Developing a Framework to Enable Collaboration in Computational Archival Science Education

RICHARD MARCIANO, GREGORY JANSEN, WILLIAM UNDERWOOD
University of Maryland

Abstract: Our IMLS-funded Symposium grant team is conducting research that explores how Computational Thinking (CT) can be incorporated into Masters-level curricula in Library and Archival Science Education and Research (CT-LASER). [1] We created an online repository called CASES (Computational Archival Science Educational System) for storing and providing access to open-source cloud-based Jupyter notebooks that record the recipes for performing archival tasks on digital records. The goal is to facilitate a collaborative network of educators and practitioners that learn from one another through the creation and dissemination of computational case studies and lesson plans. This paper presents the CASES web platform as a service for showcasing, sharing, and teaching the computational practices that archivists and researchers are increasingly applying to digital archival materials. The web-based service consists of project descriptions, lesson plans, and CASE files. This service will help develop and share the building blocks of an integrated library and archival Masters-level educational curriculum to educate the next generation of librarians and archivists in the computational treatments of collections.

Introduction

Archival collections are increasingly comprised of digital materials. This can include native digital entities (e.g. emails, tweets) or non-digital artifacts converted into digital formats for long term storage (e.g. scans of images). "The use of emergent technologies have profoundly altered the nature of archives, by disrupting how information is created, recorded, captured, encoded, curated, shared, made available and used" (Goudarouli 2019). These changes are affecting libraries and archives of all sizes. This is apparent in the decision of the National Archives and Records Administration (NARA) to stop accepting paper-based records at the end of 2022 (Chappellet-Lanier 2019). "It's something NARA [needs to do] to maintain relevance… [NARA] can no longer overextend [itself] by supporting dual processes for analog and electronic records." (Heckman 2019). To support this approach NARA has indicated the need to embrace automation and computational practices such as artificial intelligence (AI).

As part of our IMLS-funded Symposium grant, we tested the impact of teaching Computational Thinking skills to iSchool students in the fall of 2019 using the Japanese American WWII Incarceration Camps collection. Konrad Aderer, the documentary filmmaker who observed the student work, commented that "the students' [computational work] uncovered significant incidents previously unknown, synthesized data to bring out meaningful themes that light the way to further study, and contributed to the understanding of Japanese American history" (Marciano 2019-2).

A basic understanding of the characteristics, strengths, and limitations of such computational artifacts is important for future archivists. At the same time that the content is becoming more dependent on technology, so too is the nature of conducting archival work. The tools and practices associated with contemporary archival activities are increasingly dependent on computing. Related to this, the way users interact with archival collections and their expectations of what is possible reflects the increasingly computationally-mediated nature of our world. Collectively, this shifting landscape of archival work means that in order for today's learners to succeed in future archival tasks, it is essential that Computational Thinking is included as part of their training (Underwood 2018).

[1] This research is sponsored by the Institute for Museum and Library Science (IMLS) under a Laura Bush 21st Century Librarian (LB21) National Forum Grant [RE-73-18-0105-18].

The concept of Computational Thinking (CT) is being introduced to Science, Technology, Engineering and Mathematics (STEM) education. A definition of Computational Thinking for mathematics and science in the form of a taxonomy has been proposed that consists of four main categories: data practices, modeling and simulation practices, computational problem solving practices, and systems thinking practices (Weintrop 2016). In formulating this taxonomy, they draw on the existing Computational Thinking literature, interviews with mathematicians and scientists, and exemplary Computational Thinking instructional materials. This work was part of an effort to infuse Computational Thinking into high school science and mathematics curricular materials. They argue for the approach of embedding Computational Thinking in mathematics and science contexts, present the taxonomy, and discuss how they envision the taxonomy being used to bring current educational efforts in line with the increasingly computational nature of modern science and mathematics.

This taxonomy consists of four general categories of computational practices: (1) data practices, (2) modeling and simulation practices, (3) computational problem solving practices, and (4) systems thinking practices. We have started remapping these concepts to archival science. In Appendix A, derived from (Marciano 2019), we describe the meaning of each of the 22 Computational Thinking practices, where we also replaced the original mathematics and science terms with archival science terms, highlighting them in bold to demonstrate the relevance of mapping Computational Thinking practices to archival science practices.

