Running Online User Studies with the reVISit Framework

Z. Cutler[1], J. Wilburn[1], H. Shrestha[2], Y. Ding[2], A. McNutt[1], L. Harrison[2], and A. Lex[1,3]

[1]University of Utah, USA, [2]Worcester Polytechnic Institute, USA, [3]Graz University of Technology, Austria

Abstract

There are currently two main approaches for running online user studies: experimenters can use commercial survey tools, which are easy to use but can be costly, hamper reproducibility, and have limitations for complex stimuli; or they can build custom software to run and instrument a study, which is a laborious and complex task. In this tutorial, we introduce participants to a new, open-source alternative: the reVISit study platform. Many studies quickly reach a burdensome level of complexity, necessitating design of stimuli and experimental tasks as well as the study UI, data hosting, participant recruiting, randomization, etc. ReVISit ameliorates these problems and allows study designers to focus more on the research questions and stimulus design. ReVISit removes the tedium of study design by providing built-in components that most studies will need. ReVISit provides a domain-specific language and a notebook-oriented library that enables study designers to quickly create studies and deploy them as publicly accessible websites. This tutorial will introduce reVISit to the visualization community and allow community members to get hands-on experience with it through a series of practical examples. Participants will improve on a study until they have developed and deployed a study of an interactive, fully instrumented data visualization.

CCS Concepts

• ***Human-centered computing*** → *User studies;*

ReVISit [CWS*26] is an open platform designed to enable the visualization and HCI community to run sophisticated user studies through a web browser. ReVISit is a $2 mio USD investment by the US National Science Foundation into the visualization community. The proposed tutorial is a critical component of the community engagement for this project: we seek to both educate the community about how to run online user studies with reVISit, and to recruit community members to use and provide feedback on reVISit. In its fourth year, reVISit is now a stable platform: it is well documented, comes with many examples, and has been used in numerous classrooms and studies, including in visualization and HCI venues, which we document at `https://revisit.dev/adoption/`. We have held reVISit tutorials at VIS24 and 25, Georgia Tech, UUtah, UNC Chapel Hill, CHI25, and EuroVIS25. Through these presentations, we have refined our content to answer common questions, made it more interactive and engaging, and developed experience in presenting our software.

Through this course, we will show researchers what reVISit is, the problems it solves, and how to use it. The course roughly follows the online tutorials that are available at `https://revisit.dev/`, with new content and activities. Participants will first learn about the whole of the reVISit system, including all the new and cutting-edge features it ships with. After this introductory presentation, our participants will get hands-on experience as we construct a study with common reVISit elements. We will then introduce advanced topics, such as sophisticated randomization strategies (e.g., attention checks, breaks), integration with crowdsourcing platforms, and best practices for reproducibility.

The tutorial will combine a lecture-style presentation, where concepts are introduced, with hands-on exercises based on prepared examples. At all times, we will have multiple team members assisting participants with the issues they encounter.

ReVISit is a community-focused open-source project. This tutorial is part of a series of initiatives and planned events for the project. Our team maintains Slack and GitHub organizations to support platform users and project development. In these forums, students and researchers receive support for edge cases (e.g., unique study designs) and common infrastructure challenges (e.g., tool configuration). This expands an individual research lab's scope of expertise and support in conducting human-centered studies. From this course, we hope to widen the community of researchers with a solid foundation capable of running their own studies and the reVISit community (visualization) more generally. Some community members may want to publish their modifications back upstream to the reVISit GitHub repository. Our team is ready to review pull requests submitted by our community and collaborate with our community members to build new features and fix bugs. Our development roadmap is publicly accessible on our GitHub organization.

