

Programming humanists - What is the role of coding literacy in DH and why does it matter?

Liliana Melgar Estrada - Utrecht University / The Netherlands Institute for Sound and Vision

Mari Wigham - Netherlands Institute for Sound and Vision

Marijn Koolen - KNAW Humanities Cluster

Introduction

There are many research questions in the digital humanities (DH) that require intensive data gathering and processing. Such projects often start with relatively simple explorations and analyses, but involve highly complex processes later on, for which digital scholars want to make use of complex algorithmic analysis tools that require a certain amount of programming. At the same time, even apparently simple data actions such as search and visualisation require an understanding of the underlying processes and software. In this round table we want to discuss the role of programming and coding literacy in the digital humanities, when and for what types of analyses this literacy is needed and how large research infrastructure projects such as CLARIAH and DARIAH can support researchers to gain the required understanding, either through collaboration or training.

There is a trade-off between simplicity and transparency of tools (Melgar et al. 2019). The higher level of transparency gained by using programming languages for the direct use and manipulation of data without the mediation of a graphical user interface (GUI), requires more advanced skills in working with that data. Many data processing tasks that are seemingly simple, which a scholar may be used to performing with a simple GUI, are actually highly complex, since the underlying assumptions of the tool builders have been abstracted away in the interface (van Zundert & Dekker, 2017). For example, search interfaces offer easy access to numerous library, archive and museum collections, but hide many details, including aspects of metadata quality and how the full-text data was cleaned, chunked and normalized before indexing the individual index terms, mostly without explaining how these details determine what the user sees and does not see.

It may not always be necessary to understand the details of what goes on under the hood, depending on how and for what purpose the results are used. But algorithms make many interpretive decisions that change the nature of the data and how it can be validly interpreted.¹ This is where user-friendly GUIs can get in the way of research, as they make complex interpretive tasks seem like simple processing steps for which the technical details do not matter. Several scholars have argued that researchers need to understand at an abstract level

¹ The effects of algorithms in data access in the context of digital humanities research should also be framed in the fields of research about algorithms which started to emerge more than a decade ago, such as “critical algorithm studies” (Kitchin, 2017), “critical code studies (CCS),” (Marino, 2014, 2016), or “rhetorical code studies” (Brock, 2019).

how tools transform data and how that affects interpretation (Schmidt 2016, Hoekstra & Koolen 2018). We argue that a programmatic approach requires the researcher to get directly involved in the decision making process and to reflect on its impact.

There is an increasing awareness of the importance of coding literacy, i.e. understanding code and being able to work with it, as part of humanities scholarship, and as an essential building block of hermeneutical interpretation in a digital context (van Zundert & Dekker, 2017). Several initiatives have emerged to integrate programming into humanities curriculums.² However, coding literacy goes far beyond the ability to code. Some scholars (e.g., Kirschenbaum, 2009; Montfort, 2015;) suggest that the kind of analytical thinking involved in programming helps developing a conceptual understanding and critical evaluation of the steps involved in working with digital sources and data. It also helps in collaborating and communicating with others by making explicit the programmatic concepts, models, and steps used in scholarly activities, thus making scholarship more transparent, and the data more reusable (Koolen et al., 2018). But there is no consensus yet on which skills are required (O’Sullivan et al. 2015).

The CLARIAH Media Suite takes an intermediate approach, combining GUIs for searching, viewing and annotating with Jupyter notebooks. These are web-based applications that allow users programmatic access to the collections and the searching and annotation functionalities, and to also share their code (Melgar et al., 2019; Wigham, Melgar and Ordelman, 2018). Data experts have prepared notebooks with generic functions for simple and advanced tasks based on input from scholars, who can run this code, adjust it by manipulating variables, but also copy and extend it to suit their needs as their research progresses. This GUI/notebook combination enables users to slowly gain familiarity with the underlying data and functionalities built in to the interface, benefits transparency (code can be shared and reviewed), provides examples for reuse, and helps develop an understanding of how code works. This hybrid approach was evaluated at the CLARIAH 2018 Summer school, where its first users found it useful but challenging.

The proposed panel has the following objectives

- To facilitate an informed and critical debate around the topic of “coding literacy” of digital humanists.
- To involve scholars, builders of infrastructure projects, and archivists/curators, in aligning their visions on how coding literacy could benefit DH research, and the ways to achieve it.
- To come up with ways in which initiatives like CLARIAH and DARIAH can support the development of coding literacy through infrastructure.