It has been shown that the twenty-two CT practices that have been identified as important in STEM education also were essential for performing archival practices when addressing digital records (Underwood 2018; Marciano 2019). It was also argued that these practices are essential elements of an archival science education in preparing students for a professional archival career.

Problem Statement

The challenge we wish to tackle is how to incorporate Computational Thinking (CT) practices into Masters-level curricula in Library and Archival Science Education and Research. The major obstacle is that the majority of Graduate School faculty in Library and Archival Studies have little or no background in Computational Thinking and the application of computational tools to archival tasks. How can we enable these faculty to effect these essential changes in Graduate education?

Our framework seeks to promote: (1) open source research and educational platforms, (2) cloud-based student learning environments, (3) new pedagogies for educating archivists in computational methods and tools, and (4) establishing a community of practice for sharing computational and archival knowledge.

We hope to demonstrate adoption of these best practices into classroom and lab settings and will measure success through the monitoring, use, and reuse of computational notebooks. The following sections discuss our approach to these problems, our initial results, findings and our conclusion.

Approach or Method

In order to facilitate the greatest exposure to Computational Thinking and methods in classrooms, team projects, and independent study, we opted to create educational materials in the Jupyter Notebook format. Our choice of this platform is motivated by the fact that "Jupyter is a free, open-source, interactive web tool known as a computational notebook, which researchers can use to combine software code, computational output, explanatory text and multimedia resources in a single document" (Perkel 2018).

This blog-like format is a perfect site in which to smoothly blend together theory and practice, in the form of explanatory text alongside working blocks of example code and the resulting visual output. Jupyter lab notebook technology evolved from a prior Python-based document format called "IPython", originally developed as an interactive online computing environment. The Jupyter project expanded this notebook

technology to suit the needs of interactive data science and scientific computing and it now supports any programming language as well. [2]

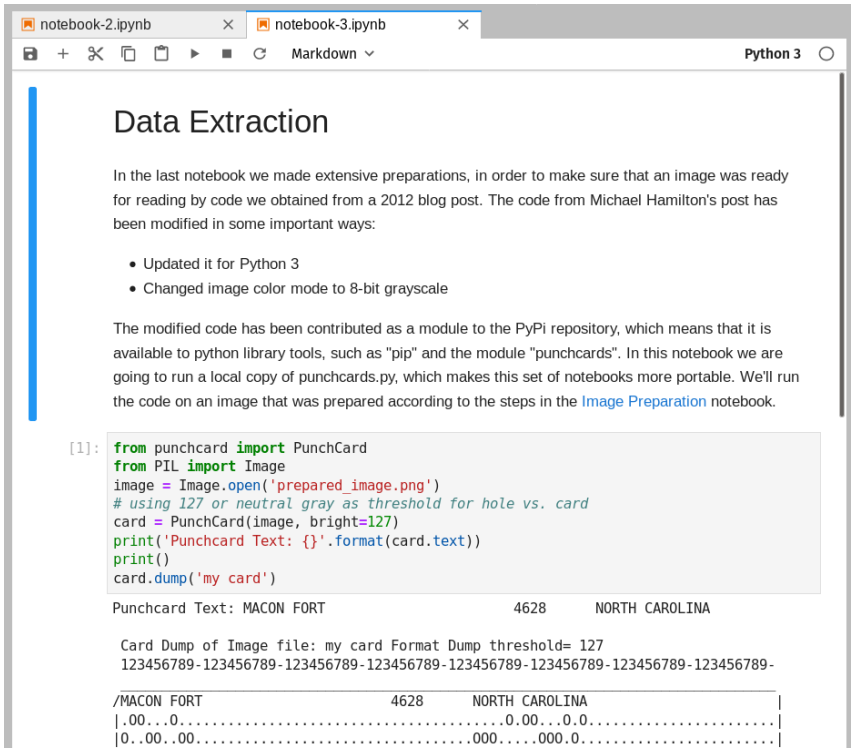

Figure 1. A notebook showing narrative text, code, and code output.

