TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Julius Felchow

SLR-Toolkit
Informal Documentation

Task: A modern approach to chart creation

Outline

1. Task
2. BIRT Project and Vega.js
3. Usage
4. Development Guide
5. Future Work
6. Contact

Task

Task

1. Replace the old charting engine
2. Create a new approach for the charting
   a. for the Bar and Bubble charts
   b. with a fresh and recent framework
   c. with export of the charts as images (svg)

BIRT Project and Vega.js

BIRT Project and Vega.js

BIRT Project - https://www.eclipse.org/birt/

- Library that provides means for charting
- Somewhat deprecated and needs some freshing up
- The active development for the BIRT project has ceased, there are major updates for issues but no new features are introduced (last release: https://projects.eclipse.org/projects/birt/releases/4.8.0)

Vega.js - https://vega.github.io/vega/

- Framework that enables javascript based charting in the browser
- defines JSON schemas that describe the behaviour of the chart
- based on d3
- a lot of configuration features but better accessible and easier to enhance than d3 (which has a steep learning curve)

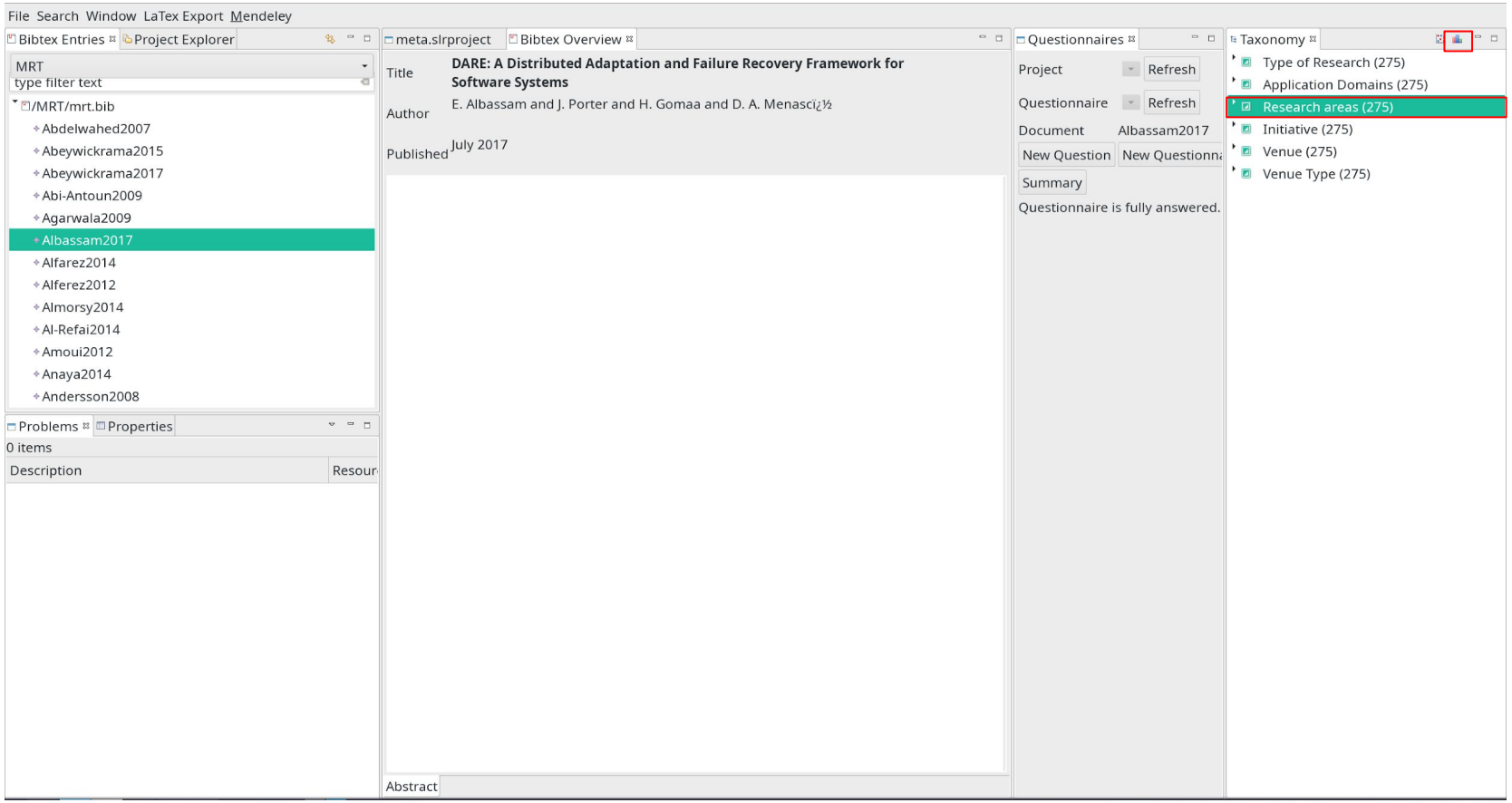
Usage

Usage

choose a value for the bar chart

Pick a value via left clicking it in the taxonomy view.

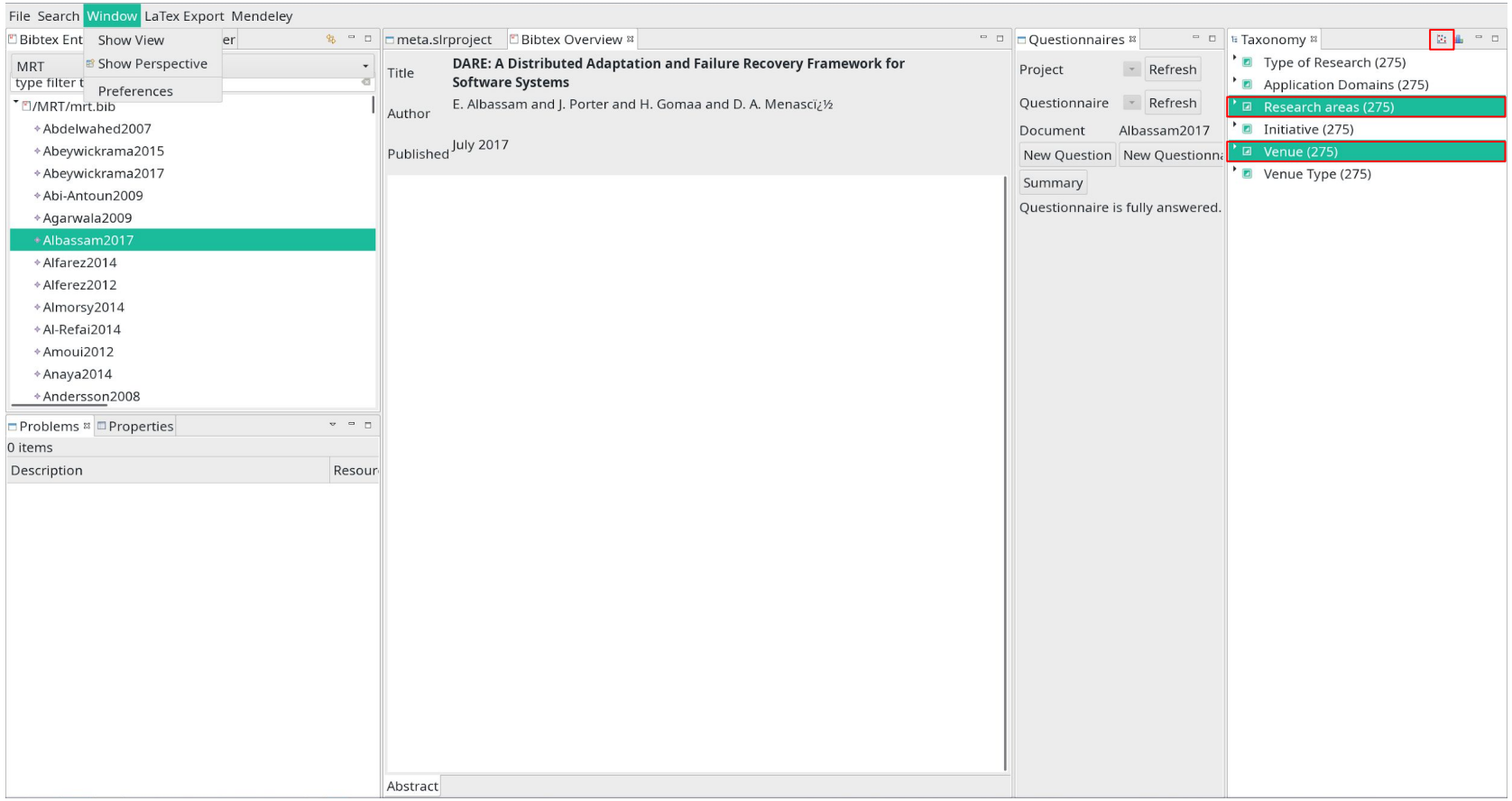
To create the bar chart, click the button in the tab bar



Usage

choose multiple values for the bubble chart

Pick a value via left clicking it in the taxonomy view and click on another value while holding the Control key.

