Developing a Software Typology and User Experience Framework for Science

Lavanya Ramakrishnan, Daniel Gunter, Sarah Poon
Lawrence Berkeley National Laboratory

1. Research Questions

Software is ubiquitous and is critical to all scientific research from data collection at experiments or instruments to running large simulations or data analysis on large supercomputers. Scientific software often emerges as an effort to solve a specific need and evolves with the complex and iterative scientific inquiry and needs of the project. Software requires ongoing usability and user experience (UX) improvements in order to be a reliable, sustainable resource for user communities. Today, the user experience and user interfaces is often an afterthought in later stages of the project. This may occur for many reasons, such as the users and user requirements not being clearly defined and evolving with the project, tight budget constraints, or that the developers themselves are the primary users of the software.

Our team at Lawrence Berkeley National Laboratory (Berkeley Lab, LBNL) proposes to research and develop a *typology* and *design framework* for scientific software that, in conjunction, will improve user experience, software quality, and software sustainability. The primary research questions that this work will address are:

- What are the characteristics of scientific work and software and how can they be classified into a *typology* to better improve user experience and software sustainability?
- What are the elements of a *design framework* for user experience of scientific software?

Background. There is a substantial focus on improving the usability and reliability of commercial software through user experience efforts. User experience research, in particular, helps ground product designs in realistic user needs through qualitative methods such as interviews, observations, and usability evaluations [1]. In recent years, these approaches have seen some success in scientific contexts [2], [3]. However, traditional software engineering and

UX methods or processes [4] cannot be directly applied to scientific projects because of the flexible and fluid nature of work and personnel roles; this has been a barrier for wider adoption. As software has become critical to scientific innovation, usability is crucial for ensuring reproducibility and broadening access to downstream researchers and decision makers. Today, scientific software programs and projects do not have the methods, processes, or best practices that are necessary to ensure high quality usable software. The proposed work targets this particular gap by development of a typology and design system for scientific software.

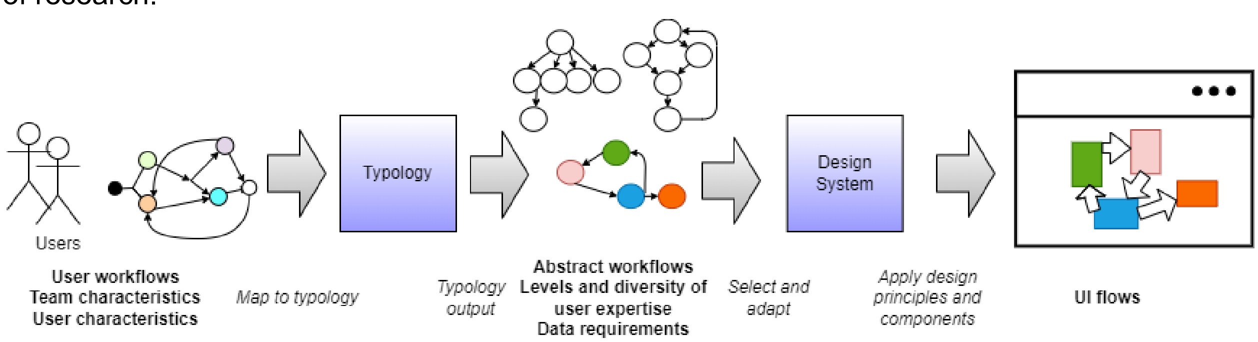
Scientific Ecosystems. Berkeley Lab is a multiprogram science laboratory that conducts scientific research on behalf of the Department of Energy seeking science solutions to some of the greatest problems facing humankind. Berkeley Lab is home to many scientific user facilities and large programs and projects. Team Science that brings together teams of individuals with different fields of expertise underpins all scientific work at Berkeley Lab. Thus, it is common to see domain scientists working alongside computer scientists and software engineers, and more recently UX researchers, to build tools and software. Berkeley Lab is home to many science user facilities, projects, and programs which provides us a rich ecosystem to study and address these challenges. We have identified nine projects and the five user facilities for this study. These projects are diverse in their sciences and project characteristics and thus representative of the larger scientific ecosystem in the community.

We use the term “scientific workflows” to describe a coherent unit of scientific work. Workflows are composed of multiple steps, such as: data collection through experiments and observations, data movement over a wide area network, running simulations and data analyses on desktops or large-scale computing systems at High Performance Computing (HPC) centers. Graphs or Directed Acyclic Graphs (DAGs) are used to capture the workflow.

In the last decade, the growth in scientific data and the ubiquity of computing have resulted in changes in the user interaction model. As the diversity and expertise of the user groups has changed, traditional text command-lines have been increasingly supplanted with web based interfaces. Scientific web interfaces are now used to access facilities, instruments, data, and computing. These web applications are often custom designed from generic technologies to fit the unique characteristics of the workflows and infrastructure (e.g., large-scale networking and compute infrastructure, domain specific data, and data collection methods). 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, for the scientists, an inconsistent and inefficient user experience

Proposed Work. Our proposed research tries to address these challenges in the scientific ecosystems through two key outputs: a typology of software and a design system. We will study scientific workflows and associated software stacks to generate a typology. By mapping project characteristics to typology categories, we will identify common patterns and construct abstract workflows that will be represented as UI flows (how a user progresses through a UI) in our design system. (Figure 1). The typology will also capture various characteristics of projects and include decision tools that allow projects to use the typology to make decisions on project investments. The design system will provide users with basic tools to get started on UI.

Figure 1. Captures the conceptual flow of our project and some of the outputs from each stage of research.



Multiple Uses of Typology of Software. A software typology will separate essential common features of scientific workflows from non-essential differences, providing a foundation for building tools that are reusable and adaptable. In addition to informing the development of the design system, the typology can help guide related aspects of scientific software development.

For example, we believe that typology can guide projects on when and how much to invest in UX for scientific projects. There are often a significant number of unknowns (e.g., features, target users) at the start of a scientific project; this discourages UI/UX investment decisions. However, delaying the introduction of UI/UX can negatively impact the outcome.