Interaction with example code is the key to successful learning in the notebook environment. Students obtain their own working copies of these notebooks, in which they are free to alter both the text and the code. They can tweak variables or alter the programming logic, then observe the results. They can add their own notes to the text of the page. In this way they are encouraged to interrogate their computational subject and experience coding as a form of play. This leads to a deeper understanding of the motives behind code structures as they are able test new hypotheses and observe the results in rapid iterations.

We have explored notebook use by:

1. Deploying an experimental Jupyter Lab server. This is a web-hosted notebook environment that is set up and ready when the student arrives so that they can load any of our example projects or create original notebooks from scratch. Once a notebook is loaded into Jupyter Lab, the user can immediately begin interactive use, adding notes while running and re-running code blocks without further ceremony or preparation. This environment is available around the clock and not just during class time. While our school-hosted environment is convenient, there are many other ways that students can use notebooks.
2. Having students run a Jupyter notebook environment on their own computers. We were even surprised by a student who found the Jupyter environment for running notebooks on their mobile phone.
3. Hosting notebook content through a free, third-party website, MyBinder.org, that anyone can launch.

With these many available environments, we see that the computational notebook is a highly portable format for building learning scenarios. The technical challenges of hosting notebooks are surprisingly few,

[2] Jupyter Notebooks. See: https://jupyter.org.

which allows us to focus instead on notebook composition and lesson plans. Since our objective is to promote a community of users, a number of deployment options are worth further exploring: gaining support through campus IT, and looking into a networked consortial model across campuses.

Results

In order to promote the adoption of this work through a network of users and contributors, a significant dimension that emerged was the importance of creating a consistent and reusable notebook format, through the adoption of a design template. These templates create consistency in the student experience, promote common design patterns, and lend themselves to search through common terms and search criteria.

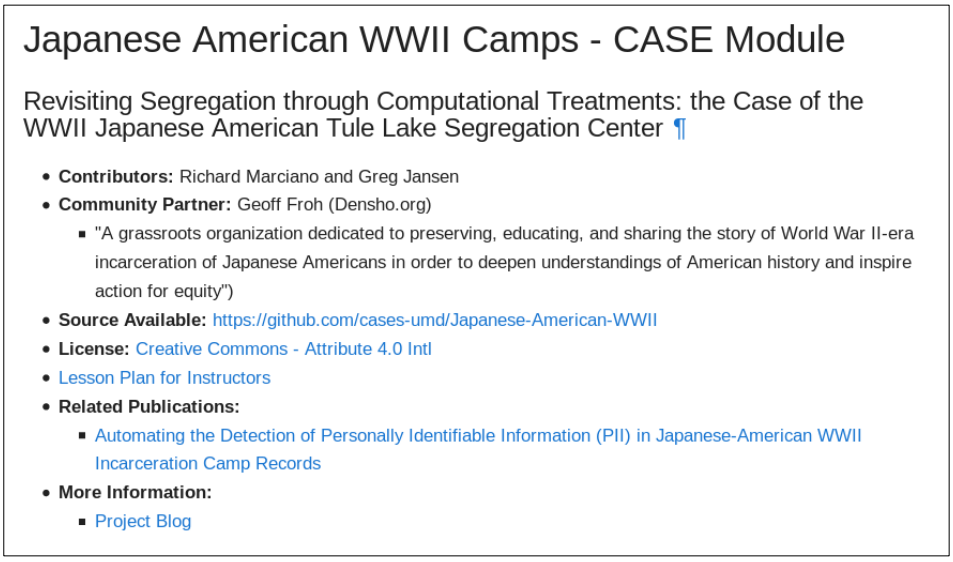

Figure 2. A CASE Module home page.

We decided to call our notebook packages a CASE module, short for Computational Archival Science Education module. Our CASE Module template consists of a set of related notebook pages, including a home page with top-level orientation sections for overview, authorship, attributions, licensing, listing of code sources, learning objectives, project background, and links to further reading. The home page then lists a series of hands-on notebooks that often follow a common sequence, from overview of source data, to data cleaning, data exploration, computational treatments, and visualization. In addition, our hands-on notebooks include some techniques used to create student engagement and interactivity. These include prompts for student reflection, blank spaces for making notes, and suggested coding exercises. We continue to find promising new ways to incorporate best practices for the use of notebooks in the educational setting (Barba 2019).

