Sustainable Growth of Multilingual Open Publishing Projects: The Case of *Programming Historian*

RIVA QUIROGA, ANISA HAWES, ANNA-MARIA SICHANI, AND CHARLOTTE CHEVRIE

Abstract: This article will discuss the sustainability challenges faced by multilingual publishing projects. Taking as its example the case of *Programming Historian*–a Diamond Open Access, peer-reviewed journal of methods for digital humanists–we first discuss why being a truly multilingual project involves more than just publishing in different languages. It requires an awareness of English-language dominance, digital monolingualism, and the need to not only translate but also localize the content. We then explain how this approach to open multilingual publishing has informed our edi-torial, technical and social guidelines, and helped to shape our workflows. Finally, we discuss the evolving challenges of managing such a project. In particular, we focus on how to reconcile the project's growth with the sustainability of our infrastructure.

Keywords: multilingual publishing, digital humanities, open scholarship, project sustainability

Digital publishing has been a "disruptive innovation" within the ecosystem of scholarly communication and publishing. It has introduced new challenges to the economic, social, and infrastructural aspects of academic knowledge production and dissemina-tion, especially in the fields of digital scholarship and digital humanities (Eve and Gray 2020). The current reality of electronic communication and Open Access is a catalyst for rapid transformations in modes of research and learning and has played a pivotal role in facilitating a collaborative and multilingual approach to scholarly publishing. This enables scholars to collaborate globally and exchange resources and ideas across diverse contexts, flowing against the overwhelming hegemony of anglophone influence in scholarly publishing.

Although the discussions around multilingualism are not new in the otherwise largely English-language-dominated field of digital humanities, they have gained centrality only recently.[1] It is in the last couple of years that the impact of monolingualism on digital practices in the humanities has been more openly recognized as a setback (Galina 2013; Fiormonte 2015; Mahony 2018). The term and concept of "multilingual digital humanities" was introduced to highlight the linguistic diversity of projects, datasets, and practitioners within the field (Ortega 2019; Nilsson-Fernàndez and Dombrowski 2022; Viola and Spence 2023). Issues of language sensitivity and diversity in research, teaching, and knowledge dissemination have also been raised in order to create a more inclusive definition of multilingualism in digital humanities (Gil and Ortega 2016; Horváth 2021; Spence and Brandão 2021).

The digital humanities community has been active in setting up forums to support and promote multilingualism at the policy level. For example, the Standing Committee on Multi-Lingualism and Multi-Culturalism of the Alliance of Digital Humanities Organizations (ADHO) is "charged with developing and promoting policies in ADHO and its constituent organisations that will help them to become more linguistically and culturally inclusive."[2] The Global Outlook::Digital Humanities (GO::DH) special interest group of ADHO is working to "break down barriers that hinder communication and collaboration."[3] Meanwhile, the recently created Multilingual DH community interest group in the UK-Ireland Digital Humanities Association "seeks to foster and expand multilingual digital research and practices within the UK and Ireland's community of digital humanities researchers and related practitioners."[4]

In this article, we present the development of *Programming Historian*'s multilingual approach to publishing and discuss the challenges to its sustainability. Given the open nature of the project—its code, content, and team discussions are openly available on GitHub—we are able to analyze how guidelines and documentation have been modified throughout the years we've worked within that platform. We can also piece together our decisions by exploring the history of our discussions held in GitHub "issues," a dedicated space which the platform offers as a forum for collaborators on a project.

This article is organized into three main sections. First, we introduce *Programming Historian* as a project, presenting our approach to multilingual publishing. Second, we explain how this approach to open multilingual publishing has informed our editorial, technical, and social guidelines and helped to shape our workflows. And last, we explore

1. Today's digital humanities community traces its roots back to corpus and computational linguistics, fields that have long engaged with multilinguality in corpora building, annotation, and analysis.
2. https://adho.org/leadership/multilingualism-multiculturalism-committee/
3. http://www.globaloutlookdh.org
4. https://digitalhumanities-uk-ie.org/community-interest-groups/multilingual-dh/

the importance of ensuring the project's sustainability. We consider in particular the opportunities and the challenges involved in expanding the project's linguistic reach and usage.

1. From Monolingual to Multilingual

Programming Historian is a Diamond Open Access suite of journals that publishes article-length lessons on digital techniques and workflows. Our lessons are created by a global community of authors, editors, peer reviewers, and translators. They've collaborated to create more than 240 lessons across our four journals: *Programming Historian in English*, *Programming Historian en español*, *Programming Historian en français*, and *Programming Historian em português*.

Each journal operates as an editorially independent "publication" with a unique ISSN and its own editorial team. As a publisher, we have aligned our publication workflows to ensure the consistency and high quality of the lessons published across our multilingual directory.

Programming Historian's approach to multilingual publishing has evolved over the years. The project was launched in 2008 by William J. Turkel and Alan MacEachern as a series of blog posts which taught the programming language Python to English-speaking historians. In 2012, the team expanded, and an editorial board was appointed to create a digital humanities methods journal. This expansion not only broadened the thematic scope of the lessons but also involved designing an open editorial workflow and implementing a technical infrastructure to host and manage the journal online.

