**MOTIVATION**

- Growing amount of data stored in RDF, e.g., within the LOD cloud, but gaining insights from data mostly reserved for tech-savvy users
- Information visualization as approach to address end-users but current RDF browsers are at most graph- or text-based and lack of assistance features like visualization recommendation

**GOALS**

- Formalization of visualization knowledge and indexing of graphic representations according identified key concepts
- Specification of a user-centered, semantics-based information visualization workflow
- Definition of a flexible, web-based software architecture which supports the workflow

**APPROACH**

- Using a semantic model, called VISualization Ontology (1), to capture visualization knowledge and to annotate data (3) and visualization components (4)
- Automatic mapping from data to widgets based on annotations as well as visualization knowledge (2) and context information (5)

**VISUALIZATION WORKFLOW**

- Workflow geared towards end-users to identify interesting areas within the data (3), select and configure appropriate visualization widgets (5, 7), and view and understand the data (9)
- System-side functionality making use of VISO, e.g., to reduce the data set (4), recommend visualization widgets (6), or to track the usage of the components (10)

**COMPONENT-BASED ARCHITECTURE**

- Based on the CRUISe ecosystem allowing for the dynamic, context-aware composition of web applications building on distributed building blocks
- VISO (1) as glue between the Data Repository (2), which augments, manages, and filters the RDF data, and the visualization widgets (3) managed within the Component Repository (4)
- Mashup Runtime Environment (5) provides the interactive VizBoard application accessing the repositories and the context service (6)

**IMPLEMENTATION**

- Extending the CRUISe ecosystem, e.g., the component description language and Component Repository, according to visualization-specific needs
- Data Repository as Java-based RESTful web service building Jena TDB triple store for RDF handling
- User interface components based on JavaScript libraries like D3.js and partly Adobe Flash, e.g., the visualization selection
- Demo and further information at http://www.vizboard.de