Clusters of key characteristics in the typology such as the size of the collaboration and target audience, the domain expertise of the software team (i.e., are the developers also "power users"), and the level of technical knowledge of the audience (in terms of experience programming and using compute infrastructure) can be correlated with the scale of UX and the scope of UI work needed for a software project. For example, the software for the Dark Energy Spectroscopic Instrument (DESI) was developed by and for power users and leveraged the learnings from prior cosmology surveys (including prior UX efforts), and needed less UX investment as a result. For the Materials Project, the diversity of the audience -- from theorists like the development team to experimentalists with skepticism about theoretical results -- meant that a great deal of UX and UI effort was required. In this case, the project team performed significant UX (without giving it that name) in a previous iteration while developing a product for an industrial partner, but still needed a large investment in UX and UI to apply this knowledge and experience to create an open-source product for the broader materials science community.

The typology could be also used to develop guidance in areas such as the level of software robustness needed and the types of workloads that next generation compute resources need to support. Finally, the typology can help scientific projects to frame what dimensions need to be uncovered during their initial discovery phases. For example, projects might identify that the target audience and their scientific workflows have not been well defined but are important for specifying the needed software. Guidance on methods, such as interviews to explore divergent ideas and rapid prototyping, can be developed and provided to help drive this discovery.

Design Systems to reduce the entry barrier for UX/UI. We believe that a design system for science, built atop a software typology, can help address the challenges in the scientific environment by providing standard components and higher-level "flows" for scientific user interfaces that can be reused and customized. Design systems, which are a collection of standardized user interface (UI) components and guidelines, have been developed in industry to address similar challenges [5]. 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. Developing a design system for science is essential to address usability while reducing cost of UX.

2. State of the Research

We examine our prior work and other research that serve as a foundation for this proposal.

Our Prior Work. Our previous research methods focus on understanding as well as verifying and validating the context of use and social challenges that often impact software design and development in science environments. Our previous work has included understanding and improving usability considerations of a workflow application programming interface [6], examining the human aspects of reproducibility [7], understanding data change [2], [8], and creating multi-dimensional classification models for scientific workflows [9]. We have also examined the software life cycle in science environments and identified the differences with commercial software development to outline ten principles we have developed to guide user engagement and software development based on our experiences [3] and develop multi-method user research processes specifically suitable for scientific environments [2], [8].

In our prior research, we described the high performance computing (HPC) socio-technical ecosystem as consisting of three primary user roles: research scientists, software engineers, and HPC staff [10], [11]. We found that software code is written by all three roles, with research scientists' coding efforts focusing on achieving scientific goals, and software engineers and HPC staff efforts help these codes run in a robust, efficient, and maintainable way. Yet, examining different dimensions in our typology may show a different makeup in these role dynamics. For example, we have worked with science collaborations with high levels of technical knowledge where most of the code base was written by the domain scientists, with minimal guidance from software engineers and HPC staff. As the technical abilities of science collaborations grow over time, the amount of software written by domain scientists versus software engineers, or the roles of personnel, can shift. Our proposed research will characterize the structure and evolution of the scientific socio-technical ecosystems in a broad set of scientific software teams, including smaller teams, teams with a high level of programming expertise, and projects that may not use large-scale computing resources.

Understanding Scientific Software Development. Previous research investigates software development practices by scientists themselves and in collaboration with software engineers [12], [4]. Software engineering research underscores that the practices and cultures of scientific software development work are distinct from commercial software engineering due to domain specific needs and challenges successfully fostering collaboration between scientists and engineers [4], [13], [14]. Scientific work does not usually follow traditional software engineering “best practices” and in many cases these practices are not applicable since end products or operational characteristics of the software are not known in advance [15]. Previous research has investigated the functional quality of software and its impact on reproducibility of findings [16], [17] which are important considerations for understanding the scientific software ecosystem. Additionally, studies have framed software as a material developed and used by stakeholders in the conduct of the scientific method [14] [18], [19] while also pointing out the cultural factors influencing the sharing of scientific software [20], [21]. Our prior work and these studies provide us a strong foundation to consider the various facets in a software typology.

User research in scientific software. User research has been shown to be valuable for scientific software [22]–[27] User research has been applied to various scientific domains including experimental cosmology, energy consumption, and biological, light microscopy projects [28], [29]. The importance of taking into account human factors in data systems has been identified before [30], [31]–[33]. While user research is increasingly recognized as important for scientific software, there are no guidelines, best practices, or frameworks that target the unique requirements of scientific software. Our proposed work will heavily enhance and spearhead human factor considerations in scientific software design and development.

Software Classifications, typology, and frameworks. There have been some works that have looked at various ways to classify and characterize software. Prior work has developed large-scale analysis of software usage and citation practices through a knowledge graph of software mentions and metadata generated using machine learning [34]. Typologies for software agents based on attributes and scope [35] and existing and proven strategies for achieving interoperability for large-scale, software-intensive systems have been proposed [36].

Studies have developed a systematic mapping study on the use of software engineering for scientific application development and their impact on software quality [37] and characterized the state-of-the-art research on software engineering taxonomies [38]. Previous work has also looked at software sustainability e.g., considering a schematic of activities in research software sustainability, and formalizing it in a directed graph model [39]. A framework to categorize different types of measures for the scientific software ecosystem and their relationships around funding, development, and scientific use has been developed [40]. Efforts have also been undertaken to introduce a framework for automatically extracting scientific software metadata from its documentation [41]. The focus of our typology is different but will benefit from these discussions on software related taxonomies and typologies.

3. Proposal Team Qualifications

Our user research (UX) team at Berkeley Lab focuses on user experience for scientific software. We work closely with scientific groups from various scientific domains including earth and environmental science, genomics, material sciences, and astrophysics, among others. Our work encompasses research, analysis, design, and implementation over a wide range of projects and user facilities (Tables 1 and 2). These projects have helped us build a repertoire of artifacts and experiences over a broad range of projects with different goals, user populations and needs, and project sizes that serve as a foundation for the proposed project.

