Design Systems for Science

Sarah Poon, Drew Paine, Lavanya Ramakrishnan, Dan Gunter
Scientific Data Division, Lawrence Berkeley National Laboratory
[sspoon, pained, lramakrishnan, dkgunter]@lbl.gov

The growth in scientific data has resulted in fundamental changes in the user interaction model in the scientific software ecosystem. While traditionally command-line interfaces were commonplace, increasingly web based interfaces have become the primary mode of interactions for users. Scientific web interfaces are now used to access facilities, instruments, data, and computing across the DOE complex. These web applications are often custom designed to fit the unique characteristics of the workflows and infrastructure they support. The custom nature of these applications makes it difficult to reuse and repurpose code, particularly across scientific domains and facilities. This leads to challenges scaling software development efforts and results in an inconsistent user experience when you consider these applications as a collective. Design systems, which are a collection of standardized UI components and guidelines, have been developed in industry to address these same issues. We believe that a design system for science can help address these challenges in the DOE environment by providing a codified standard for scientific user interfaces that can be reused and customized.

Challenge: Scientists commonly interface with several software applications (often simultaneously) in order to conduct their work. For example, a scientist running an experiment at a lightsource might use the lightsource control software, the software at a supercomputing facility to process or archive the data, data archives to retrieve reference data, and software developed by their project to analyze the results. This software includes GUI components written by different software teams and is likely to have little standardization across the suite of these tools. This can lead to poor learnability, a higher risk of user errors, and an overall inconsistent user experience.

Several factors require scientific software to be customized in order to be effective, including differences in compute infrastructure, domain specific data, and data collection methods. This encourages a pattern of creating from the ground up bespoke software that cannot be easily adapted for another use. This is both inefficient and non-scalable from a development standpoint.

Shared code libraries can help with developer efficiency and scaling. When it comes to collaborative, web-based software, several frontend frameworks such as Bootstrap [1] and Material Design [2] also lower the burden of developing custom software by providing standardized graphic design and UI components. However, these components are often low level, such as checkboxes and radio buttons, and do not provide sufficient guidance on how to combine these components to create usable workflows. In addition, these frameworks are typically optimized for commercial applications and need to be adapted for scientific purposes. For example, science applications often require data density in their displays, whereas commercial applications favor greater use of negative space and larger fonts for improved readability. Finally, these frameworks are often missing components that rarely appear in commercial applications but are more common in scientific applications, such as ways to inspect files in scientific data formats such as HDF5 and FITS.

Opportunity: Despite the highly customized quality of scientific software, in our experience designing and developing scientific applications for a broad range of domains, from biological and earth sciences to experimental facilities, we have noticed a common set of workflow patterns. These patterns include parameterizing solvers, preparing data for archives, and conducting real time data taking. Even across different domains and at different facilities, we've found these patterns share similar workflows and user interface patterns. We can take advantage of these patterns to create customizable templates and guidance in the form of a design system for scientific applications. The process of creating the design system will subsume a detailed study of the similarities and differences in user interfaces for scientific workflows across groups and domains.

Incorporating a design system into the development process benefits both the developers and the end scientific users. A shared design library reduces the number of design decisions a developer must make and code and allows the work to scale in a standardized way. It also ensures that the software is more easily maintained and updated to reflect best practices. Creating a more consistent user experience through design systems improves learnability, time efficiency in completing tasks, and leads to fewer errors.

Maturity: A design system is a set of reusable components and layouts that can be combined to create most user interfaces. Commonly used in industry, the purpose of a design system is to provide a unified language for cross functional teams to create consistent products and make design and development teams more efficient[3]. In addition, they codify usability principles and best practices backed by extensive user research to ensure an intuitive, reliable user experience. Design systems can contain style guides, component repositories, and pattern libraries. Examining examples from industry, Material Design and its implementations as well as Bootstrap primarily focus on the style guide and component repository elements, whereas Salesforce's Lightning Design provides some additional pattern libraries [4].

A design system for science applications would be an adaptation and extension of these existing systems, optimizing the graphic design elements to better serve the needs of science use cases and fleshing out and codifying the common scientific workflow patterns into reusable and customizable templates. It would not eliminate the need to build custom UIs for each science use case, but would make the design and development effort more efficient and maintainable and create more consistency across the DOE ecosystem. Evaluating the impact of a design system for science would involve examining gains in developer efficiency as well as factors in usability. To evaluate efficiency, we can compare the amount of development time needed to create a new web application against prior experiences developing similar applications. From a usability perspective, we can conduct evaluations to determine the impact across the collective tools used by science projects.

We believe that the role of collaborative, web based applications will only increase in helping the DOE achieve its scientific mission. A design system for science can play a significant role in ensuring that the development effort is efficient, scalable, and maintainable and that the user experience is consistent across the DOE ecosystem.

References

1. https://getbootstrap.com
2. https://material.io/
3. Therese Fesseden, 2021. Design Systems 101. Retrieved from https://www.nngroup.com/articles/design-systems-101/
4. https://www.lightningdesignsystem.com/guidelines/overview/