Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 98 (2016) 100 – 106

Procedia Computer Science

The 7th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2016)

A Review of Latest Web Tools and Libraries for State-of-the-art Visualization

Farrukh Shahzad[c,1,*], Tarek R. Sheltami[a], Elhadi M. Shakshuki[b], Omar Shaikh[d]

[a]*Department of Computer Engineering, KFUPM, Dhahran, KSA*
[b]*Jodrey School of Computer Science, Acadia University, Wolfville, NS Canada B4P 2R6*
[c]*Visonics, Houston-TX, USA 77082*
[d]*Dhahran High School, Al-Khobar 31952, Saudi Arabia*

Abstract

Most of the existing visualization and simulation applications run on the client machine and require an installation process. Browser based interactive visualizer for scientific and medical applications remain an unheard concept despite all advancements in computer and software technology and it remains a fairly difficult process to quickly prototype a visualization on a PC or a smart device. In this paper, we review and employed state-of-the-art web technologies, third-party libraries and frameworks to compare and develop some interactive browser-based, mobile friendly web applications. These latest web technologies have the potential to fulfill the promise of interactive browser based custom visualization applications. We presented and compared some of the latest web based tools available today. We also introduced couple of lightweight and interactive web based visualizer and simulator tools which are under development.

Peer-review under responsibility of the Program Chairs

Keywords: Browser-based Visualization; Web based Application; JavaScript; Simulation

1. Introduction

There is a greater demand for custom on-demand high performance visualization and simulation applications for medical, engineering, scientific and industrial problems. Existing visualization and simulation applications run on the client machine and require an installation process. Browser based interactive visualizer for scientific and medical applications remain an unheard concept despite all advancements in computer and software technology and it remains a fairly difficult process to quickly prototype a visualization on a PC or a smart device. The development of a reliable and robust large-scale system requires that design concepts are visualized in some digital form, before implementation. Simulations provide a cost effective and feasible method of examining the correctness and scalability of the system

* Corresponding author. Tel.: +1-713-261-7494.
E-mail address: farrukh.shahzad@visonics.net

Peer-review under responsibility of the Program Chairs
doi:10.1016/j.procs.2016.09.017

before deployment. Similarly, educational and Science, Technology, Engineering and Math (STEM) Visualization allow smart and interactive teaching in classrooms or for web based tutorials.

In the past, we have developed few visualization and simulation tools. Pymote 2.0 is an extension of Pymote which is a high level Python library for event based simulation of distributed algorithms in wireless networks[1]. We implemented modules for propagation, energy consumption and mobility models. We also added graphing and data collection modules to enhance the Pymote base functionality and modified existing modules for node, network, algorithm and logging to support the extended framework.[2,3]. The extended framework utilizes the python Matplotlib package[4] and the innovative charting library provided by Highsoft[5], which is free to use for personal and academic purposes. The output format includes Comma-Separated Values (CSV), Portable Network Graphics (PNG), high-quality Scalable Vector Graphics (SVG) and Portable Document Format (PDF) which can directly be inserted into Latex and other publishing applications. HyperText Markup Language (HTML) files are also created with embedded JavaScript (JS) and Cascading Style Sheets (CSS) for interactive plotting which is needed for presentations and on-line content.

In this work, we review the following state-of-the-art web technologies, third-party libraries and frameworks that can be utilized to fulfill the promise of interactive browser based custom visualization applications. These libraries use SVG standard libraries for appending and manipulating SVG elements, which is supported in almost all modern browsers, smart phones and tablets. These libraries and frameworks are developed using pure JavaScript, so users get interactivity without requiring round-trips to servers and without any additional plugins.

- **HTML5** is a larger set of technologies (including CSS3) that allows more diverse and powerful Web sites and applications.
- **jQuery** is a fast, small, and feature-rich JavaScript (JS) library which allow write less, do more[6].
- **D3.js** is a JavaScript library for manipulating documents based on data[7].
- **GoJS** is a JavaScript library for implementing interactive flowcharts, organizational hierarchies, trees and other complex displays[8].
- **JSmol** is an open-source JS library and HTML5 viewer for chemical structures in 3D[9].
- **Highchart** makes it easy to set up interactive charts in web pages[5].
- **Bootstrap** is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web[10].

The idea is to employ these state-of-the-art visualization, presentation, animation, and simulation technologies to create a desired custom solution. Software and application developers under the guidance of domain experts in medical, engineering and scientific profession, together can create some great visualization and simulation solutions in areas as diverse as organism/molecular simulation, Nano-technology, Material Sciences, Medical visualization, Astrophysical visualization, and Chemical or Biological visualization.

