Saving Data Journalism - Abstract

New York University Division of Libraries requests $49,557 for a planning grant to collaborate with the NYU Arthur L. Carter Journalism Institute, the NYU Center for Data Science, and ProPublica in order to develop a prototype for a software tool to capture and preserve data journalism websites in a scholarly archive. Storytelling with data has revolutionized modern reporting, and the dramatic increase in the production and popularity of data journalism projects can be seen both at news startups such as FiveThirtyEight and Vox.com, as well as at legacy news organizations like *The New York Times*, *The Washington Post*, and *The Wall Street Journal*. These stories are an important part of the historical record, yet due to their technological complexity they cannot currently be archived or preserved at libraries, newsrooms, or cultural institutions. As such, they are disappearing.

We plan to focus initially on a single kind of data journalism artifact, a database-reliant journalism story known as a news application or "news app." These projects are based on dynamic websites, often custom built by the news agency to allow readers to query, filter, and manipulate data to tell a story. Each news app has multiple components: a database; the data in the database; the graphical interface that appears in the browser; text; images; and other multimedia components. For future scholars to see the same look, feel, and functionality of the news application, all of these components must be saved and reassembled in the same way. Web crawlers only capture snapshots of the "front end" of a news app, the graphically appealing part that is seen through a browser, but cannot reach the "back end" data in the database. For this reason, they cannot be fully captured by any current web archiving technology.

We propose a prototype to capture dynamic websites through emulation, which will add to an existing open source computational reproducibility tool, ReproZip. The extension will be designed to quickly and easily pack and unpack dynamic websites, enabling the first large-scale collection, preservation, and discovery of these objects.

This work will be developed under the auspices of a planning grant and will take one year to implement. ProPublica has permitted the use of one of its data journalism projects, Dollars for Docs, as a test case. The prototype will be optimized to capture and preserve Dollars for Docs. After developing this minimum viable product, the team will communicate the findings to the larger digital archiving community and will seek additional funding to generalize the findings and apply the preservation strategies to additional data journalism sites.

Saving Data Journalism:
A prototype to enable systematic collection & preservation of dynamic news websites

Statement of National Need

From interactive visualizations to easy-to-use databases, storytelling with data has revolutionized modern reporting. Influential data journalism projects include the Panama Papers, "The Color of Debt" by ProPublica, and "Gun Deaths in Your District" by *The Guardian*. These paradigm-changing investigative stories made issues accessible to the public, brought down corrupt government officials, and prompted international changes in transparency, security, and human rights. However, many of these projects are too technologically complicated to be captured or archived by libraries, information vendors, or any current web archiving technology. As a result, they are disappearing.

New York University Libraries requests $49,557 from the IMLS National Leadership Grants for Libraries program for a one-year planning project to collaborate with the Arthur L. Carter Journalism Institute at NYU, the NYU Center for Data Science, and ProPublica to build a prototype software tool to capture dynamic data journalism websites and preserve them in a scholarly archive. The prototype will add to an existing computational reproducibility tool, ReproZip (Chirigati et al, 2016). The proposed extension will be designed to quickly and easily pack and unpack dynamic websites, enabling the first large-scale collection, preservation, and discovery of these projects. During the planning phase of this work, we will design and test the extension with the goal of building a scalable, customizable, preservation-friendly tool that would have the potential for wide adaptation in newsrooms across the country.

We plan to focus initially on a single kind of data journalism artifact, a database-reliant journalism story known as a news application or "news app." These projects are often custom built by the news agency to allow readers to query, filter, and manipulate data to tell a story. Each news app has multiple components: a database; the data in the database; the graphical interface that appears in the browser; text; images; and other multimedia components (Broussard, 2015). For future scholars to see the same look, feel, and functionality of the news application, all of these components must be saved and reassembled in the same way. Web crawlers only capture snapshots of the "front end" of a news app, the graphically appealing part that is seen through a browser, but not the "back end" data in the database. For this reason, the sites cannot be fully captured by any current web archiving technology (Boss & Broussard, 2017). ReproZip is a back-end tool that only preserves the database and server side of a news app. We propose to create a prototype that brings together front-end crawling with back-end reproduction in a single software package.

Urgency of need

Unless we act quickly to devise new ways of capturing and preserving data journalism, these important innovations will be lost to future historians. Many important works are already broken or gone. The "Fatal Flights" news app from the *Washington Post* was a cutting-edge story when it was first published in 2009, but today none of its data visualizations display. Similarly, the website for the *Philadelphia Inquirer's* high-profile 1997 investigation "Blackhawk Down" still exists, but few, if any, of the graphics, video, audio, or other multimedia components are viewable, as the software to read them is long out of date (Hansen & Paul, 2017). *The Guardian* coverage of the United Kingdom Member of Parliament (MP) expenses scandal of 2009 was internationally lauded for its innovative investigative reporting and data analysis, but fewer than 10 years later, the original websites for the project are either broken or nonexistent (Daniel & Flew, 2010).