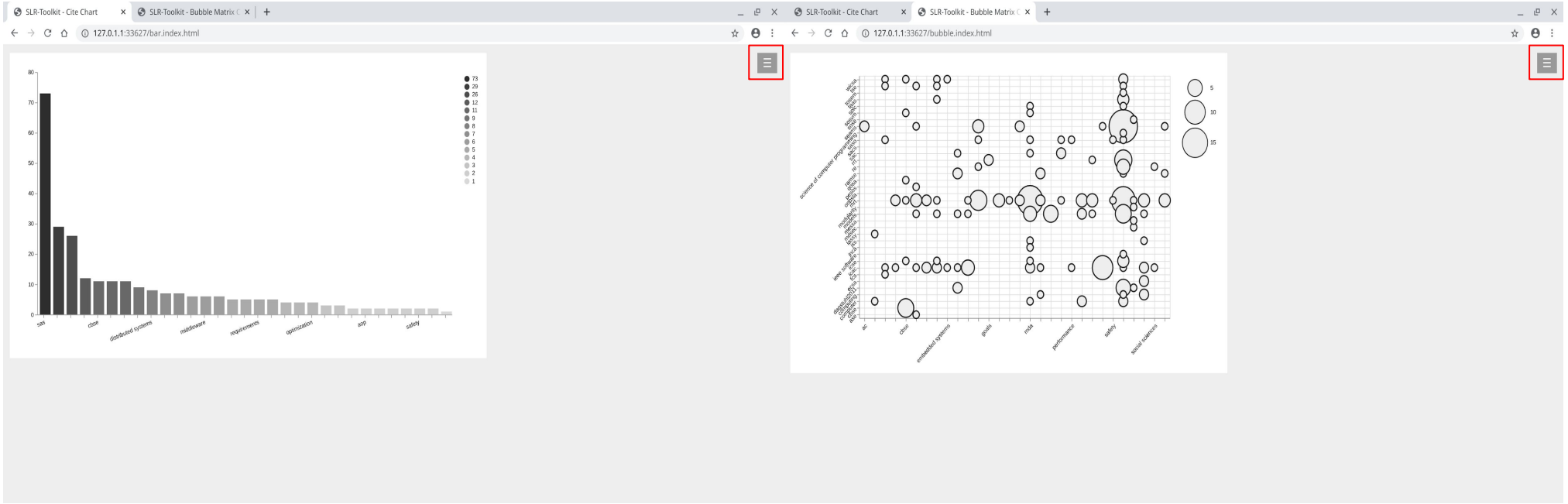


Usage

View the chart and edit it

This will open up the browser on the respective chart

Open the settings via the hamburger button

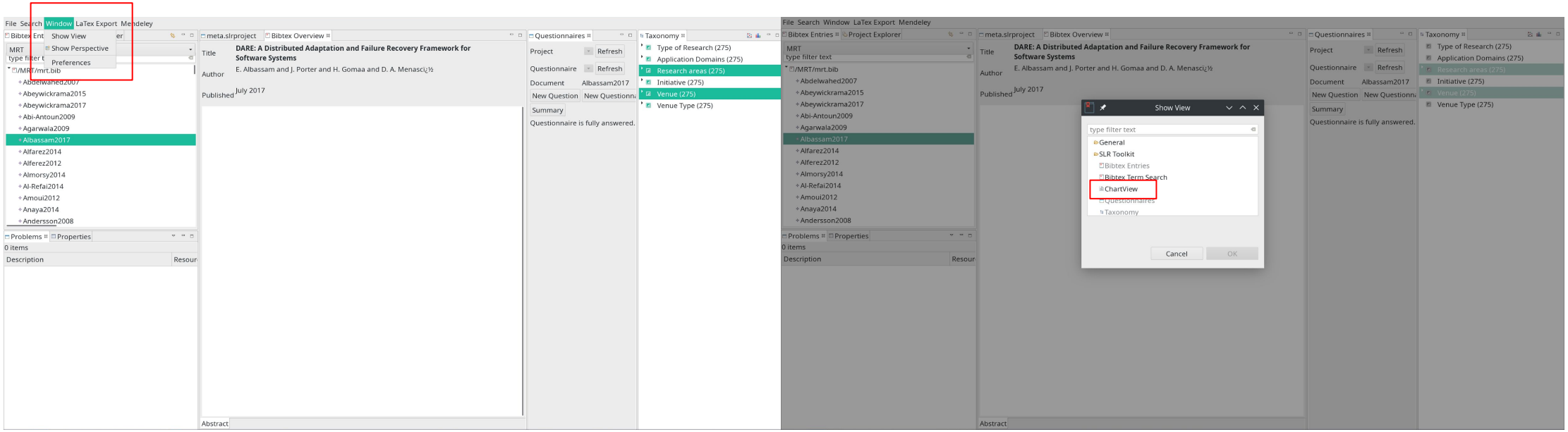


Usage
Editing and export

Edit the chart via the different options in the blocks
Download the chart as SVG via the button that says SVG
Download the chart data as CSV file

Usage
Old charting

The old charting is still available in the toolkit via the Show view menu



Development Guide

Server

Small Jetty Server serves the web content.

Started in the Activator of the taxonomy-ui-plugin on a random, free port.

Webroot can be found in the working directory:
[dev:`../runtime-slr.toolkit/webapp`; release: `./workspace/webapp`]

With the start of the server, the necessary javascript libraries (for offline use) are written from the `taxonomy-ui/src/html` directory to the webroot.

Dataflow

where does it come from and where does it go (where did you come from dataflow?)

The data is provided by the Taxonomy list.

Whenever one of the “Create Chart” buttons is pressed, the current state of the taxonomy list data is written into the corresponding `*.data.csv` file in the webapp directory.

The javascript loads the data and displays it in the chart.

The javascript does not manipulate the csv-file in anyway, all the changes (with filtering of the selection) to the data are only within the browser session.

Development Guide

how to add new data or charts

To visualize new data in the charts, it needs to be written in the correct format [bar: `category,amount`; bubblechart: `category1,category2,amount`] into the csv file.

To add another chart, multiple steps need to be taken:

- to the html folder in the taxonomy-ui/src directory:
  - add a new `*.vg.json`-schema that defines the graph needs to be added
    - refer to https://vega.github.io/vega/examples/ for examples
  - add a new `*.data.csv` with the data (or reuse an old one)