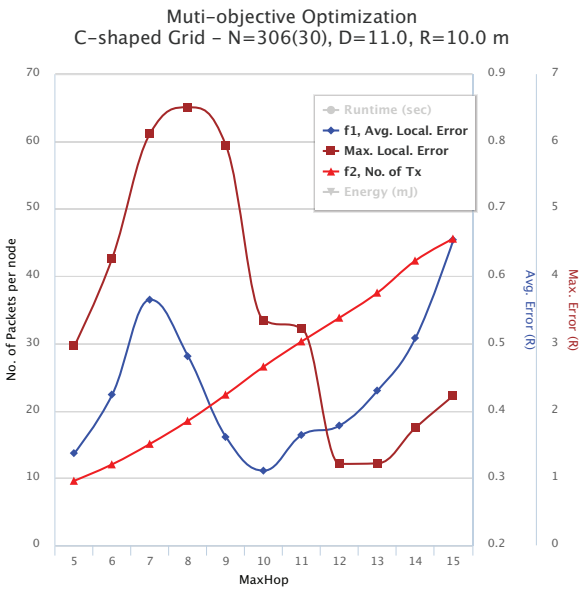
We also present the design and implementation of a generic topology generator, as part of an interactive browser-based network visualizer and simulator called *Visual-Netsim*. We also introduce an innovative web-based visual flow chart drawing application. These are lightweight utilities that allow quick and powerful web-based custom visualization and drawing.

2. JavaScript Libraries

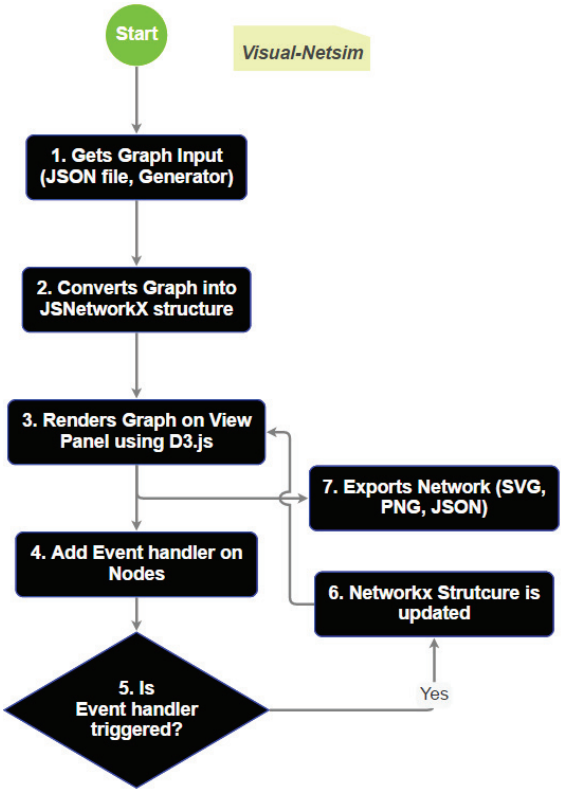
In this section, we review some of the JavaScript libraries which are designed for modern browsers and smart devices. These libraries exploited the advancement in HTML5, CSS3 and SVG and provide an Application Programming Interface (API) for developer to create web-based mobile-friendly 2D/3D diagrams and animations.

2.1. HighChart

Highcharts is a pure JS charting library, offering an easy way of adding interactive web-based charts. It is designed from ground up with mobile browsers in mind, everything from multitouch zooming to touch-friendly tooltips responds great on mobile platforms. In modern browsers graphs are rendered in SVG, with VML support for legacy



(a) An Interactive plot generated using HighCharts.



(b) A diagram created using GoJS.

Fig. 1: Interactive diagrams created using HighCharts and GoJS

browsers. Highcharts mature, and modular codebase is also the foundation of its two others products, Highstock JS (for financial data) and Highmaps JS (for location/geo-spatial data). It can be employed to visualize engineering, scientific and experimental data seamlessly using interactive 2-D and 3-D line, spline, area, areaspline, column, bar, pie, scatter, angular gauges, arearange, areasplinerange, columnrange, bubble, box plot, error bars, and polar charts. Fig. 1a shows a chart with multiple plots. Notice that some plots are disabled (grayed out). Highcharts provide interactive plot enable/disable, zooming, panning and cross hairline support. Furthermore, developer have full control on colors, fonts, legends, axis labels, etc. The web based chart can be saved in JPEG, PNG, SVG and PDF formats.

