

GUILHERME ORLANDINI HEURICH. *CODERSPEAK: THE LANGUAGE OF COMPUTER PROGRAMMERS.* LONDON: UCL PRESS 2024. 214 P. ISBN: 9781800085992

XINYUAN WANG¹

Guilherme Orlandini Heurich presents the world of Rubyist computer programmers in this vivid and refreshing ethnography. Ruby, which he calls a marginalized computing language and a hallmark of the Web 2.0 era (34), is his case for examining programming as a socio-technical collective practice. He defines 'coderspeak' as a language that pretends to be both human and machinic, building on perceived kinship between computing processes and human cognition. Emerging in the late 1970s as part of a rising computational culture, programming languages like Ruby, he argues, transcend the scale of technical tools and become cultural products.

Heurich's central argument—that programming cannot be understood without the cultural and social context in which it occurs—is carried through an ethnography of Ruby programmers, drawing on interviews with figures including Ruby's creator and the author's own experience working at a tech company – Upstream. The anthropological framing is successful in shifting attention from abstract programming to the social processes behind it, revealing how coding practices are shaped by collaboration, identity, and politics. This approach contributes meaningfully to anthropology's engagement with technology, and invites discussions between natural and artificial linguistics, though its impact on computer science is less certain.

Heurich blends technical explanation with ethnographic insight, even structuring parts of his prose in the style of Ruby code. Themes include open-source software as a 'recursive public of a massive cultural shift' (27), collaborative working nature, and the gender and racial dynamics of tech communities. Heurich seeks to show that programming is as much about people and culture as it is about code. Behind the phenomena of technical innovations, it is the hugely complex social and technical process which anthropologists should be looking at (28). His observations aim to change the way we think about knowledge and power in the establishment of global internet infrastructures. The argument appears successfully, particularly in emphasizing the social dimensions of programming, which are often overlooked and technically black-boxed. By using an anthropological lens, Heurich sheds light on the human factors that shape coding practices, contributing valuable insights to both anthropology and the history of computer science.

Heurich systematically documents the genealogical era when Ruby was used as main programming language and on what applications was the language distributed in comparison to other popular programming languages (34). He traces the 1960s and 1970s – the time when 'computing would start to become masculinised' (125). Then the early 2000s sees 'the emergence of free software and a politics of community building', while the 2010s saw the

¹ School of Anthropology and Museum Ethnography, University of Oxford. Email: connie4anthro@gmail.com

dominance of big tech in every aspect of society when '[c]ode was still play, but it had become money too' (101). Touches on capitalism and neoliberalism enrich the anthropological perspectives of this book, yet limited criticism is applied. One can see a compassionate view among the honored history of free software communities, and a potential critical lens applied to the privatization enforced by big tech capitals, although it is not further discussed or engaged with. For instance, the brief but incisive case of GitHub's centralized control over open-source projects hints at a stronger critique that is never fully developed. A potential case study of Github (169) is concise and forceful to the topic: 'GitHub hosts the majority of open-source projects in the world, but GitHub itself is not open source. What if GitHub decides to shut its services down?' While these themes are persuasively explored, the book engages less deeply with economic anthropology than it could.

Discussions of minority status are tied to Ruby's own position in the programming world and to the background of its creator, Matz, a Japanese Mormon. Matz likens his lifelong experience as a religious and linguistic minority to Ruby's trajectory, from obscurity to popularity and back to niche status, seeing this cyclical marginality as natural and even positive. The book uses this analogy to connect personal identity, linguistic hegemony, and software community culture, including contrasts between 'people-focused' and 'technology-focused' hacker conferences and the conflicts between English hegemony and Japanese creators and the core team of Ruby spaces. These reflections link sociolinguistics with computing practice, revealing how both natural and artificial languages shape community boundaries.

The Ruby community's approach to language design—eschewing prescriptive tools in favor of diversity—exemplifies the book's concern with resisting technical monocultures. Here Heurich's anthropological lens is effective, showing how design philosophies mirror cultural values. Yet the book leaves open whether such pluralism can withstand broader market pressures and infrastructural dependencies. For anthropology, *Coderspeak* demonstrates how ethnographic methods can be applied to the study of technical fields. It bridges the gap between the technical and the social, offering a fresh perspective on the anthropology of technology. The book's success ultimately depends on its interdisciplinary reach and its ability to engage both anthropologists and technologists in a meaningful dialogue. Although, it is worth questioning whether its contribution is as groundbreaking in computer science itself as for anthropology, where the technical aspects of programming are more often prioritized over social analysis.

The book closes with an open question: if Ruby fades from use, is 'coderspeak' still a language without speakers? This liminal framing reinforces the work's central concern with the boundaries between natural and artificial language. By consistently interrogating how code operates as a communicative practice, Heurich offers a compelling, if sometimes under-critiqued, account of programming as cultural work.

Overall, *Coderspeak* succeeds as an anthropological study, offering fresh insights into the social dimensions of programming and valuable case material for teaching and research. Its main limitation is a reluctance to push its critiques of power and capital to the same depth as its ethnographic description—a choice that makes the book accessible and engaging but leaves some of its most provocative points only partially explored.

This work is copyright of the author. It has been published by JASO under a Creative Commons Attribution-NonCommercial-NonDerivatives ShareAlike License (CC BY NC ND 4.0) that allows others to share the work with an acknowledgement of the work's authorship and initial publication in this journal as long as it is non-commercial and that those using the work must agree to distribute it under the same license as the original. <https://creativecommons.org/licenses/by-nc-nd/4.0/>