DOI: 10.2312/evt.20261002

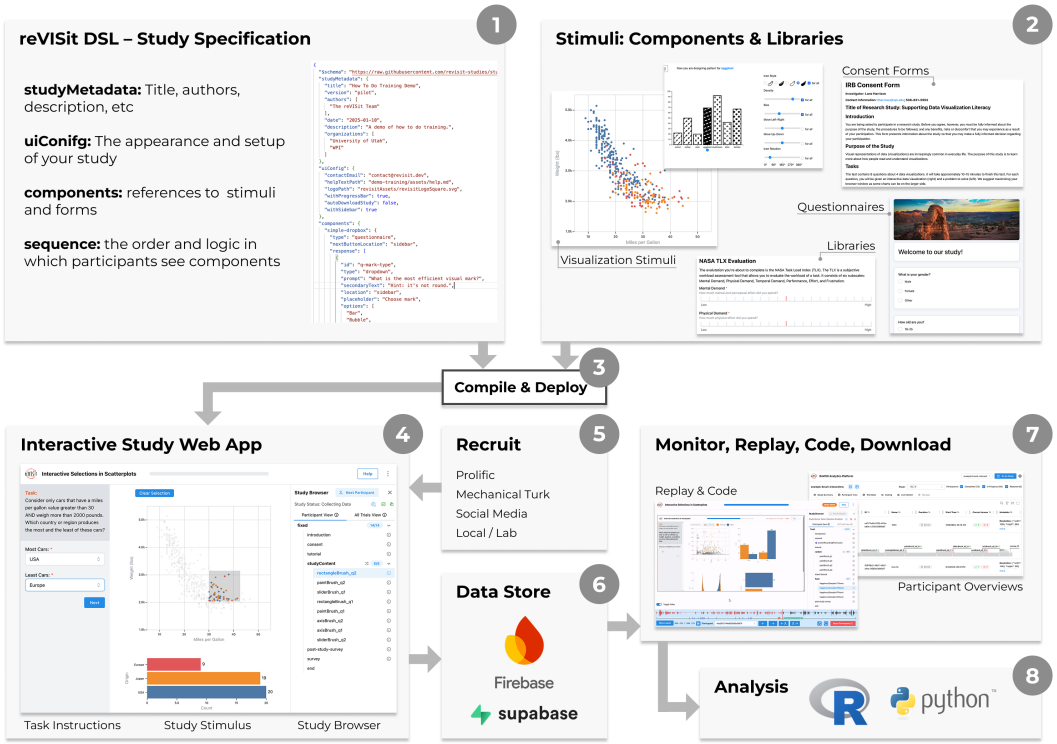

Figure 1: *Overview of the reVISit platform [CWS*26]. An experimenter generates a **(1) study reVISit specification** and **(2) study stimuli and components**. These are compiled to a deployable **(3) web app**. Data from the study can be **(4) downloaded** at the end of a trial, or **stored on a server**. Collected data can be **(5) analyzed** with external software (e.g., R or SPSS), or examined using the reVISit analytics interface.*

1. Intended Audience

Our ideal audience are researchers who want to run reproducible online user studies, generally characterized as people with **intermediate background knowledge**. Ideally, they would have some knowledge of JSON, web development, and study design principles, but our guided walkthroughs will assume no prerequisite knowledge. By the end of this course, these participants will be able to use the reVISit framework to build complex, open, and reproducible studies with custom interactive stimuli and advanced randomization strategies. They will understand how to recruit participants and how to export their data for more in-depth analysis.

2. Tutorial Content

Introduction to reVISit (40 min). We start the course with an introduction to reVISit. We will first address the challenges that currently exist for running visualization user studies in the browser, how the reVISit platform helps to alleviate these problems, and how it can be used to collect rich data. Next, we will discuss the novel features that make reVISit unique, including think-aloud support [CHNL25], participant replay, and provenance tracking.

ReVISit is designed to enable long-term reproducibility. In this course section, we will also discuss how reVISit experiments can be anonymously shared with reviewers and the general public, so that everyone can evaluate exactly what happened in a study. When reVISit code and data are published, others can also easily re-use parts of experiments or re-analyze the data. We consider these aspects a key differentiator to commercial platforms like Qualtrics.

Getting Started with reVISit (35 min). In this module, participants will learn how to clone, build, modify, and deploy an existing study. The standard pipeline for working with reVISit is to start by forking and cloning the study repository. Participants will then install all the necessary packages using yarn and serve the development server on localhost. Once dependencies are installed, participants will see a list of demo studies that come with the reVISit repository, including a placeholder for the tutorial content, which we will work together to fill out. Once the code is opened up locally, we will take a deep dive into the reVISit spec, a JSON-based Domain-Specific Language that reVISit uses to compile and build a study. Figure 1 shows an overview of the reVISit framework and the process for deploying a study. We then continue by creating our first study components: an introduction and a help page. These two MarkDown files serve as a simple introduction to our study and a help file for any study participant throughout the study. It is important to note that reVISit seeks to mitigate several of the pain points of experiment management and hosting. For this phase, participants are producing and editing simple text files, which reVISit compiles into a deployable study. Based on this initial work, we will introduce course participants to reVISit's admin functionalities, which target various pain points, such as the ability to skip questions or trial blocks to rapidly debug

experiment stimuli, to see the progress of the study, etc.

Break

Creating Interactive Stimuli (30 min). After the break, participants will learn how to integrate interactive stimuli such as data visualizations into reVISit. We will first introduce the basic principles of how such stimuli are integrated with reVISit, and explain how data is passed between revisit and study components. We'll start with an example in vanilla JavaScript, and will then continue to code up the same example in React. The features reVISit provides for each of these interactive component types are similar, so this should be a gentle introduction to working with JavaScript and React.