The loss of these works is substantial, and soon more journalistic projects will disappear with the sun-setting of Adobe's Flash software. Adobe recently announced that as of 2020, it will no longer update or distribute the Flash Player (Adobe Systems Incorporated, 2017). This is a looming crisis for the archiving and preservation of interactive websites. Half a decade of interactive websites, including thousands of data journalism stories, were built in this now

outdated software; The New York Times found in a content audit of their digital assets that a full 19 percent of their interactive projects (comprising more than 40,000 URLs) relied on Flash (Heideman, 2017). It is imperative that libraries and cultural memory institutions find a solution for capturing and archiving these works as soon as possible.

Building on current web archiving technologies

To our knowledge, our proposed extension to ReproZip would be the first full-stack emulation archiving tool ever built for websites. But even though this will represent a new technology, one of its strengths is that it will build on and align with other important digital archiving projects and initiatives. Web archiving technology, such as the Heritrix tool that powers the Wayback Machine at the Internet Archive, has been very successful in capturing static Web pages. However, it is not able to capture the complex interactions and functionality of news applications and other dynamic websites that contain social media feeds, interactive maps, visualizations, and the ability to query information from a database (Boss & Broussard, 2017). Our tool will still build on Heritrix functionality in that it will also create a WARC file of the front-end in the ReproZip package. Another notable and important technology in this field is Webrecorder, an open-source tool that can capture dynamic website interactions by "recording" network traffic and processes in the browser (Boucher, 2016). A major limitation to this tool is that it cannot *automatically* capture all possible interactions. Web archiving using Webrecorder must be done manually, meaning that the user has to click on each link of a page to archive it. This is not feasible for news applications, as there are hundreds of thousands of queries possible on any given data journalism project. In this project we will incorporate the front-end, dynamic crawling technology of Webrecorder with the back-end capture of ReproZip to create a single archivable, preservable, and emulatable file.

Direct beneficiaries of this work include legacy and startup news organizations, future historians, and the general public. Of these audiences, newsrooms stand to benefit soonest. They are losing their born-digital interactive content at an alarming rate, yet few, if any, staff in modern newsrooms wake up each day thinking about digital archives and preservation. This is a direct result of the shuttering of newsroom libraries, a trend that began in the early 2000s when budget cuts and the disruption of the publishing industry lead newsrooms to close their libraries and lay off their archivists (Hansen & Paul, 2002; 2017). The absence of librarians and archivists in newsrooms has had dire consequences for born-digital news in particular. In order to better understand the nature of this problem, our team conducted an international survey that collected data on how news applications are built, maintained, and saved. The results, which will be published in a forthcoming issue of the journal *Digital Journalism*, helped us determine the coding languages and platforms most commonly used to build these projects. The most telling and disturbing finding in the survey related to the question, "Does the news organization or institution have an archiving system for news apps?" Of 36 organizations, only two, The New York Times and Politico, responded "yes" (Boss et al., 2017). In a follow-up conversation with a team of developers and editors at The New York Times, our team discovered an extremely fragile, ad hoc archiving strategy that lacks documentation, is maintained and understood by only one person, and does not address any aspect of digital preservation or impending software obsolescence.

Project Design

The goal of this project is to build an extension to the existing computational reproducibility tool, ReproZip, that will quickly pack and unpack dynamic data journalism websites into a single archivable, describable, and preservable .rpz file. This plan supports the goals of the IMLS Content & Collections initiative in that the software will enable librarians and archivists to capture and preserve dynamic cultural and research materials. By developing open software for the preservation of complex objects, libraries and archives will be better positioned to collect not only news applications, but also various research outputs, time-based media/artwork, and even computational environments.

In the fall of 2016, our team tested ReproZip as a capture and archiving tool in a pilot experiment with ProPublica on their news app "Dollars for Docs" (Groeger et al, 2010). While ReproZip successfully captured all of the app's back end files, it was unable to capture remote front end files (e.g. CSS, JavaScript), so the packaging step was incomplete. Our proposed extension to ReproZip will be designed to make it easier to pack dynamic websites specifically, will provide a better experience when setting up the website again from the artifact, and will capture the missing front-end pieces that the current tool cannot.

One assumption of this research, and of any emulation-based approach to archiving dynamic websites, is that the content creators (in this case, newsrooms and data journalists) will be willing and able to self-archive their work. The ReproZip extension and emulation tool we plan to build must be executed on a computer or server with access to the original software environment in which the news app was built. Therefore, data journalists would have to add a new step to their workflow when they build and publish their stories. It is important that this project solicit input from these stakeholders, as their buy-in will be critical to the success of the tool. However, we cannot solicit their input until we have a workable prototype for them to test. We propose to build an extension to ReproZip that can capture a news application in its entirety. Once we have done so, the next step, for which we plan to request project grant funds, will be to conduct user testing and quality assurance of the ReproZip extension on a variety of news applications at different newsrooms. During that larger phase of the project we could then solicit extensive feedback on the quality, convenience, and usability of the tool, and refine the design to fit the workflows of data journalists at newsrooms nationally and internationally.

