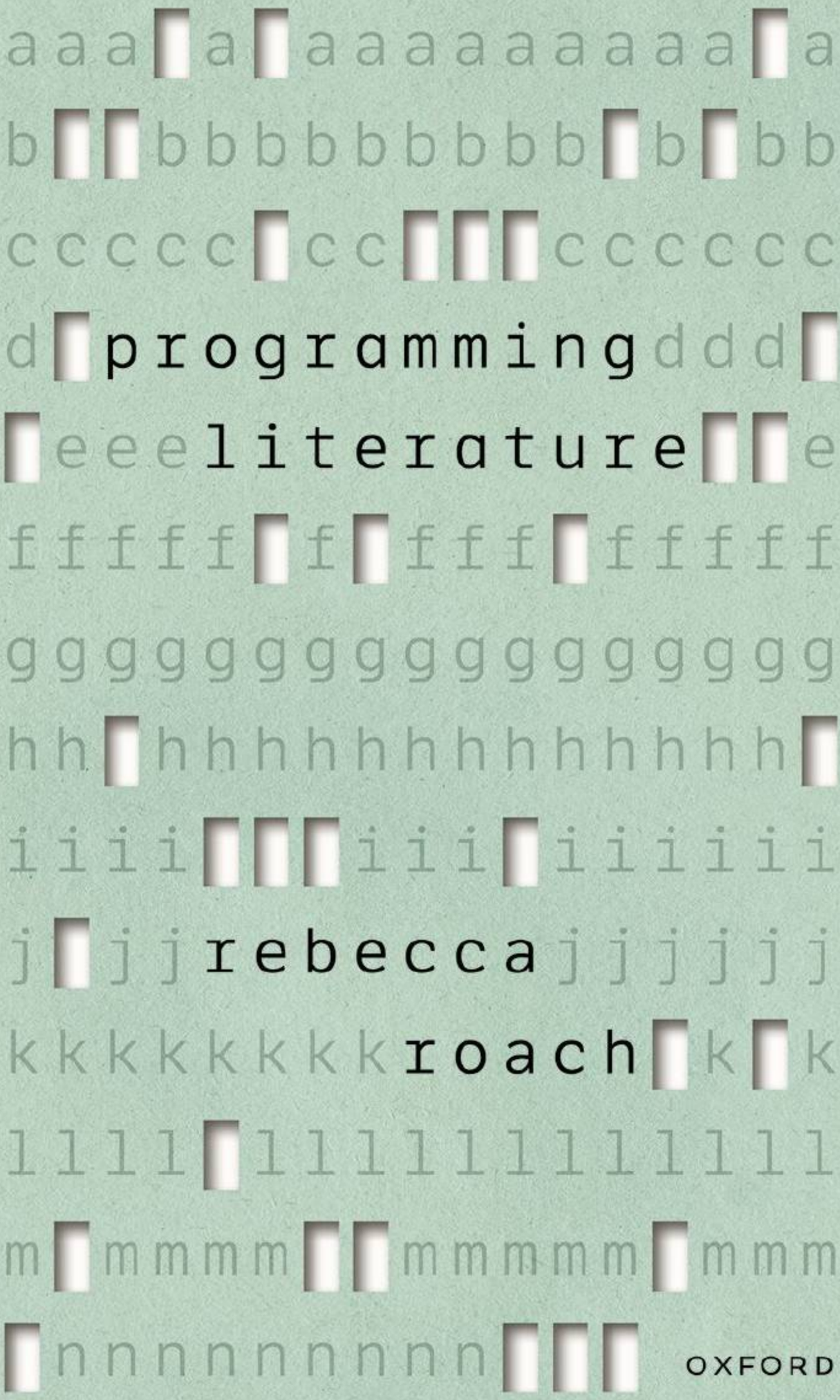

programming
literature
rebecca
roach
OXFORD

Programming Literature

Programming Literature

REBECCA ROACH

OXFORD
UNIVERSITY PRESS

OXFORD
UNIVERSITY PRESS

Great Clarendon Street, Oxford, OX2 6DP,
United Kingdom

Oxford University Press is a department of the University of Oxford.
It furthers the University's objective of excellence in research, scholarship,
and education by publishing worldwide. Oxford is a registered trade mark of
Oxford University Press in the UK and in certain other countries.

© Rebecca Roach 2025

The moral rights of the author have been asserted.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, transmitted, used for text and data mining, or used for training artificial intelligence, in any form or by any means, without the prior permission in writing of Oxford University Press, or as expressly permitted by law, by licence or under terms agreed with the appropriate reprographics rights organization. Enquiries concerning reproduction outside the scope of the above should be sent to the Rights Department, Oxford University Press, at the address above.

You must not circulate this work in any other form
and you must impose this same condition on any acquirer.

Published in the United States of America by Oxford University Press
198 Madison Avenue, New York, NY 10016, United States of America

British Library Cataloguing in Publication Data
Data available

Library of Congress Control Number: 2025939252

ISBN 9780198976561

DOI: 10.1093/9780198976578.001.0001

Printed and bound by
CPI Group (UK) Ltd., Croydon, CR0 4YY

The manufacturer's authorised representative in the EU for product safety is
Oxford University Press España S.A. of Parque Empresarial San Fernando de Henares,
Avenida de Castilla, 2 – 28830 Madrid (www.oup.es/en or product.safety@oup.com).
OUP España S.A. also acts as importer into Spain of products made by the manufacturer.

Links to third party websites are provided by Oxford in good faith and
for information only. Oxford disclaims any responsibility for the materials
contained in any third party website referenced in this work.

For D, E, & F

Acknowledgements

Just as this book engages with the messy bits and bytes of life, so too it is the product of that messiness, and I am so thankful for all the people who have lived it with me. The book started life back in 2015 when I knew very little about computing and was intrigued by a piece of code printout in an archive that I couldn't read. Learning my way out of that illiteracy has been a joy—it gave me something to ask my (software architect) now-husband about on our first date, and it seems to have sustained us ever since. The book was agreeably slowed by two children, less agreeably by a pandemic, and always enriched by the conversations of colleagues, students, archivists, friends, and family. In particular, my colleagues and students, first at King's College London and then at the University of Birmingham, have made this book so much better, as have the suggestions and ideas of participants at research events and informal coffees too numerous to mention, but whose generous contributions are, for me, the highlight of academic community. Kaitlin Staudt and Alys Moody read every word of this, so huge thanks for what has also been a big departure from literature for them too. I wish that Laura Marcus was here to admit bemusement at where this research has ended up. Outside of academia, I have been graced with a family of coders, and would like to thank Tom Roach, Caitlin Roach, Martin Roach, Chris Kerr, and especially David Illsley, for humouring my inquiries and patiently reminding me that (despite my occasional outburst) not everyone in the sector is a tech bro. Thanks too to my non-coding family, related and otherwise, who have made this project joyful and feasible.

This book began as a very different project, which (much to my chagrin at the time) split into two different monographs in 2023. The earliest stages of that project were supported by a European Research Council Grant 'Ego Media: The impact of new media on forms and practices of self-presentation', 2014–2019, grant no. 340331, and a Harry Ransom Center research fellowship (2016). Subsequently, a travel stipend from the Rockefeller Archive Center, research assistance from the University of Birmingham, and the significant support of a Leverhulme Trust Research Fellowship have been vital to the book's completion. The Leverhulme Trust was especially supportive when faced with the many changes to my fellowship plans that maternity leave

and breastfeeding entailed, and I am very grateful to them for making that complicated time as stress-free as possible.

For permission to reproduce archival materials, I would like to thank J. M. Coetzee, Joseph O'Rourke, Carcanet Press, and the estates of Christine Brooke-Rose, Hugh Kenner, and William Gaddis. I am grateful to the curators and staff, whose research and technical expertise was invaluable, at: the British Library; the Centre for Computing History, Cambridge; the Charles Babbage Institute, University of Minnesota; the George Padmore Institute; the Harry Ransom Centre, University of Texas at Austin; the Internet Archive; the John Rylands Research Institute and Library, University of Manchester; the Library of Congress; Special Collections at Washington University, St. Louis; and likely more that have slipped my mind (I plead lack of sleep with newborns, but my huge apologies for such lapses). I am also indebted to the individuals who shared information from their past, enlightened me about technicalities of obsolete code, and responded to my questions with patience and grace, including Robert A. Amsler, Jim Browne, Damien Burke, Nell Dale, Barry Landy, Jessica Meyerson, and others.

Portions of chapter one appeared first in the articles 'J. M. Coetzee's Aesthetic Automatism', *Modern Fiction Studies*, 65.2 (2019), 308–37; and '"Hero and Bad Motherland": J. M. Coetzee's Computational Critique', *Contemporary Literature*, 59.1 (2018), 80–111. Material from chapter two first appeared in 'Expanding the History of Humanities Computing: Hugh Kenner's Conversational Criticism' in *Digital Scholarship in the Humanities*, 38.2 (2023), 679–95. I am beholden to the editors and to the anonymous reviewers for their enthusiastic feedback and constructive suggestions. I am also grateful to Jack McNichol at Oxford University Press for his zeal and to my two reviewers there who restored my confidence and gave such valuable and heartening feedback.

Contents

List of Figures xi

Introduction: Literate Programming 1

1. J. M. Coetzee's Aesthetic Automatism 23

2. Hugh Kenner's Hobby 64

3. Christine Brooke-Rose's Towers of Babel 93

4. Kamau Brathwaite: Creolized Computing and Processual Aesthetics 126

5. William Gaddis, Technical Writer 156

Conclusion: Documenting Difficulty 183

Works Cited 189

Index 203

List of Figures

1.1 Excerpt from 'Line Generator', May 1963, Container Obs 143, J. M. Coetzee Papers. Image courtesy of the Harry Ransom Center, The University of Texas at Austin. 54

2.1 'A Travesty Generator for Micros', Folder 3, Box 36, Hugh Kenner Papers. Image courtesy of the Harry Ransom Center, The University of Texas at Austin. 80

3.1 Christine Brooke-Rose, 'Manuscript draft of *Remake*', '*Remake*, Original handwritten manuscript, in two notebooks, 1992–1993, vol. 2', Folder 2, Box 23, Christine Brooke-Rose Papers. Image courtesy of the Harry Ransom Center, The University of Texas at Austin. 98

3.2 Christine Brooke-Rose, 'Typescript revisions to *Remake*', '*Remake*, early drafts Typescript with holograph revisions', Folder 4, Box 23, Christine Brooke-Rose Papers. Image courtesy of the Harry Ransom Center, The University of Texas at Austin. 99

Introduction

Literate Programming

Eminent computer scientist, Donald Knuth, thinks that programming should be more like literature. Across his career, Knuth has tried to persuade his colleagues to 'treat a program as a piece of literature, addressed to human beings rather than to a computer'.[1] The programmer, he explains, should conceive of herself as akin to an 'essayist, whose main concern is with exposition and excellence of style'.[2] In focusing on exposition, style, and the reader, Knuth's analogy draws on components that regularly feature in literary–critical analysis and will be familiar to literary scholars. We have, it would seem, a nice argument for the analogousness of programming and literature and a nice argument for literature's distinctiveness in a computational world where its value is often questioned, and from a decorated computer scientist at that.

Part of the impact of this analogy comes from the perceived differences between code and literature. While programming and (literary) writing are exercises in symbol manipulation, they are usually held up as opposites. Knuth is aware of this; he deliberately couches his paradigm as unconventional: 'I must warn the reader to discount much of what I shall say as the ravings of a fanatic who thinks he has just seen a great light.'[3] Ravings aside, things start to get more difficult the moment we push his analogy further. Look at that first quote again, and particularly that little comma: 'treat a program as a piece of literature, addressed to human beings rather than to a computer'. The logic of the syntax suggests that literature is characterized by its address to people, implying that literature addressed to a machine is a tautology. It is a distinction that makes sense within computer science, but if you are situated in literary studies, it is a surprising characterization of literature on which to insist: both banal and thought-provoking.

[1] Donald E. Knuth, 'Literate Programming', *The Computer Journal*, 27.2 (1984), 97–111, 97. doi: 10.1093/comjnl/27.2.97.

[2] Knuth, 'Literate Programming', 97.

[3] Knuth, 'Literate Programming', 97.

We might further wonder at Knuth's term for this programming paradigm: 'literate programming'. Not 'literary' programming but 'literate'. Literate suggests competency—in reading and writing, or in a more specific domain of knowledge ('literate in programming', for example). Literature, meanwhile, conjures up two, potentially diverging, meanings: written work of aesthetic merit ('English literature') and the general body of writing on a specific subject ('the literature on programming'). Knuth's term seems to privilege the former, even as it cleaves to the latter in its justification, thus leaving open the question of just how distinctive and autonomous 'literature' ultimately is in this analogy.

The stakes get only more confusing when Knuth explains his methodology: one 'that combines a programming language with a documentation language, thereby making programs more robust, more portable, more easily maintained, and arguably more fun to write than programs that are written only in a high-level language.'[4] The terminology is likely confusing for literary scholars—what is the difference between programming and documentation languages and how exactly is literature tied to robustness, portability, and maintenance? It seems to imply a framework that Knuth is looking to overturn—an attempt to reprogramme the discussion or, to put it in literary terms, 'make it new' in code.

In some respects, the disciplinary incomprehension that Knuth's analogy seems to highlight is easily overcome. His methodology can be readily explained in terms more familiar to literary scholars. Programming is traditionally divided between, firstly, the code that is executable (i.e. can be 'read' and enacted) by the machine and, secondly, comment, or 'documentation', that is not executable and instead aimed at helping the human reader to understand the code.[5] In literary terms, we have a text and its interpretative glossing. Knuth is promoting a type of programming that takes a leaf out of literary studies' historic commitment to textual interpretation for human readers in an industry that has often devalued these features (in favour, for example, of machine execution time) with associated costs for the preservation and understanding of programmes (i.e. maintenance, portability, etc.).[6] Literature, for Knuth, becomes a metonym for a certain system of value.

[4] Knuth, 'Literate Programming', 97.

[5] We will get on to the distinction between high level and machine languages in Chapter 1.

[6] In Chapter 5 and the conclusion, I make the point that documentation is a major route by which programmes insert themselves into legal, political, and ethical frameworks, and through which technology is mediated.

