Visualisierung von Datenstrukturen für Graphen-Layout-Algorithmen

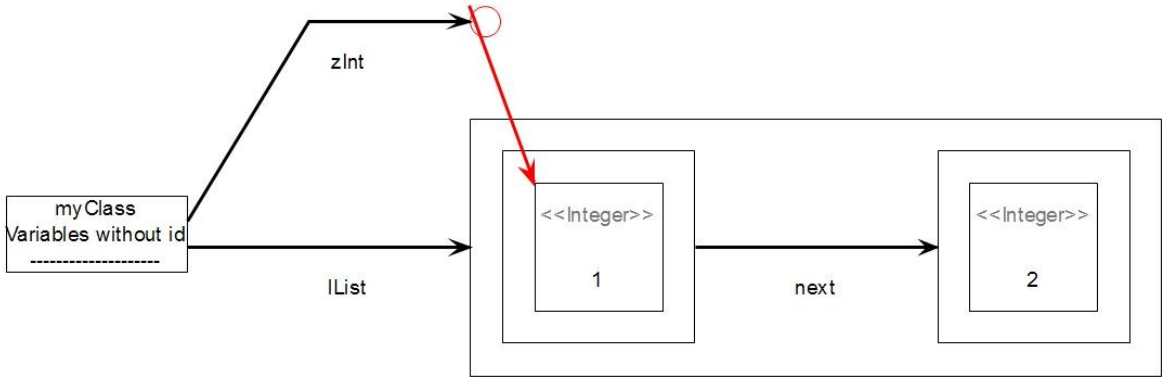
Tibor Toepffer

Department of Computer Science
Christian-Albrechts-Universität zu Kiel

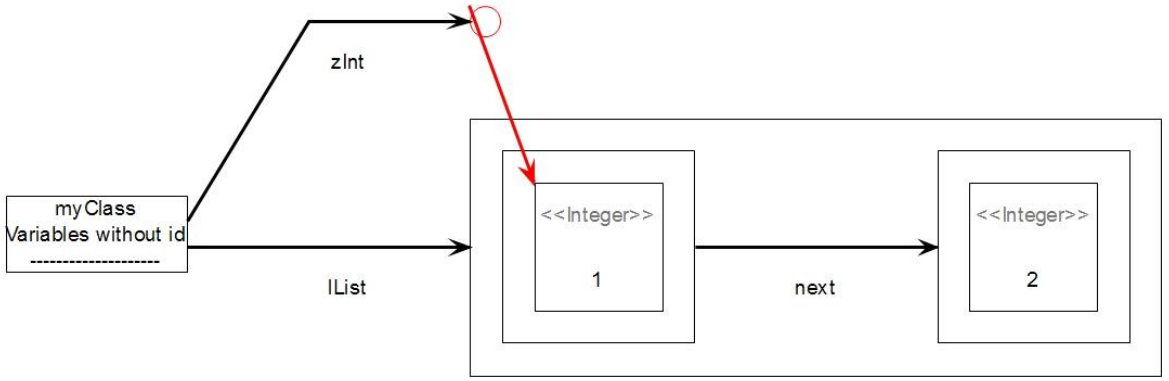
12. Februar 2013

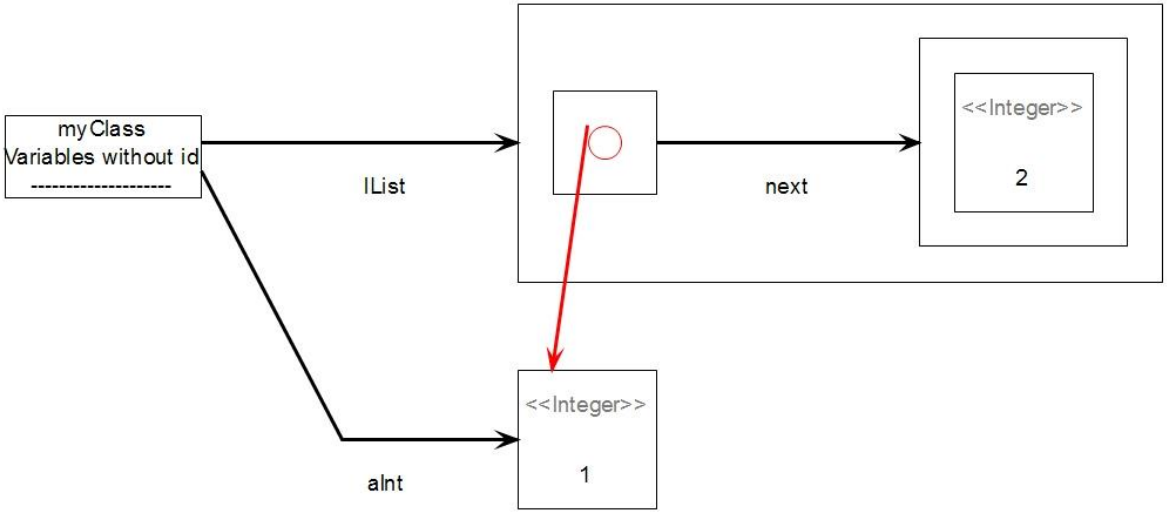
Agenda

Gleiches ist nicht gleich

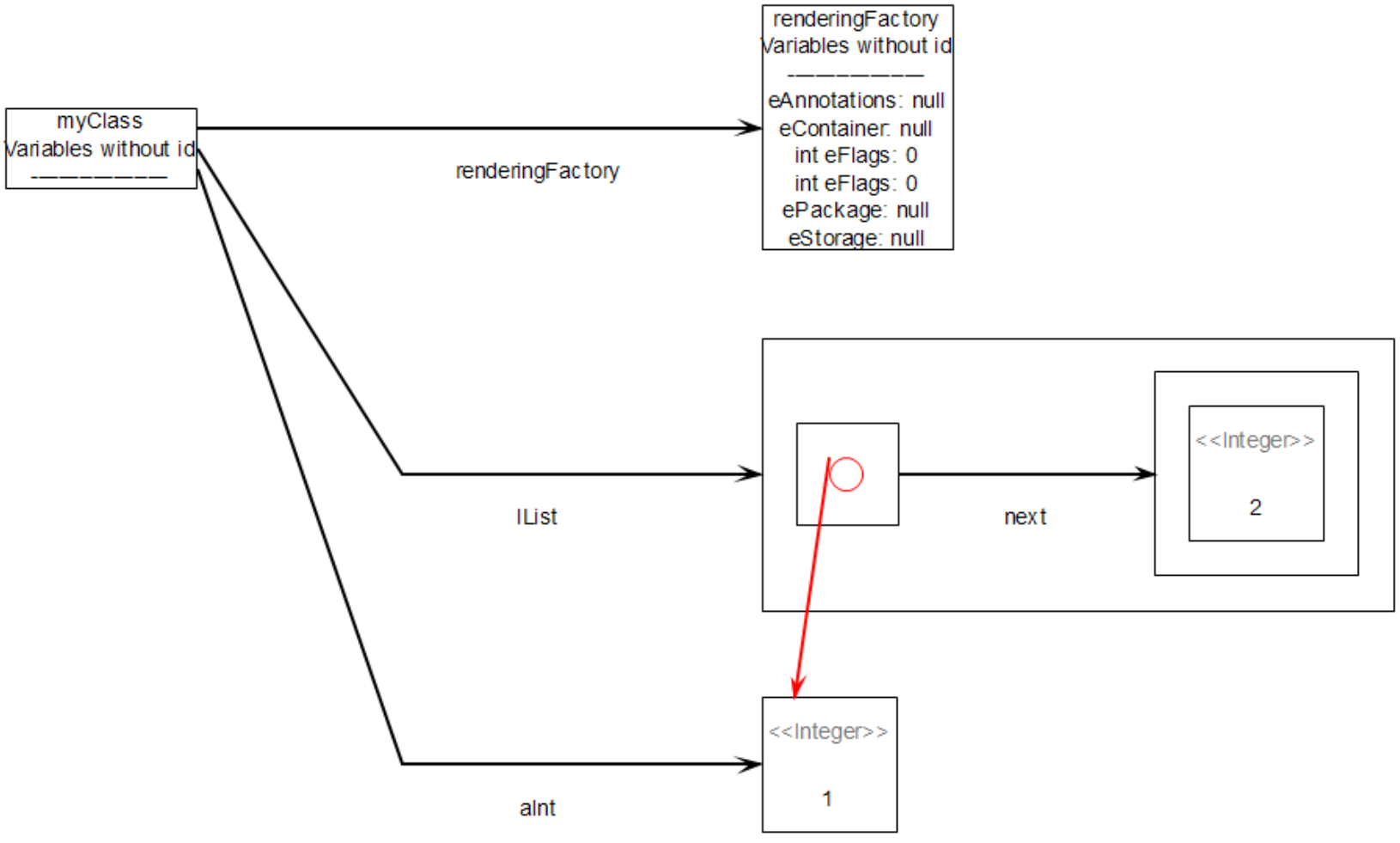


Gleiches ist nicht gleich

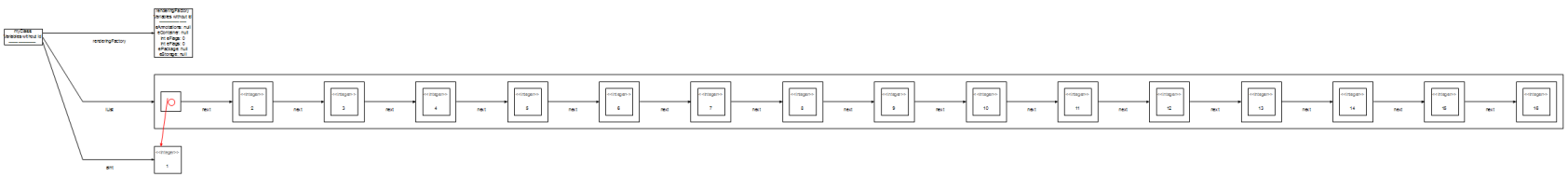




Darstellung irrelevanter Daten



Darstellung großer Datenmengen



Agenda

Graphen-Layout-Algorithmen

Einige der Arbeitsgruppe Echtzeitsysteme und Eingebettete Systeme implementierte Graphen-Layout-Algorithmen:

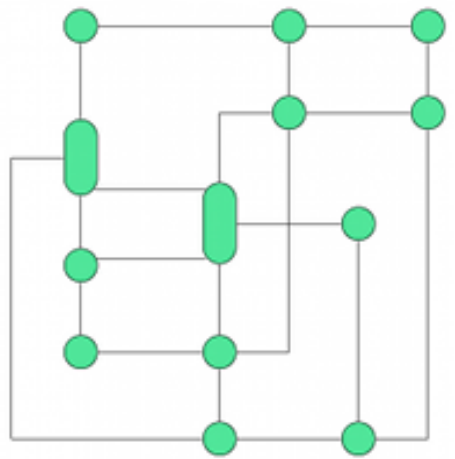
- Planarisation - PGraph [1]
- Kräftebasiert - FGraph [2]
- Ebenenansatz - LGraph [3]

