Files\\Literature\\Bostock and Heer - 2009 - Protovis A Graphical Toolkit for Visualization - § 3 references coded [0.59% Coverage]

Reference 1 - 0.13% Coverage

a gap between the notational efficiency of high-level visualization systems and the expressiveness and accessibility of low-level graphical systems.

Reference 2 - 0.33% Coverage

Both charting software and analytical tools such as Tableau are successful in practice, but their expressiveness is limited. We wanted to provide tools that enable more low-level control of the design. Unlike existing toolkits primarily suited to software engineers, we created Protovis to make interactive visualization more accessible to web and interaction designers.

Reference 3 - 0.12% Coverage

As we developed Protovis, we sought to meet our design goals of creating an expressive, efficient, and accessible visualization tool.

Files\\Literature\\Heer and Bostock - 2010 - Declarative Language Design for Interactive Visual - § 1 reference coded [0.57% Coverage]

Reference 1 - 0.57% Coverage

Moreover, contemporary visualization design tools must address a
number of new technical challenges. Not least among these is the increasing heterogeneity of commodity hardware and interactive devices. Visualization tools should ideally support interfaces ranging from traditional desktop applications, to browser-based web clients, to multi-touch mobile devices. Furthermore, visualization tools should effectively capitalize on hardware trends such as multi-core computing and specialized graphics hardware. While point designs exist for each of these areas, the field currently lacks a consistent approach to visualization design and deployment across heterogeneous platforms.

Files\\Literature\\Li et al. - 2018 - ECharts A declarative framework for rapid constru - § 2 references coded [0.45% Coverage]

Reference 1 - 0.10% Coverage

The motivation is driven by three goals: easy-touse, rich built-in interactions, and high performance.

Reference 2 - 0.35% Coverage

We argue that the flexibility and complexity of visual de-
sign should not be limited by the requirement on programming skills (Heer et al., 2008). The essential motivation of this work is to fill this gap through a declarative object option and composable visualization components, which are modeled with the user-configurable declarative object option.

Files\\Literature\\Logre and Déry-Pinna - 2018 - MDE in Support of Visualization Systems Design a - § 9 references coded [2.20% Coverage]

Reference 1 - 0.27% Coverage

The increasing amount of data produced and continuous development of new visualizations exacerbate the difficulty of designing such dashboards, while the visualization need is broaden to specialist and non-specialist final users. In this context, we offer a multi-user approach, based on Model Driven Engineering (MDE)

Reference 2 - 0.10% Coverage

The idea is for the designer to express the visualization need by characterization, according to a given taxonomy.

Reference 3 - 0.37% Coverage

An analytic visualization system captures the business knowledge of the users, using proven indicators to detect abnormalities in order to support them in the decision making process. In this context, the user tasks and the proper data sets to exploit are defined during the design of the system. To support the diagnosis goal, the design of such systems relies on the identification of the relevant visualizations to use and their connections.

Reference 4 - 0.36% 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. A classical way to support the design of complex systems (cf. (i)) in the Model-Driven Engineering (MDE) paradigm is the use of a Domain Specific Language (DSL).

Reference 5 - 0.15% Coverage

While the abstract syntax captures the structure of the domain in a meta-model, a concrete syntax allows the design of conform models through textual or graphical representations.

Reference 6 - 0.15% Coverage

The corner stone of our contribution is to apply MDE to the design of visualization systems, to provide to each user the relevant support to ease their task in the diagnosis process.

Reference 7 - 0.47% Coverage

Within the MDE paradigm, the use of models provides a higher abstraction space to design the system, it diminishes the cost of modifications and helps to quickly produce high fidelity prototypes, and handles the variability of concrete solutions. In our case, it allows to reason about abstract visualizations (i.e., a visual space designed to highlight identified characteristics or properties of its input data), instead of enduring the technological complexity of numerous concrete widgets (i.e., specific implementations of a visualization, unit of code).

Reference 8 - 0.23% Coverage

While the relevant indicators for a given diagnosis task are defined, the users needs evolve with their degree of expertise, the object under study, and the context of use. Therefore, there is no such thing as an absolute, ideal visualization system even in the analytic paradigm.

