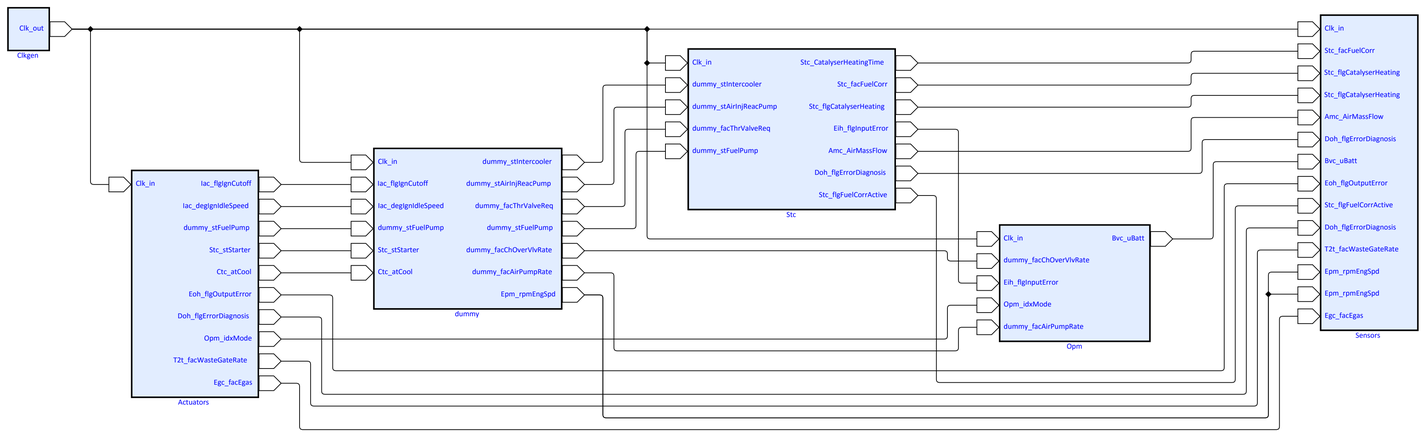
Edge Bundling
for Dataflow Diagrams

Kiel University
Faculty of Engineering
Department of Computer Science

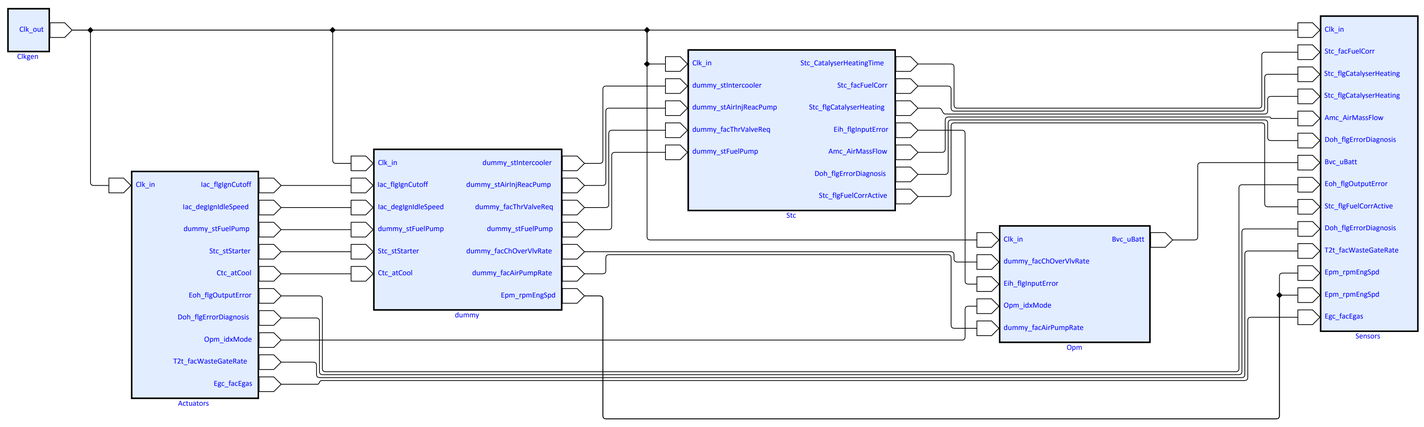
Input Graph

The large number of connections makes the diagram hard to read. Often only the connectivity between pairs of nodes is of interest in order to understand the flow of data through the system.