Table 1: Exemplar projects we have partnered with or are currently involved in that we will interface with for the proposed project. Personnel connections are also identified in the table.

	Description and Website	**Personnel**
ESS-DIVE	The Environmental Systems Science Data Infrastructure for a Virtual Ecosystem (ESS-DIVE) is a data repository for Earth and environmental sciences. https://ess-dive.lbl.gov	Ramakrishnan, Poon support the ESS-DIVE UX activities, as needed.
NAWI	The National Alliance for Water Innovation (NAWI)'s Data Management and Analysis research is developing models and a user interface to standardize the assessment of the performance and benefits of new technologies for treating non-traditional waters. https://www.nawihub.org/	Gunter is a Co-PI in the NAWI team. Ramakrishnan supports the UX team.
PARETO	The Produced Water Application for Beneficial Reuse, Environmental Impact and Treatment Optimization (PARETO) is creating an open-source produced water decision-support application. https://www.project-pareto.org/	Gunter is a Co-PI on the PARETO team and, Ramakrishnan, Poon are active team members.
KBase	The Department of Energy Systems Biology Knowledgebase (KBase) is a software and data science platform designed to meet the grand challenge of systems biology: predicting and designing biological function. https://kbase.us	Gunter, Poon were previously part of the team. Our team continues to interact with KBase members regularly.
NMDC	National Microbiome Data Collaborative (NMDC)'s mission is to work with the community to iteratively develop and pilot an integrated, open	Ramakrishnan and Poon consult with the team on UX activities

	source microbiome science gateway that leverages existing resources and enables comprehensive access to multidisciplinary microbiome data and standardized, reproducible data products. https://microbiomedata.org/	and engage periodically with the NMDC team.
MP	The Materials Project (MP) is a widely used platform that provides open web-based access to computed information on known and predicted materials as well as powerful analysis tools to inspire and design novel materials. https://materialsproject.org	Gunter was part of the MP team and continues to engage with project participants on a regular basis.
FRONTIER	FRONTIER is a decision support system for assessing natural hazard impacts and restoration pathways for island(ed) communities.	Poon participates in the FRONTIER team UX design.
DESI	Dark Energy Survey Instrument (DESI) will measure the effect of dark energy on the expansion of the universe. https://www.desi.lbl.gov/	Poon, Ramakrishnan, Gunter work closely with DESI personnel.

Table 2: User Facilities at Berkeley Lab

Advanced Light Source (ALS)	Synchrotron light source capabilities and expertise available to a broad scientific community. https://als.lbl.gov/	Poon participated in the activities for the facility UIs.
National Center for Electron Microscopy (NCEM)	Cutting-edge instrumentation, techniques and expertise required for exceptionally high-resolution imaging and analytical characterization of a broad array of materials. https://foundry.lbl.gov/about/facilities/the-national-center-for-electron-microscopy-ncem/	Ramakrishnan has active collaborations with scientists at NCEM.
Joint Genome Institute (JGI)	Advance genomics related to clean energy generation and environmental characterization and cleanup. https://jgi.doe.gov/	Ramakrishnan has previously helped with the UX activities.
National Energy Research Scientific Computing Center (NERSC)	Primary scientific computing facility for the Office of Science in the U.S. Department of Energy. https://www.nersc.gov/	Ramakrishnan and Poon support NERSC UX activities.
Energy Sciences Network (ESnet)	Provides the high-bandwidth, reliable connections that link scientists at national laboratories, universities, and other research	Ramakrishnan actively collaborates with members of the facility.

	institutions, enabling scientific collaboration. https://www.es.net/	

PI **Ramakrishnan** is a Senior Scientist and Division Deputy in the Scientific Data Division at LBNL. Her research focuses on developing methods and tools to manage workflows and data while working closely with scientific groups and influencing the design of next-generation large-scale computing systems. Ramakrishnan has a research program that focuses on ensuring scientific pipelines are usable while also running efficiently on current and future supercomputing hardware resources and distributed systems. She leads the scientific UX team at Berkeley Lab focusing on studying and enumerating the way that scientists and communities use data and workflows to build usable tools for science projects and programs.

PI **Gunter** leads the Integrated Data Frameworks group at LBNL, which includes staff computer scientists, software engineers, UI/UX developers and designers, and domain experts in earth sciences, chemical engineering, and water treatment. He is the LBNL PI on several projects in the area of chemical and process engineering in the energy/water space: the Institute for the Design of Advanced Energy Systems (IDAES), Design Integration and Synthesis Platform to Advance Tightly Coupled Hybrid Energy Systems (DISPATCHES), National Alliance for Water Innovation (NAWI), and Produced Water Optimization Initiative (PARETO). Gunter is leading software development and user interface development in those projects.

PI **Poon** has been working as a UX designer and UI engineer at LBNL for over ten years. She partners closely with science projects to understand their workflow patterns and needs and translates these learnings into user interfaces that help them accomplish their scientific goals. Her design portfolio covers a variety of domains, such as astrophysics, climate science, and computational biology, as well as user facilities such as the Advanced Light Source (ALS) and the National Energy Research Scientific Computing Center (NERSC) supercomputing facility. While working in industry, she was an early adopter of and contributed to the development of design systems for expert use software.

4. Research Methodology

Figure 2: Detailed conceptual view of our research.

