Files\\Literature\\Ledur et al. - 2017 - A High-Level DSL for Geospatial Visualizations wit - § 2 references coded [0.65% Coverage]

Reference 1 - 0.44% Coverage

Among the techniques of geospatial data analysis, data visualization can help domain users to quickly gain insights [4]. The tools available presently do not provide the necessary abstractions in the pre-processing step of the visualization pipeline [5]. Thus, even though data visualization offers many benefits, its generation is still remains a challenge [6].

Reference 2 - 0.21% Coverage

Domain users have difficulties dealing with a large amount of data, since it demands high costs and a great deal of programming effort to process and manipulate the raw data.

Files\\Literature\\Li et al. - 2018 - ECharts A declarative framework for rapid constru - § 1 reference coded [0.27% Coverage]

Reference 1 - 0.27% Coverage

While there have been a dozen of authoring systems and programming toolkits for visual design and development (Mei et al., 2018), it is still cumbersome for users, such as data analysts or interface designers, to rapidly implement a web-based and interactive visualization

Files\\Literature\\Logre and Déry-Pinna - 2018 - MDE in Support of Visualization Systems Design a - § 2 references coded [0.38% Coverage]

Reference 1 - 0.23% Coverage

The context of visualization systems provides two challenges : (i) the support of the system design, i.e., a modeling challenge, and (ii) the choice of relevant visualizations among the available widgets offered by visualization libraries, i.e., a variability challenge.

Reference 2 - 0.15% Coverage

the main challenge is for the system designer to choose appropriate visualizations. To ease this choice, we offer to constitute a tooled catalog of available visualizations.

Files\\Literature\\Logre et al. - 2014 - Sensor Data Visualisation A Composition-Based App - § 2 references coded [1.21% Coverage]

Reference 1 - 0.97% Coverage

The main challenge to tackle is then to support the intrinsic variability of this domain. This variability is twofold and thus triggers two concurrent challenges: (i) each user wants to use a dashboard dedicated to her very own needs, and (ii) visualisation libraries used at runtime provide different visualisation widgets to be used to implement such dashboards. In this context, model-driven engineering approaches can support the first challenge by capturing concepts used by the dashboard designers and providing appropriate tool support. To tackle the second challenge, Software Product Lines (SPLs) are defined as “a set of software-intensive systems that share a common, managed set of features and that are developed from a common set of core assets in a prescribed way” [2]

Reference 2 - 0.24% Coverage

two major issues: (i) there were almost as many dashboards needed as interviewed users and (ii) at the implementation level, developing such dashboards is error-prone and time-consuming.

Files\\Literature\\Morgan et al. - 2017 - VizDSL Towards a Graphical Visualisation Language - § 2 references coded [1.37% Coverage]

Reference 1 - 0.84% Coverage

Commercial visualisation tools, such as Tableau, Spotfire or QlikView, use a graphical UI and are user friendly, offer a wide range of visualisation types and techniques and can be very customizable. These tools are strongly data-oriented, in that visualisation is restricted to representing data content, rather than being capable of representing inherent data structures at a higher level of abstraction. Although many standalone packages support the use of multiple data sources, the lack of semantic representation means that visualisations of heterogeneous data sources are often
presented as a conglomerate of apparently unrelated records.

Reference 2 - 0.53% Coverage

There are a number of textual visualisation DSLs available, however, none of them have been designed with the needs of enterprise integration and interoperability in mind and they all require some coding experience. We have identified a need for a graphical visualisation DSL which supports semantic visualisation in order to further the aims of model-driven development in integration and interoperability.

Files\\Literature\\Satyanarayan et al. - 2017 - Vega-Lite A Grammar of Interactive Graphics - § 1 reference coded [0.05% Coverage]

Reference 1 - 0.05% Coverage

existing high-level languages provide limited support for interactivity.

Files\\Literature\\Teng et al. - 2021 - Sketch2Vis Generating Data Visualizations from Ha - § 1 reference coded [0.31% Coverage]

Reference 1 - 0.31% Coverage

Adopting an intermediate DSL (rather than directly relying on native visualization code for the visualization specification) enables decouples the deep learning model from the source code generation.

Files\\Literature\\Vázquez-Ingelmo et al. - 2018 - Domain engineering for generating dashboards to an - § 1 reference coded [0.43% Coverage]

Reference 1 - 0.43% Coverage

managing sets of differing requirements, focusing on the reutilization and composition of base software assets (also known as core assets) to improve scalability and maintainability of particular products that share commonalities.