[1] nach John Boyer und Wendy Myrvold, 2004
[2] Unterstütze Modelle: Eades (1984) sowie Fruchterman-Reingold (1991)
[3] Kozo Sugiyama, Shojiro Tagawa, und Mitsuhiko Toda, 1981

Graphen-Layout-Algorithmen

Einige der Arbeitsgruppe Echtzeitsysteme und Eingebettete Systeme implementierte Graphen-Layout-Algorithmen:

- Planarisation - PGraph [1]
- Kräftebasiert - FGraph [2]
- Ebenenansatz - LGraph [3]

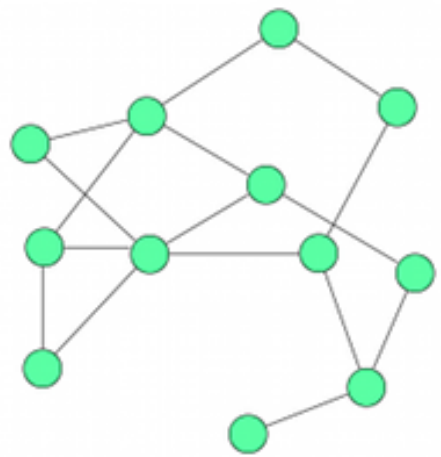


[1] nach John Boyer und Wendy Myrvold, 2004
[2] Unterstütze Modelle: Eades (1984) sowie Fruchterman-Reingold (1991)
[3] Kozo Sugiyama, Shojiro Tagawa, und Mitsuhiko Toda, 1981

Graphen-Layout-Algorithmen

Einige der Arbeitsgruppe Echtzeitsysteme und Eingebettete Systeme implementierte Graphen-Layout-Algorithmen:

- Planarisation - PGraph [1]
- Kräftebasiert - FGraph [2]
- Ebenenansatz - LGraph [3]

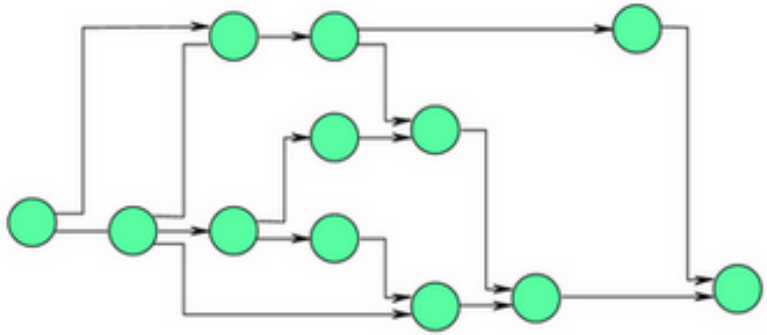


[1] nach John Boyer und Wendy Myrvold, 2004
[2] Unterstütze Modelle: Eades (1984) sowie Fruchterman-Reingold (1991)
[3] Kozo Sugiyama, Shojiro Tagawa, und Mitsuhiko Toda, 1981

Graphen-Layout-Algorithmen

Einige der Arbeitsgruppe Echtzeitsysteme und Eingebettete Systeme implementierte Graphen-Layout-Algorithmen:

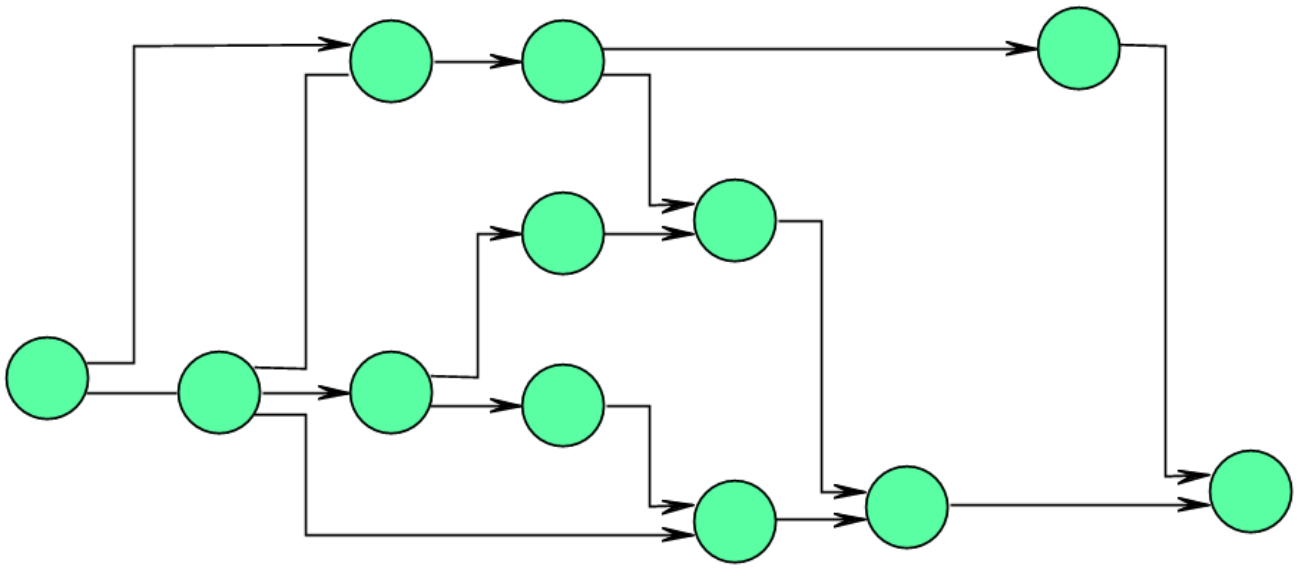
- Planarisation - PGraph [1]
- Kräftebasiert - FGraph [2]
- Ebenenansatz - LGraph [3]



[1] nach John Boyer und Wendy Myrvold, 2004
[2] Unterstütze Modelle: Eades (1984) sowie Fruchterman-Reingold (1991)
[3] Kozo Sugiyama, Shojiro Tagawa, und Mitsuhiko Toda, 1981

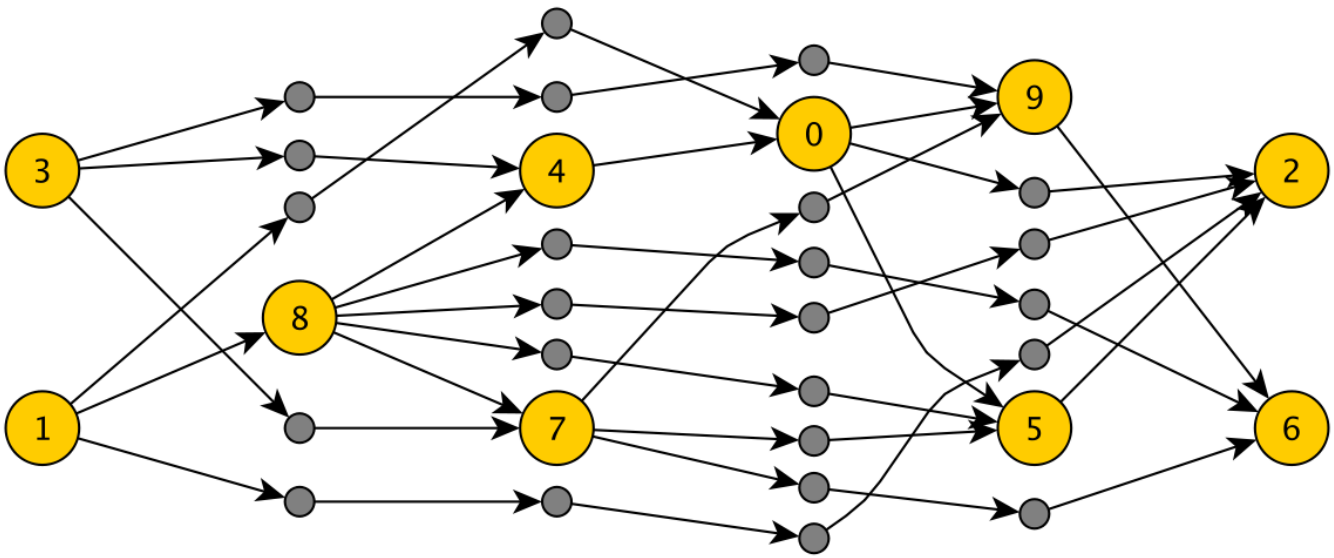
Vorstellung des LGraph

Intern genutztes Graphenformat eines Algorithmus zum Layout von Graphen nach dem Ebenenansatz von Kozo Sugiyama et al.