Literate programming is also analogous to literature for what it values: form. It is a kind of programming that privileges explication and incorporates that into the form of the programme itself. The programmer will resemble an author, who 'with thesaurus in hand, chooses the names of variables carefully and explains what each variable means'. The goal is 'a program that is comprehensible because its concepts have been introduced in an order that is best for human understanding, using a mixture of formal and informal methods that reinforce each other'.[7] In characterizing literature by its attention to form and its meaning for the reader, Knuth is using a conventional definition, albeit in less familiar terms.

In other respects, though, the analogy that Knuth draws works to highlight the difficulties and rewards of thinking across disciplines. The conflation of literate and literary is one that some literary critics might balk against, privileging as it does competencies over texts and remaining mute over the place of aesthetic merit. This conflation is confusing to such scholars because Knuth is very explicitly drawing on aesthetic merit in his appeal. Terms have distinct back histories and definitional nuances in different disciplines and this substitution doesn't work for literary scholars. Thinking across disciplines requires attending to these intricacies and to their larger import; as enthusiastic a raving madman as Knuth is, he is talking to colleagues in computer science rather than a cross-disciplinary audience. Knuth turns to literature as a means to encourage his discipline to value and practice programming in a different way.

Knuth does, however, offer the literary scholar something via his analogy. We have here a theory of literature's affordances—affordance being a term that is often deployed in human–computer interaction and design to describe 'the potential uses or actions latent in materials and designs' and that is, as Caroline Levine has noted, equally useful in thinking about aesthetic forms.[8] We might, ultimately, accept or reject Knuth's conception, but the analogy is generative. It pushes us to consider what exactly we think literature is, what it does, how we interact with it—and perhaps also rethink computing in the process. For all code's supposed singularity, programming is recognizably a linguistic exercise and is surrounded by the documents and writing that mediate computing: Knuth's argument, published as an article in a paper-based journal, is evidence of this inextricability. This

[7] Knuth, 'Literate Programming', 97.

[8] Caroline Levine, *Forms: Whole, Rhythm, Hierarchy, Network* (Princeton: Princeton University Press, 2015), 6.

enmeshment in language is something that computing and literature share. While we often conceive of the two as opposites, it can be very valuable to see a discipline through alien eyes and to recognize that our assumptions (about literature) are sometimes just that. In this book, I pursue this line.

Programmes for Human Readers

Before J. M. Coetzee was a writer of Nobel Prize–winning novels, he was a programmer for one of the most significant Cold War supercomputer projects in Britain. During the Second World War, experimental writer Christine Brooke-Rose worked in cryptanalysis with computer pioneer Alan Turing. When not teaching literature or opining about modernism as a public intellectual, literary critic Hugh Kenner was a devoted computer hobbyist. Literary scholars have often not known what to make of these 'other careers': intimidating, irrelevant, outside of the scope of literary studies, surely?

In this book, I start from a different assumption, one born of Knuth's analogy: that, far from irrelevant, these experiences were significant in the development of these writers' literary projects. Rather than assuming that computing represented a day job that funded the writing of a novel, or a machine for efficient transcription, I ask: what might we learn by attending to individual writers who have engaged deeply with computer technologies and cultures? What effects might this have had on their understanding of their own literary projects? And more expansively, what possible relations do these writers conceive between computing and literature?

The title of this book gestures towards this latter question. The ambiguity is deliberate. Is programming a kind of literature? Is literature programmable? What defines literature and programming? Are they activities, objects, or forms? There are multiple ways in which relations between the two have been conceived and analogies and disanalogies might be drawn. The consequences are not negligible or historical, but questions we are grappling with today. This is a set of debates that are disciplinary (the value of the humanities in a digital society), but also reflect more foundational concerns around the aesthetic, cultural, and ethical assumptions we build into our computational texts (or the beliefs guiding Silicon Valley); how computation affects the kinds of labour we value (whether in the automation of human labour or the widening use of generative artificial intelligence); and indeed what it means to be human (readers) in a digitally mediated world.

Through a focus on these writers, this book lays out a history of many of these debates, while outlining how conceptually executable this analogous thinking has and can been.

Reprogramming Literature

Programming Literature brings together a group of people best known today for their literary endeavours and details the specifics of their, sometimes surprising, engagement with programming and computing more broadly. The writers I discuss are characterized by their lengthy, sometimes decades-long, commitment to computation and by their willingness to take it seriously as a conceptual and material endeavour. The specific nature of that engagement varies—from professional programmer to hobbyist, from a theoretical interest in code to a practical interest in word processing. That variety is deliberate. I trace the specific material and intellectual journeys of these writers as a recovery exercise, to tell the untold stories of their computer work.

Methodologically, I have been driven by a desire to pin down the particularities of writers' engagement with specific devices, technologies, and paradigms. I think it matters if a writer's experience of computing is in devising an operating system for a supercomputer in a Cold War military facility or in soldering together components of an electronic calculator at home in the basement of an evening. Computers can be experienced as objects, effects, interfaces, processes, ideas, and cultures, and writers can think with or against them in very different ways. This book tries to engage with that nitty-gritty and the idea that only in taking that detail seriously can we think with these writers to better understand how they engage with computing.

Such an approach is partly born out of an attempt to reprogramme—deprogramme perhaps—a certain argument about computing and its cultural import. There has long been a tendency, which we see in public discourse, in industry chatter, and in some academic writing, to overgeneralize discussions of computing and to think ahistorically about the technologies, practices, cultures, and media that have made up 'the digital' (that definite article does a lot of work). There is a certain idea that has long cloaked digital computing. It stems from mathematician Alan Turing's idea, put forward in the late 1930s, that a machine capable of calculating any computable sequence was theoretically possible: the so-called universal

Turing machine.[9] This theoretical construction of the universal machine would be influential in the development of stored digital computing; the concept has additionally become a basis by which we understand digital computing in terms of universality of practice, materiality, and context—Lydia Liu pointedly notes, for example, that '*the central problem of the digital revolution is the recasting of the human mind as a psychic machine on the model of the computing machine*'.[10] The abstraction processes inherent in programming further encourage this sleight of thought; back in 2011, Wendy Chun noted that 'software is unique in its status as metaphor for metaphor itself. As a universal imitator/machine, it encapsulates a logic of general substitutability'.[11]

Such mental algorithms have proven extraordinarily productive (as many of these writers' activities demonstrate), but they also have unintended consequences. As historians of computing themselves point out, they obscure the very different ways in which we have understood these technologies in specific moments and locations.[12] It might seem self-evident that programming in the 1960s for a corporate behemoth in its London offices, or saving files on a desktop computer with an intermittent power supply in the 1980s in Irish Town, Jamaica, could produce different understandings of computing, but all too often we stretch for universal accounts, inadvertently tagging these heterogeneous experiences as obsolescent.

[9] A. M. Turing, 'On Computable Numbers, with an Application to the Entscheidungsproblem', *Proceedings of the London Mathematical Society*, s2–42 (1937), 230–65. doi: 10.1112/plms/s2-42.1.230.

[10] Lydia H. Liu, *The Freudian Robot: Digital Media and the Future of the Unconscious* (Chicago: University of Chicago Press, 2011), 34, her italics.

[11] Wendy Hui Kyong Chun, *Programmed Visions: Software and Memory* (Cambridge, MA: MIT Press, 2011), 2.

[12] There has been a real push over the last ten or so years to expand the remit of histories of computing to be more (geographically) global in focus and attentive to the contributions of 'minorities': women, people of colour, those working in linguistic contexts that are not English, or outside of large (corporate and government) institutions. Eden Medina, *Cybernetic Revolutionaries: Technology and Politics in Allende's Chile* (Cambridge, MA: MIT Press, 2011); and James W. Cortada, *The Digital Flood: The Diffusion of Information Technology across the United States, Europe and Asia* (New York: Oxford University Press, 2012) offer early examples. Recent notable books include: Dinesh C. Sharma, *The Outsourcer: The Story of India's IT Revolution* (Cambridge: MA: MIT Press, 2015); Benjamin Peters, *How Not to Network a Nation: The Uneasy History of the Soviet Internet* (Cambridge, MA: MIT Press, 2016); Julien Mailland and Kevin Driscoll, *Minitel: Welcome to the Internet* (Cambridge, MA: MIT Press, 2017); Mar Hicks, *Programmed Inequality: How Britain Discarded Women Technologists and Lost Its Edge in Computing* (Cambridge, MA: MIT Press, 2018); Joy Lisi Rankin, *A People's Histories of Computing in the United States* (Cambridge, MA: Harvard University Press, 2018); and Charlton D. McIlwain, *Black Software: The Internet and Racial Justice, from the AfroNet to Black Lives Matter* (New York: Oxford University Press, 2020).

Practically speaking, my commitment to detailing material and personal histories has entailed extensive use of archival research. I draw on the personal papers and files of many of the authors here, and their publishers, as well as the multiple collections related to computer history held by (among others) the Charles Babbage Institute at the University of Minnesota. From these materials, I can narrate details of these writers' computer activities, clarify the historical record, and demonstrate significant influences on their practice. This enables me to tell the often unknown but crucial story of how these individuals engaged with computation.

As is likely clear, this is an account aimed at literary scholars rather than specialists in computing (although I hope these readers won't be put off). I don't assume any technical familiarity but labour to translate computing's material histories, theoretical frameworks, and disciplinary quirks for readers. This means that I do spend time explaining certain concepts and operations and detailing various computational devices, information that may be superfluous for some readers who bring their own theoretical and practical expertise. Contrastingly, I assume a degree of familiarity with literary methods, contexts, and language appropriate to readers versed in literary studies. When I began the research for this project, I was a literary scholar and a novice in computer science and history. Much of the explanation that follows tracks my own education—both self-directed and heavily reliant on the gracious assistance of more knowledgeable colleagues, friends, and family. Inevitably, I will have got the balance between assumption and explanation wrong on occasion, my apologies in advance. Interdisciplinary work is gratifying and difficult.

In what follows, I utilize traditional literary methods, applying them both to traditional texts, such as novels, memoirs, poetry, and literary criticism, and to less familiar texts, such as computer programmes and technical documentation. In so doing, I seek to demonstrate the value of applying literary–critical methods and value systems to these less conventional documents and objects, adopting as it were Knuth's insight. I hope to underline the importance of the humanities as a mode of knowledge generation while expanding their object remit.

I am not alone in such an endeavour. *Programming Literature* situates literature in a wider media ecology and, despite the misleading nomenclature, will be recognizable to book historians for being a contribution to that field. It follows in the footsteps of the exceptional work of scholars such as N. Katherine Hayles, Lisa Gitelman, Matthew Kirschenbaum, Whitney Trettien, and others who have used book history approaches to particularize

the practices, media, and contexts that surround reading, writing, and informational ecosystems—computational or otherwise.[13] In what follows, I supplement textual interpretation with 'media specific analysis', Hayles's term for 'a mode of critical attention which recognizes that all texts are instantiated and that the nature of the medium in which they are instantiated matters', and Matthew Kirschenbaum's recognition that, while computer writing technologies 'present an illusion of immateriality', the computer's formal environment is a built one and digital inscription never immaterial.[14] However, I also recognize that (computer) media can be as much a mental and social creation as a physical instantiation. Lisa Gitelman has spoken of media expansively as 'socially realized structures of communication, where structures include both technological forms and their associated protocols, and where communication is a cultural practice'.[15] It is a definition I find generative, encouraging us to contemplate the interplay of the material and the abstract, the protocological and the cultural, as it affects our understanding of computing as a medial and formal resource in a specific moment and location.[16]

In adopting approaches from book history and literary criticism, this book is an enthusiastic defence of the humanities in an environment in which they are sometimes denigrated. In following these writers' intellectual and practical engagement, the book is a demonstration of the value (social, economic, aesthetic, and political) to be had in applying the same rigor to the products and processes of the tech sector as we do to those classed 'literary'. Knuth was right. If we grant computer programmes the same cultural value as a piece of literature, we can start to attend to and critically analyse the worlds they represent, the effects they produce, the (non-human) reader–users

[13] N. Katherine Hayles, *My Mother Was a Computer: Digital Subjects and Literary Texts* (Chicago: University of Chicago Press, 2005); Lisa Gitelman, *Paper Knowledge: Toward a Media History of Documents* (Durham, NC: Duke University Press, 2014); Matthew G. Kirschenbaum, *Track Changes: A Literary History of Word Processing* (Cambridge, MA: Harvard University Press, 2016); and Whitney Trettien, *Cut/Copy/Paste: Fragments from the History of Bookwork* (Minneapolis: University of Minnesota Press, 2021).

[14] N. Katherine Hayles, 'Print Is Flat, Code Is Deep: The Importance of Media-Specific Analysis', *Poetics Today* 25.1 (Spring 2004), 67–90; and Matthew G. Kirschenbaum, *Mechanisms: New Media, and the Forensic Imagination* (Cambridge, MA: MIT Press, 2008), 135.

[15] Lisa Gitelman, *Always Already New: Media, History, and the Data of Culture* (Cambridge, MA: MIT Press, 2006), 7.

[16] Some interesting work in this vein that specifically focuses on questions of agency and control (often drawing on critical and political theory) includes Alexander R. Galloway, *Protocol: How Control Exists after Decentralization* (Cambridge, MA: MIT Press, 2004); Seb Franklin, *Control: Digitality as Cultural Logic* (Cambridge, MA: MIT Press, 2015); and the writings of David M. Berry.

they imagine. Mark C. Marino, in his book *Critical Code Studies*, has made a persuasive case for adopting literary–critical methods in the study of code.[17] I embrace that idea here but expand my remit beyond code to argue that in so doing we can better understand the social, aesthetic, political, and economic import of computing in its specific incarnations, in its algorithms, interfaces, and platforms. We need to better understand what affordances programmes and interfaces and machines offer—or specific people have perceived them to offer in specific moments—and what the consequences are for our assumptions about the relation of computing and literature (here broadly conceived in line with Knuth's use of the term) in our world. We need to better understand what it is that a programme executes in addition to a calculation, and literary–critical methods provide a crucial avenue for so doing. Literary enquiry offers a means, I contend, to reprogramme our current narratives about computing.

Unprogrammatic Literature

Despite such abstract ideals, I don't want to lose sight of the literature, nor the import of my findings for literary enquiry. The materially driven approach I take here is as useful to our conceptions of literature as it is to our conceptions of computing. Literature becomes, in this view, part of a media ecology that includes computing, but a part that, as Knuth recognized, commands (command lines?) a certain attention and status.