Another assumption of this research is the ongoing development of ReproZip. The core components of the tool must be maintained and new plugins must be developed whenever needed (e.g.: to support new virtualization and emulation software). This is a safe assumption because ReproZip has institutional support and a growing and vibrant community that helps make sure the tool is maintained. ReproZip's continued existence ensures that news apps will be properly preserved and reproduced for years to come.

Outcomes of this project include an open-source prototype for an extension of ReproZip and documentation of its ability to pack and unpack interactive data journalism projects. The new extension will be composed of two essential components:

1. ReproZip & Webrecorder Integration. While ReproZip traces the back end of the web server, Webrecorder will execute simultaneously in order to record the front end files that come from external locations. Those two sets of files will be consolidated into one ReproZip package.
2. ReproZip Unpacker. When a secondary user replays the ReproZip package, a proxy integrated with Webrecorder Player will sit in front of the unpacked application, to direct requests either to the WARC contents or the application's environment (in the ReproZip file).

To bring more attention to the need to archive news apps and to advance work on a solution, another outcome of this project will be the dissemination of our research through popular channels and traditional academic venues. We have communicated about our work in two widely-read online publications, Harvard University's *Nieman Lab* and European Journalism Centre's *Data-Driven Journalism* (Wang, 2017; Broussard & Boss, 2017), as well as through multiple academic presentations and publications (Boss & Broussard, 2016; Boss & Broussard, 2017, Boss et al., 2017, Boss, Broussard & Revear, 2016; Broussard, 2015). We plan to continue these discussions at the next meeting of the National Institute of Computer Assisted Reporting (NICAR) conference and the annual Computation + Journalism symposium, which have a combined audience of more than 1,000. We will also submit more proposals to conferences related to digital archiving and preservation, including the annual "Dodging the Memory Hole" conference dedicated to saving online news, the Digital Preservation conference sponsored by the National Digital Stewardship Alliance, and the annual International Federation of Libraries Association (IFLA) News Media conference. As part of our digital outreach, we will share our findings via Twitter, where we have a combined total of about 4,000 followers, and where we actively read and post about digital archiving. Finally, we also plan to write a series of blog posts on the NYU Center for Data Science blog to engage with the reproducible science community.

ReproZip is a well-used and supported tool. It has been in development since 2012 and is used in domains from digital humanities to data science. This plan would add functionality to extend use to journalists creating digital stories with complex new technologies, not easily preserved or captured with current web archiving technologies. As the Keeping Emulation Environments Portable project recommends, any solution being developed for long-term preservation will be built with its future migration in mind, and thus must be open-source, and ideally built on open-source dependencies (European Commission, 2014). ReproZip and its dependencies are open-source and minimal. Additionally, ReproZip packages are generalized enough that they can be read by a multitude of virtualization applications, and aren't tied to any specific software. ReproZip meets all of the recommendations for a long-term, preservation-friendly system that can be updated, altered, and maintained by the larger libraries and museums community. This extension to ReproZip would enable the first large-scale collection, preservation, and discovery of complex interactive websites.

We plan to build this tool in conversation with several prominent news organizations producing data journalism, with the goal that newsrooms become incentivized, willing, and able to adopt a self-archiving workflow using ReproZip. We will be working directly with ProPublica on making sure our solution works for data journalists. The project graduate student will test and update the prototype, then create documentation and a flowchart of the workflow of using this tool with time estimates for every step. We will then solicit feedback on this workflow and the usability of the ReproZip extension from news organizations both at conferences and in informal conversations. This extension to ReproZip could also be updated or modified by other organizations in keeping with ReproZip's current license, which allows distribution and modification of ReproZip for noncommercial or commercial purposes. Additionally, we plan to reach out to other strategic partners within the digital archiving community for feedback and testing. We will find these partners through a scan of the membership in the Software Preservation Network.

Potential risks to the project are centered on technical challenges. Our team of experienced software developers and technical professionals are prepared to manage challenges. The underlying software architecture of Webrecorder and ReproZip may require adaptation in order to make the two programs compatible. To mitigate this risk, we have introduced the Webrecorder and ReproZip developers to ensure open communication and that they can work through technical issues that arise. To address intellectual property issues, we are working with NYU's general counsel to draft an agreement that will allow news organizations to use the technology to share their archives.