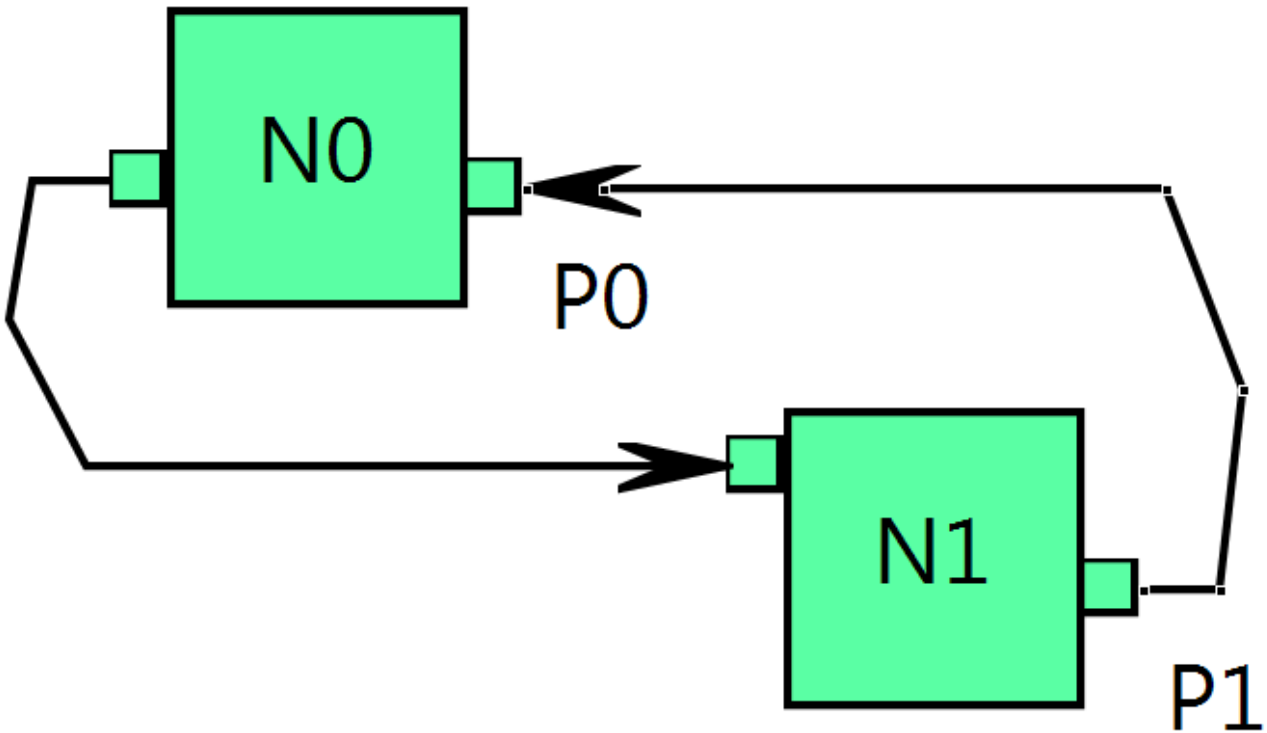
Hilfsmittel Dummy-Nodes

Wärend der Layout-Phase werden Dummy-Nodes eingefügt.



[3] Grafik aus dem Confluence der Arbeitsgruppe Echtzeitsysteme und Eingebettete Systeme

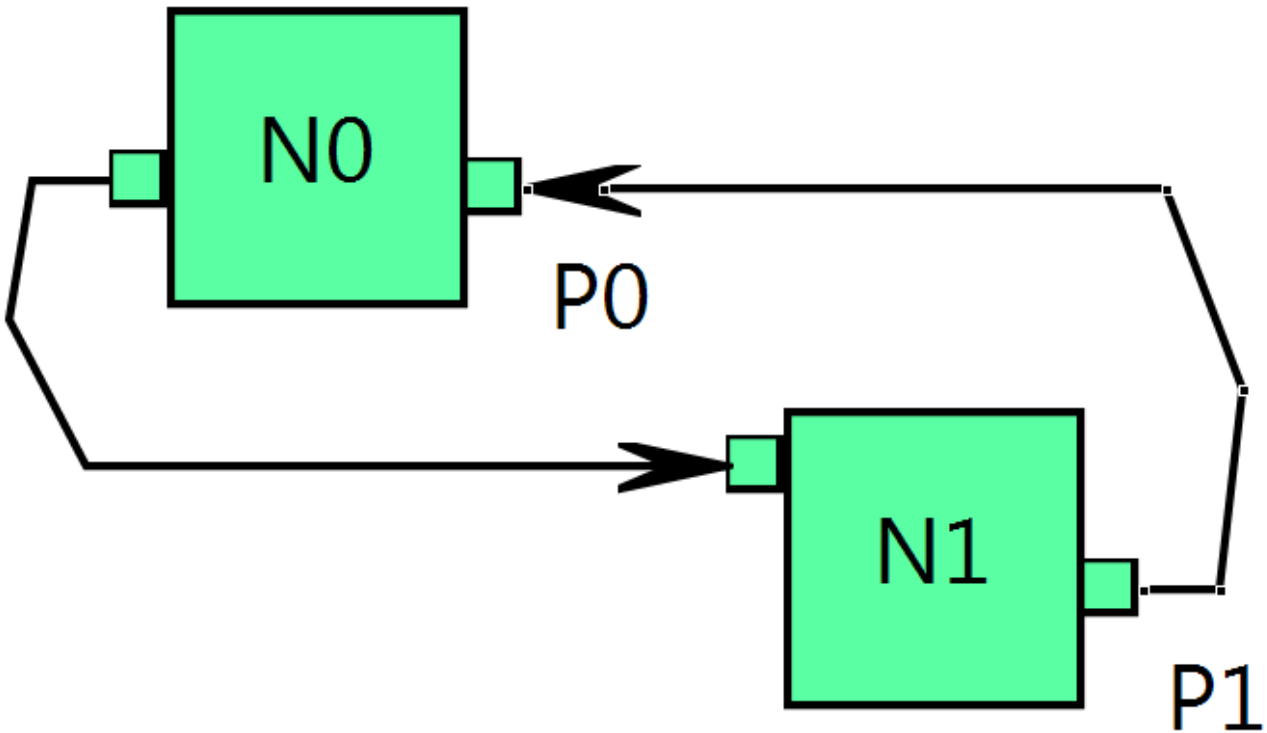
Einfacher Beispielgraph



Wärend des Layouts (zum Zeitpunkt der Betrachtung) werden

- in jede Kante ein Dummy-Node eingefügt sein
- sich je ein echter und ein Dummy-Knoten auf einem Layer befinden