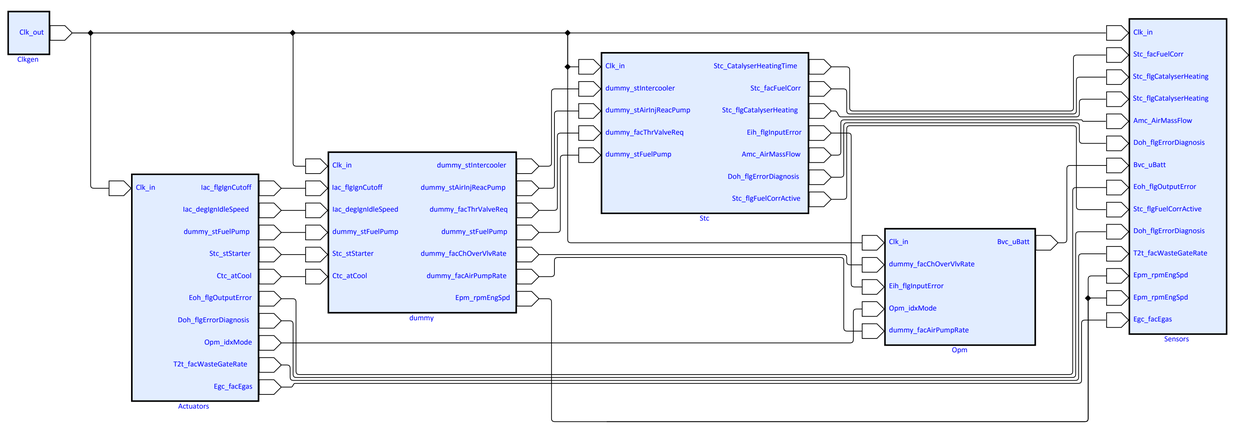
Bundled Edges

Drawing edges close to each other if they connect the same pair of nodes tidies up the diagram significantly. Node positions are not changed to preserve the user's mental map. Additionally, it allows keeping bundled and unbundled edges within the same diagram. The user can interactively un-bundle an edge within an editor and inspect the specific connections.



Bundled Edges & Compacted Node Positions

The space freed by bundling the edges can be used to compute new, more compact positions for nodes.



Formalization

Dataflow diagrams are *graphs* with *nodes* and *edges*. Edges are connected to nodes via *ports*.

Graph

$G = (V, P, \pi, E)$
$\pi : P \mapsto V$
$E \subseteq 2^P$

Hyperedge

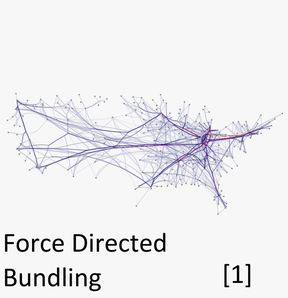
Given $e \in E$
$|e| = 2$: e is called *simple* edge
$|e| > 2$: e is called *hyperedge*

Edge Bundle

Given nodes $s, t \in V$, an *edge bundle* b is a set of edges such that
$\forall e \in b \; \exists \{p, q\} \subseteq e \; : \; \pi(p) = s \wedge \pi(q) = t$

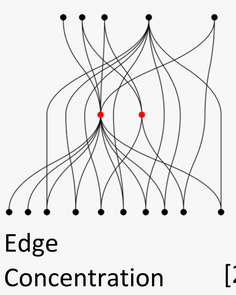
An edge can only be part of one bundle.
If $\forall e \in b : |e| = 2$, b is called *simple* bundle, i.e. it does not contain hyperedges.

Selected Literature



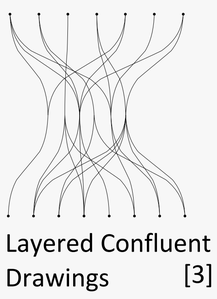
Force Directed Bundling [1]

Edges are modeled as flexible springs attracting one another. Smooth, easy to follow bundles are desired.
Unambiguity is negligible, the primary goal is to expose high-level connectivity.



Edge Concentration [2]

Based on the layer-based approach, bicliques between adjacent layers are replaced with concentration nodes. The goal is to minimizie the number of edges and/or the number of edge crossings. Concentration nodes are placed in newly added layers.



Layered Confluent Drawings [3]

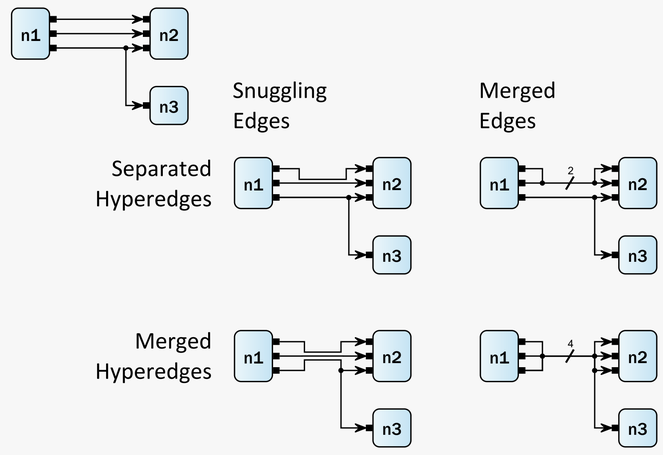
Contrary to the previous methods, bicliques are replaced by confluent tracks that are part of the edge routing. The existing layers are preserved.

