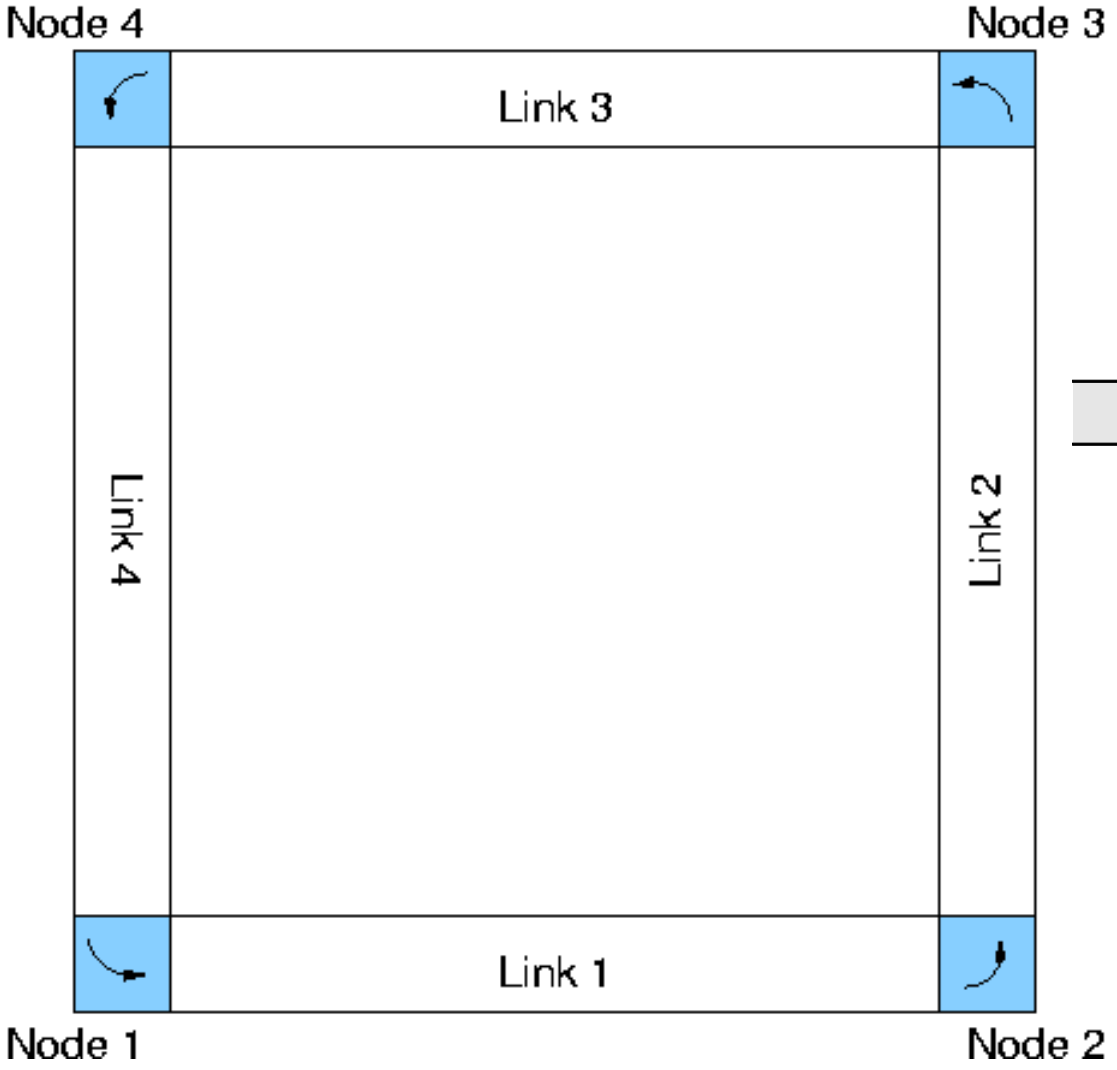
- Parameters

```
<Link>
   <id>                </id>
   <source>            </source>
   <destination>       </destination>
   <length>            </length>
   <lanes>             </lanes>
   <velocity>          </velocity>
</Link>
```

Traffic Networks – Turns

- Turns are crucial part for connecting Links
- In real life restrictions for making a turn at intersections
- Turns model these restrictions
- Parameters for Turns:
  - From
  - Via
  - To
- These three nodes are sufficient for modeling a Turn

Traffic Networks – Turns (2)



From	Via	To
1	2	3
2	3	4
3	4	1
4	1	2

Traffic Networks – Turns (3)

- Parameters

```
<Turn>
   <from>  </from>
   <via>   </via>
   <to>    </to>
</Turn>
```

Java und XML

- Scenarios, activities and networks are given in XML format
- Important to design and implement an appropriate *parser*
- Various generic XML-Parsers exist for Java
  - Can be adapted easily
- Two Types of Parsers:
  - Simple API for XML (SAX)
  - Document Object Model (DOM)
  - Both are part of Sun's Java 2 Plattform (since JDK 1.4)

Simple API for XML

- Based on Event Model
  - Sequential Reading of XML-File
  - Appropriate events are triggered
- Random access not possible
- More complex to implement compared to DOM

Document Object Model

- Object-oriented Representation of an XML-File
- DOM provides and API for altering, accessing the content of a file
- XML-Document is stored in a tree-like structure
- Access to elements easier compared to SAX
- Higher memory consumption

Questions?