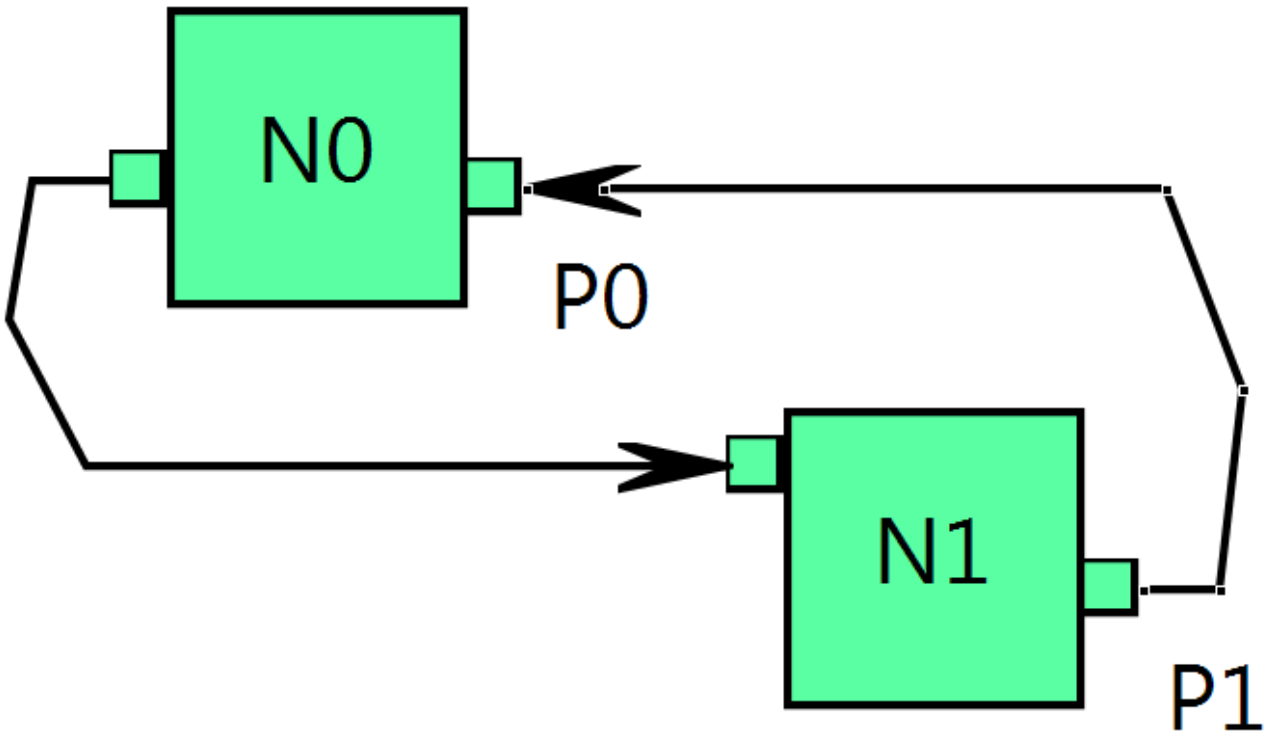
Einfacher Beispielgraph



Wärend des Layouts (zum Zeitpunkt der Betrachtung) werden

- in jede Kante ein Dummy-Node eingefügt sein
- sich je ein echter und ein Dummy-Knoten auf einem Layer befinden

Einfacher Beispielgraph



Wärend des Layouts (zum Zeitpunkt der Betrachtung) werden

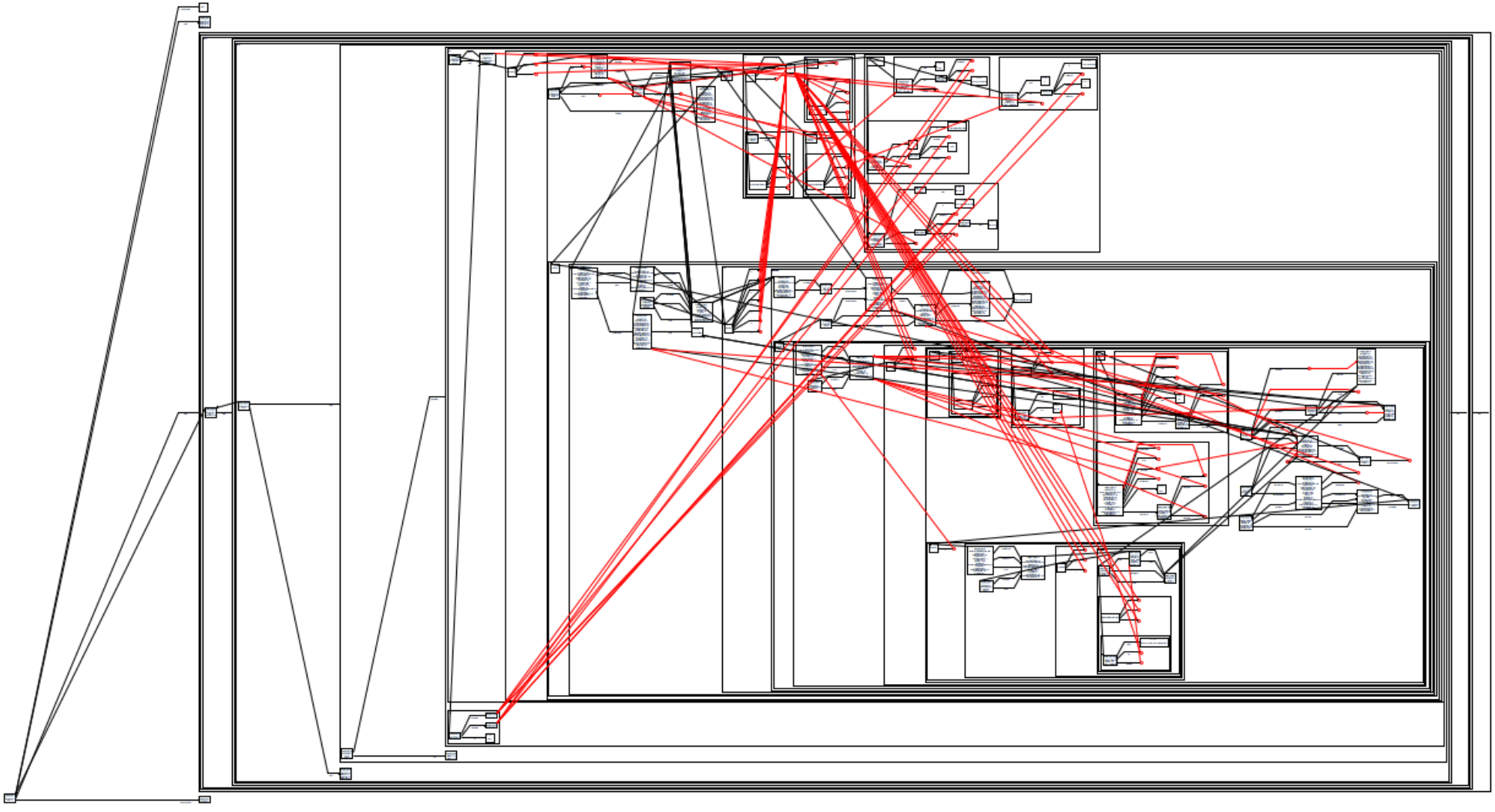
- in jede Kante ein Dummy-Node eingefügt sein
- sich je ein echter und ein Dummy-Knoten auf einem Layer befinden