Below is a summary of the project plan:

Preparation and Evaluation
Weeks 1-4 (May 2018)

- Evaluate web crawlers and proxy options
- Hire developer and graduate student

Tool Creation and Development
Weeks 5-19 (June 2018-September 2018)

- Integrate Webrecorder crawler with ReproZip (developer)
- Create ReproZip/Webrecorder unpacker (developer)
- Create documentation

Testing and Communication
Weeks 9-19 (July 2018-September 2018)

- Test, iterate, and validate with ProPublica
- Solicit community feedback
- Document findings (graduate student)

Assessment and Dissemination

Weeks 19-50 (September 2018-April 2019)

- Widely circulate code and documentation
- Publish and disseminate findings
- Create a plan to preserve more apps, pending resource availability
- Apply for additional funding

The project will analyze the nature and structure of data journalism objects and environments and identify and assess potential technical solutions that can translate the most widely used formats and software into curatable forms. Full documentation and reporting will make this information widely available for review in the community. An open-source prototype will be built to demonstrate how universal tools can be created for a core set of functions and then adapted for a variety of projects. This prototype will be widely disseminated through publications and presentations that will engage the journalism and information science communities as well as the data curation profession.

National Impact

Our approach to saving news applications will allow a significant shift in libraries' strategies for archiving digital objects. Historically, the technique libraries use in archiving has been migration: print newspapers migrated to microform, and then microform newspapers to digitized PDF/A files; or 16mm film reels to VHS tapes, DVDs, and streaming MP4 files. This strategy has limitations, but has been largely successful for physical materials and static digital objects like text or images. For dynamic digital objects, the amount of time, work, and organization involved in migrating or upgrading every software dependency is impractical, if not unfeasible. Software packages and computational platforms change vastly over time. Digital archivists believe that to save these apps for the long term, we must emulate, not migrate, the object in its computational environment (Granger, 2000; Johnston, 2014; Rechert et al, 2012; Von Suchodoletz & Van der Hoeven, 2009). This new project will take an innovative kind of emulation that has been pioneered by the computational research community and will enable this technique to be extended to the journalism community and beyond.

ReproZip, the tool we plan to extend, can reproduce a myriad of applications, from databases to interactive visualization systems. With our planned extension, it will allow dynamic digital websites, such as news applications, to be emulated. It will produce a file type, a .rpz, which will be suitable for long-term preservation, and years from now could be opened and unpacked to emulate a website produced today. The .rpz file that ReproZip produces will contain all of the files, programs, and data needed to recreate the website, as well as the precise version of the operating system, software libraries, and other software dependencies needed for the site to display. Because our solution is generalized and not dependent on any specific technology, it can be used to preserve any dynamic digital website, irrespective of the database system, the location of the files, or the required software dependencies. The .rpz file is not dependent on any emulator or virtualization software either: the tool will capture enough information and metadata from the digital objects, which can then be translated to any technology. Such features provide stronger long-term preservation guarantees and increase the impact of our solution, as the ReproZip product does not become obsolete if an emulation or virtualization software is no longer maintained. Our project thus fits with the IMLS agency-level goal to "improve preservation, conservation, and care of the nation's content and collections."

The impact of this project will extend well beyond news applications. Database-reliant journalism stories are but one example of the sophisticated digital projects that news organizations produce. Data visualizations need to be preserved too, as do newer forms like virtual reality storytelling. A tool that could quickly and simply pack and unpack these projects, irrespective of the operating system or other dependencies, would be the first of its kind and could enable the systematic collection of hundreds of thousands of dynamic digital objects nationally and internationally. In keeping with the focus and goals of the "Curating Collections" project category, the success of this project would enable new processes and workflows for the collection of this content on a mass scale. Once we

develop a workflow for journalism, the findings will be generalizable to other similar disciplines. Digital humanities and art preservation, for example, are struggling to preserve digital artifacts similar to news apps. Our open-source solution will address their needs as well.

Close to the conclusion of the award, we will release our tool as open-source, which will allow different institutions to not only use it but also help us maintain it, creating a collaboration network and community around the tool that is beneficial and impactful to all. The current increasing user base and collaborators of ReproZip will also be helpful in making sure that the core components of the tool are maintained and that new plugins are developed long after the award is concluded (e.g.: to support new virtualization and emulation softwares). Wikis and forums will also be created to keep the collaborations and interactions alive. Ultimately, we will provide, as a product of this award, complete step-by-step documentation on how to use and develop upon our proposed solution, thus allowing those who are not familiar with ReproZip to easily archive their digital objects. The documentation will allow the tool to be adapted to multiple institutions and communities. In addition, our tool will also create metadata to describe these objects in a discovery tool so that the public can find and access them in library catalogs, repositories, and Internet archives. All these factors, together with the generality of our solution, will help preserve and discover news applications and, indeed, any digital object on an unprecedented scale.