In the course of this book, we will see that working with computers can lead writers to conceive of literature in various different computational terms—as machine, as technology, or as a set of practices, frameworks, and processes. Through their work with computers, all of the writers discussed here also evaluate the facility of the codex (literature's pre-eminent technology) to inscribe, store, and transmit knowledge, and the associated cultural prestige it has historically claimed. They contemplate it in relation to other media—how its perceived affordances have shifted with the advent of electronic digital computing technologies and how that in turn has affected our understanding of literature (its autonomy, singularity, and function) writ large. In working with computers and programmes, these writers contemplate the affordances of language, forms, objects, and even literary–critical methods, just as much as they contemplate the affordances of code,

[17] Mark C. Marino, *Critical Code Studies* (Cambridge, MA: MIT Press, 2020).

interfaces, and machines. Their literary and computational productions reflect this broad thinking.

Programming Literature then offers an account of a group of authors who have grappled long and hard with questions about the technological forms, protocols, and media through which literature and computing are materialized. However, this is deliberately *not* a programmatic account of 'literature in the age of computing'. If I am being wry, I would say that it's a product of my suspicion of any definite article hovering, drone-like, over computers. More seriously though, I am concerned that such grandiose accounts merely reinforce the universalizing rhetoric that often accompanies computing and that can encourage us to lose sight of the heterogeneity and historical specificity with which writers have understood relations between literature and computing. Abstractions miss things.

In emphasizing this variety, I am also seeking to reprogramme (or perhaps debug) a trend within Anglophone literary studies to privilege formal innovation in conception of relations between the two. This is partly an accident of history. At mid-century, the rising interest in formal logic and information theory—a dematerialized, probabilistic conception of communication—which would prove vital for intellectual advances in digital computing, tracked literary modernism's own ascendency and institutionalization.[18] These developments in (what would become) computer science and literary formalism 'shared theoretical assumptions', as Dennis Tenen points out—even as those theoretical assumptions also became the grounds for disciplinary tussles over the extent of the former's domain, as when Warren Weaver, influential proponent of information theory, argued that 'entropy not only speaks the language of mathematics; it also speaks the language of language'.[19] Liu has persuasively highlighted direct lines influence between the two—for example from James Joyce's experimental fiction to information

[18] See N. Katherine Hayles, *How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics* (Chicago: University of Chicago Press, 1999) for an account of the intellectual emergence of information theory, cybernetics, and the import for literature. See also Heather A. Love, *Cybernetic Aesthetics: Modernist Networks of Information and Data* (Cambridge: Cambridge University Press, 2024). Noting the shared trajectories of T. S. Eliot and Norbert Wiener, she asks, 'What new lines of affiliation might we discover if we took seriously the suggestion that the core work of these two thinkers—one inclined to literary experimentation and the other to mathematics, probability, and statistics—stems from similar questions?', 6. One such hope she articulates is that by 'locating a cultural pre-history to cybernetics' technological innovations in modernist literature's formal experiments' we might 'cultivate new interdisciplinary connections…showcasing the potential for STEM fields to inform literary study and for literary analysis to alter STEM frameworks' (9). I hope so too.

[19] Dennis Tenen, *Plain Text: The Poetics of Computation* (Stanford: Stanford University Press, 2017), 61; and Warren Weaver, 'Recent Contributions to the Mathematical Theory of

theorist Claude Shannon's conception of Printed English, a method for determining statistically 'how much information is produced on the average for every letter of a text in that language.'[20] (We could add to this the pleasing fact that eminent computer scientist and son of Bloomsbury, Christopher Strachey, would publish some of the earliest computer poetry in the modernist little magazine *Encounter*, alongside a short story by William Faulkner.[21]) Meanwhile in her monograph *Digital Modernism: Making It New in New Media*, Jessica Pressman traces the intellectual lineage of literary modernism—represented by Ezra Pound's famed call to 'make it new'—through to contemporary born–digital experimental literature. She not only demonstrates the modernist roots of digital literature but also argues that 'modernism is centrally about media.'[22]

While I don't disagree with such accounts (far from it), I also think that our tendency to focus on formal connections means that we can sometimes lose sight of the fact that this is but one understanding of computing's relation to literature and literary enquiry. Samuel Beckett's fascination with replicating logic trees in his prose fiction (see Chapters 1 and 2 for different writers' takes on this) is intriguing, but it isn't the only way we can think about relations. Frustrations over printer errors or local computer culture might be equally as generative.

There are, I contend, a myriad of ways in which computing has affected the literary project since mid-century; *Programming Literature* demonstrates that relations between the two have been imagined very differently by different authors. This book is a celebration of that unprogrammatic thinking: of what we often lose to the abstraction that programming entails. It is a celebration of the complicated messiness of our world (and the agential possibilities that therein lie).

How then to address what might seem like a central design feature of the book, or at least a serious problem with the dataset? The writers I feature—J. M. Coetzee, Christine Brooke-Rose, William Gaddis, Kamau Brathwaite,

Communication', in *The Mathematical Theory of Communication*, eds. Claude E. Shannon and Warren Weaver (1949) (Urbana: University of Illinois Press, 1963), 94–117, 117.

[20] Liu, *Freudian Robot*, 34; Claude Shannon, 'Prediction and Entropy in Printed English', *The Bell System Technical Journal*, 30 (January 1951), 50–64, 50.

[21] Christopher Strachey, 'The "Thinking" Machine', *Encounter* (October 1954), 25+.

[22] Jessica Pressman, *Digital Modernism: Making It New in New Media* (New York: Oxford University Press, 2014), 4. There is an expansive scholarship on literary modernism's engagement with new media broadly, much of which Pressman cites. See also C. T. Funkhouser, *Prehistoric Digital Poetry: An Archaeology of Forms, 1959–1995* (Tuscaloosa: University of Alabama Press, 2007).

and Hugh Kenner—are all deeply imbued in an Anglophone modernist literature tradition. The first three, aside from their very explicit engagement with their modernist predecessors, grappled with the descriptor 'difficult' across their careers, a term often deployed by their readers, critics, and publishers, to signal them as inheritors of this tradition. Brathwaite, although rarely discussed as difficult, has nevertheless also been characterized as a key proponent of modernism in the Caribbean and would himself point to the importance of these writers to his literary endeavour.[23] Finally, Hugh Kenner would play a central role in the institutionalization of a particular strand of modernist literature through his championing of 'international modernism' and by authoring foundational critical works such as *The Pound Era* (1971).

Although this book does not advocate for a programmatic account of literature, the authors featured here all use literary modernism as a starting point for thinking about relations between literature and computing. There is pattern: in eschewing mimetic theories of representation (and thus foregrounding debates about the form and function of literature) and championing difficulty (and implicitly the role of the reader), literary modernism has been a conceptually useful resource for this set of writers, as they consider literature's affordances, programming, executability, and value. Literary modernism's promotion of the literary endeavour as 'difficult'—as entailing complex processes and procedures in its production and reception—offers an important backdrop for these writers, shortening the conceptual leap across to a domain that similarly conceives of its own endeavour as functional—and often functionally difficult at that. Literary modernism's difficulty makes it easy for these writers to think analogously.

This does not, however, produce a programmatic account of literature. The writers discussed in this book form a cohort in their willingness to conceive of computing itself as a difficult project, implicitly rejecting ideas that treat it as transparent, immaterial, easy, or simply a tool.[24] (Just using word processing software doesn't necessarily result in the kind of conceptual reflections I am keen to examine here.) Yet I bristle in the face of general

[23] 'What T.S. Eliot did for Caribbean poetry and Caribbean literature was to introduce the notion of the speaking voice', Edward Kamau Brathwaite, *History of the Voice: The Development of Nation Language in Anglophone Caribbean Poetry* (London: New Beacon Press, 1984), 30.

[24] On transparency as a design imperative, see Jay David Bolter and Richard Grusin, *Remediation: Understanding New Media* (Cambridge, MA: MIT Press, 1998); and Lori Emerson, *Reading Writing Interfaces: From the Digital to the Bookbound* (Minneapolis: University of Minnesota Press, 2014).

labels to describe these writers, for example as 'technomodernist'. Mark McGurl's term 'reasserts the obvious continuity of much postwar American fiction with the modernist project of systematic experimentation with narrative form, even as it registers a growing acknowledgment of the scandalous continuity of the literary *techne* (craft) with technology in the grosser sense'.[25] Partly I am uncomfortable with the tone here and partly the emphasis on formal experimentation, but mostly, it's that I think that, bar a willingness to engage with computer difficulty, it is the variety that marks these writers out. Kamau Brathwaite and Hugh Kenner would both, no doubt, be uncomfortable with the notion that computing is transparent, but for very different reasons (as we shall see, one might take issue on the grounds of the perceived 'difficulty' of literature, the other of computing). I want to hold on to such unprogrammatic bits—the dissonance between Brathwaite and Kenner's perspectives on relations, the particularity of their work, and how their ideas came to be.

Historical Remediations

To conceive of these writers' perspectives—and the relations they produce—we need to understand the material, specific circumstances in which these writers are operating. We need access to archives, to practices, to contextual knowledge that we have often jettisoned in favour of universals, abstractions, and high-level narratives.[26] The material matters: it is something that early programmers were extremely aware of as they wrestled with memory limitation, bit-length, and processing time in the design process, as we shall see. For all our abstractions and formalisms, the form of a programme, like the form of a poem, is built on its substrate.[27]

But such an archivally based project is difficult. I don't say this to frame my efforts as heroic (for one thing, I was pregnant for many of these archival trips, and nothing punctures such gutsy claims as having to sprint from the reading room to vomit in a bin). Rather I say this to acknowledge that

[25] Mark McGurl, *The Program Era*: *Postwar Fiction and the Rise of Creative Writing* (Cambridge, MA: Harvard University Press, 2009), 42.

[26] The implicit hierarchies in levels of abstraction in computing are usefully addressed in Matthew Kirschenbaum's *Mechanisms*.

[27] Whitney Trettien's article 'Substrate, Platform, Interface, Format' offers a valuable definitional and background account of the term(s), *Textual Cultures*, 16.1 (2023), 286–312. doi: 10.14434/tc.v16i1.36107.

difficulty takes forms other than aesthetic and conceptual, and those forms also matter.

In keeping close to the material and the specific, this project is shaped by the construction, limitations, and accessibility of archives of computer and literary history. Archives, conceived both as paper and as structured bits (i.e. datasets), are programmed and programmatic in the sense of being constructed, designed, and—definitely *not* transparent—interfaces for knowledge production.[28] The question of what constitutes an archive of programming is one that this book contemplates; I start from the premise that the material specificity I seek in reading rooms is also the product itself of infrastructures, practices, and media. Archival research is medially, institutionally, and resource bounded in ways that we often only acknowledge in the acknowledgements—the funders, employers, libraries, and communities that made this book possible. Yet it is worth remembering that our archival practices and systems were designed for paper-based knowledge—and for a specific subset of collections deemed valuable enough to construct, conserve, place, and access.[29]

When conducting archival research on computing, the difficulties are medially and temporally acute. The cycle of (planned) obsolescence for computer technologies and knowledge is rapid and can affect access for all but the most resource rich institutions and scholars—who are themselves often struggling not just with whether the disk drive will spin this time around but with difficult questions about what 'access' might mean remediated through the archive. It isn't just a question of file access or physical storage; the affordances of a particular computer technology shift at a much faster pace than writing or the codex has done historically.[30] We tend not to value old tech, and are less likely to retain it, to document the circumstances around its production and use, or to invest in the 'obsolescent' technical knowledge required to interact with it (COBOL, a 1960s programming language, is just not cool to learn today, even as it powers many of our global

[28] See, e.g. Andrew Prescott and Alison Wiggins, eds., *Archives: Power, Truth, and Fiction* (Oxford: Oxford University Press, 2023) whose extensive volume begins from the premise that 'Archives have never been more complex, expansive, or ubiquitous. Gargantuan in scale and conception yet never sufficient or complete, the archive is on the one hand a space for empowerment and expression and on the other an instrument of constraint and repression' (abstract).

[29] For a fascinating treatment of this topic, see Amy Hildreth Chen, *Placing Papers: The American Literary Archives Market* (Amherst: University of Massachusetts Press, 2020).

[30] See Wendy Hui Kyong Chun, *Updating to Remain the Same: Habitual New Media* (Cambridge, MA: MIT Press, 2016) for a reflection on the cultural import of this cycle.

financial systems). This tendency is exacerbated by a tech industry that is corporately minded and future orientated, in ways that often preclude valuing historical perspectives or making them accessible.[31] The residual assumption in some quarters that everything worth knowing is on the Internet so why bother with archives (problematic as they can be) is one that has hampered efforts in preserving our digital past.[32]

The case studies presented here then are emphatically the product of archival mediation and my methodological commitments. The writers included here all have substantive and relatively accessible (to comparatively resource-flush scholars at least) personal archives, with the significant exception of Brathwaite, who offers a counter-example as his personal archive and collections were purportedly destroyed in a 1988 hurricane (see Chapter 4). When making my selection of case studies, I was also constrained by archival factors.[33] Thomas Pynchon, having worked as a technical writer for Boeing Aircraft, might have made for an interesting case, but as I write these lines, the papers are not yet available to researchers (sold to the Huntington Library in late 2022).[34] Other archives that might tell the story of writers' engagement are private (I am looking at you Richard Powers), locked up in corporate archives (IBM never responded to my requests) or have been misplaced, thrown out, or allowed to degrade. It takes a certain person and situation to retain a life's old tech and documentation, let alone place it in an archive. What I am saying is that this book grapples with the materiality, protocols, technologies, and contexts of archives as much as with computing. It contemplates 'programming archives' in all the possible relations that such a formulation might suggest.

[31] Initiatives such as the establishment of the Computer History Museum, the Charles Babbage Institute, the Smithsonian's Computer Oral History Collection project, and the instigation of the *IEEE Annals of the History of Computing* (1979–) have been important in focusing (US) efforts in this area.

[32] Those long-standing efforts such as the Text Encoding Initiative and other large-scale projects that have sought to bring humanities and library science expertise to the creation of digital resources and tools should be a direct counter to these facile narratives.

