**Edge Bundling for Dataflow Diagrams**

**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 makes the diagram less confusing. 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 more compact positions for nodes.

**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 co-exist.

**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 n1 to n4.

**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.

2. **Select Horizontal Segments**
   An auxiliary graph is used to find a feasible horizontal segment. The segment must be surrounded by enough space to accommodate the (possibly) editing bundle and the bundle's overall length should be short. A weighted shortest path based on past 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, since 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 reusing 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.

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

**Selected Literature**

<table>
<thead>
<tr>
<th>Reference</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Force Directed Bundling</td>
</tr>
<tr>
<td>[2]</td>
<td>Layered Confluent Drawings</td>
</tr>
<tr>
<td>[3]</td>
<td></td>
</tr>
</tbody>
</table>

**Contact Persons**
- Ulf Riegel, Christoph Daniel Schulze, Carsten Sprung, Nina Wechselberg, Reinhard von Hanxleden
- Department of Computer Science, Kiel University
- Olshausenstr. 40, 24098 Kiel, Germany
- Phone: +49 (0) 431 880-7279
- Fax: +49 (0) 431 880-7615
- urpb / sil / csp / nwb / rth / informatik.uni-kiel.de
- http://www.informatik.uni-kiel.de/ihpsys

**Project Information**
- ELK: Eclipse Layout Kernel
- https://eclipse.org/ekl
- https://github.com/eclipse/ekl

---

Poster presented at the 24th International Symposium on Graph Drawing and Network Visualization (GD 2016), © 2016 Ulf Riegel, Christoph Daniel Schulze, Carsten Sprung, Nina Wechselberg, and Reinhard von Hanxleden.