To evaluate the effectiveness of our solution, we plan to work closely with different institutions and departments inside NYU, including the NYU Arthur L. Carter Journalism Institute, and with our journalism industry partner ProPublica. They will help us ensure that sure our prototype is effective and that the workflow we develop can be disseminated, reproduced, and reused by other communities.

Our benchmarks and performance measures include:

- We will use Dollars for Docs, a news app from ProPublica, to test our prototype. Dollars for Docs is based on widely used technologies in libraries and data journalism (including Elasticsearch and SQL), making it a good initial candidate for our testing phase. First, we will use ReproZip to pack Dollars for Docs, keeping track of which components cannot be correctly captured and why. We will also use ReproZip to unpack and emulate Dollars for Docs in different operating systems and computational environments, and validate the results with ProPublica, who will give us precise feedback on which aspects are being correctly preserved and which ones need more attention throughout the project. We will keep track of the components and aspects that are not being fully preserved (e.g.: remote files), making sure these are taken into account on our ReproZip extension.
- Once we have our new ReproZip extension developed, we will pack and unpack Dollars for Docs again and get new feedback from our collaborators. If time and resources allow, we will test our prototype on other news applications / technologies.
- During the testing phase, we will ask our collaborators to evaluate the user-friendliness of our prototype.

Looking ahead, we have been exploring different possible partnerships to scale the project. We have had a number of informal conversations with potential collaborators, including the Internet Archive, Rhizome, and the Software Preservation Network. Such partnerships might take different forms. In the project phase, and once we reach clear milestones with NYU and ProPublica, we will formalize these partnerships and bring in more journalism organizations and university libraries.

Journalism is often called the first draft of history. Right now, an increasing amount of innovative journalism is at risk of being lost. As archivists, librarians, and data scientists, we are committed to ensuring that history drafted today remains accessible far into the future.

References

Adobe Systems Incorporated. (2017, July 25). Flash & The Future of Interactive Content. Retrieved December 30, 2017 from: https://theblog.adobe.com/adobe-flash-update/

Boss, K. & Broussard, M. (2016, April). *Challenges facing the preservation of born-digital news applications*. Staats- und Universitätsbibliothek Hamburg Carl von Ossietzky. Paper presented at the International Federation of Libraries Association, News Media Section, Hamburg, Germany.

Boss, K. & Broussard, M. (2017). Challenges of archiving and preserving born-digital news applications. *IFLA Journal*, *43*(2), 150–157. https://doi.org/10.1177/0340035216686355

Boss, K., Broussard, M., Chirigati, F., Rampin, R. & Steeves, V. (2017, November). Saving news applications: Describing, archiving and preserving dynamic data journalism. In Dodging the Memory Hole 2017: Saving Online News. Symposium conducted at the Internet Archive, San Francisco, CA.

Boss, K., Broussard, M., & Revear, E. (2016, November). Strategies in saving dynamic, born-digital content. In Helen Tibbo, UNC-Chapel Hill (chair), Digital Preservation 2016: Building communities of practice. Symposium conducted at the meeting of the National Digital Stewardship Alliance, Milwaukee, WI.

Boucher, B. (2016, January 4). Rhizome Wins $600,000 Mellon Grant. Retrieved January 10, 2018, from https://news.artnet.com/art-world/rhizome-wins-mellon-foundation-grant-401906

Broussard, M. (2015). Preserving News Apps Present Huge Challenges. *Newspaper Research Journal, 36* (3).

Broussard, M., & Boss, K. (2017, July 10). Save the Data: Future-proofing Data Journalism. Retrieved January 10, 2018, from http://datadrivenjournalism.net/news_and_analysis/save_the_data_future_proofing_data_journalism

Chirigati et al., "ReproZip: Computational Reproducibility With Ease," SIGMOD, San Francisco, USA, 2016, 2085–88.

Daniel, A., & Flew, T. (2010). The Guardian reportage of the UK MP expenses scandal: a case study of computational journalism. *Record of the Communications Policy and Research Forum,* 186–194. Retrieved from http://www.networkinsight.org/events/cprf10.html/group/6

European Commission. Keeping Emulation Environments Portable. (2014, November). Retrieved March 13, 2016, from http://cordis.europa.eu/project/rcn/89496_en.html

Granger, S. (2000, October). Emulation as a Digital Preservation Strategy. *D-Lib Magazine*, 6(10). Retrieved from http://www.dlib.org/dlib/october00/granger/10granger.html

Groeger, L., Ornstein, C., Tigas, M., & Jones, R. G. (2010). Dollars for Docs. Retrieved December 12, 2015, from https://projects.propublica.org/docdollars/

Hansen, K. A. & Paul, N. (2002). Reclaiming News Libraries. *Library Journal, 127*(6), 44.

Hansen, K. A. & Paul, N. (2017). *Future-Proofing the News: Preserving the First Draft of History*. Lanham: Rowman & Littlefield Publishers.