Provenance Tracking for Interactive Stimuli (15 min). ReVISit automatically stores study-level provenance, such as start/end time of each task, responses, task order, and window events such as browsed in/out and mouse move. Such data are particularly useful in the study design process, for example, to determine if participants are spending sufficient time in training and course content for a given study. Additionally, reVISit allows for easy integration with Trrack [CGL20], a state-based provenance tracking library developed by our team. Trrack can be used to store provenance on a task level, allowing researchers to make their own decisions about what parts of their system they want to track. If a study designer uses Trrack in their stimuli, reVISit will show a timeline of all interactions with the stimuli from the participant. This provenance data stream enables teams to better understand any potential issues with their stimuli and how participants are completing tasks. In the course, we will introduce Trrack first with a simple example — an extension of our interactive stimuli developed previously. This will help users understand both Trrack and how it integrates. At this point, we will generate some data by taking the study so that we can demonstrate the participant replay feature, which leverages Trrack to rehydrate the stimulus state.

Experiment Design (20 min). In this section, we will describe how reVISit supports different experimental designs by managing the flow of a study. Up to this point, we will have used only a linear sequence of study components, such as consent, training, and stimuli. In this module, we will introduce more advanced strategies for study design, including randomization, breaks, dynamic flows, and attention checks. reVISit allows for multiple types of randomization and complex study design by allowing the user to create their study order via a system of blocks, which we will demonstrate throughout the course. Each block contains a list of trials, and has a defined order using a linear, random, or Latin square approach. We will discuss when each randomization strategy is useful, and how to implement them. Blocks can be arbitrarily nested, allowing for any type of complex study design. Users can also define sampling for each block, enabling designs such as 'a participant sees 2 out of these 5 tasks'. Each block can also have interruptions added into the tasks, either randomly or at a consistent interval, which is useful for giving participants breaks, or for attention checks. A common problem in a crowd-sourced study is handling complex randomization when a participant quits an ongoing study. We will demonstrate how reVISit overcomes such situations through, for example, automatically re-assigning incomplete Latin square rows to later participants. Finally, we will introduce reVISitPy, our Python bindings that facilitate factorial designs. We will demonstrate how reVISitPy streamlines the entire workflow, from initial design and data collection to preliminary piloting and analysis, directly in a notebook.

Community Feedback (10 minutes). ReVISit is a community-focused open-source project. At the end of the tutorial, we will elicit comments and suggestions from participants to help us improve reVISit and help us shape our roadmap for the future.

3. Tutorial Presenters

Zach Cutler is a PhD student at the Visualization Design and HAVOC labs at the University of Utah. Zach has extensive experience with integrating provenance tracking with user studies on the web.

Lane Harrison is an associate professor of computer science in the Department of Computer Science at Worcester Polytechnic Institute. Lane directs the Visualization and Information Equity lab at WPI (VIEW), where he and students build visualization systems to better understand how people engage and use visualizations they encounter in day-to-day life. Lane has built prior visualization experiment frameworks (experimentr), which have been used in 10+ VIS and CHI papers beyond his own work and served as part of the inspiration for the reVISit initiative.

Alexander Lex is a professor of human-computer interaction at Graz University of Technology, and holds an appointment at University of Utah's Scientific Computing and Imaging Institute and the Kahlert School of Computing. Alex directs the Visualization Design Lab and conducts research on visualization methods and systems to help solve today's data analysis problems in the sciences. Alex has experience with building infrastructure and methods for provenance tracking and user studies.

4. Equipment

This tutorial will require a projector and WiFi. The tutorial participants will follow along, including cloning the repository and modifying the code to build their studies.

5. Acknowledgments

This work is supported by the National Science Foundation (2213756 & 2213757).

References

[CGL20] CUTLER Z. T., GADHAVE K., LEX A.: Trrack: A Library for Provenance Tracking in Web-Based Visualizations. In *IEEE Visualization Conference (VIS)* (2020), pp. 116–120. doi:10.1109/VIS47514.2020.00030. 3

[CHNL25] CUTLER Z., HARRISON L., NOBRE C., LEX A.: Crowdsourced Think-Aloud Studies. In *ACM Conference on Human Factors in Computing Systems (CHI)* (2025). doi:10.1145/3706598.3714305. 2

[CWS*26] CUTLER Z., WILBURN J., SHRESTHA H., DING Y., BOLLEN B., NADIB K. A., HE T., MCNUTT A., HARRISON L., LEX A.: Revisit 2: A full experiment life cycle user study framework. *IEEE Transactions on Visualization and Computer Graphics* (2026). 1, 2