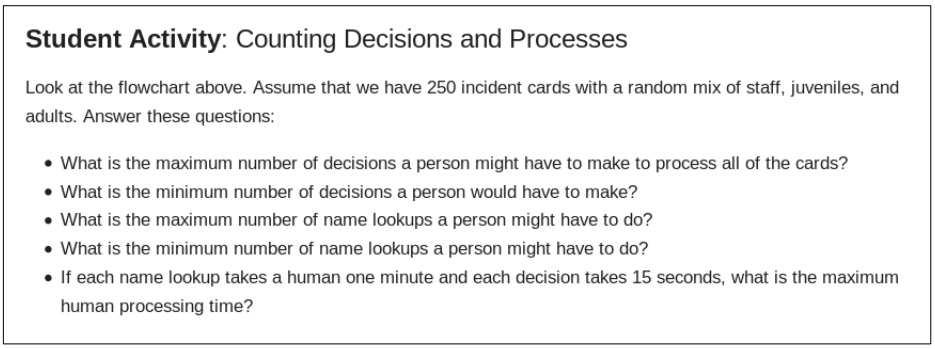

Figure 3. This section prompts the student to reflect on the efficiency of algorithms.

We currently have seven published CASE modules and several more in progress. These include projects from our research center that span a range of social justice issues, which also motivate student engagement with the archival materials and historical topics. We intend that our seven modules are the beginning of a discipline-wide effort to document and share the methods and thinking of computational archival science. To that end we have created a showcase or clearinghouse for sharing CASE modules. The current website is at http://cases.umd.edu. This site showcases the notebook exemplars that we have created so far, letting visitors read or preview their content. The CASE modules are tagged with topics from three areas: (1) We list each of the major Computational Thinking concepts used, taking these from David Weintrop's taxonomy of computation thinking (Weintrop 2016); (2) We also tag the modules according to their incorporation of archival practices, taken from the Society of American Archivists (SAA) Guidelines for a Graduate Program in Archival Studies (GPAS), (Benoit 2019); (3) Finally, but perhaps most importantly, we tag each module with a list of the ethical concerns that are germane to the project. These are the ethical issues that an archivist must keep in mind in this and similar projects. They may arise from the computational techniques, the current political context, or the historical context of the chosen materials. While we are settling on a clear taxonomy of ethical concerns regarding archival materials and computational methods, we are proceeding to identify and document relevant issues.