  - add a html `*.index.html` that displays the web content
- write the html and json file in the Activator to the webapp
- add a new handler (refer to the OpenCiteHandler.java) to write the data and open the browser

Development Guide

how to change the chart

To change the default values for the configuration, edit the corresponding values in the settings:

- within the javascript in the html is a function - ViewModel
- in the function are the default values for the chart
- if the changed value is not in the settings, edit the corresponding part in the json file

Development Guide

how to add settings for the chart configuration

The settings in the side bar are wired to the chart.

When the javascript is executed, the default chart settings are saved in the variable `spec`.
When a setting is changed, the html element is listened to in the ViewModel and the value then written to the spec. Afterwards the graph is rerendered. (Thats why there is a short flash when changing a setting.)

To add a new setting:

- add a html element to the sidebar with a data-bind value set
- Listen to that value in the ViewModel
- add it to the spec
    - for a new setting, that is not in the spec yet, check https://vega.github.io/vega/docs/ for changeable attributes

Future Work

Future Work

These are possible ideas where the current state of the Javascript charting could be enhanced:

- Saving of chart settings in the browser → export/import and enabling a default configuration or the possibility to choose between different configurations
- Integration of a browser window into the eclipse client
  - https://github.com/maketechnology/chromium.swt looks promising but couldn't be used due to incompatibility with the older Eclipse Version - maybe usable in a later version of the toolkit
- Integration of chart export of the new charts into the Latex generation

Contact

Contact

If you run into problems feel free to contact me:

- Mail: julius.felchow@gmail.com
- Github: @pibebtol