Introduction to Mascopt: a library for graph manipulation

Bruno Bongiovanni, Jean-François Lalande
Michel Syska, Yann Verhoeven

Projet Mascotte, I3S-INRIA
Sophia-Antipolis, France

CRESCCO 6-8 december 2003

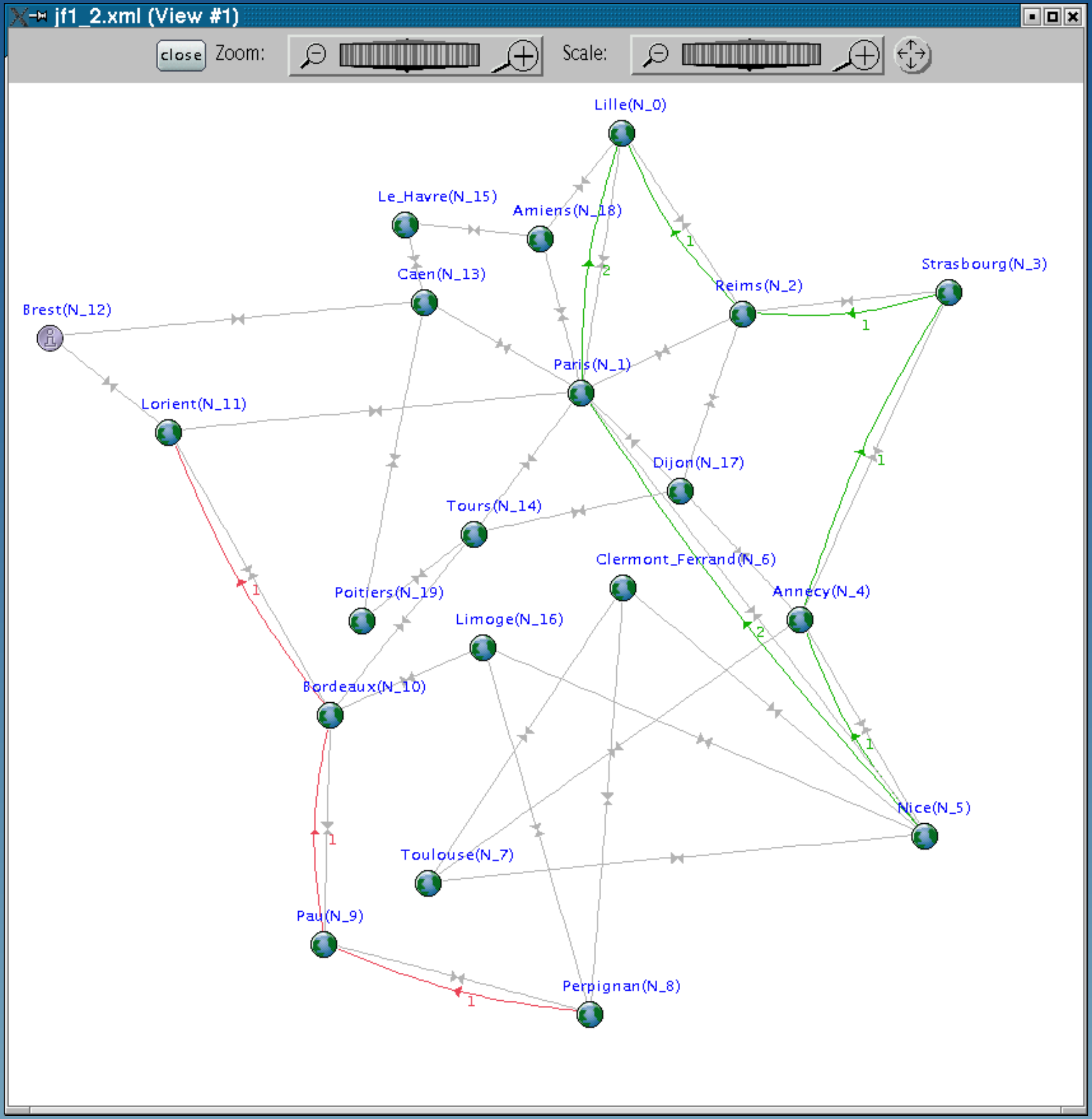
History

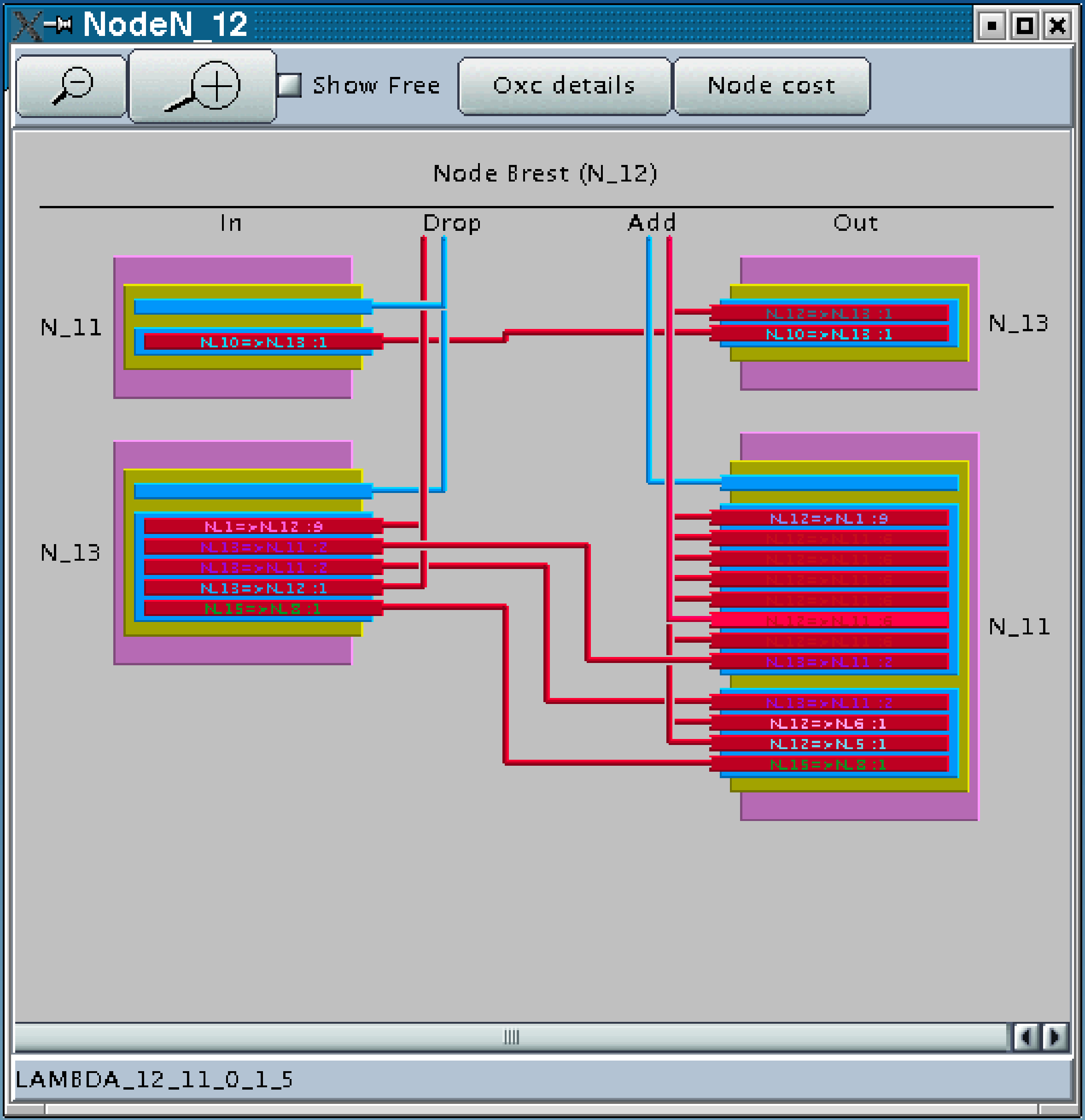
- Why Mascopt ?
- Our experience with WDM networks
  - ⋆ Porto project
  - ⋆ Alcatel and France Telecom R&D
  - ⋆ Routing static requests
  - ⋆ Grooming of the traffic
  - ⋆ Protection against failures of cables
- What we have learnt
  - ⋆ Program dedicated to WDM networks
  - ⋆ C++ code / Java GUI

Comments:

Mascopt started just after the end of the RNRT Porto project. Porto was dedicated to solve some problems such as routing, grooming, or protection problems on WDM networks. As the code was very specialized, it was not easily reusable for other kinds of networks. The code was also split between C++ and Java languages which generates interaction and portability problems.

