

Documentation for repository **Visualisation**

Created by Manuel Schrick
2018 - June - 13th

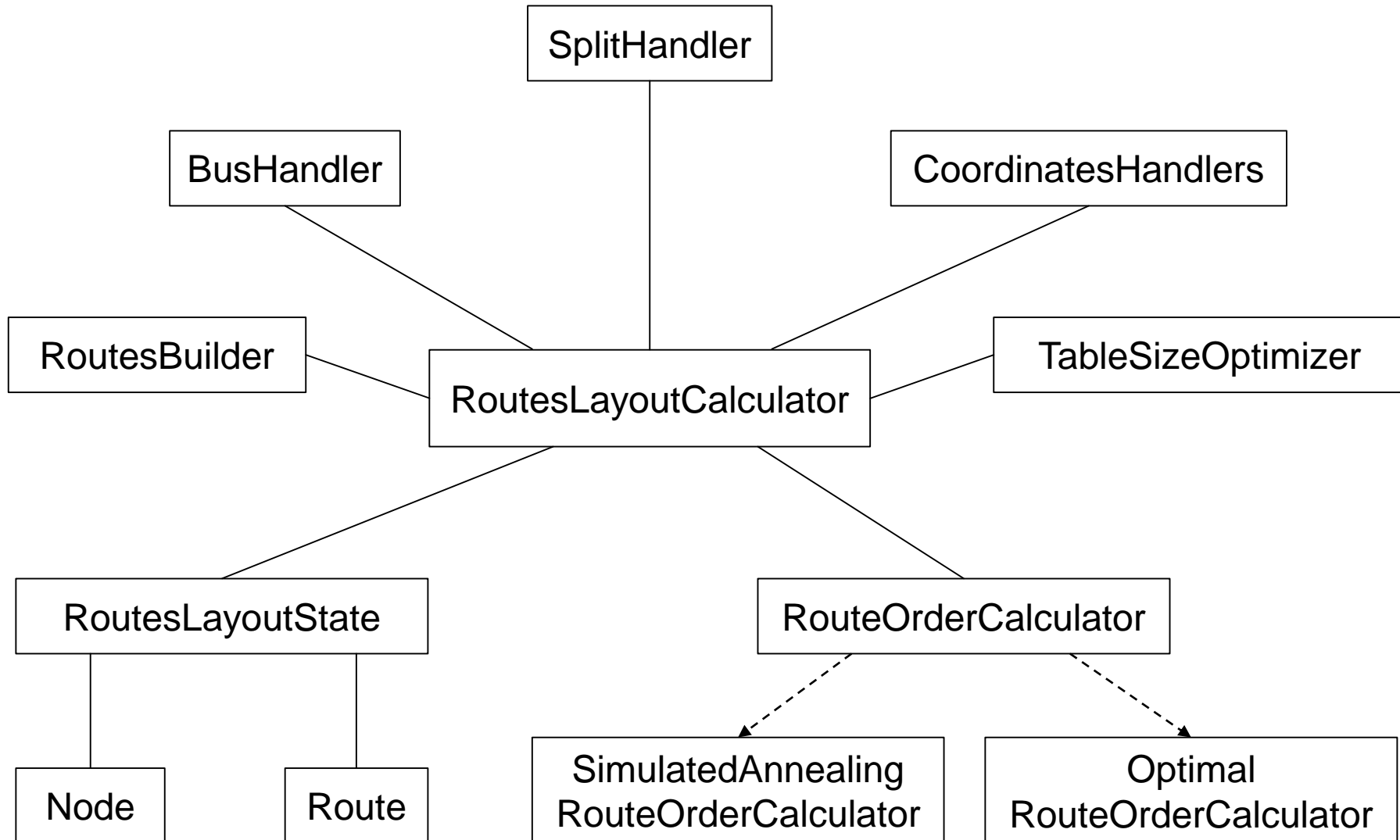
Purpose

- Automatically generate graphical layout for EmbeddedMontiArc models
- Generated Layouts:
 - can be embedded into the Online IDE
 - allow users to navigate through the different levels of the model
 - IDE opens the respective source file automatically
 - allow users to view the model with different depth of detail

Most Important Classes

- RoutesLayoutCalculator
- RoutesLayoutState
- RoutesBuilder
- SimulatedAnnealingRouteOrderCalculator
- TableSizeOptimizer
- Handlers:
 - BusHandler
 - SplitHandler
 - CoordinatesHandlers
- Node
- Route