[33] Ralph Ellison might offer an interesting addendum to this book, albeit a writer whose interest was mainly focused on the computer as a compositional tool. Interested readers are advised to consult Jeff Noh's fantastic article, 'Ralph Ellison's Computer Memory', *Contemporary Literature*, 62.4 (Winter 2021), 527–57.

[34] See 'News Release—The Huntington Acquires Thomas Pynchon Archive', Huntington.org (22 December 2022), https://huntington.org/news/news-release-huntington-acquires-thomas-pynchon-archive. The possibilities have been recognized although at this stage research often takes the form of deploying computational tools to examine Pynchon's work: e.g. Katie Muth, 'The Grammars of the System: Thomas Pynchon at Boeing', *Textual Practice*, 33.3 (2019), 473–93; and Erik Ketzan, *Thomas Pynchon and the Digital Humanities: Computational Approaches to Style* (London: Bloomsbury, 2021).

There is also another key institutional mediator in this book: the modern university. That is not because these writers always accessed computers through universities—though in the mainframe era when computers were room-sized, institutional affiliation was key to accessing the objects, practices, and people that constituted computing as it was formalizing as a discipline and sector. Rather, all of my case studies, in addition to being authors, were critics and teachers of literature in an era in which literary modernism's objects and critical methods were being institutionalized and the writer's technical expertise was codified and professionalized.[35] These writers were authoring texts that were recognizable to that institution—recognition that in turn helped to guarantee their own critical reception (and eventual institutional acquisition of their personal papers). As significant, such an institutional environment had another effect: these writers' conception of literature's relation with computing was as likely to turn on disciplinary and pedagogical questions—themselves shaped by the literary–critical tenants of close reading and aesthetic autonomy that literary modernism championed—as it was formal ones. Today scholars might argue fiercely over whether it is possible to close read a programme, as Knuth implies it might be, but that argument is itself shaped by the institutionalized legacy of modernist conceptions of the nature and methods of literary enquiry.[36] In foregrounding this, I want to acknowledge the function of the university as an institution, while also encouraging us to think outside its (albeit often productive) strictures in our understanding of how literature and computing might relate.

The authors I discuss are all positioned firmly within the legacies of literary modernism as teachers and writers, but geographically, their locations varied. Hugh Kenner was a Canadian who institutionally hopped around

[35] This context is outlined by Gerald Graff in his classic *Professing Literature: An Institutional History, Twentieth Anniversary Edition* (Chicago: University of Chicago Press, 2007). For a very different perspective, see Evan Kindley, *Poet–Critics and the Administration of Culture* (Cambridge, MA: Harvard University Press, 2017).

[36] e.g., David M. Berry, Dennis Tenen, Martin Paul Eve, and others have all championed 'close reading' as a method appropriate for the study of computational 'texts'—and, implicitly or explicitly, as evidence of the wider valence of the humanities—but such a critical approach implies a parity between language and code that, as John Cayley has pointed out, is highly disputable. David M. Berry, 'Tracing "Toxicity" Through Code: Towards a Method of Explainability and Interpretability in Software', *DHQ* 17.2 (2023), http://www.digitalhumanities.org/dhq/vol/17/2/000706/000706.html; Tenen, *Plain Text*; Martin Paul Eve, *Close Reading with Computers: Textual Scholarship, Computational Formalism, and David Mitchell's Cloud Atlas* (Stanford: Stanford University Press, 2019); John Cayley, 'The Code Is Not the Text (Unless It Is the Text)', *Electronic Book Review* (10 September 2002), electronicbookreview.com/essay/the-code-is-not-the-text-unless-it-is-the-text.

the USA while often focusing his attention on literary culture across the Atlantic. Christine Brooke-Rose's upbringing and influences spanned the English Channel. J. M. Coetzee's career took him to four continents, from Africa to Europe, North American, and Australasia. Kamau Brathwaite would be similarly mobile—the Caribbean archipelago, the Gold Coast, the UK, and the USA. Only William Gaddis would be based primarily in the USA (although he would travel to Europe and North Africa early in his career and later to East Asia and Australasia as a cultural ambassador). There is something suggestive in this: a propensity to cross geographical boundaries matched by disciplinary peripeteia.

Taken together, this geographical range is itself productive, allowing me to demonstrate the importance of national computer contexts ('combined and uneven development' occurs in computing as much as literature[37]), while also illuminating the significant reach of computation and its attendant import for literary endeavour. Despite this geographical range, this book does not aspire to be global in its coverage. It is significantly restricted by its focus on literature in English for, while (in a modern act of linguistic imperialism) English is the de facto global language for computing, there are many stories to be told about literature and computing in other linguistic contexts.[38] It is also worth flagging that this Anglophone focus reflects archival holdings and scholarship on the history of computing that has been dominated by accounts of a few American firms (IBM in particular), although this is, pleasingly, changing.

The temporal scope of this book stretches from the Second World War through to the turn of the millennium: from the first programmable electronic digital computer to the World Wide Web. The authors under discussion were born between 1922 and 1940, young enough to be middle-aged when mainframe computers went mainstream in the 1960s, and young enough at heart to enthuse over the PC (personal computing) revolution of the 1980s. Despite the half-century span, this isn't a consummate history of computing in that many events, machines, and persons often considered significant to the emergence and shaping of the computing sector do not appear here. Neither John von Neumann nor Steve Jobs features. Nor does the Burroughs Corporation, an early giant in the industry, which might

[37] WReC (Warwick Research Collective), *Combined and Uneven Development: Towards a New Theory of World-Literature* (Liverpool: Liverpool University Press, 2015).

[38] See Mark C. Marino's chapter on this in *Critical Code Studies*, 129–60. Rita Raley also wrote eloquently on this topic back in 2003: 'Machine Translation and Global English', *The Yale Journal of Criticism*, 16.2 (2003), 291–313.

seem surprising given the connection to the Beat writer William Burroughs (his grandfather founded the firm), but the link is more enticing than actual given that the family had divested in the 1920s.[39] Neither is this a fulsome history of literature over the last fifty or so years. Drama doesn't get a look in for one (unfairly ignoring Brenda Laurel's seminal text *Computers as Theatre* [1991], which revolutionized the field of human–computer interaction with its central metaphor).[40] Significant names in the history of literature don't appear. Such omissions reflect my interest in tracing the specific material and intellectual journeys of the writers featured here, not because their individual trajectories are necessarily representative of a wider sector, but because their stories are suggestive in their very singularity. They help us to move beyond programmatic accounts of computing history and literature's relations to it.

My date range is also influenced by my topic. I only touch upon the World Wide Web and don't discuss social media and 'Web 2.0' at all. Clearly, there is potential to bring this account up to date (the writer Naomi Alderman springs to mind here—author of prize-winning novels such as *The Power* [2016] and the popular running app *Zombies, Run!* [2012]). My reasoning might seem a little contradictory: I seek to highlight the degree to which contemporary cross-fertilization between literature and computation has a long history that is not necessarily oriented around formal correspondence or innovation. The novelties of electronic and hypertextual literature in the 1990s and the reimaginings of reading and writing enabled by our electronic devices and web platforms today are thrilling, but their medial disruptiveness can distract us from the quieter innovations[41]—those published in the traditional codex, those of a methodological or pedagogical orientation, and those so accepted as to seem banal.[42] I want to see that longer history for what it is and for how it has shaped our understanding of how literature and computing might relate.

[39] Albeit that family wealth provided Burroughs with the funds to write and travel—computer-funded literature in a manner somewhat akin to that discussed in Chapter 5.

[40] Brenda Laurel, *Computers as Theatre* (Menlo Park, CA: Addison-Wesley Publishing, 1991).

[41] I highly recommend the work of the Electronic Literature Organization (www.eliterature.org) and scholars associated with the field.

[42] A term with its own historical lineage, see Zara Dinnen, *The Digital Banal: New Media and American Literature and Culture* (New York: Columbia University Press, 2018).

Subroutines

The book is structured in five chapters—subroutines in a programme we might say. Chapter 1, *J. M. Coetzee's Aesthetic Automatism*, focuses on South African author and programmer J. M. Coetzee. While Coetzee's authorial career and his novels are well-known, his precise experiences as a programmer for IBM and working on Britain's foremost supercomputer project in the early 1960s are less so. This chapter draws on extensive archival material to recount this unfamiliar story across devices, technologies, and activities. In so doing, it tracks his unusually diverse programming career, which included mundane office work with punched cards, as well as operation system design on a machine for the British military—a system he also used, out of hours, to generate poetry.

The early and diverse nature of Coetzee's experiences in the industry means that he offers a useful introduction to many of the machines, concepts, and possible relations between computing and literature that other authors would also explore. In contrast to narratives that see this professional experience as the limits of his engagement with computing, the chapter outlines Coetzee's academic work in humanities computing, the forerunner to today's digital humanities, in both the USA and South Africa in the 1970s, before focusing on the implications of the broader context of the South African apartheid regime's reliance on computation for his understanding of the value of literature. In doing so, it demonstrates how central Coetzee's practical and theoretical programming experience was to the emergence of his own literary praxis. The chapter contends that we should understand Coetzee as having developed a platform of 'aesthetic automatism'—an update of modernist aesthetic autonomy that conceives of literature as being not only autonomous from computing but also a vital stalwart against the 'binary thinking' it promotes.

Chapter 1 presents a case study in which literature and computing represent opposite ways of viewing the world—Coetzee's project exemplifies a strand of thinking that lies behind much contemporary debate about the critical value of the arts and humanities in a digital world. Chapter 2, *Hugh Kenner's Hobby*, by contrast upends that disanalogy. Focusing on the career of the renowned critic of literary modernism and dedicated computer hobbyist, the chapter demonstrates that for Hugh Kenner computing is not heavily associated with the militarized nation state but, thanks to his involvement in the amateur communities that would facilitate the 'PC Revolution', conceived as a democratizing force and model for literary–critical praxis.

Bringing together hitherto scattered and unnoticed details of Kenner's activities across three decades, Chapter 2 tells the story of his hobbyist efforts as amateur programmer, writer of technical manuals, and reviewer of books for the popular computing magazine *Byte*. In so doing, the chapter argues that the hobbyist approach he utilized there proved a spur to his own pedagogical and critical practice—revealing a surprisingly anti-elitist strain in the father of modernist criticism. In Kenner's conception, the analogy is between computing and literary studies, rather than literature, and one that turns less on aesthetics than expertise, professionalization, and pedagogical theory.

Kenner would no doubt be disappointed by the afterlife of his project: while computing's libertarian potential continues to be trumpeted by Silicon Valley, literary studies has become, in many estimations, further entrenched in its ivory tower. In Kenner's story, we encounter a relation between literary criticism and computing that did not gain traction—disciplinarily literary enquiry is often defined by its contrast with tech culture, a problematic relation when literary criticism makes claims for the valence of its skills and modes of reading outside of the academy. The relation Kenner endorsed is one that advocates of literary studies might want to revisit today.

If Chapter 2 reclaimed a forgotten relation, Chapter 3, *Christine Brooke-Rose's Towers of Babel*, showcases a conceptual analogy that is so commonplace as to be almost unrecognizable: that programming is a linguistic activity. Computer code is written, read, and interpreted; we write in programming languages. This understanding of programming underpins many of the methodological justifications for literary–critical approaches to digital technology: these are texts, peculiarly executable and machine-readable texts, certainly, but amenable to the methods of literary enquiry, in particular close reading.[43] The chapter focuses on experimental writer Christine Brooke-Rose, as a means of defamiliarizing and interrogating this analogy.

Christine Brooke-Rose worked at Bletchley Park under Alan Turing during the Second World War; a novelist was present for the birth of electronic computing. Forty years later, she would write a series of 'difficult' novels, *Amalgamemnon* (1984), *Xorandor* (1986), and *Verbivore* (1990), and a memoir, *Remake* (1996), that experimented with programming languages. For many literary critics, these novels respond to her early awareness that computing shares with literature a linguistic commonality. But in 1945, this was not the

[43] This helps to explain the historical dominance of literary–critical approaches in the digital humanities broadly, and in endeavours such as *Critical Code Studies*. The assumption also lies behind Donald Knuth's 'Art of Computer Programming' project.

case; back then, computers were calculating machines and programming hadn't yet been invented. This chapter draws on Brooke-Rose's personal archives, and those of her publisher, to make a case that it was her career with words—as a reviewer for British periodicals, as a translator, as a literature professor in Paris after the May 1968 student riots, and as a novelist experimenting within a modernist tradition—that would shape her understanding of programming languages. Indeed, her novels are in many ways an attempt to refute the underlying conception of language (itself, as we shall see, influenced by a modernist conception of difficulty) that computing developed after 1945.

Chapter 3, then, reveals the historicity of our contemporary assumption that literature and computing are comparable in their shared use of language. In so doing, it illuminates the degree to which many contemporary arguments over literary studies' value (or lack thereof) are in fact turf wars—disagreements over whose methods, assumptions, and perspectives should be adopted to study this shared object. It also reminds us that the analogy can be dropped when its usefulness wains, leaving us with the question: what then for literary studies and computing?

One potential response is offered in Chapter 4, *Kamau Brathwaite: Creolized Computing and Processual Aesthetics*. It takes as its focus Caribbean poet and scholar Kamau Brathwaite's exuberant experiments with PC desktop publishing—his 'SycoraX aesthetics'—in the 1980s. Brathwaite came to the machine not as an early adopter or programmer, but as the purchaser of American-made mass market items and inexpert user of word processing software. This chapter draws upon archival discoveries and an expanding body of scholarship on the Digital Caribbean, attending to the specifics of the computing culture and intellectual tradition in which Brathwaite was enmeshed. It argues that Brathwaite conceived of his word processing activities as one means of decolonizing the modernist literary project. Crucially, for Brathwaite, the physical book, the text on the printed page, becomes representative of imperial culture with digital inscription offering a resistive alternative.

In Chapter 4, the distinction Brathwaite draws between literature and creolized computing is oriented around format and the kinds of processual interaction that reading and inscription can enable or foreclose. The interest in interaction provoked by PC innovations was not restricted to Brathwaite; it has defamiliarized the codex for book historians and renewed attention on the affordances of literature's most familiar format. Simultaneously, such untethering literature from its traditional format has stimulated scholarly interest in its processual aspects—what literature, and literary enquiry, does in the world. Inbuilt in Brathwaite's perspective, however, is an insistence on the political and cultural repercussions of (literary) interaction. As this

chapter indicates, in a contemporary environment in which literary enquiry is sometimes conceived as a digital bulwark—a means of facilitating social justice, community, or empathy—Brathwaite offers a significant alternative voice and a valuable point of origin as we assess debates around not only the object but also the methods, authority, and scope of literary studies today.