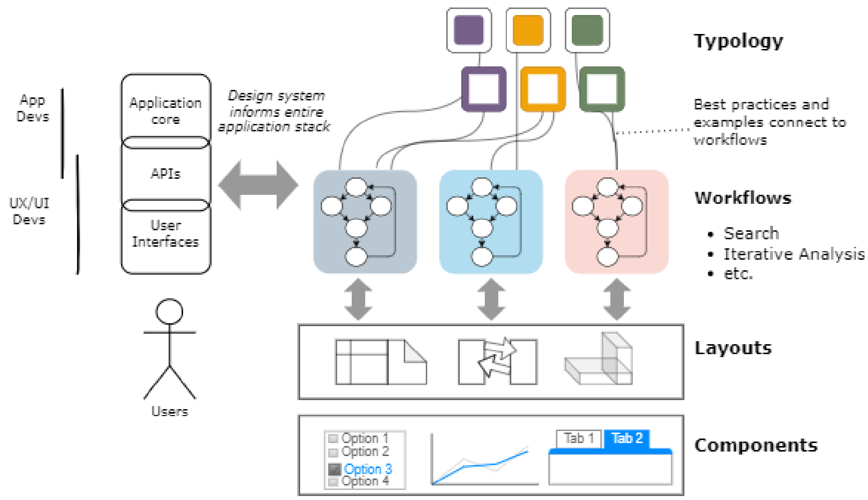


Figure 2 captures the detailed conceptual view of our research. Our research focuses on two key areas: 1) develop a typology that classifies workflows and the software systems based on a qualitative investigation of end-to-end scientific workflows and software systems and 2) develop a design system that captures the user interface flows, layout, and components for the major classes in the typology. The typology coupled with a design system will lay the foundation that is necessary for scientific ecosystems to provide a rich user experience that is critical for accelerating important scientific discoveries.

4.1. Development of Typology of Scientific Software

We will use traditional user experience research methods such as interviews, surveys, participant observation, and focus groups to understand the life cycle, malleability, and evolution of scientific software and the role of UX. Our goal in this study will be to cover the broad range of software from user scripts, collaborative code bases, libraries, and software stacks. We will also include the context around the software including interdependencies, open source and software sustainability practices, roles of the developers, organizational practices around software, and the culture of the teams (e.g., trust). We aim to identify and characterize dimensions that are specifically useful for impacting the design and development of scientific software in the context of usability and software sustainability.

User Research. User research will serve the foundation necessary to better understand the scientific practices and gather insights into the way scientists interact with software and user interfaces. Our qualitative user research will use multiple methods including semi-structured interviews, observation of scientific work, analysis of existing tools and infrastructures.

We will start with a broad survey that will draw upon existing data and projects (since 2014) that we have worked and/or work closely with (select projects and information listed in Tables 1 and 3) and also closely with staff and users from our user facilities (Table 2 and 4). We will gather and analyze artifacts (e.g., current UIs, user scripts, GitHub repositories) to understand the state of software systems and user interfaces. We will examine the evolution of the software and the UI and consider the inflection points in the project that impacted designs and outcomes.

We have identified projects and programs based on the year the project was started, size of the project, user base, the functionality provided by user interfaces, and characteristics of their software licensing and data. We have selected two small (< 10 staff), two medium (10 to 50), five large (>50) projects, based on project staff composition and size. There are four small (<100), three medium (100-1K), and two large projects (<1K) when we look at the user base size. Eight out of our nine projects are open source and five provide open data access, two have a mix of public and private data, and one is entirely private. This provides a diverse set of projects that we can study in detail to capture various aspects of scientific software. In addition to the nine projects, we will also study the five user facilities that typically have a more heterogeneous set of users with a wider range of skill sets in the users.

Next, we will interview select projects, with diversity across anticipated dimensions of the typology (Table 3). We will investigate the domain, types of workflows supported, team size, relationship between science teams and software teams, repository size, software stack, user base, compute and data resources used. We will augment these with a heuristic evaluation of interfaces and usability interviews. Questions we will address through these studies will include tasks supported by the software, missing functionality, and unofficial workarounds used to support software that are common in scientific ecosystems. The output from our user research will include expanding Tables 3 and 4, developing journey maps (capturing user actions in a timeline), and capturing project timelines.

Development of Scientific Software and Workflow Classification and Typology. We will use our user study to develop a categorization on basic dimensions and follow it up with qualitative thematic analysis [42] to identify most relevant or new dimensions.

Through our collective work designing and developing scientific software since 2014, we have recognized that common scientific workflow patterns exist, even across different scientific domains and facilities. Based on our prior experiences, we hypothesize that a workflow pattern is a key dimension in categorizing scientific software and developing a typology. The workflow patterns we have identified thus far include: a) parameterizing an algorithm and comparing runs; b) curating dataset metadata for archives; c) real time instrument operation; d) visual analysis; e) lab notebooks and related methods for interactively building analysis workflows with code; f) scientific database search; g) scientific catalogs; h) tracking objects of interest.

Our ongoing work will involve doing a broad survey of scientific software to evaluate this hypothesis, uncovering additional workflow patterns, and identifying other dimensions that are useful for software development. We will develop a typology that will capture a set of abstract workflows and its associated characteristics in a framework that can be used to inform UX investment decisions and will inform the design system. We will demonstrate a decision framework built on top of the typology to make decisions around UX investment.

4.2. Design System

We will develop a first version of the design system [5] for scientific software interfaces that will be overlayed on top of the typology of scientific software. Figure 3 shows an example of a *UI flow* that was developed for a specific workflow.

Table 3: Preliminary mapping of projects to anticipated important dimensions of the typology.