Darstellung in der Variables View

(x)= Variables Breakpoints

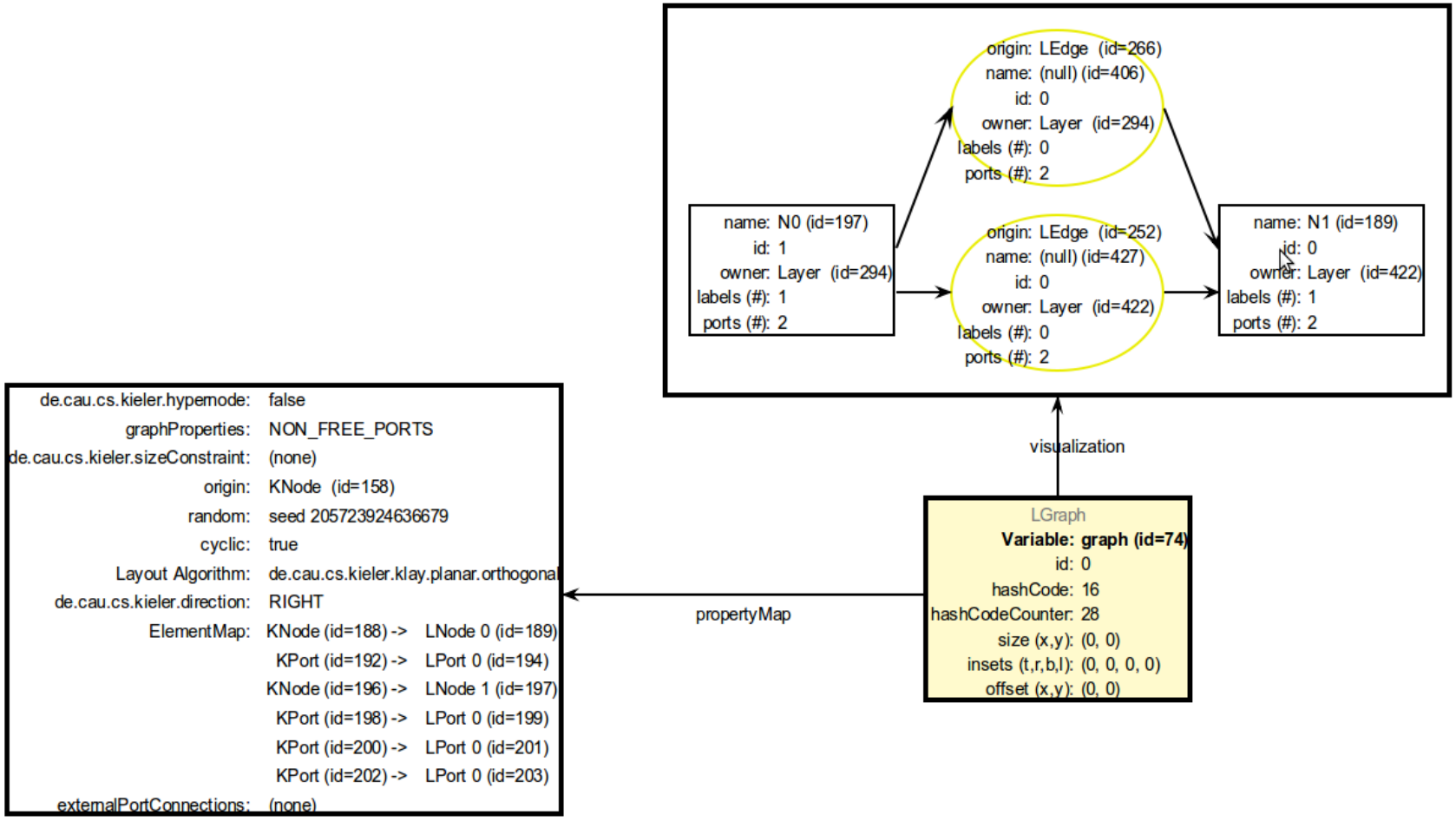
Name	Value
graph	LGraph (id=88)
hashCode	16
hashCodeCounter	LGraphElement$HashCod...
id	0
insets	LInsets$Double (id=108)
layerlessNodes	LinkedList<E> (id=111)
layers	LinkedList<E> (id=113)
[0]	Layer (id=141)
hashCode	17
id	0
nodes	LinkedList<E> (id=143)
[0]	LNode (id=157)
hashCode	8
id	1
insets	LInsets$Double (id=167)
labels	LinkedList<E> (id=168)
margin	LInsets$Double (id=169)
owner	Layer (id=141)
ports	LinkedList<E> (id=170)
[0]	LPort (id=175)
anchor	KVector (id=178)
anchor	KVector (id=178)
hashCode	9
id	0
incomingEdges	LinkedList<E> (id=179)
labels	LinkedList<E> (id=180)
margin	LInsets$Double (id=181)
outgoingEdges	LinkedList<E> (id=182)
[0]	LEdge (id=190)
bendPoints	KVectorChain (id=192)
hashCode	14
id	0
labels	LinkedList<E> (id=194)
propertyMap	HashMap<K,V> (id=195)
source	LPort (id=175)
target	LPort (id=196)
anchor	KVector (id=197)
hashCode	26
id	0
incomingEdges	LinkedList<E> (id=198)
labels	LinkedList<E> (id=199)
margin	LInsets$Double (id=200)
outgoingEdges	LinkedList<E> (id=202)
owner	LNode (id=203)