The editorial board opted to make the lessons available under an open license. The possibility of use and reuse that this kind of licensing allows encouraged Victor Gayol, a Mexican historian, to translate in early 2016 some of the Python lessons into Spanish and publish them on his blog, *Cibercliografía*.[5] There, he explains, "As all the content of *Programming Historian* is available under a CC BY license, initially, we were interested in adapting the content. However, the simplicity with which Turkel and Crymble present the topic rather encouraged us to make a translation, with annotations and adaptations, but respecting the form of the original English text as much as possible."[6]

This translation caught the attention of *Programming Historian*'s editorial board.[7] Their discussion about how best to support these kinds of translation efforts led them

5. https://web.archive.org/web/20160602013401/; https://cibercliografia.org/manuales/introduccion-a-python/
6. The message was originally posted in Spanish: "Como todo el contenido de Programming Historian está disponible bajo una licencia Creative Commons—by, en un primer momento nos interesó hacer una adaptación del contenido. Sin embargo, la sencillez con la que está expuesto el tema por Turkel y Crymbe nos invitó más bien a hacer una traducción, con anotaciones y adaptaciones, pero respetando en lo posible la forma del texto original en inglés."
7. Discussion available at https://github.com/programminghistorian/jekyll/issues/246.

to open a call for a Spanish-language editor.[8] Although the original plan was to appoint just one editor, the success of the call resulted in adding three new members to the project. In August 2016, Victor Gayol, Maria José Afanador-Llach, and Antonio Rojas Castro were appointed to create a new team of editors with the goal of making "*Programming Historian* lessons more accessible to Spanish speakers and to begin expanding the community beyond the site's primary audience of English-speaking scholars in North America and Europe."[9] To accomplish this, their first task was to translate the website's core pages and some of the existing lessons into Spanish. In March 2017, they launched the Spanish version of the site,[10] which included the translation of the whole Introduction to Python series.[11]

Shortly after the creation of the new team, discussions about how to address more specific regional learning needs highlighted the importance of supplementing translations with original lessons in Spanish. As a result, an open-ended call for tutorials in Spanish was launched in April 2018, and, in August of that year, a writing workshop funded by the British Academy was held at Universidad de Los Andes, in Bogotá, Colombia. There, 22 Spanish-speaking humanities scholars from across the Americas (Colombia, Chile, Argentina, Brazil, Cuba, Mexico, Canada, and the United States) came together with the shared objective of writing tutorials that targeted research needs in Latin America and the Hispanic world.[12] In April 2019, the first original lesson in Spanish was published.[13]

As Maria José Afanador-Llach explains when analyzing the context of digital humanities in Colombia, "the type of 'digital' work being done in the region differs from the priorities of the Global North" (Crymble and Afanador-Llach 2021, 91). For instance, the work of "cultural memory organisations has focused on accessibility and conservation rather than data mining" (92). A recent study that analyzed web traffic data of Programming Historian lessons has further concluded that "digital humanities educators should adopt practices that acknowledge local needs and differences and make space for local experts to define and best serve those needs" (Crymble and Im 2023, 253).

The publication of that first original lesson in 2019 was a milestone for the project and also a prompt for the Spanish team to reflect on the nature of *Programming*

8. Call available at http://programminghistorian.org/posts/spanish-editor.
9. Post with the announcement available at http://programminghistorian.org/posts/announcing-new-team-spanish-language-editors.
10. Announcing post available at http://programminghistorian.org/posts/launch-PH-Spanish.
11. The first lesson of the series is available at https://doi.org/10.46430/phes0016.
12. A full report is available at Adam Crymble and Maria José Afanador-Llach, "Writing Workshop Report Bogotá, Colombia," *Programming Historian* (blog), August 8, 2018, https://programminghistorian.org/posts/bogota-workshop-report.
13. Interestingly, the first original lesson was about the creation of a parallel corpus for translations: https://doi.org/10.46430/phes0044.

Historian en español. Was it simply a translation of *Programming Historian in English*? As Victor Gayol commented in a GitHub issue where the project's goals were discussed:

> [W]e have embedded ourselves in a previously determined structure, which is not bad, but it invites us to reflect. There is an example: the reflections we took over last year for the DH2018 conference. There were several things discussed, such as the digital divide. If we can achieve an agreement of our goals as [*Programming Historian en español*] beyond being a mere reflection of a predetermined structure, it would be fantastic. And it implies to think what were the goals we have when creating [*Programming Historian en español*]? It is not a mere translation of a given thing, it is a dialogue with the cultures and the possibilities of the digital between the north and the south (to put it quickly).[14]

The establishment of the Spanish team paved the way for the addition of further language teams. *Programming Historian en français* was established in 2019, followed by *Programming Historian em português* in 2021. In both cases, their initial focus was on translating existing tutorials, but they also soon began developing original lessons to cover topics relevant to their own communities.

The project's multilingual growth cemented our commitment to acknowledging diversity as an essential part of removing barriers to knowledge production and access in open digital scholarship (Crymble 2016). The findings of a 2016 survey helped *Programming Historian* devise a diversity policy stipulating that members from any one gender or any one nationality should not make up more than 50% + 1 of the editorial board.[15] This policy ensures that the project can continue to benefit from diverse viewpoints while celebrating linguistic, cultural, and national diversity in practice. Anytime the board grows or shrinks, we re-examine our makeup to ensure that we are representative of our diverse international audience. Honoring these commitments is an ongoing process which requires continual effort. For example, when opening a call for editors, we actively reach out to potential members of yet unrepresented languages and countries.

2. Advocacy in Action

Programming Historian's commitment to multilingualism is layered and a matter of identity. Since the establishment of our multilingual publications, we have recognized how reciprocal translations between journals enrich our learning offer. This translation task can be seen as a way to enrich other language publications but also as a means to

14. Full discussion available at https://github.com/programminghistorian/jekyll/issues/1179.
15. https://programminghistorian.org/posts/PH-commitment-to-diversity

explore and highlight the limits and challenges of multilingualism in digital scholarship and pedagogy.