The Porto software





LAMBDA_12_11_0_1_5

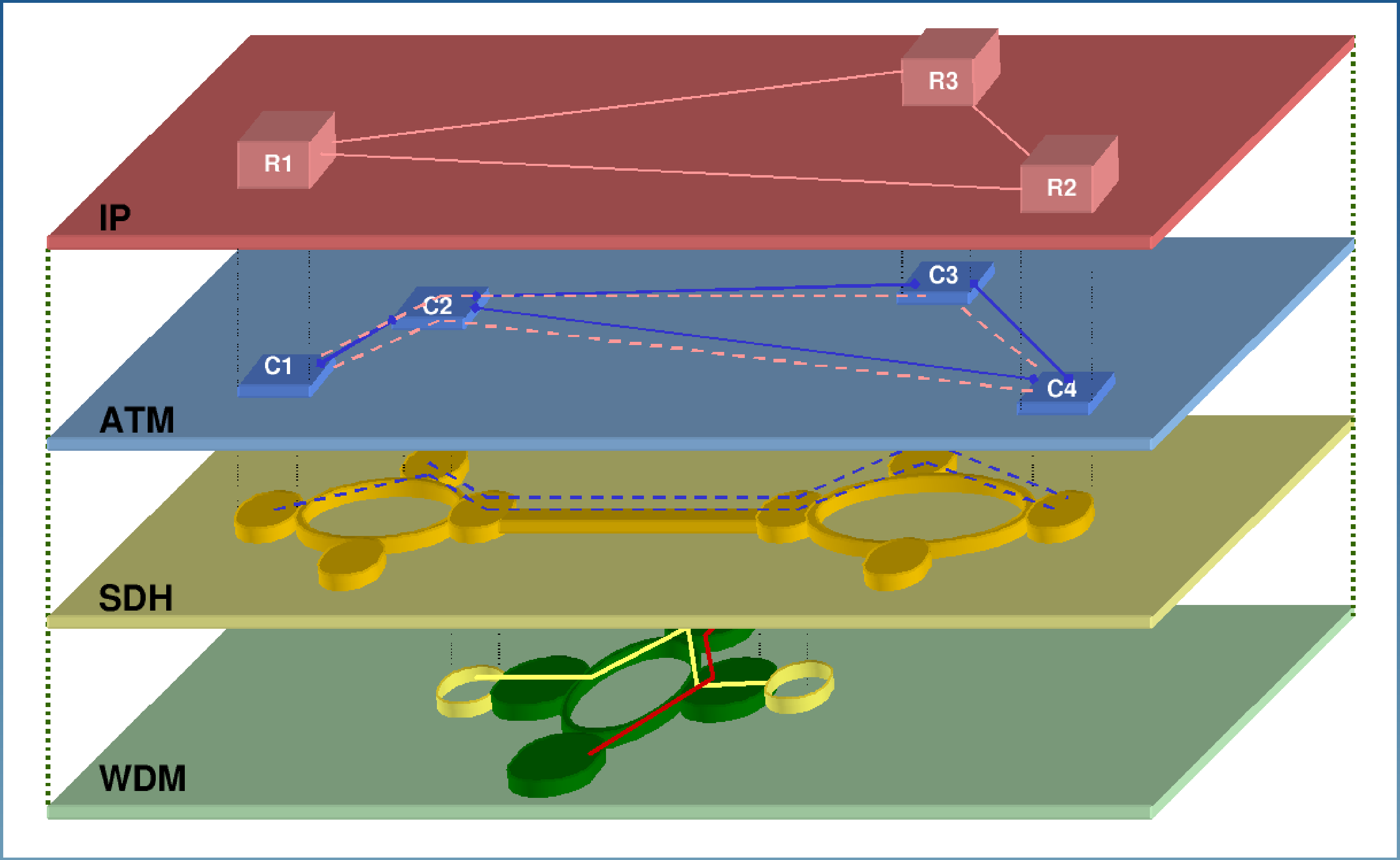
Comments:

These two screenshots show how the data managed by the Porto software is dedicated to WDM networks. The second one shows the lightpaths crossing a node and changing of fibers.

The third drawing shows the different types of networks we want to deals with.

The mascopt code is designed to be able to represent different kinds of networks, from ATM to IP networks.