Heideman, J. (2017, November). URLs should never die; retiring old technology while preserving The New York Times' first draft of history. In *Dodging the Memory Hole 2017: Saving Online News.* Forum conducted at the Internet Archive, San Francisco, CA.

Howard, A. (2014). The Art and Science of Data-Driven Journalism (pp. 1–145). Tow Center for Digital Journalism.

Johnston, L. (2014, February 11). Considering Emulation for Digital Preservation. Retrieved March 30, 2014, from http://blogs.loc.gov/digitalpreservation/2014/02/considering-emulation-for-digital-preservation/

Johnston, L. (2014, March 11). Preserving News Apps. Retrieved March 30, 2014, from http://blogs.loc.gov/digitalpreservation/2014/03/preserving-news-apps/

Rechert, K., Valizada, I., von Suchodoletz, D., & Latocha, J. (2012). bwFLA - A Functional Approach to Digital Preservation. *PIK - Praxis Der Informationsverarbeitung Und Kommunikation, 35*(4), 259–267. https://doi.org/10.1515/pik-2012-0044

Wang, S. (2017, September 28). The internet isn't forever. Is there an effective way to preserve great online interactives and news apps? Retrieved January 2, 2018, from http://www.niemanlab.org/2017/09/the-internet-isnt-forever-is-there-an-effective-way-to-preserve-great-online-interactives-and-news-apps/

Von Suchodoletz, D., & Van der Hoeven, J. (2009). Emulation: From Digital Artefact to Remotely Rendered Environments. *International Journal of Digital Curation*, *4*(3), 146–155. https://doi.org/10.2218/ijdc.v4i3.118

New York University Division of Libraries SCHEDULE OF COMPLETION: *Saving Data Journalism*

Major Tasks	Date In-Progress												Assigned Project Staff							
PREPARATION + EVALUATION																				
Evaluate web crawlers	☒													■	■	■	■			
Evaluate proxy options	☒													■	■	■	■			
Hire graduate student		☒	☒										■	■				■		
Hire software developer		☒	☒										■	■	■	■	■	■		
TOOL CREATION + DEVELOPMENT																				
Crawler/ReproZip integration + unpacker creation		☒	☒	☒	☒													■	■	
Documentation creation		☒	☒	☒	☒													■	■	■
TESTING + COMMUNICATION																				
Test, iterate, validate with ProPublica			☒	☒	☒											■	■	■	■	■
Solicit community feedback			☒	☒	☒									■	■	■	■	■		
Document findings			☒	☒	☒													■	■	■
ASSESSMENT + DISSEMINATION																				
Widely circulate code + documentation					☒	☒	☒	☒	☒	☒	☒			■		■	■	■		
Publish + disseminate findings					☒	☒	☒	☒	☒	☒	☒	☒		■	■		■	■	■	
Create future app creation plan							☒	☒	☒	☒	☒	☒		■	■	■	■	■		
Apply for additional project funding					☒	☒	☒	☒	☒				■	■	■	■	■	■		
Project wrap-up + final report											☒	☒	■	■	■	■	■	■		
Project Title: Saving Data Journalism: A prototype to enable systematic collection preservation of dynamic websites **Principal Investigator/Project Director:** Michael Stoller	May-18	Jun-18	Jul-18	Aug-18	Sep-18	Oct-18	Nov-18	Dec-18	Jan-19	Feb-19	Mar-19	Apr-19	**PI/Project Director**	**Project Manager**	**Meredith Broussard**	**Fernando Chirigati**	**Remi Rampin**	**Victoria Steeves**	**Software Developer**	**Graduate Student**

DIGITAL PRODUCT FORM

Introduction
The Institute of Museum and Library Services (IMLS) is committed to expanding public access to federally funded digital products (i.e., digital content, resources, assets, software, and datasets). The products you create with IMLS funding require careful stewardship to protect and enhance their value, and they should be freely and readily available for use and re-use by libraries, archives, museums, and the public. However, applying these principles to the development and management of digital products can be challenging. Because technology is dynamic and because we do not want to inhibit innovation, we do not want to prescribe set standards and practices that could become quickly outdated. Instead, we ask that you answer questions that address specific aspects of creating and managing digital products. Like all components of your IMLS application, your answers will be used by IMLS staff and by expert peer reviewers to evaluate your application, and they will be important in determining whether your project will be funded.

Instructions

- ☐ Please check here if you have reviewed Parts I, II, III, and IV below and you have determined that your proposal does NOT involve the creation of digital products (i.e., digital content, resources, assets, software, or datasets). You must still submit this Digital Product Form with your proposal even if you check this box, because this Digital Product Form is a Required Document.