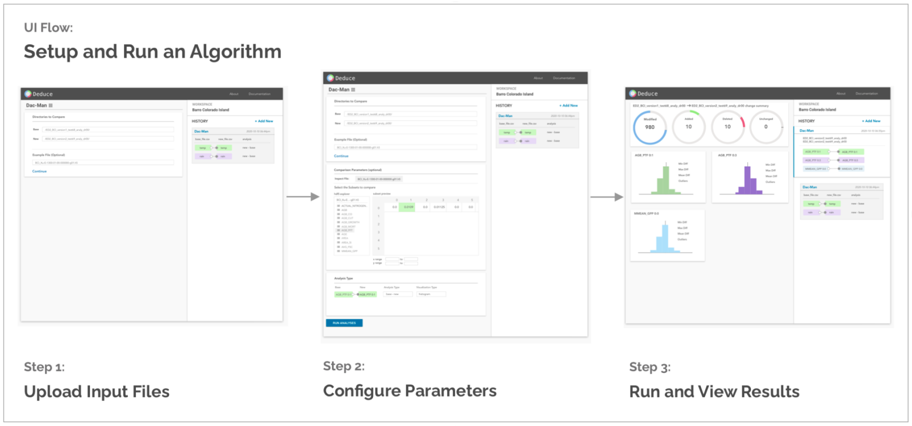
Project	**Project Started**	**Project staff composition (Small <10, Medium 10-50, Large > 50)**	**User Base (Small < 100, Medium 100-1K, Large > 1K)**	**User Interfaces**	**Software (Open Source, Private)**	**Data (Open data, Private)**
ESS-DIVE	2017	Medium sized project where code is developed and maintained largely by software developers	The active user base of the project is small with the total user base being medium. The users vary from users who rely on user interfaces to basic coding.	Repository search and data submission tools	Based on Metacat, a flexible, open source metadata catalog and data repository software	Data once published is openly available - no legally protected data
NAWI	2019	Large project composed of a number of small to medium sub-projects (we are mostly concerned with the WaterTAP software)	Users of the WaterTAP software are restricted researchers on related funding (25-50)	Desktop application for model analysis	Open-source, in GitHub	Public data
PARETO	2021	Small project composed of software developers and domain experts building code together	Initially a small user base of 10-20 companies.	Desktop application for running optimizations	Open-source, in GitHub	Legally protected data from collaborations with industry
IDAES	2016	Large project team across 3 national labs and 4 universities.	Medium user base of developers, students, and some industry stakeholders.	Desktop or Jupyter Notebook model visualization tool	Open-source, in GitHub	Mix of public data and legally protected data from collaborations with industry
KBase	2011	Medium to large sized project where code is developed by an interdisciplinary team	The total registered users is over 20K.	Tools for searching and analyzing data and tracking scientific workflows.	Open-source.	Data once published is openly available - no legally protected data
NMDC	2019	Medium to large sized	Small to medium sized user	Data portal for	Open-source (not	Data once

		project where code is developed by an interdisciplinary team	base at this point but expected to grow in the coming years.	searching and discovering data.	released yet)	published is openly available.
MP	2011	Medium size team, with large group of collaborators in theoretical and experimental materials science.	The total registered user base is over 100K. There are thousands of users active in a given month.	UI and API to search for materials and tools for data analyses	Open-source	Open data
FRONTIER	2021	Small to medium sized project with a separate software development team.	The user base is small, primarily consisting of local utility planners and policymakers with no coding experience.	Decision support tool for comparing resilience strategies.	Private	Private
DESI	2020	Large international collaboration from over 70 institutions.	Primary users are collaboration members but published data is meant for the general public.	Experts use tools and public archives.	Open-source, in GitHub	Public once published - no legally protected data

Table 4: Preliminary mapping of facilities to anticipated important dimensions of the typology. All facilities serve a large number of users > 1K.

Facility	Year established	User Interfaces
ALS	1993	Proposal and experiment safety portals; Beamlines have instrument operation and analysis tools
NCEM	1983	Limited interfaces to apply for accounts
JGI	1974	Multiple user interfaces for managing allocations, data.
NERSC	1974	Command line interfaces and web tools for monitoring and managing compute resources and allocations.
ESnet	1986	Command line interfaces and web tools for monitoring and managing network resources

Figure 3: An example of a UI Flow. Generizalied UI Flows will be developed.



A design system is commonly used in industry to provide a unified language for cross functional teams to create consistent products and make design and development teams more efficient [3]. Design systems can contain style guides, component repositories, and pattern libraries. For example, Material Design [43] and Bootstrap [44], both commercially developed, are primarily style guides with component repository elements, and Salesforce's Lightning Design provides some additional pattern libraries [4]. 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 [45] and FITS [46].

While there are differences across scientific disciplines, there are often a common set of data and workflow patterns. We can take advantage of these patterns to create customizable templates in the form of a design system for user interfaces for science. A design system for science applications would be an adaptation and extension of existing systems for scientific workflows and use cases, 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, but would make the design and development effort more efficient and maintainable and create more consistency across the scientific ecosystem.

The design system will consist of UI flows, layouts, and components and usage guidelines. UI flows represent the flows through the software while layouts represent how an individual page or step is organized (e.g., data dashboards, configuration panels). Components are lower level pieces, such as forms, tables, node diagrams, and equation editors.

A key question we will also need to address is how to evaluate the effectiveness of the design system for science. This is an area that is under explored and we will start with usability evaluation and explore other methods as appropriate. 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.

We will use the typology to develop common UI flows and provide guidelines on when and how to use the UI workflows. We will distribute Figma [47] files showing workflows as a set of user flows and JavaScript React templates for selected workflows. Figma is a collaborative, cloud-based sketching tool for creating mockups and click-through prototypes that is easy to share and disseminate. React is our chosen frontend framework because it has become a popular framework within the scientific software community. For example, NERSC and the Materials Project, among others, use React. Its component-based architecture maps well to design systems and the idea of reusable UI features. Finally, we will also work with one or two projects to use the early versions of the design system.

5. Work Plan

This project brings together a unique diverse team of personnel, building on previous work and existing strong collaborations with science projects. PI Ramakrishnan will lead and manage the overall project. The project will be managed through weekly teleconferences where progress on activities, partnerships, challenges, and risks will be discussed. We will work closely with partners identified in Tables 1 and 2 above (Letters of support included in Appendix). We will hire an UX researcher who has experience with user research processes and design (job posting is currently and description included in Appendix).

Study existing artifacts [July’22 - Aug’22] [Poon, Gunter, Ramakrishnan]. We will conduct a broad survey of scientific software projects, scoped to projects within LBL, covering multiple domains and user facilities. During this process, we aim to gather artifacts from these projects, including screenshots of existing interfaces and lists of code repositories and project websites. Existing interfaces will undergo an expert heuristic evaluation and potentially usability studies in order to identify opportunities for improving potential UI patterns.