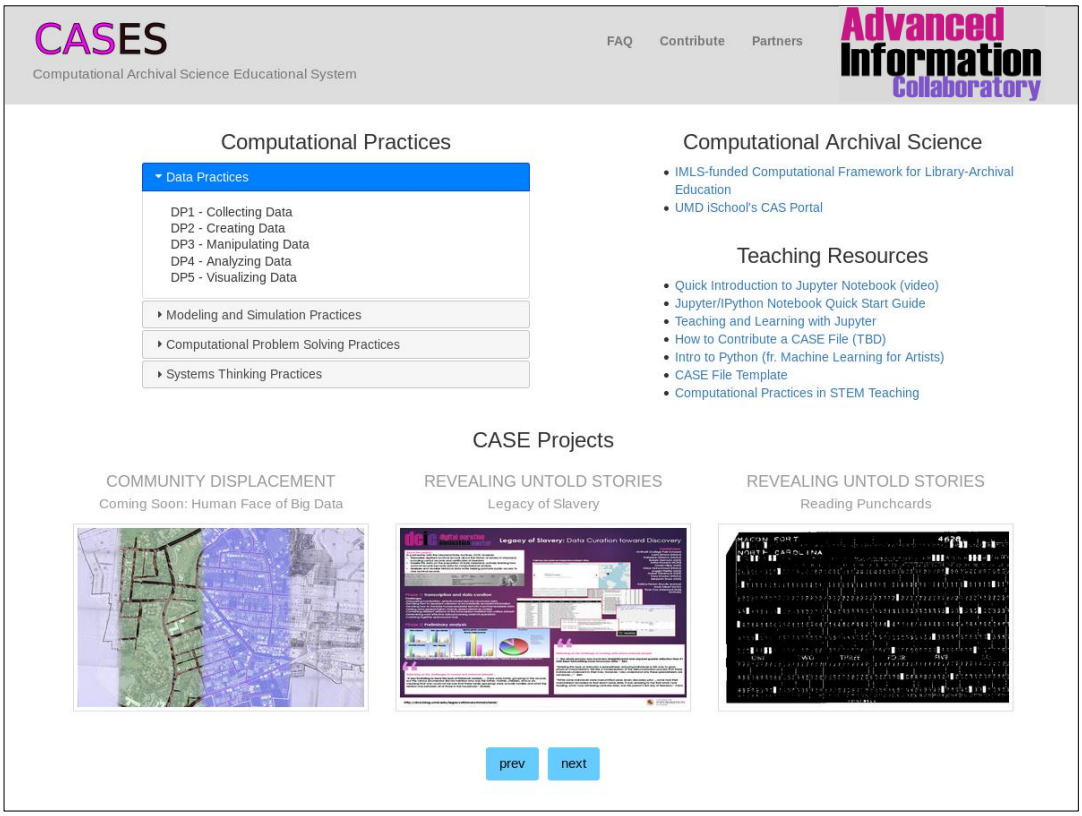

Figure 4. The CASES website showing a carousel of notebooks for browsing.

The CASES website is ready for outside contribution of CASE modules from archival educators and practitioners. The site can easily include any publicly available module that is made available via a GitHub repository. Our own modules are all published via GitHub and are ready for collaboration, including suggested changes from both students and other educators in the form of pull requests or the filing of GitHub issues. We encourage readers to visit the CASES website, preview the notebooks, and take the next step to check out a notebook for themselves, perhaps most easily via the MyBinder service. Computational and science educators are rapidly adopting Jupyter for teaching and it is becoming widely used in these classrooms, for developing teaching materials and sharing lessons, and for creating computational stories.

Findings

The results of that project were:

- Jupyter Notebooks were a useful tool for recording and organizing the tasks, data, tools and results of performing archival tasks on digital records.
- An on-line repository called CASES (Computational Archival Science Educational System) was constructed for storing and providing access to Jupyter notebooks that recorded the results of performing archival tasks on digital records.
- A two-day symposium was conducted at the University of Maryland in April 2019 in conjunction with the 2019 iConference in which the results of this research were presented to some 50 national and international Archival and Library Graduate School Faculty and Practitioners in Libraries, Archives and Museums.

The idea for an online national collaborative network pilot proposal originated from feedback provided by educators and practitioners from this Workshop.

Conclusion

This work focused on mapping out the theoretical framework for Computational Thinking in Archival Science Education and the development of a software infrastructure framework for conducting these learning experiences. However, we further realize that the community network building effort is key.

Towards these goals, we are now planning to pilot an online national collaborative network of educators and practitioners to enable the sharing and dissemination of computational case studies and lesson plans through an open source cloud-based platform based on Jupyter Notebooks. The goal is to change the current MLIS-level educational culture and build a community network that supports educators and practitioners in training the next generation of library and archives leaders. We have identified an initial network of six iSchool Archival Science programs with educators at all ranks and a network of four archival practitioners willing to partner to speed up the development of real-word case studies that can be tested and deployed in a classroom setting.

Our goals are to:

- Further research how to deepen the tagging of the CASE modules with topics from our three areas: *archival practices*, *Computational Thinking practices*, and *ethics and values considerations*, with a particular focus on developing or identifying a taxonomy for representing ethical concerns in Computational Archival Science. This relates research in computational metadata.
- Enhance the CASES repository to include lesson plans as well as cases, and index it using the CT taxonomy. This relates to developing research and educational infrastructure.