Such focus is reconfigured in Chapter 5 as a discussion around the value of different kinds of writing—what is worthy of the human reader's attention, as Knuth had it. Chapter 5, *William Gaddis, Technical Writer*, discusses the American author. Drawing on Gaddis's abundant archives, the chapter tells the story of his work as an author of technical literature—documentation, user manuals, and sales material—for the computing colossus IBM (among other companies) when he wasn't writing novels. I make the case that the technical writing that Gaddis was producing, itself a mediator of computer technology in society, would play a vital role in the development of Gaddis's own literary project. In particular, his notoriously difficult style was developed out of a desire to rebut technical writing's commitment to transparency. For Gaddis, and for some of his peers, computing's (abundant) mediations in the second half of the twentieth century operate, crucially, as literature's foil. It is an argument that has gained significant traction, if little critique.

A short conclusion, *Documenting Difficulty*, turns from Gaddis to reflect briefly on a topic that held Knuth's interest, namely documentation. It contemplates the critical import of the mass of material that can surround computing. I point out that the turn away from the idea of the universal machine should not lead us to valorize documentation in and of itself, especially in a moment in which concerns around artificial intelligence's own 'difficulty'—its opacity and inscrutability for human readers—have positioned documentation as an ethical remedy. Literary enquiry may well offer tools and expertise for engaging with such difficulty, but that doesn't mean that we should accept such an argument—or justification for the discipline—at face value, nor the account of literary value that it implies. As this book hopefully makes clear, multiple and shifting relations have existed between literature and programming, and that history has simultaneously created and constrained current conceptions of both literary enquiry and computing. Donald Knuth might have called for a new kind of programming, but in examining and reflecting on the history archived in these pages, and by dwelling in the (digitally) material, we might also discover new relations for programming literature to code going forward.

Programming Literature. Rebecca Roach, Oxford University Press. © Rebecca Roach 2025.
DOI: 10.1093/9780198976578.003.0001

1

J. M. Coetzee's Aesthetic Automatism

J. M. Coetzee's international reputation rests on his prize-winning fictions, autobiographical writings, and critical works—an oeuvre that won him a Nobel. His reputation could just of easily have been based on his pioneering work in computing, employed as he was as a programmer on one of Britain's most advanced supercomputing projects in the 1960s. It isn't as if he has concealed his contributions: the broad details are hiding in plain sight, recounted in his fictionalized autobiography *Youth* (2002). But it isn't a story that we tend to repeat. Why is this second story not so well-known? Why, when we talk of Coetzee, is this history often relegated to a curious side note—as the mundane day job and rejected career of the aspiring novelist? Why can't literature and programming dwell happily in the house of two cultures?

In this chapter, I want to suggest that they can. I recount Coetzee's extensive—decades long—engagement with computing and argue that his years in the industry had a foundational influence on his writing. In so doing, I introduce many of the concepts and equipment that we will return to throughout this book. I further suggest that the common contemporary notion that literature and computing make for awkward housemates, an idea heavily promoted and politicized by Coetzee, is not backed up by the evidence, at least not in his archive.

Critics have often noted Coetzee's background in mathematics. Taking their cue from Coetzee himself (who links mathematics and poetry as 'two rarefied forms of symbolic activity'[1]), they frame the relationship as one of conceptual influence. However, as Coetzee's biographer J. C. Kannemeyer mused, 'one wonders…how congenial the abstract ratiocination of mathematics can be for the verbal artist who draws on the concrete world for his raw material and means of representation, and who has to think with his

[1] J. M. Coetzee, 'Review of *Strange Attractors: Poems of Love and Mathematics*, ed. Sarah Glaz and Joanne Growney', *Notices of the American Mathematical Society*, 56.8 (2009), 944–6, 944.

senses'.[2] Taking a more historical and material perspective, I deliberately focus on the crucial early years of Coetzee's engagement with applied mathematics and the material affordances of computing.[3] I deploy this perspective to argue that this led Coetzee to develop a literary platform that I call 'aesthetic automatism'. My term draws on what Coetzee describes, with deep ambivalence, as the Prague school's notion of automatization, or 'the process by which repeatedly used speech forms wear a neural rut'. Developing out of—and yet often in opposition to—the idea that 'automatized speech is speech that speaks its speaker', Coetzee's platform of aesthetic automatism, as I conceive it, involves a conception of literary form and language that recasts modernism's aesthetic autonomy within a mid-century context in which computing formalizations automated many processes, including those of reading and writing.[4]

The development of Coetzee's platform of aesthetic automatism forms the basis of *Youth*, the fictional narrative of an 'ignorant provincial' and aspirant poet who struggles miserably to find his literary voice among a cacophony of inherited aestheticist and modernist strictures.[5] The work seeks to resolve such demands for modernist impersonality, verbal 'compression', and 'fine discriminations' within the context of automation; as the narrator ponders, 'might it not be argued that the invention of computers has changed the nature of art, by making the author and the condition of the author's heart irrelevant?'[6] Although written almost forty years after his time working in the computing industry, *Youth* encourages readers to consider the idea that Coetzee's 'famed impersonality' is in actuality 'an *achievement*' that has its roots as much in his work with computer programming as it does in his engagement with high modernism.[7]

[2] J. C. Kannemeyer, *J. M. Coetzee: A Life in Writing*, trans. Michiel Heyns (UK: Scribe, 2012), 88.

[3] Coetzee's computing experience is notable for occurring at a moment before the distinctions on which platform, software, and critical code studies rely today were clearly articulated.

[4] J. M. Coetzee, 'The First Sentence of Yvonne Burgess' *The Strike*' [1976]. Reprinted in J. M. Coetzee, *Doubling the Point: Essays and Interviews*, ed. David Attwell (Cambridge, MA: Harvard University Press, 1992), 91–3, 93. In arguing for Coetzee's platform as one that sought to recast modernist autonomy, I don't wish to downplay the complexity of his own relationship to literary modernism (in its various incarnations), a topic that has itself proven particularly fertile for scholars including: Jarad Zimbler, *J. M. Coetzee and the Politics of Style* (Cambridge: Cambridge University Press, 2014); David James, *Modernist Futures: Innovation and Inheritance in the Contemporary Novel* (Cambridge: Cambridge University Press, 2012); and Derek Attridge, *J. M. Coetzee and the Ethics of Reading: Literature in the Event* (Chicago: University of Chicago Press, 2004).

[5] J. M. Coetzee, *Youth* [2002] (London: Vintage, 2003), 51.

[6] Coetzee, *Youth*, 61, 135, 161.

[7] David Attwell, *J. M. Coetzee and the Life of Writing: Face to Face with Time* (Oxford: Oxford University Press, 2015), 32, 33.

Certainly, modernism's own engagement with new media has been widely discussed in scholarship. Jessica Pressman in particular argues persuasively that close reading was itself the product of modernism's 'mediatized moment' and that Marshall McLuhan's media theory, which strongly influenced Coetzee, itself adapted 'the New Critical method of focusing on form' in order to think 'critically about forms of media'.[8] Moreover, as I mentioned in the introduction, mid-century information theory, pioneered by figures such as Claude Shannon, Norbert Wiener, and Warren Weaver, had also led to an imbrication of modernist and computational ideas (whether via computational linguistic projects or 'cybernetics'—the theory of control and communication in systems).[9] In Coetzee's early computational experiments, he engages enthusiastically with such formalist projects, in turn exemplifying what Hugh Kenner terms 'art in a closed field', or the relentless drive towards abstraction via permutation exhibited in writers such as Samuel Beckett and James Joyce (for more on which, see Chapter 2). Indeed, in 1977, Coetzee published an essay in *PMLA* on the 'poetics of failure' he sees in works such as Beckett's *The Unnamable* and Vladimir Nabokov's *Pale Fire*. 'The poetics of these works', Coetzee explains, constitutes 'a program for constructing artifacts out of an endlessly regressive, etiolated self-consciousness lost in the labyrinth of language and endlessly failing to erect itself into autonomy'.[10]

Yet Coetzee's work also clearly articulates the limits of such projects, limits born not of modernism's 'exhaustion' but rather of the automation of language that computational media—via a cybernetic theory of language that is dematerialized and probabilistic—frequently embody and promulgate.[11] As I show in the first half of this chapter, Coetzee might characterize his art as 'spare prose and a spare, thrifty world', but it is highly constructed, built

[8] Jessica Pressman, *Digital Modernism: Making It New in New Media* (New York: Oxford University Press, 2014), 11, 25.

[9] See Heather A. Love, *Cybernetic Aesthetics: Modernist Networks of Information and Data* (Cambridge: Cambridge University Press, 2024). I outline the contributions of modernist critic I. A. Richards in developing Communication Theory (later renamed information theory) in the 1930s and 1940s in Rebecca Roach, 'Cross-Cultural Communication Theory: Basic English and Machine Translation at the Rockefeller Foundation', Rockefeller Archive Center Research Report (19 December 2019), https://rockarch.issuelab.org/resource/cross-cultural-communication-theory-basic-english-and-machine-translation-at-the-rockefeller-foundation.html and a forthcoming project on 'Machine Talk'.

[10] J. M. Coetzee, 'Achterberg's "Ballade van de gasfitter": The Mystery of I and You', *PMLA*, 92.2 (1977), 285–96, 293.

[11] John Barth quoted in Brian McHale, *Constructing Postmodernism* (London: Routledge, 1992), 28.

in part via a deeply material understanding of language born of his own engagement with computing.[12]

In making this case, I highlight another problem. While *Youth* confronts, head-on, the relationship between political power and computing in a global Cold War context, it is contrastingly silent on the role of computers in supporting the South African apartheid regime. Such an imbalance is troubling, bringing accusations of political quietude. Its significance only grows as we trace the extent of Coetzee's own engagement with computing across his career. The second half of this chapter turns then to address this imbalance explicitly and considers the national context out of which Coetzee's aesthetic automatism develops. In so doing, it underlines the degree to which the analogies and disanalogies that are drawn between literature and computing always do political work—even as they make claims to autonomy.

FORTRAN Thinking: Coetzee the Programmer

Coetzee's employment in the computer industry occurred over a four-year period in the early 1960s, when he was based in London. Before immigrating to the United Kingdom, he graduated from the University of Cape Town (UCT) with honours in both English and Mathematics. His undergraduate experiences guided his later interests. On the literary side, he studied stylistics, philology, literature across periods, and more contemporary modernist writers and their criticism; he also took noncredit classes in imaginative writing with the poet Guy Howarth.[13] For his mathematics degree, he studied under Douglas Sears and Stanley Skewes, the latter renowned for discovering the Skewes number and an associate of Alan Turing before the Second World War. Coetzee's courses in pure mathematics, applied mathematics, and mathematical statistics also gave him a firm grounding in the ideas that formed the basis of computer work.[14] However, it was in the London office of American computing giant IBM that he first obtained a position as a programmer.

This was 1962. The computer industry was in transition. Digital computing use was expanding from its origins in wartime military research and early use in mathematical applications, to a much wider deployment in

[12] Coetzee, *Doubling the Point*, 20.
[13] Kannemeyer, *J. M. Coetzee*, 89–96.
[14] Kannemeyer, *J. M. Coetzee*, 87–9.

business, academic research, and beyond. The social, economic, and cultural implications of computation were being hotly contested in the public sphere. At the same time, the pace of technological innovation had been rapid: it was only eleven years since the delivery of the UNIVAC 1, the first commercial machine; in hardware, the move had only just been made from vacuum tubes to the more reliable transistors. As the market for computation grew, so did the realization that programming was a pinch point in the industry's expansion. The new markets required expensive specialized software, but there was a dearth of good programmers to write it. Part of the difficulty, as computer historian Nathan Ensmenger notes, is that defining and then identifying a good programmer proved more difficult than many companies anticipated. Far from being glorified clerical work, good programming was considered a 'black art'.[15] In their search for programmers, companies like IBM hired people from a diverse range of backgrounds, including poets, chess players, and mathematicians. As a chess player, mathematics graduate, and aspiring poet, Coetzee was an attractive candidate.

When Coetzee joined IBM, the company—having expanded from being a purveyor of tabulators in the 1950s—was the global leader in the booming electronic computer industry. In 1960, IBM launched what would become the classic mainframe computer: the 7090. Despite being as big as a room and priced at almost $3 million, the computer was a commercial success thanks to its speed, its architecture, and the range of tasks it could perform. The 7094, for example, had a huge (for the time) 36-bit word length, which made it suitable for scientific calculations and a large core memory. It was often run in conjunction with the smaller 1401 model, with which Coetzee spent most of his time working. Announced in 1959 and designed for the business world, the 1401 demonstrated to the industry that there was a sizeable commercial market for computers. Within IBM, for reasons of efficiency (calculation 'run time' on the 7090 was expensive), programmes were often developed on the 1401 using punched cards and then, once perfected, transferred onto magnetic tape and run on the 7090.

Working on these machines, Coetzee likely conceived of computing as a multimedia activity with very clear constraints. Far from the immediacy of today's user experience, Coetzee's computer interfacing at IBM involved interacting with a host of different objects: switches and drum storage units,

[15] Nathan Ensmenger, *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise* (Cambridge, MA: The MIT Press, 2010), 40. Ensmenger takes the term 'black art' from computer luminary John Backus.

console keyboard, punched cards, magnetic tape, and continuous form printer paper. Memory was limited, computation entailed the correct sequential inputting of notched cards for batch processing, and programming was based around the linear and temporally discrete structure of such processing. In particular, only one programme could be run at a time, and it couldn't be altered half way through; when the programme finished running, the human operator had to reset the machine for the next job. Though extremely advanced technology for the time, the computer's physical limitations were obvious to those working with them: despite the multiplication of physical objects, this was an environment of scarcity. This culture resonates with Coetzee's own conception of the writing craft. His fiction regularly conceives of its medium in terms both minimalist and material. That is, while his characters desire 'a life unmediated by words',[16] Coetzee's austere prose renders these words objects 'to burnish and fondle' in *Dusklands* (1974), or stones to be itemized and manipulated into messages in *In the Heart of the Country* (1976).[17]

Meanwhile, Coetzee's experience writing programmes also likely influenced his understanding of literary language and form. Programming was essentially a two-stage process: first, the design of the programme entailed the establishment of an objective—a calculation—and the logical process needed to arrive at that objective. All logical assumptions needed to be made explicit (unlike humans, computers 'will embody no inexplicit preconceptions'[18]) and organized in an efficiently written series of logical propositions. This stage could be inscribed graphically in a series of flow charts, which Coetzee later acknowledges in his doctoral dissertation, 'ha[d] the power to condense information which language seldom equals'.[19] Alternatively, these stages could be written in pseudocode, an informal and often idiosyncratic notation system that uses the structural conventions of programming languages but is not executable and is designed to be read by humans rather than computers. Only after this step was completed could the second stage occur: coding the programme—the embodiment of the intention—into a machine-readable language for the computer to execute.