Prototype Design System UI flows [Sep’22] [Poon, UXResearcher]. Based on initial patterns that emerge from the artifact gathering and initial interviews, we will begin prototyping aspects of the design system, focusing on UI flows. The output from this activity will be Figma files containing UI flows for select scientific workflows and React based application strawman that encapsulate these flows as customizable templates. We will partner with a science project to iterate on and understand how these design system elements might work in practice.

Conduct interviews with select projects [Aug’22 - Nov’22] [Poon, UXResearcher]. Based on the outcomes of the initial artifact gathering activities, we will select projects to conduct a deep dive into their scientific workflows and processes. This will involve interviews with project scientists and software engineers, contextual inquiry to uncover supporting scripts and “shadow” systems and understand software and infrastructure use, and potentially surveys. **Analyze qualitative data and develop dimensions of typology [Dec’22-Jan’23] [UXResearcher, Poon].** We will do a thematic analysis on the interview data and artifacts to cluster aspects of the various science software and uncover patterns. In addition, for select projects, we will begin creating journey maps of their scientific workflows. A key outcome from this work will be uncovering what dimensions are most useful in informing the types of software needed to support projects' science goals. To accomplish this, we will compare the results of the thematic analysis to the design system prototype learnings, to identify dimensions that best inform design. To broaden this perspective, we will also discuss this with systems designers and software engineers to identify additional dimensions needed for their design activities.

Characterize typology and map typology to existing projects [Jan'23-Feb'23] [Gunter, Ramakrishnan]. Based on the dimensions identified in the previous activities, we will map projects and organize the dimensions into a structure. Some of these dimensions will be used to define a journey map template for the projects and develop decision tools.

Develop generic design system elements [Jan'23-Mar'23] [Poon, UXResearcher]. We will conduct a comprehensive UI audit of existing UI's, in order to build a broad inventory of components, layouts, and other UI elements, and then cluster them into common patterns. The categories of UI elements will inform the overall structure of our design system, and validate or invalidate our hypothesized organization of component, layout, and flows.

Develop Design System UI flows [Feb'23-Apr'23] [Poon, UXResearcher]. Using our inventory of UI flows and their groupings, we will generalize them into our catalog UI flow patterns. These will be codified into Figma (a collaborative interface design tool) files for design purposes. A selection of these patterns will be implemented into React (Javascript library for building web UI) application templates. We will also draft our initial documentation.

User Research feedback on the Design System with select sub-group [Apr'23-May'23] [All]. We will partner with a select projects to integrate the UI flow portion of the design system into their software stack. This work will help us uncover any missing patterns and further learn what scientists need in order to use and customize the design system for their own needs.

Update Design System and Website Launch [May'23-Jun'23] [All]. We will iterate the UI flows as well as the accompanying documentation. We will make our Figma files and React templates, with guidelines for use, available for download on our website http://ux.lbl.gov.

6. Output

We will disseminate our results at various stages of development through a public project website and a public GitHub code repository. The website will document the concepts and application of the typology and design system, making the results highly accessible and usable. We will continuously improve this documentation with usability tests and feedback. The source code and other artifacts in GitHub will be available via download or, in the future, through a package manager. A plan for long-term dissemination and sustainability of the software and documentation is beyond the scope of this one-year project. However, we will create the artifacts using common conventions for open-source repositories and data, allowing users to contribute more easily in the near-term and providing a foundation for long-term sustainability.

The proposal team will also publish research results as open-access archive reports and peer-reviewed conferences and journals, as appropriate. The proposed work will change the role of software in the scientific ecosystem and significantly improve the pace of scientific innovation and broader accessibility. We believe that the role of web based applications will only increase in helping the community achieve its scientific mission. Our strong partnerships with scientific communities ensures intellectual transfer across environments. A typology coupled with a design system will ensure that the software development effort is efficient, scalable, and maintainable and that the user experience is consistent across the scientific ecosystem.

7. Budget Justification Summary

PI Ramakrishnan (LBNL) will lead the proposed team and manage the overall project. Poon and the UX Researcher will conduct the qualitative user research. Gunter will provide a supporting role on the project helping with data analyses and as an interface to projects that will benefit from the design systems.

Empirical Research Methods

Our study is investigating scientific software ecosystems at Lawrence Berkeley National Laboratory, a US Department of Energy national lab.

Artifact Collection and Study. Our initial study will involve existing artifacts from the projects including software source code, GitHub issue and pull request conversations, examples and tutorials, presentation slides, stakeholder meeting notes, user scripts, and user interfaces. We will also rely heavily on our user research data, collected continuously since 2014, that includes hundreds of semi-structured interviews, participant observations, surveys, project artifacts including user interfaces developed) gathered from a diverse set of domains.

Interviews. We will conduct semi-structured interviews (i.e., interviews in which the overall theme and some of the questions are predetermined, while other questions are asked as part of an open-ended discussion) with project participants over a period of two to three months. We will start from existing interview transcripts from 2014 (~150 interviews)[2], [3], [6]–[8], [10], [11], [48], [49]. In addition, we expect to interview 10-25 staff, with the exact number depending on the results of the artifact and literature survey study, about their work with scientific software. Our goal is to ensure sufficient diversity in the projects and project participants. We will consider various dimensions in our selection process including characteristics of the projects, personal identity and role identity. Interviews will be recorded, transcribed, and cleaned by the team. Our interview protocol is designed to learn about various aspects of an individual's research projects, focusing on workflows and software they work with. Previous studies have discussed the optimal number of interviews for studies where you reach saturation on new insights gained and the recommended numbers can vary from 10 to 40 for a diverse study like this one [50], [51]. While scientific ecosystems are different and numbers are not entirely clear, we believe at this time that upto 25 participants will be sufficient since we have previous interviews and artifacts to build on, letting us focus in the interviews on missing knowledge.

Data Analyses. We will analyze our data using an adaptation of the grounded theory process [52], open coding transcripts for responses to our questions and emergent ideas and assess these codes in relation to the literature identified earlier. Coding enables us to distinguish various dimensions of the typology among our interviewees. We will derive common themes that will inform our typology.