References

Barba, L., Barker, L., Blank, D., Brown, J., Downey, A., George, T., Heagy, L., Mandli, K., Moore, J., Lippert, D., Niemeyer, K., Watkins, R., West, R., Wickes, E., Willing, C., and Zingale, M. 2019. Teaching and Learning with Jupyter, https://jupyter4edu.github.io/jupyter-edu-book/

Benoit, E. & Force, D. "One Size Does Not Fit All: Graduate Archival Education in the Twenty-First Century", The American Archivist Vol. 82, No. 1 Spring/Summer 2019, pp.24-52.

Chappellet-Lanier, T.2019. OMB issues guidance on NARA's transition to electronic record keeping. Fedscoop, July 1, www.fedscoop.com/nara-electronic-records-omb-guidance/, https://federalnewsnetwork.com/digital-government/2019/12/nara-to-shed-more-light-on-paper-records-phase-out-through-updated-guidance/

Goudarouli, E., Sexton, A. & J. Sheridan. 2019. The Challenge of the Digital and the Future Archive: Through the Lens of The National Archives. UK J. Philos. Technol. 32: 173, https://doi.org/10.1007/s13347-018-0333-3

Heckman J.. 2019. NARA to Shed More light on Paper Records Phase-out through Updated Guidance, https://federalnewsnetwork.com/digital-government/2019/12/nara-to-shed-more-light-on-paper-records-phase-out-through-updated-guidance/

Marciano, R., Underwood, W., Hannaee, M., Mullane, C., Singh, A. & Z. Tethong. 2018. Automating the Detection of Personally Identifiable Information (PII) in Japanese-American WWII Incarceration Camp Records. Proceedings of IEEE Big Data Conference 2018, CAS Workshop, Seattle, Washington, https://ai-collaboratory.net/wp-content/uploads/2020/03/1.Underwood.pdf

Marciano, R., et al. 2019. Reframing Digital Curation Practices through a Computational Thinking Framework. Proceedings of IEEE Big Data Conference 2019, CAS Workshop, Los Angeles, https://ai-collaboratory.net/wp-content/uploads/2020/04/ReframingDC-UsingCT_final.pdf

Marciano, R., 2019. Digital Curation Students & Filmmaker Event, https://ai-collaboratory.net/projects/japanese-american-ww2-camps/digital-curation-students-and-filmmaker-event

Perkel, J. 2018. Why Jupyter is Data Scientists' Computational Notebook of Choice, Nature Interational Journal of Science, Oct. 30, 2018, https://www.nature.com/articles/d41586-018-07196-1

Underwood, W., Weintrop, D., Kurtz, M. R. Marciano. 2018. Introducing Computational Thinking into Archival Science Education, IEEE Big Data 2018, Computational Archival Science (CAS) workshop #3, Seattle, https://ai-collaboratory.net/wp-content/uploads/2020/03/1.Underwood.pdf

Underwood, W. & R. Marciano. 2019. Computational Thinking in Archival Research and Education, IEEE Big Data 2019, Computational Archival Science (CAS) workshop #4, Los Angeles, Dec., https://ai-collaboratory.net/wp-content/uploads/2020/02/Underwood.pdf

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouillel. & U. Wilensky. 2016. Defining Computational Thinking for Mathematics and Science Classrooms, Journal of Science Education and Technology, 25(1), pp. 127–147, www.terpconnect.umd.edu/~weintrop/papers/WeintropEtAl_2015_DefiningCT.pdf

Appendix A: Summary of Computational Thinking Practices with an Archival Science Lens