Reference 9 - 0.09% Coverage

Characterization, is the constitution ofa catalog encompassing the capabilities of our solution space

Files\\Literature\\Logre et al. - 2014 - Sensor Data Visualisation A Composition-Based App - § 2 references coded [0.58% Coverage]

Reference 1 - 0.48% Coverage

The one-size-fits-all paradigm does not apply in this context, as user’s roles are variable and impact the way data should be visualised: a building manager does not need to work on the same data as classical users. This paper presents an approach based on model composition techniques to support the development of such monitoring dashboards, taking into account the domain variability.

Reference 2 - 0.10% Coverage

This variability is supported at both implementation and modelling levels.

Files\\Literature\\Morgan et al. - 2017 - VizDSL Towards a Graphical Visualisation Language - § 1 reference coded [0.42% Coverage]

Reference 1 - 0.42% Coverage

The main contribution of this paper is threefold: (1) We of complex data structures techniques against them, (2) identified some identified requirements for data visualisation to support the understanding existing and compared deficiencies of existing techniques which we propose to overcome with a new language called VizDSL,

Files\\Literature\\Rojas et al. - 2020 - Cities-Board A Framework to Automate the Developm - § 1 reference coded [0.59% Coverage]

Reference 1 - 0.59% Coverage

Cities across the world have similar, if not the same, functional and nonfunctional requirements to develop their dashboards. Software developers will face the same challenges and they are likely to provide similar solutions for each developed city dashboard. Moreover, the development of these dashboards implies a significant investment in terms of human and financial resources from cities. The automation of the development of smart cities dashboards is feasible as these visualization systems will have common requirements between cities.

Files\\Literature\\Satyanarayan and Heer - 2014 - Authoring Narrative Visualizations with Ellipsis - § 1 reference coded [0.16% Coverage]

Reference 1 - 0.16% Coverage

Design tools for narrative visualization that support this process could improve efficiency and empower journalists to collaborate with developers.

Files\\Literature\\Smeltzer and Erwig - 2018 - A domain-specific language for exploratory data vi - § 1 reference coded [0.55% Coverage]

Reference 1 - 0.55% Coverage

To support such an exploratory workflow in which design decisions can be delayed, we propose variational visualizations, which are structures that encode arbitrarily many traditional visualizations in a systematic way. Variational visualizations empower visualization authors and don’t force them to commit prematurely to particular design decisions. Variational visualizations are not limited to producing simply side-by-side comparisons of visualizations; due to their tight integration into a general visualization DSL, they facilitate versatile combinations of visualization variants such as overlays of alternative bar charts

Files\\Literature\\Smeltzer et al. - 2014 - A transformational approach to data visualization - § 1 reference coded [0.47% Coverage]

Reference 1 - 0.47% Coverage

The design of this DSL is informed by a number of more specific goals. First, we provide multiple layers of abstractions in order to support multiple use cases. Sometimes a quick oneliner is all that is necessary to get an overview of data, while more customization and detail may be desirable for a figure designed to be shared. We also intentionally keep the number of core language constructs small, which is intended to simplify systematic analysis and ease the process of defining new, custom transformations. This can also allow the user to create custom abstraction layers to match particular requirements.

Files\\Literature\\Teng et al. - 2021 - Sketch2Vis Generating Data Visualizations from Ha - § 2 references coded [1.56% Coverage]

Reference 1 - 0.86% Coverage

Although the learning curve of visualization tools have been reduced, domain experts still often require significant amounts of training to use them effectively. To reduce this learning curve even further, this paper proposes Sketch2Vis, a novel solution using deep learning techniques and tools to generate the source code for multi-platform data visualizations automatically from handdrawn sketches provided by domain experts, which is similar to how an expert might sketch on a cocktail napkin and ask a software engineer to implement the sketched visualization.

Reference 2 - 0.69% Coverage

A key problem in generating programming language code from sketches is that the models must be trained on each programming language that they target, which is tedious and error-prone. To overcome this challenge—and to support realization of the sketch using multiple visualization platforms— we employ an intermediate DSL model that represents the abstract structure of the sketch with a simple syntax that can be learned readily by a deep learning model.

Files\\Literature\\Vázquez-Ingelmo et al. - 2018 - Domain engineering for generating dashboards to an - § 1 reference coded [0.50% Coverage]

Reference 1 - 0.50% Coverage

Having a methodology to efficiently generate dashboards taking into account differing needs would add a customization layer to allow particular users to reach their own goals. This approach can be achieved through domain engineering and automatic code generation processes.