In 2021, Jennifer Isasi and Antonio Rojas Castro (2021) analyzed the 41 translations published to date by *Programming Historian en español* and identified three types of translation strategies. Level 1, "linguistic changes," concerns lessons in which only the text of the lesson is translated into Spanish, while datasets, analyzed materials, and examples used to explain methods are kept as is. Level 2, "expressive changes," defines lessons in which various strategies are used to adapt it to better suit the needs of a new audience, such as clarification, generalization, annotation, or addition of a bibliography in Spanish. Level 2 translations may also involve partially adapting the code or the dataset used in the lesson. Level 3, "substantial changes," describes lessons where major changes are required to make the content relevant to a Spanish-speaking audience: updating, addition, omission, adaptation, versioning, and change of the dataset. As this analysis shows, there has been a clear focus on localizing the learning resources to make them relevant for the new audience needs, which may differ from those of the language community to which the original lesson was addressed. This analysis also highlights how translations have been used as a way to rethink the politics of citation and knowledge production by updating the bibliography to include authors who write, in this case, in Spanish.

As Pedro Nilsson-Fernàndez and Quinn Dombrowski (2022) mention, "the presence of multilingualism in the platform is not limited to the translated tutorials; [*Programming Historian*'s] English-language tutorials create awareness of the multilingual applications of the tools referenced" (88). In fact, our commitment to multilingualism has led us to actively devising guidelines and shaping workflows that align our practice with the principles that guide us.

Language for a Global Readership

Developing guidelines that support prospective and active authors to write for a global audience has been (and continues to be) a key mechanism towards achieving this alignment. Indeed, our current author guidelines (which are produced in each of the four languages we publish) include a section titled "Write for a Global Audience" which attempts to set out some principles to guide writing that is culturally inclusive and globally accessible.[16] For instance, we caution against assuming that public figures well known in one's own communities will be familiar to others; we advise avoiding vague or

16. The full guideline is available at http://programminghistorian.org/en/author-guidelines#write-for-a-global-audience.

generalized references to geographical locations which inadvertently signal that everywhere else in the world is relative to a centralized, notional "here"; and we discourage jargon, jokes, syntactic puns, and other linguistic complexities.

Instead, we encourage authors to choose the clearest and simplest words possible to communicate their ideas—using the kinds of phrasing they might adopt if they were explaining their workflow to a colleague or peer, taking the initiative to define new concepts and preempt questions. Our readership is not only globally and culturally distributed but also broadly varied in their skill sets and learning needs. We've understood that our readers include researchers working in academic humanities and social sciences departments, librarians, archivists, and self-taught programmers honing their skills, as well as autodidacts with other specialist interests. We want language to let readers in—not erect a barrier—and that means encouraging writing which makes ideas accessible to all readers, whether they are reading in their first, second, or third language.

Multilingualism from the Proposal Phase

We do understand, however, that the length and detail of this documentation means it is not the most effective way to convey our principles to prospective authors. A recent call for new lesson proposals for our English-language journal provided an opportunity for us to review and refine our Lesson Proposal Form.[17] We want to ensure that the proposals we receive succeed in centering methods that support multilingualism and multiculturalism. *Programming Historian in English* is the eldest of our four journals. Currently, it receives the greatest number of proposals overall and a markedly higher proportion of proposals for original lessons than translations each year.[18] Having published their first two translations (both from Spanish) in 2023, the English team have articulated an ambition to increase editorial capacity for guiding translations through the pipeline in 2024. As a publisher, we recognize this is an excellent opportunity to diversify participation and foster cross-team knowledge exchange about translation workflows and best practices. The proposal phase is an ideal time to encourage the selection of case studies which can be exchanged or would be pertinent across global research contexts. We ask authors to consider sample datasets on the basis that they are Open Access and suitable for substitution, adaptation, or translation.

17. Comparison of previous and current form available at https://github.com/programminghistorian/jekyll/commit/296c2c3894b8af5fb5ff23a70cbfb9606aa84d4c.
18. To date, *Programming Historian in English* has published 105 original lessons that have provided the basis for 113 translations across our Spanish, French, and Portuguese editions.

However, there is not always a multilingual choice of dataset available. The production, preservation, and publication of large-scale data is a privilege of national archives and libraries that have substantial resources to invest in its digitization, backed by government initiatives to disseminate it openly and encourage research (Zaagsma 2023). Many nations may not have the funds, the infrastructure, or the governmental support to do so. Recognizing this global disparity is key to understanding why, sometimes, it isn't possible to identify an equivalent of the type of data required to demonstrate a method. In these cases, the pedagogical focus will need to be adapted. Keeping the original dataset in place, our translators will be encouraged to provide additional support to their readers so that they can navigate, transform, and interpret research data in a non-native language.

In other cases, our authors actively choose multilingual datasets from the onset, to demonstrate the global application of the method their lesson explores. For example, "Introduction to Jupyter Notebooks" (Dombrowski, Gniady, and Kloster 2019) centers a dataset of Italian fanfiction, and "OCR and Machine Translation" (Akhlaghi 2021) uses documents in Russian as its case study example. Elsewhere, our translators have opted to retain the original language dataset. For example, the English translation of "Sentiment Analysis with 'Syuzhet' Using R" retains the Spanish-language novel *Miau* by Benito Pérez Galdós (1888) as its case study, to introduce scholars working with non-English texts to the complexity of handling accented characters and to "the intellectual problems of applying English language algorithms to non-English texts" (Isasi 2023). Other lessons, such as "La reconnaissance automatique d'écriture à l'épreuve des langues peu dotées" (Vidal-Gorène 2023), have focused on the challenges of working with non-Latin alphabets.