If you ARE creating digital products, you must provide answers to the questions in Part I. In addition, you must also complete at least one of the subsequent sections. If you intend to create or collect digital content, resources, or assets, complete Part II. If you intend to develop software, complete Part III. If you intend to create a dataset, complete Part IV.

Part I: Intellectual Property Rights and Permissions

A.1 What will be the intellectual property status of the digital products (content, resources, assets, software, or datasets) you intend to create? Who will hold the copyright(s)? How will you explain property rights and permissions to potential users (for example, by assigning a non-restrictive license such as BSD, GNU, MIT, or Creative Commons to the product)? Explain and justify your licensing selections.

The digital products will be fully open-source and will be published online via Github and licensed under a BSD 3-Clause license. Copyright is held by NYU.

A.2 What ownership rights will your organization assert over the new digital products and what conditions will you impose on access and use? Explain and justify any terms of access and conditions of use and detail how you will notify potential users about relevant terms or conditions.

Full license details are available online at https://github.com/ViDA-NYU/reprozip/blob/1.0.x/LICENSE.txt. Users are notified automatically when any changes are made to the license or the Github repository.

A.3 If you will create any products that may involve privacy concerns, require obtaining permissions or rights, or raise any cultural sensitivities, describe the issues and how you plan to address them.

N/A

Part II: Projects Creating or Collecting Digital Content, Resources, or Assets

A. Creating or Collecting New Digital Content, Resources, or Assets

A.1 Describe the digital content, resources, or assets you will create or collect, the quantities of each type, and format you will use.

A.2 List the equipment, software, and supplies that you will use to create the content, resources, or assets, or the name of the service provider that will perform the work.

A.3 List all the digital file formats (e.g., XML, TIFF, MPEG) you plan to use, along with the relevant information about the appropriate quality standards (e.g., resolution, sampling rate, or pixel dimensions).

B. Workflow and Asset Maintenance/Preservation

B.1 Describe your quality control plan (i.e., how you will monitor and evaluate your workflow and products).

B.2 Describe your plan for preserving and maintaining digital assets during and after the award period of performance. Your plan may address storage systems, shared repositories, technical documentation, migration planning, and commitment of organizational funding for these purposes. Please note: You may charge the federal award before closeout for the costs of publication or sharing of research results if the costs are not incurred during the period of performance of the federal award (see 2 C.F.R. § 200.461).

C. Metadata

C.1 Describe how you will produce any and all technical, descriptive, administrative, or preservation metadata. Specify which standards you will use for the metadata structure (e.g., MARC, Dublin Core, Encoded Archival Description, PBCore, PREMIS) and metadata content (e.g., thesauri).

C.2 Explain your strategy for preserving and maintaining metadata created or collected during and after the award period of performance.

C.3 Explain what metadata sharing and/or other strategies you will use to facilitate widespread discovery and use of the digital content, resources, or assets created during your project (e.g., an API [Application Programming Interface], contributions to a digital platform, or other ways you might enable batch queries and retrieval of metadata).

D. Access and Use

D.1 Describe how you will make the digital content, resources, or assets available to the public. Include details such as the delivery strategy (e.g., openly available online, available to specified audiences) and underlying hardware/software platforms and infrastructure (e.g., specific digital repository software or leased services, accessibility via standard web browsers, requirements for special software tools in order to use the content).

D.2 Provide the name(s) and URL(s) (Uniform Resource Locator) for any examples of previous digital content, resources, or assets your organization has created.

Part III. Projects Developing Software

A. General Information

A.1 Describe the software you intend to create, including a summary of the major functions it will perform and the intended primary audience(s) it will serve.

Researchers at the NYU Center for Data Science have developed a tool, ReproZip, that captures and preserves the code, data, and server environment associated with a scientific experiment and adds metadata for library indexing. Our team successfully used ReproZip to partially preserve two data journalism projects. We need to build a new feature for ReproZip so it can fully preserve data journalism projects. We plan to build this feature prototype over the course of this planning grant. The software already serves scientific researchers. This new functionality will allow it to benefit libraries, digital archivists, and newsrooms.

A.2 List other existing software that wholly or partially performs the same functions, and explain how the software you intend to create is different, and justify why those differences are significant and necessary.

Conventional archiving methods include crawlers such at Archive-it or Webrecorder.io, and the automated archiving feeds of companies like Lexis-Nexis. These methods are no longer sufficient to capture projects that involve big data, databases, streaming data, or interactive graphics.

B. Technical Information

B.1 List the programming languages, platforms, software, or other applications you will use to create your software and explain why you chose them.

ReproZip is a thoroughly-tested tool that has gained significant traction in the reproducible research community. It has been used in the reproducibility section of Elsevier's Information Systems Journal; was recommended by the ACM SIGMOD Reproducibility Review; and has been listed on the Artifact Evaluation Process guidelines. It is primarily written in Python and runs on Linux.

B.2 Describe how the software you intend to create will extend or interoperate with relevant existing software.

