

City University of New York (CUNY)

CUNY Academic Works

Publications and Research

Kingsborough Community College

2022

Automation, Abstraction and Building it Ourselves

Mark E. Eaton

CUNY Kingsborough Community College

[How does access to this work benefit you? Let us know!](#)

More information about this work at: https://academicworks.cuny.edu/kb_pubs/219

Discover additional works at: <https://academicworks.cuny.edu>

This work is made publicly available by the City University of New York (CUNY).

Contact: AcademicWorks@cuny.edu

Automation, abstraction and building it ourselves

By Mark E. Eaton
Associate Professor/Reader Services Librarian
Kingsborough Community College
City University of New York

Abstract

This paper argues that indexers should work collaboratively to build software tools that support our profession. As technology automates the procedural aspects of our work, we need to respond by building tools that support the conceptual labor of indexing.

Can't computers do that?

Indexers, like crossword-makers and library catalogers, are often familiar with the layperson's well-intentioned question: "can't a computer do that job?" Despite offering reassurance that humans do have an important role to play in our jobs, we find sometimes ourselves warily looking over our shoulders to see if someone has made software that will adequately and autonomously make indexes. Our nervousness is certainly understandable. Considering that it is increasingly easy to deploy natural language processing tools, threats of automation do not seem implausible. It is easy to feel disempowered, rather than empowered by technology.

Glenda Browne describes some of the ways that computers and humans coexist in indexing today: "While to my knowledge nobody has been able to create software that effectively indexes a stand-alone book, automation has been used [in some cases] ... with human review." Furthermore, she hypothesizes that "it seems inevitable that automation will be used more in the future." (2016). Technology will probably play an important role in the future of indexing. How will this affect indexing professionals? How will we respond?

Working with technology

While our fears about new technologies are understandable, indexers also have the opportunity to work with technology, rather than just fret about it. These opportunities are considerable. For example, when I began my indexing career, I made a couple back-of-the-book indexes exclusively using MS Word. For those early indexes, I had no specialized software, just a Word document, a slightly paranoid attention to detail, and my somewhat naive ideas about how indexes should work. Needless to say, these indexes weren't the best, nor was the workflow that pleasant.

Later, I adopted Cindex, which was a huge improvement. In part, it taught me the immense value of technology that supports me as an indexer. While this sounds like a somewhat trivial example, the impact of Cindex on my workflow was unambiguous. The value of this software was that it took my Word-based workflow and partly automated it to produce a much simpler process. Many of the tedious tasks that I did in Word were easily handled by Cindex, with a wholly positive impact.

Index as concept

What I learned from this experience is that what we consider to be the "technical" or "procedural" aspects of our work largely reflect the level of automation already built into our workflow. For the most part, we don't have to worry about low-level details like typesetting anymore. Most of us don't even really have to worry that much about formatting our indexes in Word. Software

engineers often speak of this as layers of abstraction. Solving low-level programming problems (or, for that matter, indexing problems) allows those problems to be abstracted away, so that we can focus on higher-level problems. Abstraction is beneficial to indexers because, at its core, our job is conceptual. The procedural work we do supports our conceptual labor.

Since the work we do is conceptual, the tools we use will likely become more conceptual and less procedural. Computers already take care of many of the procedural aspects of our work. This “automation” is the kind of abstraction that software developers build. Abstraction is a positive contribution because instead of focusing on procedural tasks, indexers can focus on the conceptual aspects of indexing that add human value. Bill Johncocks claims that this will “free us to concentrate on the intellectual part of indexing, avoiding the error-prone drudgery” (2013).

Human indexers undeniably provide essential conceptual labor in the creation of indexes. It would be a shame to attempt to build indexing software that automates indexers out of their own area of expertise, not least because this will definitely not help anyone make better indexes. Rather, it is much more useful to build tools that indexers intuitively understand and can use conceptually. Making useful tools is largely about building software that deploys effective abstractions and metaphors (Manovich, 2013) that enhance the capabilities of the indexers who use them.

An open source future

The open source software community provides a useful model for our collaboration on indexing software. The benefits of open source methodologies are numerous, including improved sustainability, collaboration, security and access. Pragmatic arguments for open practices have been around at least since *The Cathedral and the Bazaar* (Raymond, 1999). These open approaches have become widely accepted in software development, and can be translated into building better indexing software.

Unfortunately, there aren’t many openly licensed projects in our field. For example, there are very few well-developed back-of-the-book index software projects on GitHub, a popular platform for open source repositories (for example, see: *Evaluating Variations in Language Laboratory*, 2012; Holler, 2014). Most of these public projects do not appear to be made by an indexer, nor do they appear to be actively maintained. There is an obvious lacuna here, and an opportunity for us to develop open source tools for our profession.

My rhetorical agenda is to suggest that we can come together to collaboratively build the software we need. I recognize that this may seem like a tall order, as many indexers do not see themselves as programmers, but I think we are up to the task. This is because open source software projects allow us to

do more by combining our abilities. Open source projects allow for many contributors, with varying technical skill levels, to participate. These methodologies can promote broad engagement, and produce a diversity of solutions to the common technical challenges that our profession faces.

We may be a small community with very specific needs, but my hope is that there is enough coding talent – and collaborative spirit – among us to collectively build solutions that support us all. We shouldn't necessarily be afraid of technology's impact on our profession. Our future technologies are in our hands, especially if we build them ourselves.

References

- Browne, Glenda (2016) 'Ebook Indexing Update'. *The Indexer* 34(4).
- Evaluating Variations in Language Laboratory (2012) 'Indexer: Generates a Back of the Book Index Using the Text of a Book'. <https://github.com/evllabs/Indexer>.
- Holler, Michael (2014) 'Thesis-Undergrad: Automated Back-of-the-Book Indexing'. <https://github.com/mikeholler/thesis-undergrad>.
- Johncocks, Bill (2013) 'EPUB Indexes and the Future of Indexing Revisited'. *The Indexer* 31(4).
- Manovich, Lev (2013) *Software Takes Command: Extending the Language of New Media*. London: Bloomsbury.
- Raymond, Eric S. (1999) *The Cathedral and the Bazaar: Musing on Linux and Open Source by an Accidental Revolutionary*. Cambridge, MA: O'Reilly.