This is also our opportunity to promote the choice of tools which are documented and workable in languages other than English. Some software tools offer multilingual options to experience their graphical user interface (GUI), which supports navigation of menus, options, and buttons in readers' preferred languages. Sometimes, a multilingual GUI may be offered, but full documentation isn't translated. An example of this is Voyant Tools, on which *Programming Historian en español* published an original lesson by Silvia Gutiérrez De la Torre in 2019, which has an optional Spanish interface but doesn't offer Spanish documentation. Meanwhile, the Map Warper tool has neither a multilingual GUI nor documentation, and in this case the Spanish-speaking authors of a lesson on the tool opted to create animated visual aids to support user navigation of the interface (however, this solution raises its own barriers to accessibility unless detailed, descriptive alt text is also provided).

Each of our lessons is graded on a scale of 1 to 3 where 1 indicates beginner-appropriate, 2 defines intermediate, and 3 signals a high level of difficulty. We recognize that the need to negotiate technical documentation in a language other than one's

own increases a lesson's difficulty level. So, where multilingual documentation is not available, we encourage authors to be explicit about this fact and to include multilingual references for further reading wherever possible. In particular, the Proposal Form foregrounds open tools, open source software, and open programming languages. We consider openness as an extension of our commitment to multilingualism, because choosing open recognizes the imbalance of computational and financial resources across our readership community.

Reviewing our Lesson Proposal Form prompted us to realize that, up until now, we had neglected to explicitly encourage authors to consider, at the earliest opportunity, how the lesson they would like to share with our readers could be translated and localized by other language communities in the future. In the revised form, we specifically ask prospective authors to think about multilingualism from the beginning.[19] Prompting them to provide us with information about how their method or tool could be applied or adapted for use with non-English-language data, we state our strong preference for those that could be used in multilingual research contexts. Our aim is to encourage proposals that are devised and developed for a global audience, with future translatability and multilingual researchers in mind.

These recent adjustments will also empower our managing editors and their teams in their assessment of proposals. In cases of textual analysis methods, translation can be constrained from the outset. The feasibility of a lesson's application beyond its original language and character-set can be key to our editors' decision about whether or not to invite submission.

Learning across Languages

After submission and throughout the editorial workflow, there is further work to do to ensure that the lessons we develop are convenient for translation in the future. As well as text and data assets, our lessons comprise code and visuals. Code can take a relatively "straightforward" form of raw blocks of script, prepared for readers to copy-paste and use. But these blocks frequently contain directives or "comments" preceded by a '#' which signals to the compiler not to process them as part of the executable code. These in-line comments will always need to be translated. In cases where the lesson necessitates the use of specific code libraries of which the function names and variables do not exist in translation, commentary will need to be extended to provide clarification.

19. Form available at https://programminghistorian.org/assets/forms/Lesson.Query.Form.txt.

Since 2023, we have increasingly encountered authors who wish to integrate ready-to-run computational notebooks with their lessons.[20] This trend presents a range of challenges for our publishing workflows: it requires us to manage notebooks among existing assets, render and serve them to our readers, and coordinate future maintenance to ensure their sustainability. Most pertinent here is the challenge notebooks pose for future translatability. We've drafted some guidelines for authors based on an understanding that notebook assets will form part of the package of files for translation. If commentary from lessons is substantially extended in notebooks, we foresee an unacceptable increased burden upon translators. Therefore, the guidelines include a stipulation that notebooks should consist of the code and minimal in-line comments only and an agreement that section headings and subheadings in the codebook must mirror those of the lesson to support translators' (and also eventual readers') navigation through the lesson.

Visuals within our lessons range from screenshots to hand-drawings, maps, graphs, and annotated diagrams. These, too, can raise challenges for us in translation. When there is no multilingual GUI option for a software or programming tool, screenshots will need to be adapted with annotated overlays or additional labeling. Meanwhile, plots and graphs usually feature axis labels which our translators can reproduce in-code as they work through the steps of the lesson, substituting any text input with their own translations. Recreating diagrams that include extended text (denoting processual models, for example) has proven to be more time consuming for translators, because the underlying diagram form must also be created anew. To reduce this burdensome part of the workflow, we have introduced a requirement for "mute" copies (that is, without any text) of underlying diagrams to be provided in addition to the text-populated version at the time of submission. This "mute" copy supplies a base which future translators can adapt to add their text.

3. Towards Sustainable Growth

It was after onboarding our Spanish team that concerns about the sustainability of the project's structure were first raised. On March 9, 2017, Adam Crymble opened an issue on GitHub to document a suggestion put forward by Fred Gibbs:

> We've had a suggestion that the team might be growing too big for everyone to be involved in all aspects, and that it might be time to separate the board (the part of the

20. An example of this kind of notebook is the one created by Project Jupyter: https://jupyter-notebook.readthedocs.io/en/stable/index.html.

> team that is involved with the project direction & monthly meetings) from people who want to contribute in a more focused manner (e.g., editing lessons in a specific area). This is something our team will have to discuss, but I wanted to put a ticket here so we remember to talk about it.[21]

Although the discussion did not progress initially, the consolidation of *Programming Historian en español* and the onboarding of the French team in February 2019 reignited it. This time, the discussion on GitHub questioned the sustainability of the current structure (which sometimes hindered decision-making) and sought to define language-team-specific strategic objectives.[22] The teams considered restructuring the project into two distinct tiers: (1) a "Publisher" responsible for maintaining the technical infrastructure, providing publishing services, and growing the project and (2) language-specific "Publications" responsible for producing and editing lessons. The framework emphasized editorial freedom for language-specific teams while maintaining alignment with the project's agreed principles. Having a board running the Publisher was a way to recruit people with skills that could benefit the project as a whole—for example, experts in the fields of sustainability, metadata discovery, community outreach, and financial management.