2.2. *GoJS*

GoJS is another JavaScript library for implementing interactive diagrams and visualizations on modern web browsers and platforms. GoJS allows easy construction of custom and complex diagrams of nodes, links, and groups with customizable templates and layouts. The library offers rich features like drag-and-drop, copy-and-paste, in-place text editing, tool-tips, templates, data binding and models, transactional state and undo management, palettes, event handlers, commands, and an extensible tool system for real-time custom operations on the diagram. GoJS does not depend on any JavaScript libraries or frameworks, so it should work with any web framework or with no framework at all. Fig. 1b shows an interactive and responsive flow chart designed using GoJS. This diagram allows in place editing, resizing, recoloring and dragging of each item.

2.3. D3

D3.js is a JavaScript library for manipulating documents based on data and bring it to life using HTML, SVG and CSS. D3 emphasis on web standards gives developer the powerful visualization and provide a data-driven approach to Document Object Model (DOM) manipulation. It is build on top of jQuery. It can be utilized to draw interactive diagrams and 2D/3D charts and plots.

2.4. JSmol

JSmol allows interactive display of 3D molecular structures on web browsers. JSmol is a JavaScript framework build using Python based *Pymol* and Jave based *Jmol* in 2012, to allow web developers to create applications that utilize HTML5 (or Java if available on the platform). This enables the extensive original Jmol library to be compatible with devices that no longer support Java Applets due to security concerns or for which Java is not available (such as smart phones and tablets). No hardware-based graphics acceleration is used, allowing JSmol to run in any web browser that supports HTML5 standards, and needing no server technologies for most of its operation.

JSmol provide several importing option to load the chemical structure data including Crystallographic Information File (CIF) and Protein Data Bank (PDB). The library has rich set of tools and features to manipulate, visualize, animate, simulate and export the chemical structure in real-time. Fig. 2 shows a zeolite visualized using JSmol.

3. Tools Development

We have utilized the technologies and libraries introduced in previous section and developed few custom visualization solutions.

3.1. Visual-Netsim

Visual-Netsim is a novel web based network visualizer and simulator, which includes a generic and interactive topology generator. The simulation tool renders and generates graphs in the browser using D3 library. A simple GUI allows the user to generate topologies of different shapes, and provides tools that help manipulate complex graphs on client browsers and mobile devices. This tool has its application in several domains including computer

