

CONVERT Meets KIELER: Integrating Advanced Layout Algorithms into By-Example Visualisations

Iman Avazpour¹, Ulf Rüegg², and John Grundy¹

Introduction:

- The CONcrete Visual assistEd Transformation (**CONVERT**) uses by-example model transformations in the visualisation process i.e. notation design and composition [1]
- Layouts in CONVERT are notation specific, hard to specify and not flexible (e.g. see Figure 1)
- The Kiel Integrated Environment for Layout Eclipse Rich Client (**KIELER**) framework [2] provides various layout algorithms
- We show how the layout algorithms of KIELER are married to CONVERT visualisations

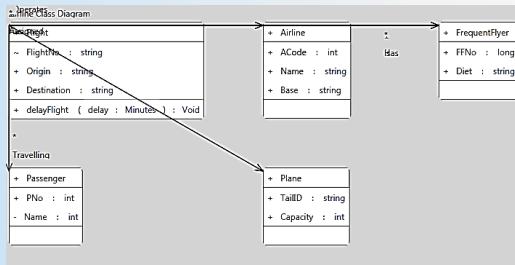


Figure 1. Class diagram visualisation with CONVERT's default layout.

Summary:

- Element positioning and Layout is crucial for visualisations
- We demonstrate the integration of KIELER's advanced layout mechanism into the by-example visualisations of CONVERT
- The example of a UML class diagram visualisation shows how class notations in CONVERT can be laid out using this approach
- This integration can be used for many alike graph based visualisations, or be applied in a similar way using other layout service providers

References:

- [1] I. Avazpour and J. Grundy, "CONVERT: A framework for complex model visualisation and transformation," in VL/HCC'12, Innsbruck, Austria, 2012, pp. 237–238.
- [2] M. Spönemann, C. D. Schulze, C. Motika, C. Schneider, and R. Von Hanxleden, "KIELER: building on automatic layout for pragmaticsaware modeling," in VL/HCC'13, San Jose, CA, USA, 2013, pp. 195–196.

The work:

- There are three steps to create visualisations in CONVERT:
 1. Generate/Reuse visual notations
 2. Map input data to notations
 3. Compose the mapped- to-input notations to generate complete visualisation (see above diagram)
 - Each notation controls positioning of its internal visual elements which allows use of third party layout algorithms
 - CONVERT uses the interface for selecting desired layout
- algorithms provided by KIELER to request positioning
 - Notations using this service send a JSON encoded graph of their internal elements to KIELER asking for the selected layout algorithm to be applied to the graph
 - KIELER enriches the graph with coordinates for every element and returns the result
 - The returned results are passed to the notation to reorganise internal visual elements (see Figure 2)

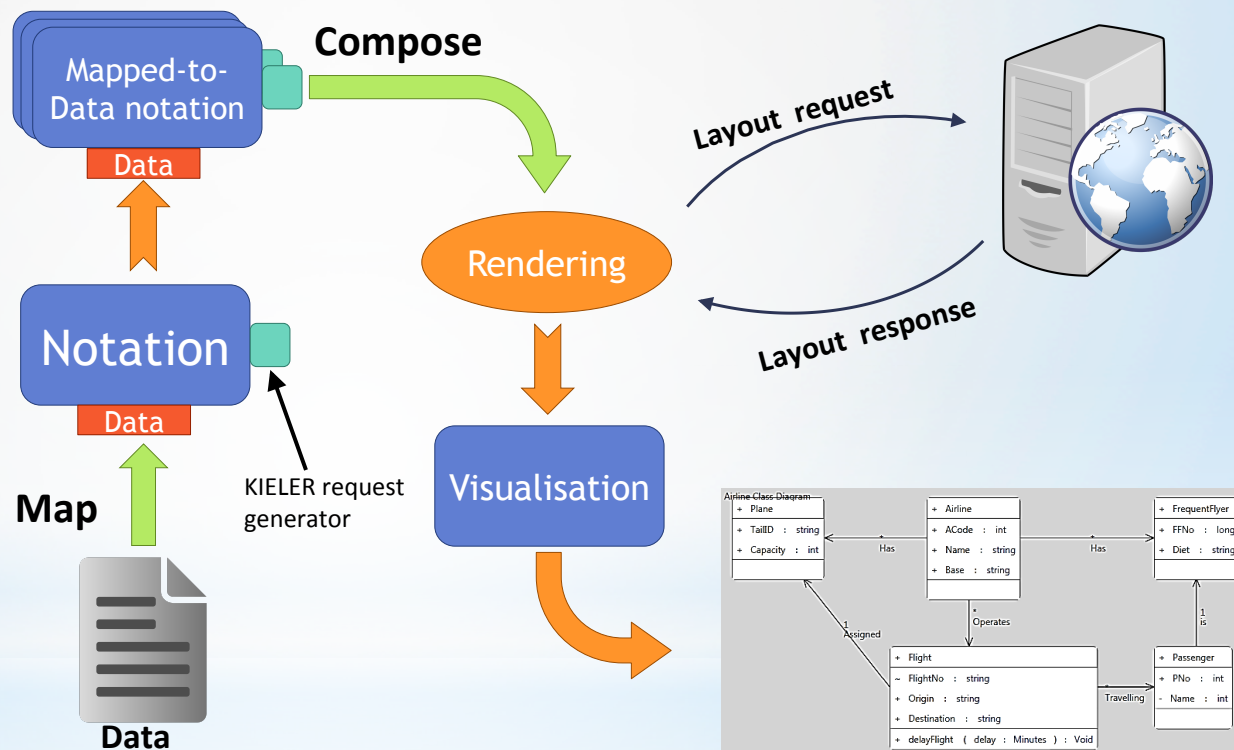



Figure 2. Class diagram visualisation using KIELER provided layout.

Author affiliations:

1  SUCCESS Centre, School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne, Australia

2  Department of Computer Science, Christian-Albrechts-Universität zu Kiel, Kiel, Germany