Darstellung mittels Standardtransformation



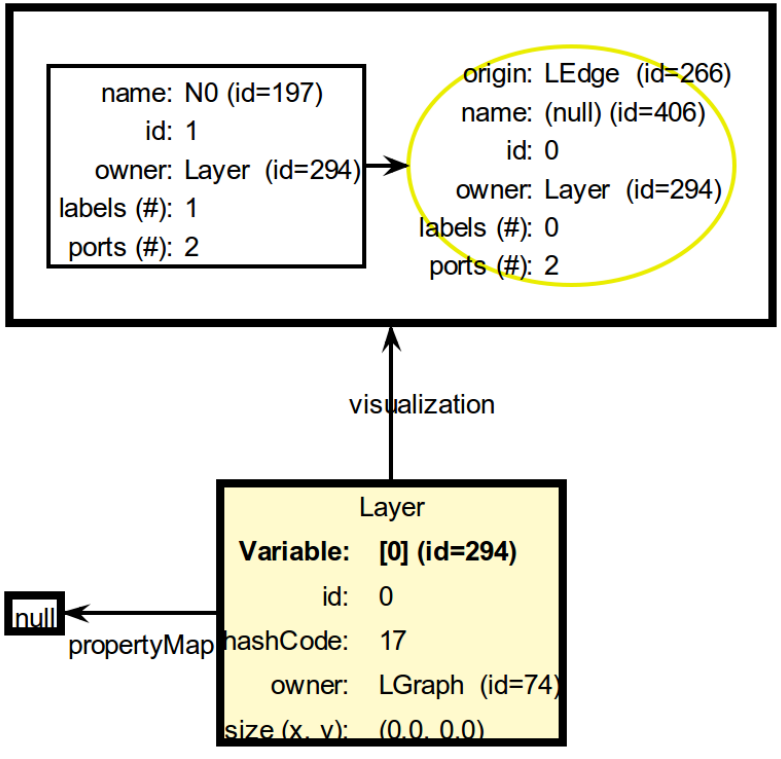
Darstellung mittels angepasster Transformationen