Networks



Mascopt objectives

- Be a set of tools for network optimization
- Be a multiplatform tool/library
- Be as modular as possible
- Have I/O facilities, Graphical User Interface

Mascopt objectives

- Be a set of tools for network optimization
- Be a multiplatform tool/library
- Be as modular as possible
- Have I/O facilities, Graphical User Interface

- To have an efficient tool box for our experiments
- We do not want to restart from scratch for every new numerical experiment

- No existing free library fitting these requirements

Mascopt Graph package

- The first step: a working graph package

Mascopt Graph package

- The first step: a working graph package

- Graph / Directed Graph
- Set of vertices, Set of edges/arcs
- Path / Directed Path

Mascopt Graph package

- The first step: a working graph package

- Graph / Directed Graph
- Set of vertices, Set of edges/arcs
- Path / Directed Path

- Sharing of objects
- G1=(V,E1), G2=(V,E2)

Comments:

The graph package implements the share of objects. The previous example constructs two graphs G1 and G2 with the same nodeset V. The removing of nodes in V affects directly the two graphs G1 and G2.

Example

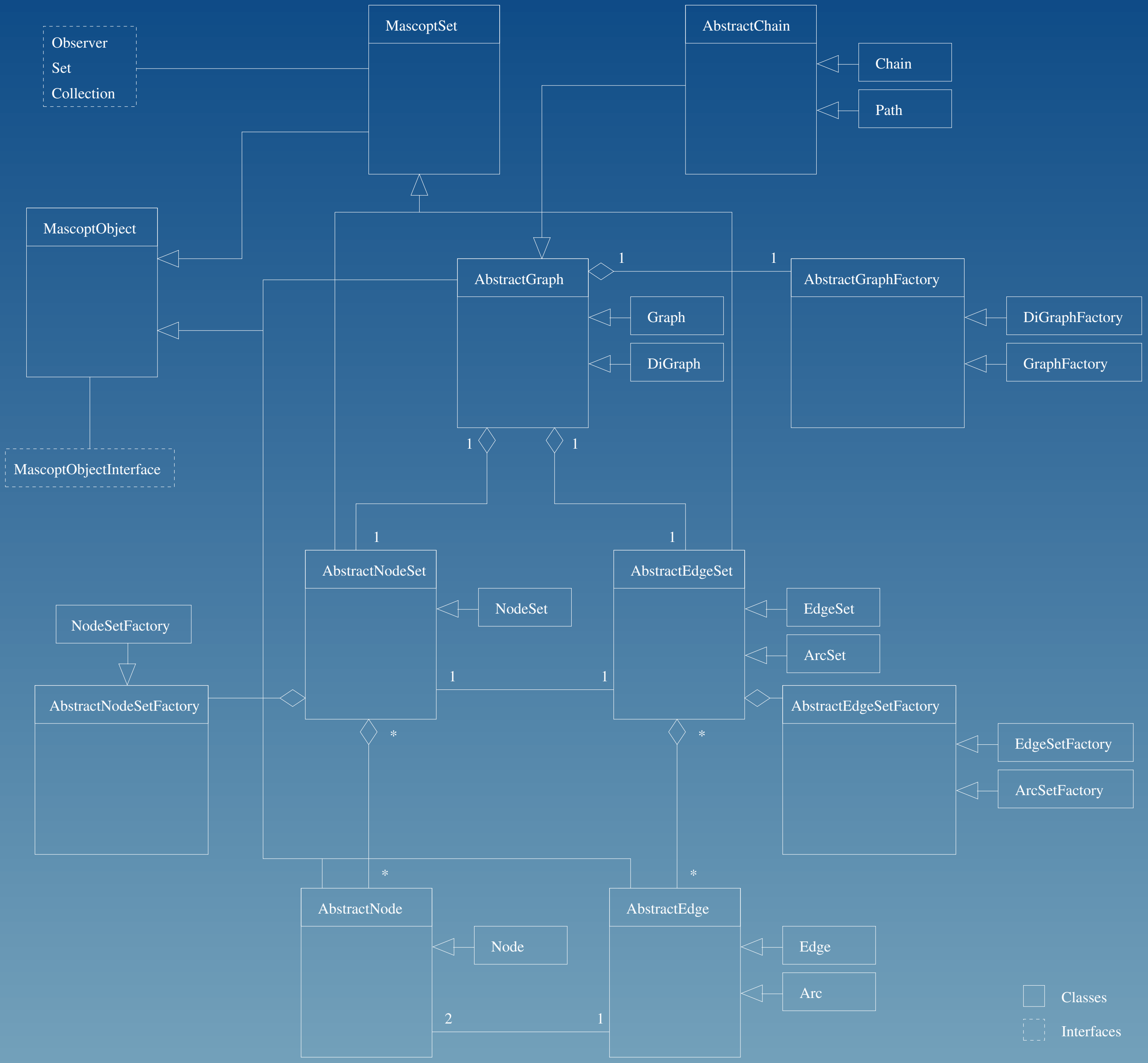
```
Vertex n1 = new Vertex();
Vertex n2 = new Vertex();
Vertex n3 = new Vertex();

VertexSet vs = new VertexSet();
vs.add(n1); vs.add(n2); vs.add(n3);

Edge e1 = new Edge(n1,n2);
Edge e2 = new Edge(n1,n3);

EdgeSet es1 = new EdgeSet(vs);
EdgeSet es2 = new EdgeSet(vs);
es1.add(e1); es2.add(e2);

Graph g1 = new Graph(vs, es1);
Graph g2 = new Graph(vs, es2);
```