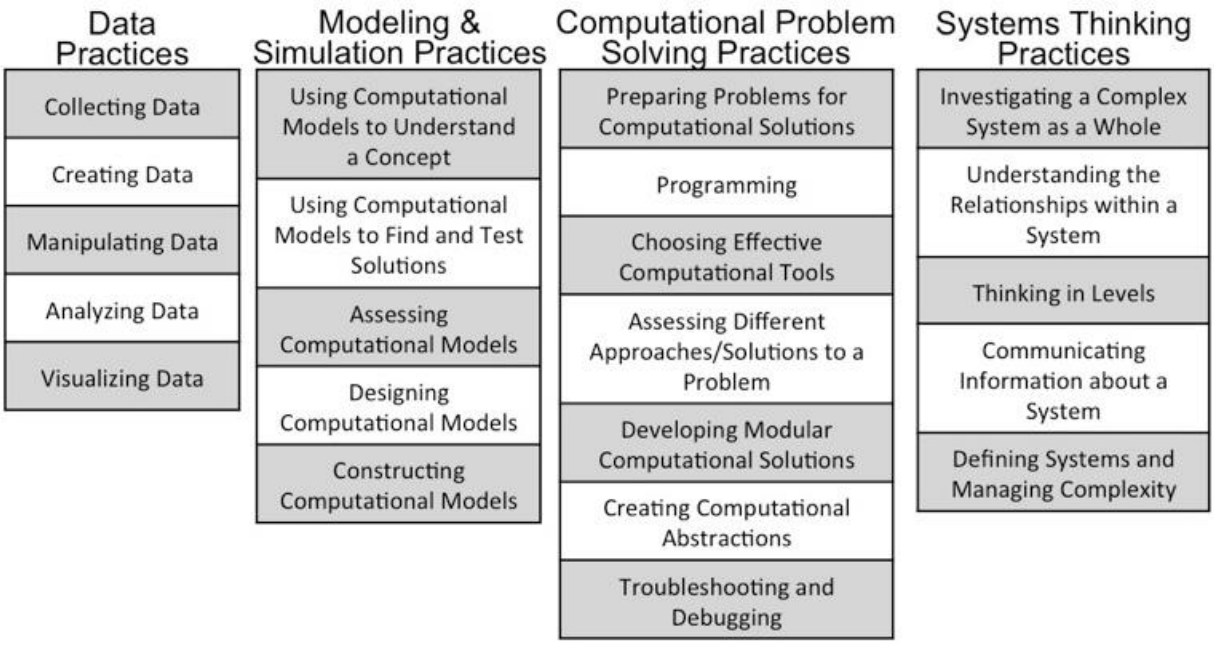

Figure 5: Computational Thinking in math and science practices taxonomy

In (Weintrop 2016) CT concepts are further refined by envisioning a set of computational practices covering: (1) data, (2) modeling and simulation, (3) computational problem solving, and (4) systems thinking. CT is a form of problem solving that uses modeling, decomposition, pattern recognition, abstraction, algorithm design, and scale [3]. We provide a summary of these 22 CT practices spread across these 4 categories next. We have started the remapping of these concepts to archival science.

We describe the meaning of each of the 22 Computational Thinking practices, where we also replaced the original mathematics and science terms with archival science terms, highlighting them in bold to demonstrate the relevance of mapping Computational Thinking practices to archival science practices (Marciano 2019).

A. Data Practices

The nature of how data are collected, created, analyzed, and shared is rapidly changing primarily due to advancements in computational technologies.

1. **Collecting Data**

   "Data are collected through observation and measurement. Computational tools play a key role in gathering and recording a variety of data across many different **archival** endeavors. Computational tools can be useful in different phases of data collection, including the design of the collection protocol, recording, and storage**."**

2. **Creating Data**

   "The increasingly computational nature of working with archival data underscores the importance of developing Computational Thinking practices in the classroom. Part of the challenge is teaching students that answers are drawn from the data available. In many cases archivists use computational tools to generate data… at scales that would otherwise be impossible."

3. **Manipulating Data**

   "Computational tools make it possible to efficiently and reliably manipulate large and complex **archival holdings**. Data manipulation includes sorting, filtering, cleaning, normalizing, and joining disparate datasets."

4. **Analyzing Data**
   "There are many strategies that can be employed when analyzing data for use in an **archival context**, including looking for patterns or anomalies, defining rules to categorize data, and identifying trends and correlations."
5. **Visualizing Data**
   "Communicating results is an essential component of **understanding archival data** and computational tools can greatly facilitate that process. Tools include both conventional visualizations such as graphs and charts, as well as dynamic, interactive displays."

B. Modeling & Simulation Practices

The ability to create, refine, and use models of **archival** phenomena is a central practice… Models can include flowcharts and diagrams.