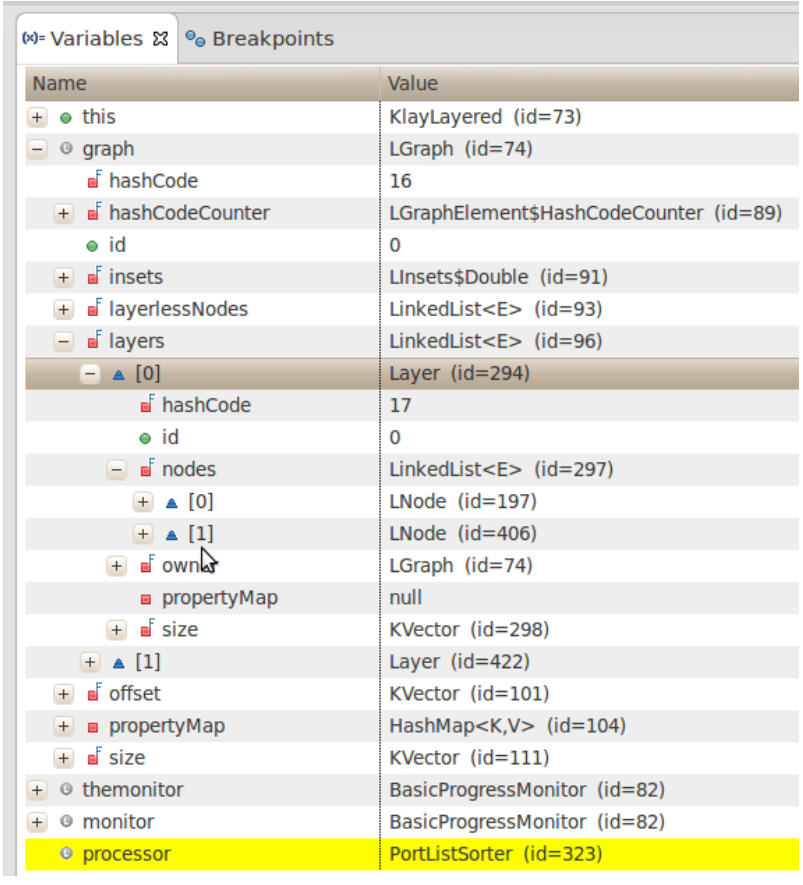
LGraph



LGraph mit angepasster Transformation

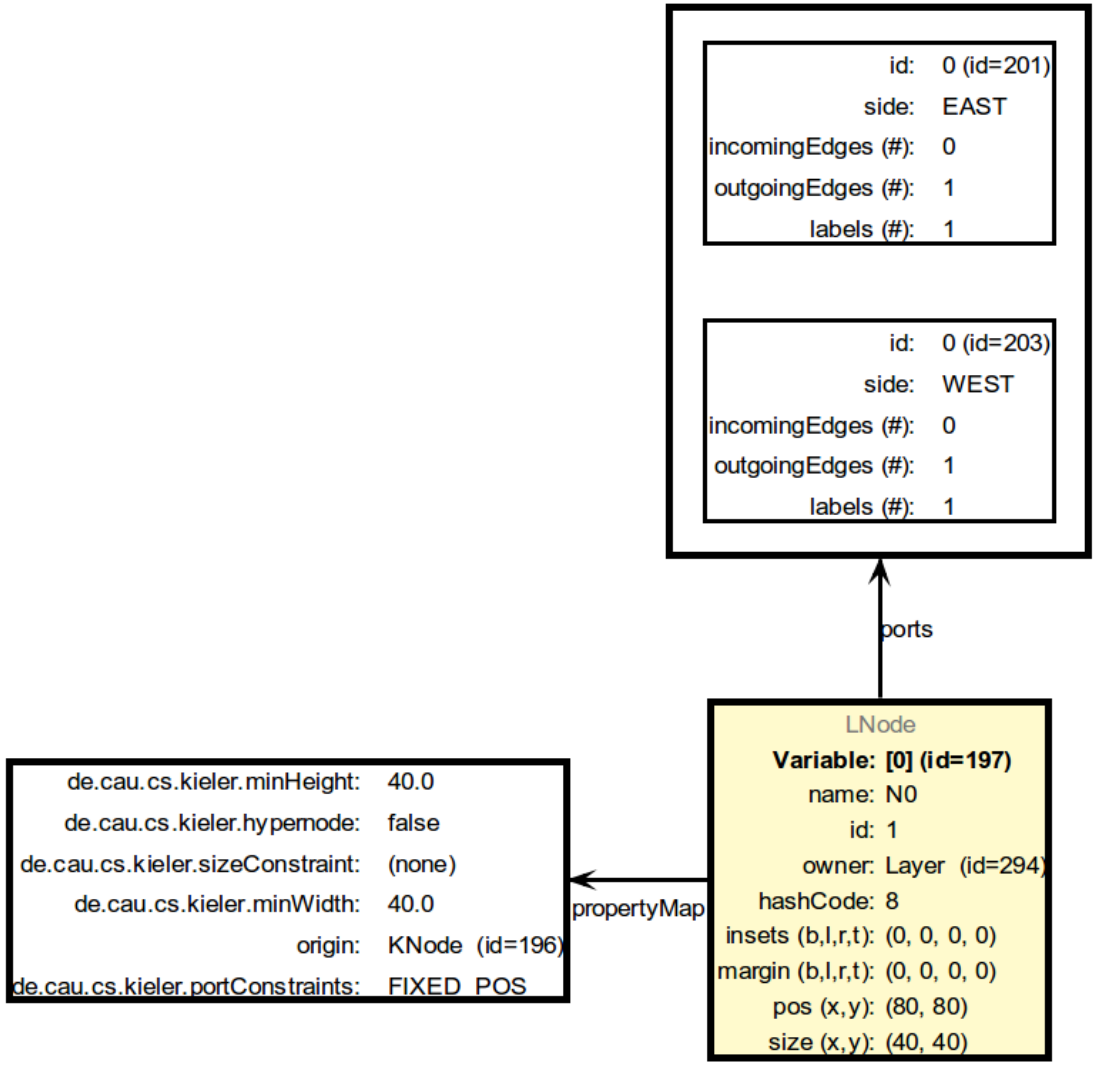
Layer





LGraph mit angepasster Transformation

LNode



Agenda

Vorteile der Visualisierung der Semantik

Individuelle Transformation für eigene Datenstrukturen ermöglicht:

- Beschränkung auf relevante Daten
- Bündeln von Informationen nach logischen Zusammenhängen
- Kodierung von Informationen in Formen und Farben
- Bedingte Formatierung

Vorteile der Visualisierung der Semantik

Individuelle Transformation für eigene Datenstrukturen ermöglicht:

- Beschränkung auf relevante Daten
- Bündeln von Informationen nach logischen Zusammenhängen
- Kodierung von Informationen in Formen und Farben
- Bedingte Formatierung

Vorteile der Visualisierung der Semantik

Individuelle Transformation für eigene Datenstrukturen ermöglicht:

- Beschränkung auf relevante Daten
- Bündeln von Informationen nach logischen Zusammenhängen
- Kodierung von Informationen in Formen und Farben
- Bedingte Formatierung

Vorteile der Visualisierung der Semantik

Individuelle Transformation für eigene Datenstrukturen ermöglicht:

- Beschränkung auf relevante Daten
- Bündeln von Informationen nach logischen Zusammenhängen
- Kodierung von Informationen in Formen und Farben
- Bedingte Formatierung

Vorteile der Visualisierung der Semantik

Individuelle Transformation für eigene Datenstrukturen ermöglicht:

- Beschränkung auf relevante Daten
- Bündeln von Informationen nach logischen Zusammenhängen
- Kodierung von Informationen in Formen und Farben
- Bedingte Formatierung

Nachteile und Ausblick

- Nachteile der eigenen Transformationen
  - Aufwand eigene Transformationen zu schreiben
  - Welche Informationen sind wirklich wichtig?
- Ausblick und weiter Arbeit
  - Benutzerinteraktion mit der Grafik
  - Bereitstellung einer konfortablen Utility-Klasse

Nachteile und Ausblick

- Nachteile der eigenen Transformationen
  - Aufwand eigene Transformationen zu schreiben
  - Welche Informationen sind wirklich wichtig?
- Ausblick und weiter Arbeit
  - Benutzerinteraktion mit der Grafik
  - Bereitstellung einer konfortablen Utility-Klasse

Nachteile und Ausblick

- Nachteile der eigenen Transformationen
  - Aufwand eigene Transformationen zu schreiben
  - Welche Informationen sind wirklich wichtig?
- Ausblick und weiter Arbeit
  - Benutzerinteraktion mit der Grafik
  - Bereitstellung einer konfortablen Utility-Klasse

Nachteile und Ausblick

- Nachteile der eigenen Transformationen
  - Aufwand eigene Transformationen zu schreiben
  - Welche Informationen sind wirklich wichtig?
- Ausblick und weiter Arbeit
  - Benutzerinteraktion mit der Grafik
  - Bereitstellung einer konfortablen Utility-Klasse

Nachteile und Ausblick

- Nachteile der eigenen Transformationen
  - Aufwand eigene Transformationen zu schreiben
  - Welche Informationen sind wirklich wichtig?
- Ausblick und weiter Arbeit
  - Benutzerinteraktion mit der Grafik
  - Bereitstellung einer konfortablen Utility-Klasse

Nachteile und Ausblick

- Nachteile der eigenen Transformationen
  - Aufwand eigene Transformationen zu schreiben
  - Welche Informationen sind wirklich wichtig?
- Ausblick und weiter Arbeit
  - Benutzerinteraktion mit der Grafik
  - Bereitstellung einer konfortablen Utility-Klasse

End

Any questions?