When Coetzee worked at IBM, programmes were usually written in FORTRAN. The first high-level language, FORmula TRANslation was

[16] J. M. Coetzee, *In the Heart of the Country* [1976] (London: Vintage, 2004), 147.
[17] J. M. Coetzee, *Dusklands* [1974] (London: Vintage, 1998), 43.
[18] Roger Needham, Letter to the Editor, *Times Literary Supplement* (8 June 1962), 429.
[19] J. M. Coetzee, 'The English Fiction of Samuel Beckett: An Essay in Stylistic Analysis', PhD dissertation, University of Texas at Austin, Austin, 1969, 148–9.

developed in 1957. Computer instructions were previously written in so-called machine code, which consisted of strings of binary or hexadecimal numbers. These numbers are executable by the computer but difficult for the human coder to write and read (the opposite of Knuth's vision). Machine code is also specific to the individual computer. With the advent of compilers, which automated code conversion, programmers could write the code in a more human-friendly language that was based on mathematical formulae. FORTRAN, like most formal languages, deviated from natural language in notable ways: it was designed to avoid ambiguity and redundancy, but it also promoted literalness. FORTRAN also emphasized mathematical and logical propositions. In using FORTRAN, Coetzee worked at one level of abstraction from the computer's machine code, in a language that utilized 'arithmetical statements', 'efficiency statements', and 'subprograms' (i.e. small nested programmes).[20] While they could be intellectually stimulating, both stages of programming involved rule-bound composition that relied heavily on the either/or logic of Boolean algebra and drew on a limited range of statements to execute functions. Such experience likely prompted Coetzee to consider the operations of natural language and the affordances of literary language.

Overall, Coetzee seems to have found working at IBM to be intellectually restrictive, a sentiment that directly contradicts much of the cultural hype generated by IBM's company mantra, 'THINK'. As *Time* triumphantly declared in 1955, IBM's mantra indicates its ambition 'to mechanize hundreds of processes.… Thus liberated from grinding routine, man can put his own brain to work on problems requiring a function beyond the capabilities of the machine: creative thought'.[21] As an employee rather than a customer, Coetzee was not 'liberated from' but rather functioned in service to the 'grinding routine' of the computer. In a letter, Coetzee describes the experience as 'sterile'.[22] In *Youth* too, the narrator damns the entire enterprise: 'The more he has to do with computing, the more it seems to him like chess: a tight little world defined by made-up rules.'[23]

[20] IBM, *Reference Manual 709/7090 FORTRAN Programming System* (New York: IBM, 1961), 5.

[21] Quoted in Kevin Maney, Steve Hamm, and Jeffrey M. O'Brien, *Making the World Work Better: The Ideas That Shaped a Century and a Company* (New York: IBM Press, 2011), 16.

[22] J. M. Coetzee to Stanley Skewes (6 December 1964), Box 104, Folder 1, J. M. Coetzee Papers, Harry Ransom Center, University of Texas at Austin (hereafter JMCP).

[23] Coetzee, *Youth*, 149.

It is perhaps no surprise, then, that in 1963 Coetzee resigned from IBM. Though returning temporarily to South Africa, he remained intent on pursuing various employment opportunities back in the United Kingdom. While exploring the option of teaching, he also wrote to IBM's competitors, including International Computers and Tabulators (ICT), suggesting that Coetzee disliked IBM rather than the industry as a whole. In January 1964, he was offered and accepted the job of programmer on ICT's Atlas project.

Babeling in Closed Worlds: The Atlas Project

Atlas was Britain's first supercomputer. A joint development between the University of Manchester and Ferranti—an electronics company with numerous defence contracts—the Atlas computer was designed between 1956 and 1962 under the leadership of Tom Kilburn. The computer was fast and introduced various innovations, including virtual memory and a multitasking operating system called the 'Supervisor'. Although technically impressive (it was arguably the fastest in the world at its commissioning in 1962), only three Atlas computers were produced, mainly due to their prohibitively high cost. Ferranti was interested in producing a simpler and more commercially viable model; in 1962, the company supplied the University of Cambridge's Mathematics Laboratory with Atlas hardware at discounted rates in return for research support. Called 'Titan', the commercial machine created by the Mathematics Laboratory was the prototype for the Atlas 2 supercomputer.

It was on the Titan/Atlas 2 project that Coetzee worked as a programmer (ICT absorbed Ferranti's mainframe computer interests in 1963). A far cry from the routine business processing that occupied him at IBM, the work that Coetzee conducted as part of the Titan/Atlas 2 project was at the cutting edge of computing. He worked closely with a team comprised of computer scientists and mathematicians from both ICT and Cambridge; the team's members were a veritable who's who of Britain's post-war (male) computing luminaries, including Roger Needham, David Wheeler, David Hartley, David Barron, Peter Swinnerton-Dyer, and Barry Landy.

Coetzee was part of a team working on the Titan's Supervisor. Unlike previous computers, the Atlas had deployed multiprogramming (several programmes could seemingly be run concurrently). In order to handle this innovation, the first computer operating system was developed: the so-called Atlas Supervisor. If previously a computer was 'effectively [an] empty

machine',[24] the Supervisor offered 'a radical solution to this situation, integrating the treatment of…input/output handling'. The programming entailed in developing this operating system was far from minor: thirty-five thousand machine instructions needed to be written.[25] When it came to the development of the Titan/Atlas 2, an entirely new operating system was needed. The Atlas Supervisor had exploited an innovation in the computer design: the first use of virtual memory (it automated some aspects of information storage and access). However, to save on costs, this feature was removed in its successor with the result that the Titan/Atlas 2 Supervisor could not be merely updated but had to be designed (again) from scratch.

Coetzee's work focused on conceptualizing the design and writing the code that would instruct the computer about which jobs to implement and in what order. In *Youth*, he satirizes such work by describing the machine as exhibiting 'self-consciousness of a kind': 'At regular intervals—every ten seconds, or even every second—it interrogates itself, asking itself what tasks it is performing and whether it is performing them with optimal efficiency.' With each 'swing of the magnetic tape', the operating system questions itself, moving to the strict rhythm, we might imagine, of the arms race and automation.[26] Despite the (inhuman) efficiency goals inherent in job control, the actual experience of programming on the Titan/Atlas 2 Project was far more provisional, messy, and human.

One of the most notable features of working on this project was its Babel-like multilingualism and coding inventiveness. Programming languages were proliferating during this moment thanks to the development of compilers, but these languages were not yet defined by formal standards. The collaborative but federated nature of the Atlas project meant that it was particularly notable for its programming language multiplicity and creativity. The original Atlas machines deployed Mercury Autocode, the Manchester University machine used Atlas Autocode, the University of London Atlas used a compiler–compiler, and the Harwell Atlas used a FORTRAN dialect and compiler. Meanwhile, the Titan used its own autocode (developed by David Hartley) while the new language it was to run on, 'CPL', was in

[24] Ian Pyle, 'Memories of Atlas Fortran', *Memories of the Ferranti Atlas Computer* (School of Computer Science, University of Manchester, December 2012), http://elearn.cs.man.ac.uk/~atlas/, 7.

[25] Simon Lavington, 'The Atlas Story', *Memories of the Ferranti Atlas Computer* (School of Computer Science, University of Manchester, December 2012), http://elearn.cs.man.ac.uk/~atlas/, 21.

[26] Coetzee, *Youth*, 143.

development. Coetzee, thanks to his prior work with FORTRAN, had expertise in programming with the most advanced formal language and on the Titan/Atlas 2 project would work with machine code, with languages under development, and with programmes written for the other Atlas machines in various other dialects and languages.

Working across these languages, Coetzee was likely well aware of the degree to which the specific style and structure of these early formal languages affected not only the code that could be written but also the computational functions that could be performed. As Donald Knuth and a colleague would later point out: 'After learning a high-level language, a person often tends to think mostly of improvements he or she would like to see (since all languages can be improved), and it is very easy to underestimate the difficulty of creating that language in the first place.'[27] Coetzee was uniquely positioned to experience the process of language formalization. Moreover, in moving between abstracted languages and machine code, Coetzee was likely confronted, on a daily basis, with questions about the material implications of working within formalizations and at varying degrees of abstraction. These practical questions led directly to his own doctoral work in stylostatistics and to a lifelong interest in the material origins and import of linguistic structures.

One final feature of Coetzee's employment at ICT is vital to note. In response to innovative work being done in the other Cambridge—at Massachusetts Institute of Technology—the Titan team began developing one of the earliest time-sharing systems for the Supervisor. Time-sharing built on multiprogramming by enabling multiple users to interact concurrently (so it seemed) with the same computer, a big improvement on batch processing. The work on time-sharing, however, brought with it a shift in Coetzee's responsibilities and likely an increased awareness of the potential politics of computing.

In addition to Titan, two Atlas 2 computers were being built, one of which was destined for the Atomic Weapon Research Establishment (AWRE) at Aldermaston. For security reasons, AWRE did not want to have more than one job stored in the computer's memory at any one time; time-sharing was 'a complete No No.'[28] The Aldermaston Supervisor would thus need to

[27] Donald Knuth and Luis Trabb Pardo, 'The Early Development of Programming Languages', *A History of Computing in the Twentieth Century*, eds. N. Metropolis, J. Howlett, and Gian-Carlo Rota (Academic Press, 1980), 197–273, 198.

[28] Barry Landy, 'Atlas 2 at Cambridge Mathematical Laboratory (and Aldermaston and CAD Centre)', *Memories of the Ferranti Atlas Computer* (School of Computer Science, University of Manchester, December 2012), http://elearn.cs.man.ac.uk/~atlas/, 6.

differ from the operating system being designed for Titan. In the spring of 1965, the Cambridge and ICT programmers therefore split into separate teams to focus on the two different iterations of the Supervisor. For the remainder of his time at ICT, Coetzee's programming activities focused on the Aldermaston Atlas 2.

The shift in his work had, I contend, long-term consequences for Coetzee's thinking about computing, underlining as it did the connections between his own work on multiprogramming and the wider social, ethical, and political implications of computing. Aldermaston was a high-security environment where much of the UK government's top-secret research into nuclear armaments was conducted. Much of this research, which remains classified, relied on impressive computation facilities. Aldermaston received one of the original Ferranti (Mark 1) computers, and by the time Coetzee was working on the project, Aldermaston also had an IBM Stretch supercomputer as well as remote electronic access between their computers and those held at other research sites.[29] These were innovative and expensive technologies with, for the companies providing them, high-stakes contracts. Teams from Ferranti/ICT, IBM, Aldermaston, and the Atomic Energy Research Establishment collaborated extensively around both hardware and programming to ensure that these machines worked effectively and that Britain remained at the forefront of atomic weapons research. As a site, Aldermaston epitomized the military–industrial complex that President Eisenhower had described in 1961.[30]

Notably, in the late 1950s and 1960s, the so-called Aldermaston March was the centrepiece in the Campaign for Nuclear Disarmament's annual calendar. The annual march between London and Aldermaston in Berkshire raised public awareness about and protested against global nuclear armament, and it drew upward of sixty thousand attendees. Far from a secret government facility, Aldermaston was a politically charged, visible symbol of the military–industrial complex and Cold War nuclear armament. In working on the Atlas 2 at Aldermaston, Coetzee was employed on one of the more controversial

[29] Simon Lavington, 'Ferranti Atlas 2 Computers at AWRE Aldermaston and at the CAD Centre', *Memories of the Ferranti Atlas Computer* (School of Computer Science, University of Manchester, December 2012), http://elearn.cs.man.ac.uk/~atlas/, 4, 8.

[30] While *Youth*'s narrator helps a Mr Pompret with 'wind-tunnel data' for the development of the TSR-2 bomber (83), I have found no evidence of this in Coetzee's own employment record or in publicly available records. IBM did not respond to my request for information. Damien Burke, author of *TSR2: Britain's Lost Bomber*, found no mention of IBM in his research (although he notes that IBM may have been a supplier to a supplier). Damien Burke to Rebecca Roach, 'Tsr-2', email (12 October 2017).

technical projects of his era, and he saw firsthand the ways in which computing was imbricated with—and facilitating—the politics of what computer scholar Paul N. Edwards describes as a 'closed world discourse' characterized by control.[31] This early experience would shape Coetzee's later thinking around computing's cultural import in the context of South Africa.

Coetzee's role on the Atlas project has been largely ignored by literary scholars and historians of computing alike. Lack of interest by the latter group may be attributed partly to Coetzee's decision to leave the Atlas project, which was the most exciting computer research project of the time in Britain, before its completion.[32] Scholars often interpret his move across the Atlantic to pursue a PhD in English as a severing of connections with the world of computing and evidence of his general disinterest in the discipline. This is despite the fact that his subsequent scholarly and creative works testify to his continued engagement with concerns developed while employed in the computing industry. Literary scholars have also downplayed the influence of this period on his intellectual development. Specifically, they often overlook the practicalities of working in the computer industry in the mid-1960s. When his work is discussed, it tends to be within a wider argument about Coetzee's engagement with mathematics and, more specifically, pure mathematics.[33] While this subject is clearly important to Coetzee, when scholars ignore the realities of the computer industry they miss the specificity of Coetzee's experiences and the implications of those experiences for his writing.