The methods described in section A.2 focus on the user-facing front end of websites. Scientific reproducibility software like ReproZip focuses on the back end, or server side, of complex websites. In order to fully preserve data journalism projects, a tool is needed that preserves both the front end and the back end together in a single package. We plan to extend ReproZip so that it captures the complete data journalism artifact.

B.3 Describe any underlying additional software or system dependencies necessary to run the software you intend to create.

ReproZip is a tool aimed at simplifying the process of creating reproducible experiments from command-line executions. It tracks operating system calls and creates a package that contains all the binaries, files, and dependencies required to run a given command on the author's computational environment. A reviewer can then extract the experiment in his own environment to reproduce the results, even if the environment has a different operating system from the original one.

ReproZip works by tracing the a news app's systems calls to automatically identify which files should be included. A user can review and edit this list and the metadata before creating the final package file. Packages can be reproduced in different ways, including chroot environments, Vagrant-built virtual machines, and Docker containers; more can be added through plugins.

B.4 Describe the processes you will use for development, documentation, and for maintaining and updating documentation for users of the software.

We will take advantage of Github's rich tools for version control, documentation, and communication. ReproZip's documentation will thoroughly describe how the tool operates; will describe how to use the product; and will describe possible uses and configuration. We have already developed a variety of use cases, and we intend to develop practical guides and case studies for library users.

A regularly monitored email address, users@reprozip.org, is currently used for feedback, questions, concerns, and issues. This address is also used to share use cases with the developers, as well as to report on best practices and lessons learned for reproducibility. Bugs and feature plans are tracked via GitHub issues.

B.5 Provide the name(s) and URL(s) for examples of any previous software your organization has created.

ReproZip's code, documentation, and use cases may be found at https://www.reprozip.org/ and https://github.com/ViDA-NYU/reprozip.

C. Access and Use

C.1 We expect applicants seeking federal funds for software to develop and release these products under open-source licenses to maximize access and promote reuse. What ownership rights will your organization assert over the software you intend to create, and what conditions will you impose on its access and use? Identify and explain the license under which you will release source code for the software you develop (e.g., BSD, GNU, or MIT software licenses). Explain and justify any prohibitive terms or conditions of use or access and detail how you will notify potential users about relevant terms and conditions.

The license on the front page of the source code repository currently reads as follows:

> **Copyright (C) 2014-2017, New York University**
> **Licensed under a BSD 3-Clause license. See the file LICENSE.txt for details.**
> **To suggest changes to this source code, feel free to raise a GitHub pull request. Any contributions received are assumed to be covered by the BSD 3-Clause license. We might ask you to sign a Contributor License Agreement before accepting a larger contribution.**

The software is fully open-source and does not include any prohibitive terms or conditions of use or access.

C.2 Describe how you will make the software and source code available to the public and/or its intended users.

To communicate our findings to the public, we will present the work at a variety of conferences for librarians and journalists; publish articles in the popular press; and publish a scholarly article in the journal *Digital Journalism*. Users of the software currently have access to two electronic mailing lists: reprozip-users (for questions, suggestions and discussions about using ReproZip) and reprozip-dev (for questions about ReproZip source code). We will use these lists to communicate with users.

C.3 Identify where you will deposit the source code for the software you intend to develop:

Name of publicly accessible source code repository: Github

URL: https://github.com/ViDA-NYU/reprozip

Part IV: Projects Creating Datasets

A.1 Identify the type of data you plan to collect or generate, and the purpose or intended use to which you expect it to be put. Describe the method(s) you will use and the approximate dates or intervals at which you will collect or generate it.

A.2 Does the proposed data collection or research activity require approval by any internal review panel or institutional review board (IRB)? If so, has the proposed research activity been approved? If not, what is your plan for securing approval?

A.3 Will you collect any personally identifiable information (PII), confidential information (e.g., trade secrets), or proprietary information? If so, detail the specific steps you will take to protect such information while you prepare the data files for public release (e.g., data anonymization, data suppression PII, or synthetic data).

A.4 If you will collect additional documentation, such as consent agreements, along with the data, describe plans for preserving the documentation and ensuring that its relationship to the collected data is maintained.

A.5 What methods will you use to collect or generate the data? Provide details about any technical requirements or dependencies that would be necessary for understanding, retrieving, displaying, or processing the dataset(s).

A.6 What documentation (e.g., data documentation, codebooks) will you capture or create along with the dataset(s)? Where will the documentation be stored and in what format(s)? How will you permanently associate and manage the documentation with the dataset(s) it describes?

A.7 What is your plan for archiving, managing, and disseminating data after the completion of the award-funded project?

A.8 Identify where you will deposit the dataset(s):

Name of repository:

URL:

A.9 When and how frequently will you review this data management plan? How will the implementation be monitored?