Evaluation of Typology. To evaluate the typology, we will emulate its intended use by third parties by performing a cross-cutting study that applies the typology to a number of scientific projects not initially considered, and comparing the expected workflows with those actually deployed by the project. While still mostly qualitative, we hope this study can help identify gaps and ambiguities, as well as establish a method for evolving the typology in the future. As the opportunity arises, we will also perform user research into the application of the typology for new scientific projects.

Evaluation of the Design System. We will work closely with project developers to test out the design system. We will use usability studies, collaborative design sessions, and possibly hackathons to evaluate early versions of the design systems in the context of two or three projects. Our initial guess of the two projects/facilities we will work closely with include PARETO and NERSC but this might change depending on the timelines and milestones of the scientific projects in relation to our goals. Our project choice is based on capturing multiple dimensions of project characteristics as shown in Table 3. For example, with PARETO and NERSC we can cover small and large sized projects, newer and more well-established to show the usefulness of the design system in different types and lifecycle stages of the project.

If we need additional data, we may augment this direct evaluation with speculative evaluation of how a design system might have affected an existing interface. This would be grounded in reality by restricting it to UIs for which we have been on the development team, so that we can leverage detailed knowledge of the requirements and have access to artifacts such as meeting notes and source code history.

References

[1] "What is UX research?," *Interaction Design Foundation*. https://www.interaction-design.org/literature/topics/ux-research (accessed Nov. 18, 2021).

[2] D. Paine, D. Ghoshal, and L. Ramakrishnan, "Experiences with a Flexible User Research Process to Build Data Change Tools," *Journal of Open Research Software*, vol. 8, no. 1, 2020, doi: 10.5334/jors.284.

[3] L. Ramakrishnan and D. Gunter, *Ten Principles for Creating Usable Software for Science*. 2017. doi: 10.1109/eScience.2017.34.

[4] J. Segal, "Software Development Cultures and Cooperation Problems: A Field Study of the Early Stages of Development of Software for a Scientific Community," *Computer Supported Cooperative Work*, vol. 18, no. 5, pp. 581–606, 2009, doi: 10.1007/s10606-009-9096-9.

[5] T. Fessenden, "Design systems 101," *Nielsen Norman Group*, 2021.

[6] L. Ramakrishnan, S. Poon, V. Hendrix, D. Gunter, G. Z. Pastorello, and D. Agarwal, "Experiences with User-Centered Design for the Tigres Workflow API," in *2014 IEEE 10th International Conference on e-Science*, Oct. 2014, vol. 1, pp. 290–297. doi: 10.1109/eScience.2014.56.

[7] D. Ghoshal *et al.*, "Experiences with Reproducibility: Case Studies from Scientific Workflows," in *Proceedings of the 4th International Workshop on Practical Reproducible Evaluation of Computer Systems*, Virtual Event, Sweden, Jun. 2020, pp. 3–8. doi: 10.1145/3456287.3465478.

[8] D. Paine and L. Ramakrishnan, "Surfacing Data Change in Scientific Work," in *Information in Contemporary Society*, 2019, pp. 15–26. doi: 10.1007/978-3-030-15742-5_2.

[9] L. Ramakrishnan and B. Plale, "A multi-dimensional classification model for scientific workflow characteristics," in *Proceedings of the 1st International Workshop on Workflow Approaches to New Data-centric Science*, Indianapolis, Indiana, Jun. 2010, pp. 1–12. doi: 10.1145/1833398.1833402.

[10] N.-C. Chen, S. Poon, L. Ramakrishnan, and C. Aragon, "Considering Time in Designing Large-Scale Systems for Scientific Computing," 2016.

[11] N.-C. Chen, L. Ramakrishnan, S. S. Poon, and C. Aragon, "Harnessing Complexity in High Performance Computing Ecosystems: A Complex Adaptive Systems Framework," 2019. [Online]. Available: https://scholarspace.manoa.hawaii.edu/handle/10125/60065

[12] D. Heaton and J. C. Carver, "Claims about the use of software engineering practices in science: A systematic literature review," *Information and Software Technology*, vol. 67, pp. 207–219, 2015.

[13] D. F. Kelly, "A software chasm: Software engineering and scientific computing," *Software, IEEE*, vol. 24, no. 6, pp. 120–119, 2007.

[14] "Keeping Software Present: Software as a Timely Object for STS Studies of the Digital," *digitalSTS*. pp. 423–446, 2019. doi: 10.1515/9780691190600-029.

[15] J. C. Carver, "Software engineering for computational science and engineering," *Comput. Sci. Eng.*, vol. 14, no. 2, pp. 8–11, 2012.

[16] C. Goble, "Better software, better research," *IEEE Internet Comput.*, vol. 18, no. 5, pp. 4–8, 2014.

[17] Z. Merali, "Error: why scientific programming does not compute," *Nature*, vol. 467, no. 7317, pp. 775–777, 2010.

[18] D. Paine and C. P. Lee, "'Who Has Plots?': Contextualizing Scientific Software, Practice, and Visualizations," *Proceedings of the ACM on Human-Computer Interaction*, vol. 1, no. CSCW, Nov. 2017, doi: 10.1145/3134720.

[19] M. L. Cohn, "Keeping Software Present: Software as a Timely Object for STS Studies of the