The idea of risk management was first raised as part of this discussion. For many team members, the advantages of formalizing roles, expectations, and tasks within the project were evident, but they agreed that there would also be a need for careful risk management considering the scale of the proposed changes. In particular, the group discussed risks to the stability and clarity of our editorial processes, as well as potential challenges to diversity. If we adapted our recruitment processes to align with the new structure, it would be critical to maintain a balanced power dynamic to respect and support the unique contributions of all members. Regarding diversity, Anna-Maria Sichani posted:

> I am guessing that the "Publisher" role will be very American-English populated (in terms of language/nationality) and this might be an unexpected battle for our diversity and internationalisation fronts. We need to secure that even if Publications teams are by default language specific, we somehow keep (language & national) diversity within the Publisher team as well. I think we need to revisit again our diversity algorithm here . . . or even be more creative on establishing a new diversity rule/pattern.[23]

21. Discussion available at https://github.com/programminghistorian/jekyll/issues/379.
22. Discussion available at https://github.com/programminghistorian/jekyll/issues/1183.
23. Comment available at https://github.com/programminghistorian/jekyll/issues/1183#issuecomment-470223690.

Another key discussion regarding the sustainability of the project had started a couple of months earlier in a different GitHub issue.[24] In February 2018, the editorial board shared the idea of implementing some form of sponsorship scheme, to help *Programming Historian* achieve its goal of operational sustainability. The plan was to start accepting sponsorships on a case-by-case basis to support the growth of the project. (This later transformed into our Institutional Partnership Programme, a framework which invites academic libraries and universities to contribute an annual membership fee in lieu of a subscription.)[25] As the plan progressed, we set up a "Support Us" page on our website. James Baker pointed out that this was "a good opportunity to explain the risks the project faces by being entirely volunteer driven."[26]

This discussion was pivotal for the future of *Programming Historian* for two reasons. First, it brought to light the need for a financial structure to manage the budget. Second, it underscored the need to professionalize aspects of the project that cannot rely solely on volunteer work.

Initially, we planned to seek support from our members' affiliated institutions or to explore options on crowdfunding websites. However, the bureaucratic limitations of the first approach and the taxing and money management issues of the latter led, in February 2019, to the idea of becoming a charity.[27] The next few months were dedicated by the team, especially those based in the United Kingdom, to work towards this goal. Finally, on September 6, 2019, ProgHist Ltd was incorporated in England and Wales as a company limited by guarantee. The path to becoming a charity took a little longer because the team was unsure about meeting the £5,000 per year income requirement needed to be a charity in the United Kingdom. However, in mid-2020, as new institutions joined the Institutional Partnership Program, the threshold was met, and a first attempt to become a charity was made. Unfortunately, this first attempt was unsuccessful due to issues with the submission process, highlighting the need for expert advice on such matters beyond the team's expertise.[28] Once these problems were resolved, a new submission was made, and ProgHist Ltd was finally registered as a charity in England and Wales in September 2021.

Through the Ltd, we were able to establish a financial infrastructure to manage the funds received via the Institutional Partnership Program (and other sources of donations), making it possible to professionalize certain aspects of the project by requesting

24. Original comment available at https://github.com/programminghistorian/jekyll/issues/724#issue-298999936.
25. Description of the current program available at https://programminghistorian.org/en/ipp.
26. Original comment available at https://github.com/programminghistorian/jekyll/issues/915#issuecomment-448165752.
27. Discussion available at https://github.com/programminghistorian/jekyll/issues/878.
28. The problems with the first submission are available in this discussion: https://github.com/programminghistorian/jekyll/issues/1671#issuecomment-690126554.

services in specific areas.[29] Initially, focus was placed on two key areas: copyediting and resolving bugs in lessons that exceeded the skill set or time availability of current volunteer team members. The copyediting process was first established for *Programming Historian in English*, motivated by the desire not only to ensure the quality of original lessons but also to provide an opportunity for authors whose first language was not English to contribute to the journal. In the case of Spanish, it served also as a means to connect with authors who spoke Spanish as a heritage language and wished to engage with the Spanish-speaking DH community via publishing a lesson in the journal.[30] This is a radically different approach from the one of journals published by for-profit publishers, who tend to advertise their own editing services as a product that authors should pay for (Arenas-Castro et al. 2024), using language barriers as a way to profit.

At this point, it was clear that a piece was missing: someone with a global vision of the project who could not only coordinate specific tasks but also assist in improving our workflows across all four publications. The financial stability provided by the Institutional Partnership Programme paved the way to open a position in 2021 for a publishing assistant and, in 2023, for a publishing manager. The seven years that *Programming Historian* ran as an open and multilingual volunteer-driven project with a global readership before incorporating were key to achieving financial stability. This trajectory allowed us to show prospective institutional partners the real impact that their support could have.

Complexities of Growing a Multilingual Infrastructure

Programming Historian's publishing workflows are now supported by the publishing manager and assistant, who collaborate with our four language teams to administer and produce all journals. Although our staff collectively speak only two of the journals' four languages fluently (French and English), they also succeed in providing a high standard of support to the Spanish and Portuguese teams, including the technical processing of lesson materials, small-scale adjustments to text and code, and the coordination of copyediting services. Another crucial aspect of the workflow is typesetting the submissions ahead of publication: figures must be captioned; technical terms must be highlighted; direct translations must be given where necessary; citations must be formatted in our chosen style. This work, as well as the daily management of directories

29. Individuals can support the project via Patreon: https://www.patreon.com/theprogramminghistorian. This source of income represents only 7% of ProgHist Ltd income, versus the 93% obtained via the Institutional Partnership Programme.
30. For example, David Rodríguez translated the original lesson he published in English (his first language), to Spanish (his heritage language): https://doi.org/10.46430/phes0049.

and folders within our GitHub repositories, is manageable in languages that our staff do not speak or write, because they can visually parse and type the letterforms, accents, and punctuation characters.[31] Latin languages are familiar enough that it's possible for them to use linguistic intuition to pick out key terms and understand the general structure of the lesson, especially with the help of online translation software.