Dark Arts in the Reading Room

At the end of August 1965, Coetzee sailed for the United States to pursue a PhD in English at the University of Texas (UT) at Austin, but this was not all that radical a change in direction. While it is generally known that Coetzee wrote an MA thesis on the writings of Ford Madox Ford while working as a programmer in London, his reading was as much guided by his day job as by his dissertation. In fact, Coetzee's intellectual interests over the course of

[31] Paul N. Edwards, *The Closed World: Computers and the Politics of Discourse in Cold War America* (Cambridge, MA: The MIT Press, 1996), 1.

[32] Notably, there is no mention of Coetzee's role on the Atlas project in any of the Atlas Symposium reminiscences or published histories.

[33] The most extensive treatment is Peter Johnston, '"Presences of the Infinite": J. M. Coetzee and Mathematics', PhD dissertation, University of London, London, 2013.

his doctoral study and as an early career researcher in the late 1960s and 1970s grew, at least in part, out of the work he performed at IBM and ICT and the seeming strain that work placed on modernist and New Critical conceptions of authorial impersonality, close-reading practices, and form.

At first glance, Coetzee's MA dissertation topic might suggest that his reading was little concerned with computing, focusing as it did on Ford's literary impressionism. Quoting Ford's own declared preference for 'Impressions rather than Statistics',[34] Coetzee puts forward the Poundian argument that 'selection' is the key to creativity. However, his conclusion that Ford's *The Good Soldier* was 'probably the finest example of literary pure mathematics in English' hints at Coetzee's struggle to resolve a much broader question, namely whether 'impressions or statistics might provide the most authentic "means of approaching to the heart of things" ', which, as Peter Johnston argues, 'coloured almost every aspect of his literary, linguistic, and ultimately ethical enquiry' in the 1960s and 1970s.[35]

Much of this interest is evidenced in the notes Coetzee took on IBM's green and white printer paper (scavenged from the office) when reading for the project. Along with quotations concerning literary impressionism, Coetzee's studies in the British Museum Reading Room and across the 1960s included consulting books engaged with questions around the representational and conceptual implications of a shifting media ecology. He read key works such as Hugh Kenner's *Flaubert, Joyce and Beckett* (1962), Marshall McLuhan's *The Gutenberg Galaxy* (1962), Sigfried Giedion's *Mechanization Takes Control* (1948), G. T. Guilbaud's *What Is Cybernetics?* (1960), and Walter J. Ong's *Ramus, Method, and the Decay of Dialogue* (1958). He followed work in the emerging field of cognitive science, and his notes testify to his personal interest in forms of information representation. For instance, when recording quotations from Christopher Alexander's *Notes on the Synthesis of Form* (1964), which used a mathematical framework to discuss design process, Coetzee made notes in block diagrams rather than alphabetic script. Coetzee's overriding interest was seemingly in the relationship between cognition and formal systems of representation, their media instantiation, and the implications of this for literary and critical endeavours. His thesis might have been concerned with Ford's impressionism, but

[34] J. M. Coetzee, 'The Works of Ford Madox Ford with Particular Reference to the Novels', MA thesis, University of Cape Town, Cape Town, 1963, x.

[35] Johnston, 'Presences of the Infinite', 68.

his context for this discussion was much broader; it stemmed very explicitly from the tensions his work in programming exposed.

Coetzee's intellectual engagement with the implications of formalization extended beyond the reading room. During his time in the United Kingdom, a small but growing community of artists and scholars were exploring the material and conceptual capacities of computers to generate and analyse artistic output. Coetzee followed this community closely and used his employer's equipment to experiment with the artistic possibilities computer technology offered to a young writer striving to find a literary platform and voice.

The first of his experiments was published in the March–April 1963 edition of a UCT student magazine, *The Lion & the Impala*. The 'Computer Poem', which Coetzee explicitly states was written on a 1401 computer, was the result of a 'primitive' programme and his own editing.[36] The published piece comprises the unedited and edited poems along with a glib commentary, which is in part exegesis of the creation process. As Coetzee explains, the programme used a preset vocabulary (in this case eight hundred words from Roget's *Thesaurus*) and structure to generate a list of all possible variants of word choice within these confines (2,100 poems printed at a rate of seventy-five per minute). Coetzee then 'wades' though this 'garbage', selecting and editing the output.[37]

Poem (ex computer)	**Poem (edited)**
Dawn Birds Stream	Dawn, birds, a stream, a calm morning.
Calm-Mornings	You stand among the trees alone and tense.
You) Stand-Among	You have cried.
Forest	You spend the nights away from me,
Alone Tense	Terrified, rapt,
You) Cry	Among owls and black men,
You) Spend-The-Nights	Hoping for violence.
I) Away-From	
Terrified Rapt	
Owls Blackmen	
You) Hope Violence	

[36] J. M. Coetzee, 'Computer Poem', *The Lion and the Impala*, 2.1 (1963), 12–13, 12.
[37] Coetzee, 'Computer Poem', 13.

Retaining the 'key words' so as not to 'dilute' the 'originality' of the poem, Coetzee privileges semantics (and pronouns) over syntax.[38] He shifts 'You) Cry' from simple present to present perfect, thus interrupting the potential triple repetition while retaining the majority of word choices. Overall, the grammatical shifts and the poem, with its theme of 'personal estrangement', are less interesting than the cynicism Coetzee expresses in the commentary on the role of the editor, the programme, and the computer and on the status of the work itself.[39] Coetzee dismisses the programme in its current iteration because it requires an editor with 'elementary critical abilities'; however, he does foresee a time in which a sophisticated 'hack' could run the programme and produce a decent poem without relying on editorial intervention.[40]

This published poem, with its accompanying exegesis, was not the only literary experiment that Coetzee undertook while at IBM. His archives include several printouts from this period; notably, in May 1963, he developed a poetry line generator for an IBM computer. Similar to the previous programme, this line generator used a library of one thousand words, each of which he assigned a grammatical class (pronoun, infinitive verb, preposition, noun, and modifier), to compose lines such as this one:

WE ROAM / WITH SKY / COLD
WE WANDER / ABOUT CITIES / WAITNG
YOU WAIT / BETWEN LEAVES / BURNNG[41]

The 508 lines testify to the material constraints within which Coetzee operated: the computer used 8-bit bytes, meaning that each byte (unit of memory) could contain six characters (the other two being reserved for word mark and parity indication); accordingly, those words longer than six characters had to be abbreviated.[42]

The extensive lines produced are wearisome to read, and they evince a restricted lexicon and repetitive syntax. It is at the level of coding, however, that these lines reveal their formal innovations. The line generator was composed in a combination of FORTRAN-style pseudocode and assembly code (a low level programming language); much of the programme is taken up with

[38] Coetzee, 'Computer Poem', 12.
[39] Coetzee, 'Computer Poem', 13.
[40] Coetzee, 'Computer Poem', 12.
[41] J. M. Coetzee, Untitled line generator output [1963], container Obs 143, JMCP.
[42] Forward slashes are part of the computer-produced line.

instructions for set-up, rendering the library data readable, and formatting the printed output. For example, the code 'FORMAT(1H ,A6,1X,A6,3H / ,A6, 1X,A6,3H / ,A6)' produced the word order, forward slashes, and letter spacing.[43] The necessity of such programming indicates the low level at which Coetzee operated: he had to write the subroutine to tell the computer to ignore blank lines in the input library, for example. Such programming might be 'primitive', but much of it was written entirely from scratch in an extremely time-consuming exercise that relied heavily on a human coder. This programme does, however, seek to expand on the workings of its predecessor. Coetzee wrote two subroutines to automate aspects of the programme; the second was a random number generator (RNG). This programme used prime numbers to create the semblance of randomness: the numbers were used to automate word combinations as Coetzee explored methods of writing that placed the author at one remove from the process of invention. In decoupling (albeit temporarily) creation from conscious authorial intention, the programme offers a mechanical solution to what T. S. Eliot saw as the problem of personality in poetry.

Coetzee, bolstered by the congenial atmosphere at Cambridge, continued these experiments during his time at ICT. Coetzee's experiments included attempts to use Titan/Atlas 2 machine code to produce RNG-composed lines of poetry. Using similarly repetitive, if slightly more complex, grammatical structures, Coetzee deployed more exotic vocabularies to explore the degree to which semantic effect could be automated:

> TURBID SCURF SADDEN THE CONVERSATION
> WEEKLY ATARAXY TREPAN THE DOCUMENT
> CARKING CREED VOICE THE ENTROPY[44]

Or,

IN THE BALLAD THE ANGEL NEEDS THE BIDDING. THE ADMONISHMENT IS SYMMETRICAL AND TASTE FILLS MY LAND

IN THE TWILIGHT THE POPPY CLAIMS THE CHILD. THE FLOWER IS HEAVY AND LONELINESS FILLS MY EARTH

IN THE WINTER THE FOOTFALL CHERISHS THE SNAIL. THE CICACA IS MILKY AND EFFENECENT FILLS MY FRUIT[45]

[43] J. M. Coetzee, 'Line Generator' (30 May 1963), container Obs 143, JMCP.
[44] J. M. Coetzee, Untitled and undated printout, container Obs 143, JMCP.
[45] J. M. Coetzee, Untitled printout (2 April 1965), container Obs 143, JMCP.

Or,

THE DOCUMENT BEFORE THE HONEY TRAMPLES THE CAPTAIN OF CERTAINTY. THE SWEET WALKS

COUNTERFEIT DOCUMENT MAUDLIN HONEY CLOSE CAPTAIN INCONCLUSIVE SWEETS

THE LEAF OVER THE VIOLET WANTS THE NEST OF ANXIETY. THE BLOOM DRESSES[46]

All three outputs were printed on different machines but do seem to draw on the same base vocabulary, which included a high percentage of unusual and arcane words. The number of lines generated was large; the second extract quoted above is taken from a text that extended across approximately one hundred pages. Of these pages, the first few are annotated; a red or blue pen underlines various phrases and a tiny pencil tick mark accompanies them, suggesting that Coetzee was transcribing or otherwise recording those phrases with which he was particularly taken, including 'the cicaca is milky', 'the chimera is sleeping', and 'the nude is punctual'.[47] As with his earlier experiments, the programmes here generate lines from which the poet might then identify choice phrases. The act of selection, such a key feature of the writing process for modernists like Ezra Pound and Ford, here attains full creative significance.

While these programmes might be classed as relatively unsophisticated (by Coetzee's proclaimed scale), they do demonstrate an increasing structural complexity—longer lines of output—and computational dexterity. The extant printouts indicate that Coetzee wrote some of this programming 'close to the machine', in machine code.[48] While composing the RNG for the Titan/Atlas 2, Coetzee inserted an additional handwritten line in the Atlas assembly code printout: '170 90 81 0 1C. if length>prime'.[49] While the latter part of the line is human-readable commentary, the former is the compiler-readable assembly code. At this point in his career, Coetzee was working at such a low level of programming (writing the Supervisor code during the day) that, unsurprisingly, when he came to write computer poetry at night,

[46] J. M. Coetzee, 'Generate Index' (25 March (or May?) 1965), container Obs 143, JMCP.
[47] J. M. Coetzee, Untitled printout (2 April 1965).
[48] For a discussion of the cultural kudos and intellectual appeal of programming at such 'low' levels, see Ellen Ullman, *Close to the Machine: Technophilia and Its Discontents* (San Francisco: City Lights Books, 1997).
[49] Coetzee, 'Generate Index'.

he composed at the same deep level, using executable numerical notation. Despite its late-modernist associations, this was computer poetry composed with zero abstraction.

This was also dangerously executable code. The Titan/Atlas 2 had three separate protection regimes from which programmes could run: the regular user programme, the Supervisor programme, and the interrupt code. While in the former the user could only modify memory storage areas assigned to them, the latter two provided enhanced access and execution possibilities but also brought with them a danger of a system crash should a computer programme fail. Coetzee wrote his programmes for the Supervisor mode because his work on job control made him more familiar with this mode. However, his choice to run the programme in that regime meant that no other client programmes could be run at the same time. It also brought with it a small risk that the entire supercomputer could fail, which was especially significant given that this was the height of the Cold War.

But which computer did Coetzee use? By May 1965, the Titan and Atlas 2 teams were in the process of splitting, but the dates and the paper records are inconclusive. It is probable that these computer programmes were run on the Titan as Coetzee was unlikely to have had unfettered access to a supercomputer in an atomic weapons research facility at eleven at night (although this was the usual time to perform tests). Nevertheless, it is easy to imagine the alternative, and the potential outcome it might have fostered.[50]

Remembering Computers in Texas

Coetzee's creative and intellectual interests, developed during his time in the computing industry, seem to have held his attention across decades, continents, and careers. During his doctoral studies at UT Austin between 1965 and 1968, Coetzee would pursue research at the juncture of computational linguistics and literary study. In addition to taking classes with a heavy linguistic and stylistic bent, Coetzee would read the work of Russian Formalists closely, demonstrating a commitment to pursuing methods of

[50] Moreover, Coetzee was rumoured to have been designated a 'communist' by the South African government and deemed a security risk on the Aldermaston site (Barry Landy to Rebecca Roach, 'Re: Coetzee...Again!' (1 May 2017)). Regarding the incident in *Youth* where the narrator writes to Chinese Embassy offering his services (as an English teacher) (153), I can find no evidence in the JMC archives to support or disprove its basis in fact.

literary analysis that followed the 'spirit of scientific positivism'.[51] If his computer poetry experiments highlight that many of those features and practices in which modernism was deeply invested were constructed forms with specific real-world implications, Coetzee's engagement with style can be understood in part as an attempt to retain facets of this value system within a more quantitative context. These studies provided Coetzee with crucial grounding in the conceptual and methodological issues surrounding stylistic analysis, linguistics, and literary representation across languages; they also offered a crucial complement to his already extensive knowledge of advanced programming. Together, these interests fuelled fifteen years of research examining the application of computation to the study of literature and in particular style; as one scholar put it, 'there is a strong…case for claiming J. M. Coetzee as a significant figure in the early development of digital humanities'.[52]

Coetzee's doctoral dissertation, *The English Fiction of Samuel Beckett: An Essay in Stylistic Analysis*, was the culmination of his interest in stylostatistics. The dissertation seeks to account for Beckett's increasing dissatisfaction with the English language (and his subsequent move to writing in French) by examining the author's prose style. Later, Coetzee frames the linguistic side of this interest as the query: 'if a grammar of language is to be thought of as a formalization of the processes required to decode utterances in that language, what *additional* capacities would such a grammar have to possess to allow us to decode utterances that employ novel or deviant rules?'[53] Applying stylostatistical methods to what the narrator of *Youth* would later describe as Samuel Beckett's 'classless' prose, he considers the insights that such 'objective' criticism might offer to the 'impressionistic' discipline of literary criticism.[54] Coetzee's dissertation maps syntactical units, syllabicity, sentence length, rarity of words, and even Watt's chains of logical speculation (something Kenner would also do, as we shall see in the next chapter). Yet in this work and in the published essays that derived from his studies, Coetzee ultimately concludes that the promise of objective critical insight is a hollow one.