1. **Using Computational Models to Understand a Concept**
   "Computational models that demonstrate specific ideas or phenomena can serve as powerful learning tools. Students can use computational models to deepen their understanding of archival science."
2. **Using Computational Models to Find and Test Solutions**
   "Computational models can be used to test hypotheses and discover solutions to problems. They make it possible to test many different solutions quickly, easily, and inexpensively before committing to a specific approach."
3. **Assessing Computational Models**
   "Students who have mastered this practice will be able to articulate the similarities and differences between a computational model and the phenomenon that it is modeling."
4. **Designing Computational Models**
   "Part of taking advantage of computational power… is designing new models that can be run on a computational device. Students… will be able to define the components of the model, describe how they interact, decide what data will be produced by the model."
5. **Constructing Computational Models**
   "An important practice… is the ability to create new or extend existing computational models. This requires being able to encode the model features in a way that a computer can interpret."

C. Computational Problem Solving Practices

Problem solving is central to archival inquiry.

1. **Preparing Problems for Computational Solutions**
   "While some problems naturally lend themselves to computational solutions, more often, problems must be reframed so that existing computational tools can be utilized. Strategies for doing this include decomposing problems into subproblems, reframing new problems into known problems for which computational tools already exist, and simplifying complex problems so the mapping of problem features onto computational solutions is more accessible."
2. **Computer Programming**
   "Enabling students to explore **archival problems** using computational problem solving practices such as programming, algorithm development, and creating computational abstractions. The ability to encode instructions in such a way that a computer can execute them is a powerful skill for investigating **archival problems**. Programs include ten-line Python scripts."
3. **Choosing Effective Computational Tools**
   "Students who have mastered this practice will be able to articulate the pros and cons of using various computational tools and be able to make an informed, justifiable decision."
4. **Assessing Different Approaches/Solutions to a Problem**

"When there are multiple approaches to solving a problem, or multiple solutions to choose from, it is important to be able to assess the options and make an informed decision about which route to follow. Even if two different approaches produce the same correct result, there are other dimensions that should be considered when choosing a solution or approach such as cost, time, durability, extendibility, reusability, and flexibility."

5. **Developing Modular Computational Solutions**
   "Students who have mastered this practice will be able to develop solutions that consist of modular, reusable components and take advantage of the modularity of their solution in both working on the current problem and reusing pieces of previous solutions when confronting new challenges."
6. **Creating Computational Abstractions**
   "The ability to create and use abstractions is used constantly across archival science undertakings, be it creating computational abstractions when writing a program, generating visualizations of data to communicate an idea or finding, defining the scope or scale of a problem or creating models to further explore or understand a given phenomenon."
7. **Troubleshooting and Debugging**
   "Troubleshooting broadly refers to the process of figuring out why something is not working or behaving as expected. There are a number of strategies one can employ while troubleshooting a problem, including clearly identifying the issue, systematically testing the system to isolate the source of the error, and reproducing the problem so that potential solutions can be tested reliably."

D. Systems Thinking Practices

Systems thinking analyses… focus on an inclusive examination of how the system and its constituent parts interact and relate to one another as a whole.

1. **Investigating a Complex System as a Whole**
   "Students who have mastered this practice will be able to pose questions about, design and carry out investigations on, and ultimately interpret and make sense of, the data gathered about a system as a single entity... Computational tools such models and simulations are especially useful in such investigations."
2. **Understanding the Relationships within a System**
   "Computational tools are useful for conducting such inquiry as they can provide learners with controls for isolating different elements, investigating their behaviors, and exploring how they interact with other components of the system."
3. **Thinking in Levels**
   "Students who have mastered this practice will be able to identify different levels of a given system, articulate the behavior of each level with respect to the system as a whole, and be able to move back and forth between levels, correctly attributing features of the system to the appropriate level."
4. **Communicating Information about a System**
   "Students who have mastered this practice will be able to communicate information they have learned about a system in a way that makes the information accessible to viewers who do not know the exact details of the system from which the information was drawn."
5. **Defining Systems and Managing Complexity**
   "Students who have mastered this practice will be able to define the boundaries of a system so that they can then use the resulting system as a domain for investigating a specific question as well as to identify ways to simplify an existing system without compromising its ability to be used for a specified purpose."