The administration and delivery of our publishing services depends upon effective written communication. Most of our exchanges take place in GitHub Issues where it is critical to write clearly, using simple language. Where possible, and with the support of our four journal teams, we've introduced templated comments which are translated across four languages. This helps us to ensure that all contributors understand the instructions or advice given, and our publishing team are planning to expand our use of translated templates for comments through 2024.

Despite our distribution across continents and time zones, we also regularly engage in spoken exchanges. Inter-team dialogue about general topics depends upon an agreement to use our most common shared language (English), the second or third language of the majority of our project team. Our commitment to maintaining collaborative, democratic, and open discussions means accepting this "good enough" yet imperfect solution. Its adequacy is contingent upon the continued efforts of all participants to be clear speakers and patient listeners.

We strive to cultivate a collaborative and non-judgmental environment, in which native speakers of each language can support each other by translating or reviewing the accuracy of their colleagues' work in a non-native language. Historically, our project team's volunteers have taken upon themselves the extensive work of translating contributor guidelines, as well as the administrative and structural texts required for our website. While we have always recognized and valued the skillful labor involved in translation, in the past we were unable to compensate individuals for that work. Since establishing our Institutional Partnership Programme in 2020 and professionalizing our publishing team from 2021, we have built capacity to invest in people to undertake that translation work.

Although these frameworks and practices for effective multilingual communication hold up to the demands of our four language teams, we are acutely aware that substantial commitment will be required to meaningfully and fairly adapt them as the project grows. Integrating a language which uses non-Latin script, especially, would present a challenge of a different magnitude. The sustainable growth of the project in terms of adding new languages will likely require enlarging our professional publishing team, a goal only achievable through substantial growth in the number of Institutional Partnership Programmes or by designing new ways to achieve financial sustainability.

31. https://github.com/programminghistorian

Devising a path towards growth also involves re-evaluating our current technical infrastructure. Our website was built in 2014 using Jekyll, an open and flexible static site generator which has enabled us to adhere to the principles of "minimal computing."[32] Notably, these principles make websites navigable and accessible by users with less powerful hardware or slower internet connections. Until recently, the Jekyll system has effectively accommodated our four journals and met the needs of our contributors, staff, and volunteers (Lincoln et al. 2022). However, future development of and support for Jekyll is soon to cease and its toolkit is gradually being deprecated. We are now experiencing excessively long build-times (up to 12 minutes) and encountering frequent and increasingly complex build errors that must be attended to before we can move our publishing pipeline forwards. Further scaling up our current architecture (to accommodate an additional language section, for example) would further increase build-times and slow the system as a whole.

To overcome this major issue, we are currently working on a technical infrastructure renewal project to address the limitations of our current system. Our expectation is to rebuild our website in alignment with our ethos, which means remaining committed to an open and minimal architecture. The goal is to create a website that can better meet the needs of our community through contemporary design that has low carbon intensity and adheres to accessibility best practices. To achieve this, we will need to partner with potential funders who share our vision for a robust and sustainable multilingual open publishing infrastructure for hosting our learning materials.

As things are, the inherent limitations of both our human and technical infrastructure therefore create a bottleneck in our trajectory towards growing *Programming Historian* multilingually. These limitations highlight the tension between ensuring the sustainability of a publishing project and the desire to serve a broader linguistic community. However, this issue also underscores how the power of an open license could help us to keep "the language door" open.

Harnessing the CC BY License to Empower Multilingual Growth

At its core, the *Programming Historian* project is a not-for-profit charity with a pedagogical mission: to teach and disseminate valuable digital humanities learning resources for free, to users around the world. As part of our commitment to open source values and practices, we publish all our journals under the CC BY 4.0 license. This choice

32. For more about minimal computing, see https://sas-dhrh.github.io/dhcc-toolkit/toolkit/minimal-computing.html#:~:text=Minimal%20computing%20is%20a%20set,frugal%2C%20and%20 nuanced%20digital%20 decisions.

affords us a powerful alternative for reconciling our aspiration for growth with the sustainability of our infrastructure. Perhaps a key to understanding the core principles and affordances of this license is to individually examine the constituent parts of its name, "Creative Commons Attribution."

"Creative" tells readers that they are free to adapt the original work in any way: they may use it for a different purpose, recreate it in a different language or through a different medium. This term encourages a creative reuse of the work in its original form, to expand its reach to a wider, perhaps completely new, audience. "Commons" indicates that users are free to share the original work (or their adaptation) through their own channels, even commercially. Part of a commitment to openness is accepting to share the work among the community for free. Authors remain the copyright owners, but they understand that the work is now a "common good," which belongs to all. "BY" represents the third pillar of the CC BY license: attribution. Indeed, the only restriction to the ultimate freedom given to users is that they must, whenever they either share or adapt the work, always attribute the original author and declare any modification.

A CC BY license therefore holds fast to the principle of credit where credit is due by demonstrating an effort to recognize the labor and the people involved in creating works of value, especially when they have been published Open Access, for free. Crediting the author means that they still have something to gain: in the academic world, recognition can essentially be capitalized upon through career advancement. Because of our awareness of the skill and effort involved in publishing and translating, *Programming Historian* is adamant about crediting all contributors to our lessons—not just the author. We are somewhat unusual in acknowledging the skilled labor volunteered by authors, translators, editors, and reviewers of all our lessons, a practice deeply tied to our celebration of multilingualism.