² E.g. the Python programming for the humanities course by Folgert Karsdorp and Mike Kestemont, Software Carpentry workshops at the Netherlands eScience Center, the Coding the Humanities at the University of Amsterdam, Data Scopes workshops at the KNAW Humanities Cluster.

Method and topics to be addressed

The round table's method consists of a series of questions around a number of topics.

- What do we mean by “coding literacy”?
 - Break open the concept of coding and the skills it entails, from learning to create queries, regular expressions, data models (Flanders and Jannidis, 2015) and data structures, to collaborating with programmers, to actually writing programs

- Is understanding code an essential part of thorough scholarship?
 - Does analytical and/or computational thinking influence hermeneutical work and critical thinking (Flanders and Jannidis 2015, Smith and Rieh, 2019)?
 - What are the arguments in favour and against the idea that “thinking with computers” (Montfort, 2015) can be helpful in the humanities?
 - What are the impact and consequences of learning programming in the humanities?
 - For what kind of research work is this needed?

- Can coding literacy be distributed through collaboration in multi-disciplinary groups? As O’Sullivan et al. (2015) argued: *“You do not ‘have’ to code, as long as you can work—effectively—with someone who does.”*

- How can the community be supported to incorporate this transition?
 - E.g. complementing GUIs with Jupyter notebooks and APIs. How do we evaluate whether this works?
 - Some reviewers struggle to properly assess research output and proposals that involve complex data processing. What are ways to help reviewers to fairly evaluate such work?
 - When do we reach a critical mass of code literate scholars to make it worth investing in training programs?

Panelists

The round table will include DH scholars who have written on the topic of “coding literacy”, DH scholars who already use various forms of coding and those who see little value in learning these skills for their work, as well as participants representing data providers such as libraries, archives and heritage collections.

References

- Brock, K. (2019). *Rhetorical code studies: discovering arguments in and around code* (Vols. 1–1 online resource (ix, 213 pages).). Ann Arbor: University of Michigan Press.
- Flanders, J. and Jannidis, F., 2015. Data modeling. *A new companion to digital humanities*, pp.229-237.
- Hoekstra, R. and Koolen, M., 2018. Data scopes for digital history research. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, pp.1-16.
- Kitchin, R. (2017). Thinking critically about and researching algorithms. *Information, Communication & Society*, 20(1), 14–29. <https://doi.org/10.1080/1369118X.2016.1154087>
- Marino, M. C. (2014). Field Report for Critical Code Studies, 2014 : Computational Culture. *Computational Culture: A Journal of Software Studies*, (4). Retrieved from <http://computationalculture.net/field-report-for-critical-code-studies-2014%e2%80%a8/>
- Marino, M. C. (2016). Why We Must Read the Code: The Science Wars, Episode IV. In *Debates in the Digital Humanities* (pp. 139–152). Retrieved from <http://www.jstor.org/stable/10.5749/j.ctt1cn6thb.16>
- Melgar Estrada, L., Koolen, M., van Beelen, K., Huurdeman, H., Wigham, M., Martínez Ortiz, C., & Ordelman, R. (2019). The CLARIAH Media Suite: a hybrid approach to system design in the humanities. *CHIIR 2019: ACM SIGIR Conference on Human Information Interaction and Retrieval*. Presented at the Glasgow, Scotland, UK. Retrieved from <https://dl.acm.org/citation.cfm?id=3298918>
- Montfort, N., 2015. Exploratory Programming in Digital Humanities Pedagogy and Research. *A New Companion to Digital Humanities*, pp.98-109.
- O’Sullivan, J., Jakacki, D., & Galvin, M. (2015). Programming in the Digital Humanities. *Digital Scholarship in the Humanities*, 30(suppl_1), i142–i147. <https://doi.org/10.1093/lc/fqv042>
- Schmidt, B.M., 2016. Do digital humanists need to understand algorithms?. *Debates in the Digital Humanities 2016*.
- Smith, C. L., & Rieh, S. Y. (2019). Knowledge-Context in Search Systems: Toward Information-Literate Actions. *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, 55–62. <https://doi.org/10.1145/3295750.3298940>

Wigham, M., Melgar Estrada, L., & Ordelman, R. (2018). Jupyter Notebooks for generous archive interfaces. *IEEE Big Data 2018: 3rd Computational Archival Science (CAS) Workshop*. Presented at the Seattle, WA. Retrieved from <https://dcicblog.umd.edu/cas/ieee-big-data-2018-3rd-cas-workshop/>

van Zundert, J.J. and Haentjens Dekker, R., 2017. Code, scholarship, and criticism: When is code scholarship and when is it not?. *Digital Scholarship in the Humanities*, 32(suppl_1), pp.i121-i133.