Class Relations



RoutesLayoutCalculator

- Connects all parts of the algorithm
- Is called by the main class
- Calls parts in the correct order (see “Algorithm”-slide)

RoutesLayoutState

- Represents the current state of the layout algorithm
- Contains (excerpt):
 - Topmost component of model
 - List of all nodes (in initial order)
 - List of all routes (in current order)
 - List of buses
- Is passed through most parts of the algorithm

RoutesBuilder

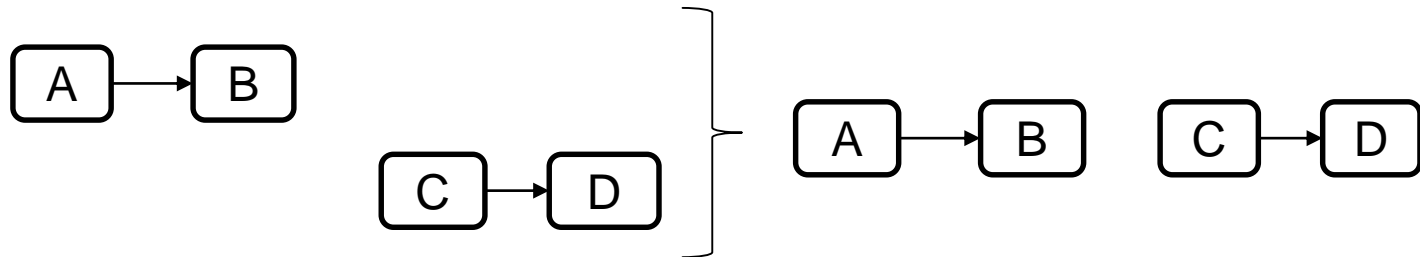
- Translates the EmbeddedMontiArc model into a graph data type (set of nodes and a set of routes) such that:
 - Every component is represented by one node (two in case of the enclosing component)
 - Every route contains one path in the graph
 - If two paths share one or two nodes these nodes are either a start or an end node of at least one of the paths
- Uses depth-first-search

SimulatedAnnealingRouteOrderCalculator

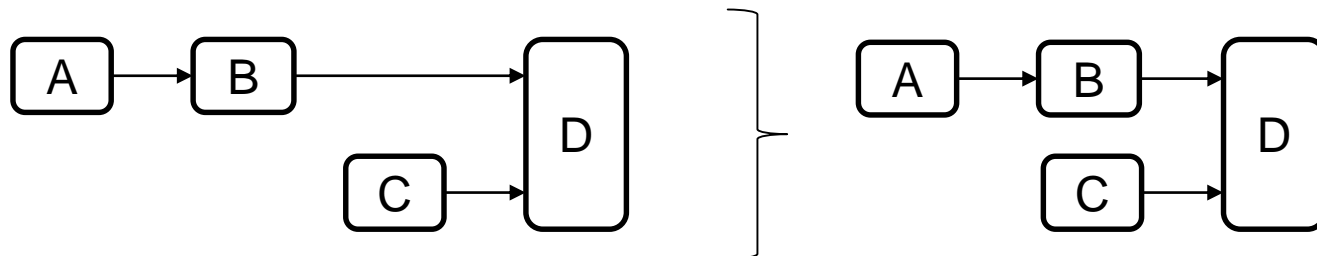
- Takes a set of routes and calculates a vertical order such that:
 - as few connectors as possible intersect
 - components are as small as possible
 - as few components as possible are crossed by a connector
- Uses Simulated Annealing ([wiki](#))

TableSizeOptimizer

- Reduces the overall size of the table layout
- Merges two rows if possible:

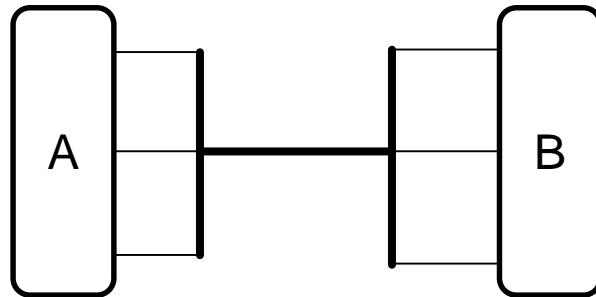


- Merges to columns if possible:



BusHandler

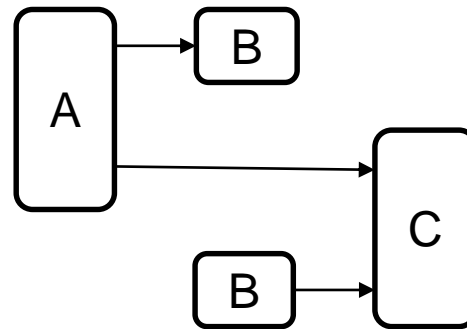
- Multiple connectors connecting the same source and target component are only once in the graph data type
- BusHandler adds missing connections and forms a bus with them:



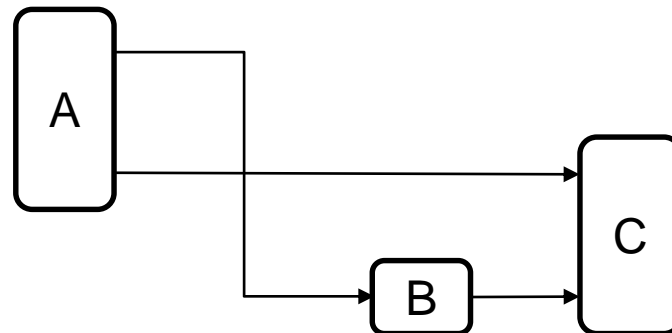
- Connections are sometimes not formed into a bus but displayed separately (see function: `shouldDecomposeBus()`)

SplitHandler

- Sometimes components end up split after the Simulated Annealing step:



- SplitHandler repairs these components by adding vertical lines:



CoordinatesHandlers

- Assign coordinates to model elements
- ComponentCoordinatesHandler
 - Width = width of column
 - Height = sum height of rows
- PortCoordinatesHandler
 - Input ports: left edge of component column, center of row
 - Output ports: right edge of component column, center of row
- ConnectorCoordinatesHandler
 - Path from source to target component
 - Considers buses, cycles, split nodes

Node & Route

- Represent the underlying graph of the model
- Node:
 - Represents a component
 - Enclosing component represented by two nodes:
 - ROOT_IN (input side)
 - ROOT_OUT (output side)
- Route
 - Represents a row in the layout
 - Contains connections displayed in that row

Algorithm

1. Translate EmbeddedMontiArc model (see RoutesBuilder)
2. Calculate route order (see SimulatedAnnealingRouteOrderCalculator)
3. Insert missing bus routes (see BusHandler)
4. Reduce overall height of table layout (see TableSizeOptimzier)
5. Repair split nodes (see SplitHandler)
6. Reduce overall width of table layout (see TableSizeOptimzier)
7. Assign coordinates to:
 - a. Components (see ComponentCoordinatesHandler)
 - b. Ports (see PortCoordinatesHandler)
 - c. Connectors (see ConnectorCoordinatesHandler)
8. Generate output files with FreeMarker templates
(see e.g. ComponentGenerator)

Tests (excerpt)

- HTMLBuilderTest
 - Tests whether .html output file is generated
- HTMLBuilderTest
 - Tests whether .html output file is generated
- BaseLayoutTest
 - Tests whether all components, ports and connectors are drawn
- BusHandlerTest
 - Tests whether buses are inserted correctly
- SplitHandlerTest
 - Tests whether split nodes are repaired correctly
- VerticalLinesTest
 - Tests whether buses and splits are displayed with vertical connector sections

Command Line Interface

- Sample call:

```
java -jar jar/svggenerator.jar  
      --input testManuelSchrick.thesisExample.thesisExample  
      --modelPath models/
```

- Options:

--input: The package of the model

--modelPath: The directory to the root folder where the model is stored

--outputPath: The path where the files should be stored (default: /output)

--recursiveDrawing: If “true” all subcomponents are drawn (default: false)

--onlineIDE: If set, click-events are handed to the IDE

(navigation might not work outside of the IDE)

Known Issues

- Ports without connectors are not drawn
- Some layouts are too large, because of overuse of buses
- Breadcrumb navigation is too wide if component names are very long

Code Quality

- Code Quality(according to codeclimate.com): C
- Test Coverage:
 - Total: 85%
 - Handwritten code: 85%
 - Generated code: (There is no code generated into target folder)