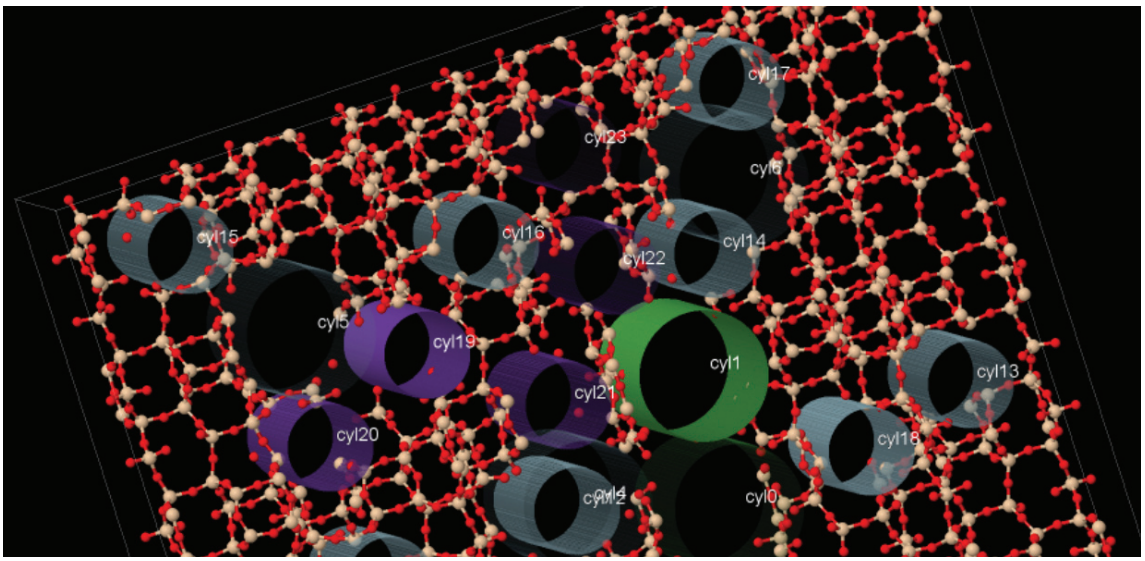


Fig. 2: A mesoporus zeolite visualized using JSmol

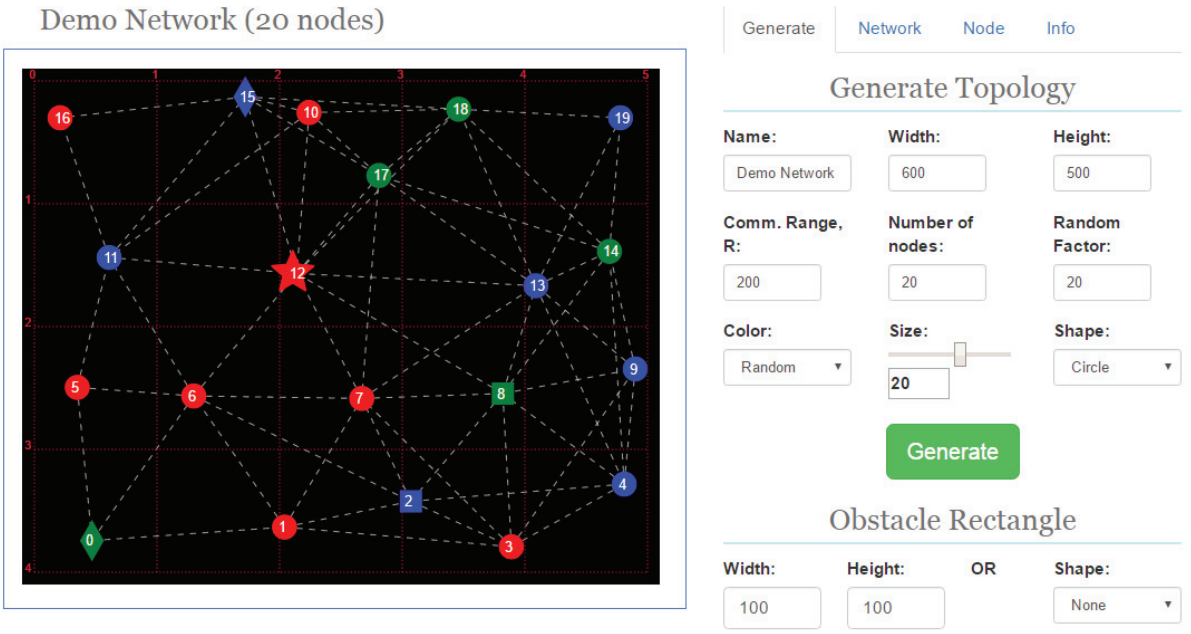
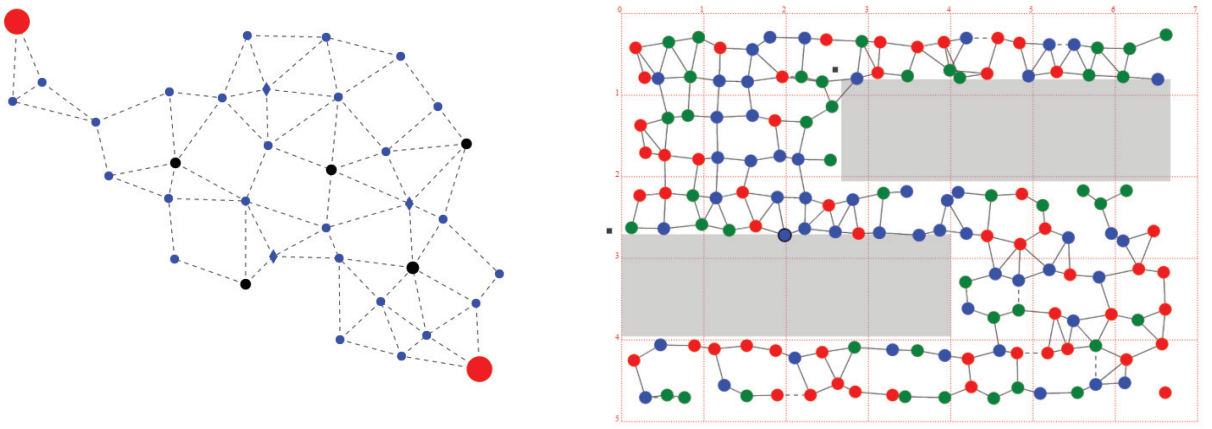