Within its terms, the CC BY license enables each of our four journals to translate, adapt, and localize what we have already published across their collective directories. Our broader impetus for choosing a license which offers the freedom of adaptation and circulation, however, is that it opens opportunities for others to do the same. Thus, our commitment to multilingualism is not solely limited to the language teams currently working under the *Programming Historian* umbrella: various non-affiliated groups and individuals have already taken up this opportunity to translate and localize some of our lessons for their own audience. As we saw above, the Spanish edition began with an ad hoc translation by the Mexican historian Victor Gayol. In 2020, Nobuhiko Kikuchi at the Kansai University Open Research Centre also translated and published six of our lessons from English into Japanese, showcased on the university-managed East Asia DH Portal.[33]

33. "The Programming Historian 日本語," 東アジアDHポータル (2020), accessed January 26, 2024, https://www.dh.ku-orcas.kansai-u.ac.jp/?cat=2.

These examples reinforce the strength of the CC BY license as a vector for multilingualism across languages and cultures. As a team who accept our own limitations, we recognize that we don't have omniscient knowledge of the digital humanities landscapes and all its cultural nuances. The Japanese initiative allowed our resources to be distributed beyond the reach of our internal spheres of expertise. By sharing ownership of our lessons with others, we allow those better-placed to choose when, where, and how to adapt and distribute them, rather than attempting to encompass adaptations within our own infrastructure.

One of the ways we have already attempted to communicate the opportunity of the CC BY license more explicitly is to develop a formalized copyright declaration form. We recognized that our previous practice of asking authors and translators to grant their permission to publish—by way of posting a comment on the issue dedicated to their lesson—was inadequate. First, we noted that contributors were being prompted to grant this permission before the process of editing, review, and revision were complete; we felt strongly that the agreement should instead be arranged when lessons are shaped and ready for publication. Second, we sought a formal way to underscore that the license under which we publish allows, invites, and encourages creative reuse. Our objective was to distill and clarify the terms of the license so that authors and translators understand their own freedoms and the freedoms that will be afforded to others upon publication of their lesson as a common good.

We invested in the translation of the Authorial Copyright and Publishing Rights form into all four journals' languages, appreciating the importance for contributors to be able to read the agreement in their preferred language before signing to confirm their understanding and grant their permission for us to be the first publisher. Within the formalized declaration, we include our own brief, context-specific summary of the license alongside links which direct to the Creative Commons organization's own short, digestible summary of the deed as well as the full legal code. The deed and the legal code are available in a range of languages, including all four of those represented by our journals. Again, this is important because it enables contributors to read the text—which is technically and semantically complex—in their preferred language.

However, increasingly we recognize that we have perhaps done too little to explain and promote the affordances of our license. We have been approached by several groups who express a keen interest in adapting, translating, or creating *Programming Historian* lessons. Their requests are most often framed within a request for formal integration into the project, which indicates that we have not made the freedoms for creative reuse with attribution entirely clear.

We can understand why these external groups might prefer to join our publishing infrastructure rather than work independently to develop their own. We have the benefit of two paid, dedicated staff; we have developed structured workflows and policies

over the years; we have an established readership; and our name is recognizable in the field. While we remain committed to extending participation and expanding global readership, our priority is to manage capacity so that existing editions are sustainable.

However, we are particularly keen to find alternative ways of encouraging, supporting, and nurturing initiatives that emerge from the opportunity offered by our CC BY license as self-organized "community teams." We would consider this investment to be appropriately justified because these independent efforts have clear potential to empower an extension of our guiding objective, which is to advance education in the digital humanities through the production of lessons published in the public domain.

The support we might offer the community teams could take several forms. One is a program of direct training that centers upon sharing the accessibility and sustainability principles and best practice workflows we have developed. For example, by teaching others how we create and embed alt text to accompany images, collate authorial copyright declarations, or generate and apply archival links, we can help to increase the likelihood of their success. This will also pave the way to easily incorporate them into our infrastructure in the future.

Another idea would be to aid promotion of community-led initiatives through our website and social media channels. We could collaborate to devise a publicity campaign including blog articles, social posts, and announcements in our quarterly newsletter to support dissemination of the new resources created by external communities. We have built a substantial network of readers, contributors, and supporters and established a wide circle of connections across various social media. These networks and platforms afford us the advantage of a "voice" that can speak to many people and organizations beyond independent groups' reach.

We'd also like to make these projects more "visible" on our website. At the moment, we simply include a citation and a link to independent initiatives within our Project Scholarship pages.[34] But community-led work could be given more prominence by dedicating a page of our own website to showcase independent groups' outputs and direct readers towards the external sites where they have been established. This page could also provide space to explain what our CC BY license invites and what forms of further support we can offer, underscoring our commitment to encouraging multilingual growth through creative reuse beyond our current infrastructure.

34. http://programminghistorian.org/en/research

4. Conclusion

Programming Historian embraces and celebrates multilingualism in its many iterations: across our multilingual and multicultural team, within our publications in four languages, and throughout our language-sensitive practices, guidelines, and workflows.

Multilingualism, in both principle and practice, supports the fulfillment of our broader objective as a publisher: to advance education in the humanities through the production of learning resources which are circulated in the public domain. Publishing lessons in languages other than English and—critically—making them available as Diamond Open Access support the extension of digital humanities skills and knowledge beyond the sphere of anglophone-dominant online learning. It is about reaching towards self-learners and under-resourced educators who seek ways to build knowledge for themselves and for a future generation of global researchers.

Over time, we've cemented our commitment to developing and disseminating accessible digital humanities learning resources to users around the world. While we are keen and enthusiastic to grow, to produce new lessons, and to involve more language communities, we are aware of the need to balance the growth of our multilingual publishing project with its sustainability.