Ink Minimization [4]

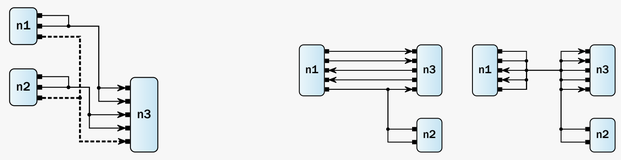
The method's input is the output of the layer-based approach. Edges are bundled if drawing them on top of each other saves ink. Within a bundle the edges are sorted and drawn a small distance apart to diminish ambiguity.

Visual Representation

Dataflow diagrams contain hyperedges that carry the same data. In contrast, edge bundles are a form of visual representation to combine otherwise unrelated edges. The question is how edge bundles and hyperedges can coexist.



Problematic Cases



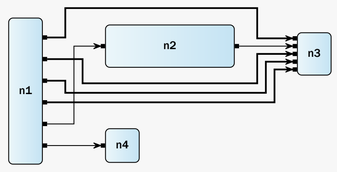
It is not clear with which bundle the hyperedge (dashed) should be combined.

The bundled drawing (right) looks confusing. It is not clear if there was a directed edge from **n2** to **n1**.

Methods

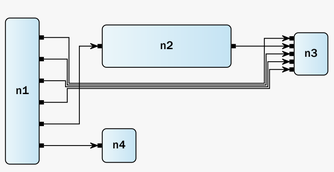
A drawing with orthogonally routed edges serves as starting point. It can be computed using the layer-based layout approach as proposed by Sugiyama et al. with several extensions.

1. Initial Drawing



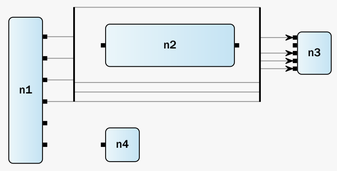
Positions for nodes and orthogonal edge routes have been calculated. Highlighted edges should be bundled.

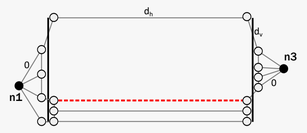
5. Final Drawing



Bundle paths have been derived from the existing edge routes.

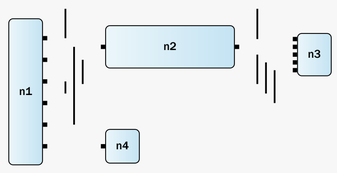
2. Select Horizontal Segments

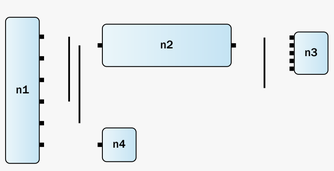




An auxilliary graph is used to find a feasible horizontal segment. The segment must be surrounded by enough space to accomodate the (possibly wider) bundle and the bundle's overall length should be short. A weighted shortest path based on pixel distances is used. In this example, the dashed segment is selected.

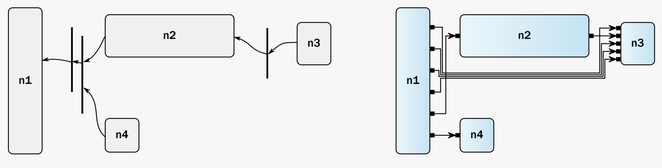
3. Order Vertical Segments





Vertical segments of edges only exist between pairs of layers. For the initial layout their order is chosen to minimize the number of edge crossings (left). We count crossings with bundled edges only once, hence a different order may be beneficial (right). Also, the height of vertical segments depends on the selected horizontal segments.

4. Position Vertical Segments and Nodes Compactly



Instead of re-using existing edge segment positions, horizontal compaction techniques can be used to find positions for the vertical segments. This optionally allows to calculate new node positions as well.

Contact Persons

Ulf Rüegg, Christoph Daniel Schulze, Carsten Sprung,
Nis Wechselberg, Reinhard von Hanxleden
Department of Computer Science, Kiel University
Olshausenstr. 40, 24098 Kiel, Germany
Phone: +49 (0) 431 880-7297
Fax: +49 (0) 431 880-7615
uru@ / cds@ / csp@ / nbw@ / rvh@informatik.uni-kiel.de
http://www.informatik.uni-kiel.de/rtsys

Literature

[1] Holten, D., & Van Wijk, J. J. (2009).
Force-Directed edge bundling for graph visualization.
Computer Graphics Forum, 28(3), 983–990.
[2] Onoue, Y., Kukimoto, N., Sakamoto, N., & Koyamada, K. (2016).
Minimizing the Number of Edges via Edge Concentration in Dense Layered Graphs.
IEEE Transactions on Visualization and Computer Graphics, 22(6), 1652–1661.
[3] Eppstein, D., Goodrich, M., & Meng, J. (2007).
Confluent layered drawings.
Algorithmica, 47(4), 439–452.
[4] Pupyrev, S., Nachmanson, L., & Kaufmann, M. (2010).
Improving layered graph layouts with edge bundling.
18th International Symposium on Graph Drawing, 329–340.

Project Information

ELK
Eclipse Layout Kernel

https://eclipse.org/elk
https://github.com/eclipse/elk

Poster presented at the 24th International Symposium on Graph Drawing and Network Visualization (GD 2016),