Observer
Set
Collection
MascoptSet
AbstractChain
Chain
Path
MascoptObject
AbstractGraph
1
1
AbstractGraphFactory
Graph
DiGraph
DiGraphFactory
GraphFactory
MascoptObjectInterface
1
1
1
1
AbstractNodeSet
AbstractEdgeSet
NodeSet
EdgeSet
ArcSet
NodeSetFactory
1
1
AbstractNodeSetFactory
AbstractEdgeSetFactory
*
*
EdgeSetFactory
ArcSetFactory
*
*
AbstractNode
AbstractEdge
Node
Edge
Arc
2
1
Classes
Interfaces

Features

- Guaranty of valid data
- Existing Basic algorithms
  - ⋆ (k) Shortest paths
  - ⋆ Minimum spanning tree
  - ⋆ Multicommodity flow
  - ⋆ Breadth/Depth first search
  - ⋆ To be continued ...
- Valuation System (String, Integer, Double)
- Interfaced with ILOG Cplex LP solver API

- We provide services: what is its quality ?

Features

- Time access vs Memory requirement

Features

- Time access vs Memory requirement
- Time to get the set of neighbors of vertex v1: $O(1)$
- A bad case: algorithm adding and removing many edges

Features

- Time access vs Memory requirement
- Time to get the set of neighbors of vertex v1: $O(1)$
- A bad case: algorithm adding and removing many edges
- Access to a value on an edge or vertex: $O(\text{number of graphs})$
- A bad case: algorithm using a lot of graphs

Features

- Time access vs Memory requirement
- Time to get the set of neighbors of vertex v1: $O(1)$
- A bad case: algorithm adding and removing many edges
- Access to a value on an edge or vertex: $O(\text{number of graphs})$
- A bad case: algorithm using a lot of graphs
- Removing a node from graph G=(V,E): update of E in time $O(log|E|)$

Features

- Time access vs Memory requirement
- Time to get the set of neighbors of vertex v1: $O(1)$
- A bad case: algorithm adding and removing many edges
- Access to a value on an edge or vertex: $O(\text{number of graphs})$
- A bad case: algorithm using a lot of graphs
- Removing a node from graph G=(V,E): update of E in time $O(log|E|)$
- The implementation implies some choices

Comments:
The library provides services and information which is built in the main classes. Each data which can be asked, can be constructed at run time or already built and available for the user. The first possibility needs some time when the user needs the data. The second possibility needs more memory requirement to be able to keep all the data in memory. According to the best choices we did to implement each service, some algorithms could run slowly if it exploits the bad cases.

XML I/O

```
<?xml version="1.0" ?>
<!DOCTYPE OBJECTS SYSTEM "ftp://ftp-sop.inria.fr/mascotte/mascopt/
                          dtd/mgl_v1.1.dtd">

<OBJECTS>
<VERTICES>
        <VERTEX id="N11">
                <POSITION>
                        <X>74.0</X>
                        <Y>60.0</Y>
                </POSITION>
        </VERTEX>
        <VERTEX id="N10">
                <POSITION>
                        <X>47.0</X>
                        <Y>232.0</Y>
                </POSITION>
        </VERTEX>
```

Comments:
This is a part of an MGL file, the i/o format for storing graphs. This XML file can be extended easily (as adding a Z coordinate) without breaking all the structure of the file.

Conclusion

- http://www-sop.inria.fr/mascotte/mascopt
- Google + Mascopt
- GPL
- Michel Syska, Yann Verhoeven

- Network packages
- Experiments
- Switch to demo ...