Digital," in *digitalSTS: A Field Guide for Science & Technology Studies*, J. Vertesi, D. Ribes, C. DiSalvo, Y. Loukissas, L. Forlano, D. K. Rosner, S. J. Jackson, and H. R. Shell, Eds. Princeton University Press, 2019, pp. 423–446. [Online]. Available: http://www.jstor.org/stable/j.ctvc77mp9.31
[20] J. Howison and J. D. Herbsleb, *Scientific Software Production: Incentives and Collaboration*. Hangzhou, China: ACM, 2011. doi: 10.1145/1958824.1958904.
[21] D. Paine, "Software and Space: Investigating How a Cosmology Research Group Enacts Infrastructure by Producing Software," PhD, University of Washington, 2016. [Online]. Available: https://hdl.handle.net/1773/36979
[22] W. Albert and T. Tullis, *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Newnes, 2013.
[23] C. Aragon and S. Poon, "The impact of usability on supernova discovery," 2007.
[24] T. Boellstorff, *Ethnography and virtual worlds: A handbook of method*. Princeton University Press, 2012.
[25] S. B. Davidson and J. Freire, "Provenance and scientific workflows: challenges and opportunities," 2008.
[26] D. Nicolini, *Practice theory, work, and organization: An introduction*. Oxford University Press, 2013.
[27] J. A. Maxwell, *Qualitative Research Design: An Interactive Approach*, 3rd ed. Sage Publications, Inc, 2012.
[28] C. R. Aragon and S. S. Poon, "Designing Scientific Workflow Management Systems for Data-Intensive Astrophysics Projects," 1998.
[29] C. Macaulay *et al.*, "Usability and user-centered design in scientific software development," *IEEE Softw.*, vol. 26, no. 1, pp. 96–102, 2009.
[30] P. Ziegler and K. R. Dittrich, "Three decades of data integration-All problems solved?," in *IFIP congress topical sessions*, 2004, pp. 3–12.
[31] A. Halevy, A. Rajaraman, and J. Ordille, "Data integration: the teenage years," in *Proceedings of the 32nd international conference on Very large data bases*, 2006, pp. 9–16.
[32] L. J. Seligman, A. Rosenthal, P. E. Lehner, and A. Smith, "Data Integration: Where Does the Time Go?," *IEEE Data Eng. Bull.*, vol. 25, no. 3, pp. 3–10, 2002.
[33] C. Goble and R. Stevens, "State of the nation in data integration for bioinformatics," *J. Biomed. Inform.*, vol. 41, no. 5, pp. 687–693, 2008.
[34] D. Schindler, F. Bensmann, S. Dietze, and F. Krüger, "The role of software in science: a knowledge graph-based analysis of software mentions in PubMed Central," *PeerJ Computer Science*, vol. 8. p. e835, 2022. doi: 10.7717/peerj-cs.835.
[35] M. S. Meenakshi and P. Anand, "A framework on typology of software agents," *Int. J. Inform. Technol.*, vol. 2, no. 2, pp. 251–254, 2010, [Online]. Available: http://csjournals.com/IJITKM/PDF%203-1/15.pdf
[36] P. H. D. Valle, L. Garcés, and E. Y. Nakagawa, "A Typology of Architectural Strategies for Interoperability," in *Proceedings of the XIII Brazilian Symposium on Software Components, Architectures, and Reuse*, Salvador, Brazil, Sep. 2019, pp. 3–12. doi: 10.1145/3357141.3357144.
[37] E.-M. Arvanitou, A. Ampatzoglou, A. Chatzigeorgiou, and J. C. Carver, "Software engineering practices for scientific software development: A systematic mapping study," *J. Syst. Softw.*, vol. 172, p. 110848, Feb. 2021, doi: 10.1016/j.jss.2020.110848.
[38] M. Usman, R. Britto, J. Börstler, and E. Mendes, "Taxonomies in software engineering: A Systematic mapping study and a revised taxonomy development method," *Information and Software Technology*, vol. 85, pp. 43–59, May 2017, doi: 10.1016/j.infsof.2017.01.006.
[39] S. Druskat and D. S. Katz, "Mapping the Research Software Sustainability Space," in *2018 IEEE 14th International Conference on e-Science (e-Science)*, Oct. 2018, pp. 25–30. doi:

10.1109/eScience.2018.00014.
[40] J. Howison, E. Deelman, M. J. McLennan, R. Ferreira da Silva, and J. D. Herbsleb, “Understanding the scientific software ecosystem and its impact: Current and future measures,” *Res. Eval.*, vol. 24, no. 4, pp. 454–470, Jul. 2015, doi: 10.1093/reseval/rvv014.
[41] A. Kelley and D. Garijo, “A framework for creating knowledge graphs of scientific software metadata,” *Quantitative Science Studies*, vol. 2, no. 4, pp. 1423–1446, Dec. 2021, doi: 10.1162/qss_a_00167.
[42] R. E. Boyatzis, *Transforming Qualitative Information: Thematic Analysis and Code Development*. SAGE, 1998. [Online]. Available: https://play.google.com/store/books/details?id=_rfClWRhIKAC
[43] “Material.io.” https://material.io/
[44] M. Otto, J. Thornton, and Bootstrap contributors, “Bootstrap.” https://getbootstrap.com (accessed Apr. 18, 2022).
[45] M. Folk, G. Heber, Q. Koziol, E. Pourmal, and D. Robinson, “An overview of the HDF5 technology suite and its applications,” in *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*, 2011, pp. 36–47.
[46] D. C. Wells and E. W. Greisen, “FITS-a flexible image transport system,” *Image processing in astronomy*, 1979, [Online]. Available: https://adsabs.harvard.edu/full/record/coll/ipia./1979/1979ipia.coll..445W.html
[47] “Figma website.” https://www.figma.com/
[48] S. Swaid, M. Maat, H. Krishnan, D. Ghoshal, and L. Ramakrishnan, “Usability heuristic evaluation of scientific data analysis and visualization tools,” 2017.
[49] D. Paine and L. Ramakrishnan, “Understanding interactive and reproducible computing with jupyter tools at facilities,” 2021, [Online]. Available: https://escholarship.org/content/qt9n11k2zm/qt9n11k2zm.pdf
[50] “How many participants for a UX interview?,” *Nielsen Norman Group*. https://www.nngroup.com/articles/interview-sample-size/ (accessed Apr. 26, 2022).
[51] “How many participants for quantitative usability studies: A summary of sample-size recommendations,” *Nielsen Norman Group*. https://www.nngroup.com/articles/summary-quant-sample-sizes/ (accessed Apr. 26, 2022).
[52] B. G. Glaser and A. L. Strauss, *The discovery of grounded theory: Strategies for qualitative research*. Routledge, 2017. [Online]. Available: https://www.taylorfrancis.com/books/mono/10.4324/9780203793206/discovery-grounded-theory-barney-glaser-anselm-strauss