Indeed, sustaining the multilingual aspect of the project is a relatively recent and evolving challenge, which demands a multifaceted and dynamic response. As we grow, we must continue to adapt to the interrelated needs of our internal infrastructure and those of the communities we aim to serve, which require careful and educated consideration. Ultimately, growing multilingually means shouldering a complex yet rewarding responsibility.

References

Akhlaghi, Andrew. 2021. "OCR and Machine Translation." *Programming Historian* 10. https://doi.org/10.46430/phen0091.

Arenas-Castro, Henry, Violeta Berdejo-Espinola, Shawan Chowdhury, Argelia Rodríguez-Contreras, Aubrie R. M. James, Nussaibah B. Raja, Emma M. Dunne, et al. 2024. "Academic Publishing Requires Linguistically Inclusive Policies." *Proceedings of the Royal Society B* 291 (2018). https://doi.org/10.1098/rspb.2023.2840.

Crymble, Adam. 2016. "Identifying and Removing Gender Barriers in Open Learning Communities: The Programming Historian." *Blended Learning in Practice* (Autumn): 49–60. http://researchprofiles.herts.ac.uk/portal/files/10476604/Blip_2016_Autumn_2016_Final_Autumn_2016.pdf.

Crymble, Adam, and Maria José Afanador-Llach. 2021. "Digital History: The Globally Unequal Promise of Digital Tools for History: UK and Colombia's Case Study." In *Teaching History for the Contemporary World: Tensions, Challenges and Classroom Experiences in Higher Education*, edited by Adele Nye and Jennifer Clark, 85–98. Singapore: Springer.

Crymble, Adam, and Charlotte M. H. Im. 2023. "Measuring Digital Humanities Learning Requirements in Spanish & English-Speaking Practitioner Communities." *International Journal of Digital Humanities* 5:253–82. https://doi.org/10.1007/s42803-023-00066-x.

Dombrowski, Quinn, Tassie Gniady, and David Kloster. 2019. "Introduction to Jupyter Notebooks." *Programming Historian* 8. https://doi.org/10.46430/phen0087.

Eve, Martin Paul, and Jonathan Gray. 2020. *Reassembling Scholarly Communications: Histories, Infrastructures, and Global Politics of Open Access*. Cambridge, MA: MIT Press.

Fiormonte, Domenico. 2015. "Towards Monocultural (Digital) Humanities?" *Infolet*, July 12, 2015. https://infolet.it/2015/07/12/monocultural-humanities/.

Galina, Isabel. 2013. "Is There Anybody Out There? Building a Global Digital Humanities Community." *Red de Humanidades Digitales*, July 19, 2013. https://humanidadesdigitales.net/is-there-anybody-out-there-building-a-global-digital-humanities-community/.

Gil, Alex, and Élika Ortega. 2016. "Global Outlooks in Digital Humanities: Multilingual Practices and Minimal Computing." In *Doing Digital Humanities, Practice, Training, Research*, edited by Constance Crompton, Richard J. Lane, and Ray Siemens, 22–34. London: Routledge.

Gutiérrez De la Torre, Silvia. 2019. "Análisis de corpus con Voyant Tools." *Programming Historian en español* 3. https://doi.org/10.46430/phes0043.

Horváth, Alíz. 2021. "Enhancing Language Inclusivity in Digital Humanities: Towards Sensitivity and Multilingualism: Includes Interviews with Erzsébet Tóth-Czifra and Cosima Wagner." *Modern Languages Open* (1). https://doi.org/10.3828/mlo.v0i0.382.

Isasi, Jennifer. 2023. "Sentiment Analysis with 'Syuzhet' Using R." Translated by Adam Crymble. *Programming Historian* 12. https://doi.org/10.46430/phen0110.

Isasi, Jennifer, and Antonio Rojas Castro. 2021. "¿Sin equivalencia? Una reflexión sobre la traducción al español de recursos educativos abiertos." *Hispania* 104 (4): 613–24. https://doi.org/10.1353/hpn.2021.0130.

Lincoln, Matthew, Jennifer Isasi, Sarah Melton, and François Dominic Laramée. 2022. "Relocating Complexity: The Programming Historian and Multilingual Static *Site Generation*." *DHQ: Digital Humanities Quarterly* 16 (2).

Mahony, Simon. 2018. "Cultural Diversity and the Digital Humanities." *Fudan Journal of the Humanities and Social Sciences* 11:371–88. https://doi.org/10.1007/s40647-018-0216-0.

Nilsson-Fernàndez, Pedro, and Quinn Dombrowski. 2022. "Multilingual Digital Humanities." In *The Bloomsbury Handbook to the Digital Humanities*, edited by James O'Sullivan, 83–92. New York: Bloomsbury Publishing.

Ortega, Élika. 2019. "*Zonas de Contacto*: A Digital Humanities Ecology of Knowledges." In *Debates in the Digital Humanities 2019*, edited by Matthew K. Gold and Lauren F. Klein. Minneapolis: University of Minnesota Press.

Spence, Paul Joseph, and Renata Brandão. 2021. "Towards Language Sensitivity and Diversity in the Digital Humanities." *Digital Studies/Le champ numérique* 11 (1). https://doi.org/10.16995/dscn.8098.

Vidal-Gorène, Chahan. 2023. "La reconnaissance automatique d'écriture à l'épreuve des langues peu dotées." *Programming Historian en français* 5. https://doi.org/10.46430/phfr0023.

Viola, Lorella, and Paul Spence, eds. 2023. *Multilingual Digital Humanities*. London: Routledge.

Zaagsma, Gerben. 2023. "Digital History and the Politics of Digitization." *Digital Scholarship in the Humanities* 38 (2): 830–51. https://doi.org/10.1093/llc/fqac050.