Fig. 3: *Visual-Netsim*: Generate Topology Tab



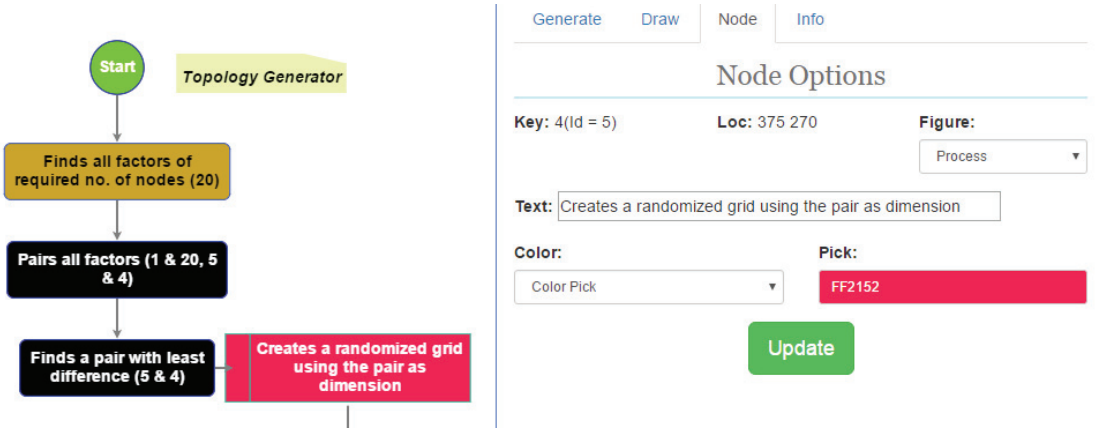
(a) Node's color, size and shape are settable

(b) S-shaped Network

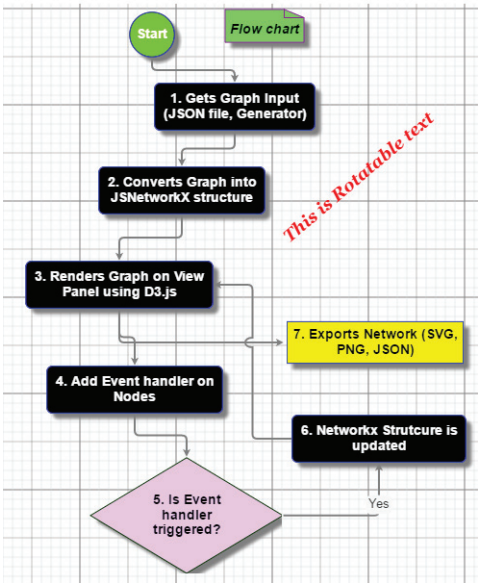
Fig. 4: Some screen output from *Visual-Netsim*

network configuration and management, sensor and ad-hoc networks, biological and social media visualizations. This application can also facilitate research and education in the related areas.

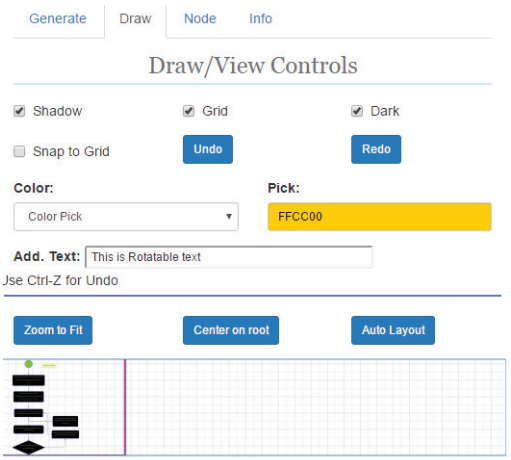
Visual-Netsim user interface provide four tabs for topology generation, network view options, node editing and information display related to the topology. The tool is capable of generating a graph on the client itself using a configurable algorithm. The generator uses several parameters, including communication range value, total number of nodes, the length and width of the network deployment area, optional obstacles dimensions, and a random factor. Since *Visual-Netsim* is a client-based application, it executes this algorithm on the user's device, not relying on a remote server for the process. There are several options to adjust and display other items like labels, grid line, background, etc. Each node's attributes can be modified as needed. The user has option to save the work in JSON format and load it later to continue working on the topology. The topology can also be exported in PNG image and high quality SVG files. Fig. 3 shows the user interface with generate Topology tab on the right. A random network



(a) A simple flow chart with modified nodes



(b) A Web Application flow diagram



(c) Draw/View control tab for quick customization

Fig. 5: Output from *Visual-Flow* showing the interface and a sample diagram.

with different color and size of nodes is shown in Fig. 4a, whereas an anisotropic S-shaped Network is shown in Fig. 4b.

3.2. *Visual-Flow*

We also utilize HTML5 and JavaScript, to design and implement an interactive mobile-friendly flow-chart and drawing application, called *Visual-Flow*. It uses GoJS library for drawing. *Visual-Flow* user interface provide tabs for flowchart or drawing generation, draw/view controls, node editing and information display related to the diagram.

A simple flow chart is shown in Fig. 5a with panel on the right showing option to modify a node. User can click either on the node on the view panel or in the info list to edit any node. A web application flow diagram is shown in Fig. 5b with some nodes dragged or modified to show the interactivity of the application. In Fig. 5c, the draw/view control tab is shown which allows quick customization of the diagram.

The tool is capable of generating custom flow charts or diagrams using pre-defined types or by dragging standard building blocks and linking them as required. A previously saved JSON file containing diagram details can also be loaded. There are several option to adjust and display items like labels, grid line, background, etc. Each node's attributes (color, type and label) can be modified as needed. There is also option to have additional labels (in-place editable and rotatable) anywhere in the diagram. The user has option to save the work in JSON format and load it later to continue working on the design. The diagram can also be exported in PNG image and high quality SVG files. The Info tab list all of the diagram's blocks or nodes and the list of all the links.

Conclusions

We presented and compared some of the latest web based tools and libraries available today for client-side browser-based visualization. We also introduced couple of lightweight and interactive web based visualizer and simulator tools which are under development. We strongly believe that there is huge potential for software developers and domain experts to collaborate and create some great custom visualization and simulation solutions in medical, engineering and scientific domain.

As future work, we will continue building on the presented visualization applications to improve user friendliness and performance as well as add new features. We are also working on some ideas to design and implement custom visualization and simulation tools for chemistry and biomedical applications.

Acknowledgments

We would like to acknowledge the support provided by the Deanship of Scientific Research at King Fahd University of Petroleum and Minerals (KFUPM) and the Deanship of Research and Graduate Studies at Acadia University.

References

1. Arbula, D., Lenac, K.. Pymote: High Level Python Library for Event-Based Simulation and Evaluation of Distributed Algorithms. *International Journal of Distributed Sensor Networks* 2013;**2013**(797354):12. doi:10.1155/2013/797354.
2. Shahzad, F., Sheltami, T.R.. An efficient MAC scheme in wireless sensor network with energy harvesting (EHWSN) for cloud based applications. In: *2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)*. IEEE. ISBN 978-1-4673-6773-8; 2015, p. 783–788. URL: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7365928`. doi:10.1109/LCNW.2015.7365928.
3. Shahzad, F., Sheltami, T.R., Shakshuki, E.M.. Effect of network topology on localization algorithm's performance. *Journal of Ambient Intelligence and Humanized Computing* 2016;**7**(3):445–454. URL: `http://link.springer.com/10.1007/s12652-016-0349-4`. doi:10.1007/s12652-016-0349-4.
4. Hunter, J.D.. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering* 2007;**9**(3):90–95. URL: `http://scitation.aip.org/content/aip/journal/cise/9/3/10.1109/MCSE.2007.55`. doi:10.1109/MCSE.2007.55.
5. Høns, T.. Highcharts, Highstock and Highmaps documentation — Highcharts. 2013. URL: `http://www.highcharts.com/docs`.
6. jQuery. 2016. URL: `https://jquery.com/`.
7. Bostock, M., Ogievetsky, V., Heer, J.. D3: Data-Driven Documents. *IEEE transactions on visualization and computer graphics* 2011; **17**(12):2301–9. URL: `http://dl.acm.org/citation.cfm?id=2068462.2068631`. doi:10.1109/TVCG.2011.185.
8. GoJS - Interactive Diagrams for JavaScript and HTML. 2016. URL: `https://www.nwoods.com/products/gojs/`.
9. Hanson, R.M., Prilusky, J., Renjian, Z., Nakane, T., Sussman, J.L.. Jsmol and the next-generation web-based representation of 3d molecular structure as applied to proteopedia. *Israel Journal of Chemistry* 2013;**53**(3-4):207–216. doi:10.1002/ijch.201300024.
10. Bootstrap - The world's most popular mobile-first and responsive front-end framework. 2016. URL: `http://getbootstrap.com/`.