[51] Boris Eikhenbaum quoted in Nicholas Gaskill, 'The Close and the Concrete: Aesthetic Formalism in Context', *New Literary History*, 47.4 (2016), 505–24, 510.

[52] Peter Johnston, 'J. M. Coetzee's Work in Stylostatistics', *DHQ*, 8.3 (2014), http://www.digitalhumanities.org/dhq/vol/8/3/000188/000188.html.

[53] J. M. Coetzee, 'Linguistics and Literature', *An Introduction to Contemporary Literary Theory*, eds. Rory Ryan and Susan Van Zyl (South Africa: A. D. Donker, 1982), 41–52, 42.

[54] Coetzee, *Youth*, 155.

For Coetzee,

> Stylostatistics oversimplifies because it is dominated by a metaphor of linearity, a conception of language as a one-dimensional stream extending in time. The origin of the metaphor probably lies in our alphabet; it has been fortified by printing technology and by the twentieth-century metaphor of the mind as a computer with an input system which reads linear strips of coded information.[55]

Evincing a keen awareness of the conceptual constraints such a metaphor places on the methodology, Coetzee uses an analogy rooted in a material process with which he is very familiar in order to make two points. Firstly, in its computer input-like orientation towards the syntagmatic, the metaphor ignores 'precisely what we are interested in', namely 'variation in a stylistic dimension'. Secondly, this metaphor implies that the experience of literature is linear, whereas Coetzee argues in his dissertation that it involves 'incessant recursion': 'as we read we are continually reformulating formal hypotheses to account for what we are reading and what we have read. Insofar as stylistic data have a formal function they too enter into these reformulations.'[56] In rejecting the metaphor of linearity, Coetzee dismisses his analogy based on one computational process and adopts another—programming's recursive reading. He might criticize stylostatistics for its conceptual limitations, but Coetzee is at this point very much open to the possibility that computation, and programming in particular, might offer positivist literary critical insight by formal analogy.

Formal, but not immaterial. Coetzee's understanding of the word, and by extension his understanding of the creative and critical processes that produce and interpret literature, is far from Information Theory's dematerialized and mathematical understanding of communication. Rather, it accords with what we know about his own experiences with computers. Coetzee's programming work did much to underline the processes of abstraction and formalization that enabled computation and the underlying material structures on which they operated to function; these were processes and material structures that stylostatistics, dependent as it was on a statistical conception of information, deemed conceptually irrelevant (if not methodologically). It is thus notable that Coetzee's dissertation turns on juxtaposing two definitions of style to wrestle with (linguistic) media determinism.

[55] Coetzee, 'English Fiction of Samuel Beckett', 160.
[56] Coetzee, 'English Fiction of Samuel Beckett', 161.

According to Coetzee, structural linguists such as Bernard Bloch understand that 'a word can be conveniently reduced, for the purposes of study, to a dimensionless and immaterial point. For Beckett, on the other hand, the "terribly arbitrary materiality of the word's surface" is, we infer, at least in 1937, a burden.'[57] While Coetzee suggests that Beckett's prose style is mathematical and that Watt is analogous to Leibniz's automaton in being 'something living encrusted on the mechanical',[58] he insists that there is a 'polar and antithetic relation' between the two notions of style.[59] To use programming terminology, we could say that Bloch happily approaches style through examination of the higher-level abstractions of FORTRAN while Beckett utterly rejects any correspondence between the machine code and its systematization as a formal language; in other words, Coetzee dismisses the efficacy of the compiler and instead argues for the imbrication of form and content.

Although comfortable moving between levels of abstraction, Coetzee refuses to conceive of language, or the literary form and style it can produce, as dematerialized. It is no surprise that he turns to Beckett's manuscript revisions to *Watt*, the physical drafts of which were held in the UT Austin library, in order to think through the import of what he describes here and in later essays as Beckett's career progression 'toward a formalization or stylization of autodestruction.'[60] In this, he places himself in opposition not only to Bloch but also to Claude Shannon. Far from conceiving of language as merely statistical or as a transparent indicator of thought or perception, Coetzee uses computation to explore the specificity of authorial choices and word use in which material form is itself a consideration.[61] More than this, Coetzee adopts Beckett's 'automatism of style' as a foundation on which to build his own creative writing practice.[62]

J. M. Coetzee's Computational Critique

Coetzee's creative dialogue with Beckett might seem, at first glance, to be an example of academic abstraction—a far cry from the realities of punched

[57] Coetzee, 'English Fiction of Samuel Beckett', 2–3.
[58] Coetzee, 'English Fiction of Samuel Beckett', 32.
[59] Coetzee, 'English Fiction of Samuel Beckett', 2.
[60] J. M. Coetzee, 'Samuel Beckett and the Temptations of Style' [1973], reprinted in *Doubling the Point*, 43–9, 45.
[61] Compare Coetzee's writing in 'English Fiction of Samuel Beckett': there are no 'invisible extension cords leading from modes of thought or perception, into which a writer can plug his pen', 89.
[62] Coetzee, 'Samuel Beckett and the Temptations', 49.

cards and the Cold War. In fact, we can see the developing politicization of Coetzee's platform of aesthetic automatism in the decade following his graduate work. This period was marked by his move to Buffalo, New York, for his first academic position, and then back to South Africa in 1971, where he would find a job at UCT and simultaneously begin to make a name for himself as a writer. In this period, Coetzee's aesthetic automatism would, I suggest, be increasingly envisaged as autonomy from the dominance of 'binary' or 'computational thinking'.[63]

Distinct from the machines that Coetzee programmed, computational thinking is 'the view that a great deal, perhaps all, of human and social experience can be explained via computational processes'.[64] Conceived as universal machines, computers would become, as I noted in the introduction, 'metaphors for metaphor itself: they embody a logic of substitution' under which important cultural distinctions could be dissolved.[65] In the post-war era, rapid advances in computing technologies were accompanied by similar advances in the application of computational thinking across the cultural and social domain: whether Noam Chomsky's theory of transformational grammar in linguistics or Hilary Putnam's classical computational theory of mind in philosophy. Often stripping away its own metaphorical origins, what N. Katherine Hayles has called the 'regime of computation' and David Golumbia terms 'the cultural logic of computation', offers a technological update to a longer tradition of the rationalist theory of mind.[66] In so doing, computational thinking has helped to promote a set of beliefs that benefits existing structures of institutional power: 'computation meshes far too easily with instrumental reason'.[67]

This set of beliefs, rather than computers themselves, would be the target of Coetzee's literary project as early as the late 1960s. Coetzee would incorporate computer processes into his fiction as a method for encouraging readers to resist the cultural automatization that computational thinking propagated. It was, however, in the context of late apartheid South Africa that

[63] I draw here on Andrew Goldstone's reading of modernist conceptions of aesthetic autonomy as autonomy *from* something, *Fictions of Autonomy: Modernism from Wilde to de Man* (Oxford: Oxford University Press, 2013), 15.

[64] David Golumbia, *The Cultural Logic of Computation* (Cambridge, MA: Harvard University Press, 2009), 8.

[65] Wendy Hui Kyong Chun, *Programmed Visions: Software and Memory* (Cambridge, MA: The MIT Press, 2011), 56.

[66] N. Katherine Hayles, *My Mother Was a Computer: Digital Subjects and Literary Texts* (Chicago: University of Chicago Press, 2005), 17.

[67] Golumbia, *Cultural Logic*, 5.

Coetzee's platform of aesthetic automatism would become more clearly articulated. In the years after the 1976 Soweta Uprising, international and internal pressures would lead to the politicization of computation, increasingly depicted as a symbol of the apartheid regime, while also placing the platform of aesthetic autonomy under strain in the literary field. Coetzee's particular conception of the relation between literature and computing, which had anticipated many of these symbolic connections in the years previously, would take on a more explicit political dimension. Both before and after 1976, his literary project would seek autonomy from the excesses of computationalism as embodied in the apartheid state. Moreover, the products of Coetzee's aesthetic automatism offer instances of 'tactical media', or media that signifies 'the intervention and disruption of a dominant semiotic regime, the temporary creation of a situation in which signs, messages and narratives are set into play and critical thinking becomes possible.'[68] Coetzee's writings work to promote a reader, and not an automaton, who can think critically and, resist the logic of binaries, substitution, and functionalism, in order to confront the institutional power of apartheid South Africa.

Coetzee's Critique of Computational Thinking

Coetzee's anxieties about computational thinking would develop as he finished up his doctoral studies and in the years after he left Austin. In a letter to *The Daily Texan*, he would subtly satirize the 'clear and systematic thinking' behind US strategy in the war: 'The Vietnamese who has seen a modern technology in action must be aware of the true direction of "the tide of history".'[69] In a climate of increasing resistance to the Vietnam War and student radicalization (his own participation in protests at Buffalo and arrest would lead to his US visa extension being refused), Coetzee would make more explicit associations between stylostatistics and right-wing political agendas: in 1971, he would suggest that the project promoted a 'universal language for the technical elite', thus tying 'succeeding generations into a twentieth century positivist mythology more tightly than natural languages tie us' to the past.[70]

[68] Rita Raley, *Tactical Media* (Minneapolis: University of Minnesota Press, 2009), 6.
[69] J. M. Coetzee, 'Misconception', Letter to the Editor, *Daily Texan* (24 October 1967).
[70] J. M. Coetzee, 'Review of *Nach Allen Regeln Der Kunst* by Wilhelm Fucks', *Style*, 5.1 (1971), 92–4, 94.

Back in South Africa in the 1970s and 1980s, he would continue to keep abreast of work around the subject, with computer scientist Joseph Weizenbaum's seminal book, *Computer Power and Human Reason: From Judgment to Calculation* (1976), a particular favourite. One of the clearest articulations of computational thinking (although the term isn't used) and its dangers, Weizenbaum's book received considerable cultural comment on publication, coming as it did from an early champion of AI, and programmer of ELIZA, one of the earliest chatbots. In his explosive book, Weizenbaum argued that 'however intelligent machines may be made to be, there are some acts of thought that *ought* to be attempted only by humans'; this book became a key reference point for Coetzee across his career.[71]

This set of beliefs would shape Coetzee's creative practice even before he engaged with Weizenbaum's work. *Dusklands*, composed of the two stories, 'The Vietnam Project' and 'The Narrative of Jacobus Coetzee', explores the ramifications of computational thinking on individuals. Drawing heavily on his own reading on computation, 'The Vietnam Project' turns to the war as a key instance of what Weizenbaum would later call the 'imperialism of instrumental reason'.[72] The subject is government analyst Eugene Dawn who recalls the so-called Whiz Kids, the desk-bound RAND corporation analysts who sought, under American Defense Secretary Robert MacNamara, to rationalize war strategy through computational analysis, with devastating human consequences. Examining the effectiveness of psychological warfare, Dawn is himself rendered 'unfit for consecutive thought' by the 'plump, incontrovertible voice of the master of statistics' on the radio and the other propaganda sources he is supposed to be analysing.[73] Struggling to produce a report for a supervisor (named 'Coetzee') whose expertise is in game theory, Dawn flails under the requirement that he 'explain in words of one syllable the kind of procedures you follow—how myths operate in human society, how signs are exchanged'.[74] While the 'message' of the newspaper is: 'I can say anything and not be moved. Watch as I permute my 52 affectless signs', for Dawn, it is precisely the disconnect between this instrumental,

[71] Joseph Weizenbaum, *Computer Power and Human Reason: From Judgment to Calculation* (New York: W. H. Freeman, 1976), 13.

[72] See e.g. Paul Erickson, Judy L. Klein, Lorraine Daston, Rebecca Lemov, Thomas Sturm, and Michael D. Gordin, *How Reason Almost Lost Its Mind: The Strange Career of Cold War Rationality* (Chicago: University of Chicago Press, 2013); and Michael Adas, *Dominance by Design: Technological Imperatives and America's Civilizing Mission* (Cambridge, MA: Belknap Press, 2006), in particular, Chapter 6: 'Machines in the Vietnam Quagmire'.

[73] Coetzee, *Dusklands*, 14.

[74] Coetzee, *Dusklands*, 4.

computational conception of human language and society and his own lived experience that leads to his own paranoia and eventual breakdown.[75]

In 'The Narrative of Jacobus Coetzee' meanwhile, Coetzee traces the origins of such computational thinking in rationalism and ties it more explicitly to colonial expansion. The story describes the journey of the eponymous character, an eighteenth-century Boer, as he travels into the southern African interior, naming and mapping it and the local inhabitants. Coetzee nods to a long tradition of justifying Western imperialism (particularly the colonization of the African continent) via claims to technological and scientific superiority.[76] In this tale, despite Jacobus Coetzee's lyrical desire for the gun to stand 'for the hope that there exists that which is other than oneself', as 'our mediator with the world', the gun is not an indicator of cultural superiority but merely brute power.[77] Moreover, the frontiersman's sense of superiority might be rooted in condescension of the 'Bushman' whose language 'does not include a procedure for counting', but he uses numbers mainly to quantify his own atrocities. 'Every wild creature I kill crosses the boundary between wilderness and number. I have presided over the becoming number of ten thousand creature.' Far from evidencing the superiority of Western instrumental reason, Coetzee's 'hero of enumeration' comes to represent the disparity between rationalism and the unethical, colonialist uses to which it can be put.[78] Neither 'The Narrative of Jacobus Coetzee' nor 'The Vietnam Project' discusses computers outright, but Coetzee's first published work has as its target the dangerous logic of computational thinking and the rationalism on which it rests.

Computerizing Apartheid and the Politicization of Aesthetic Automatism

As part of a seminar series he taught late in his career on the topic of 'Literature and Confinement', Coetzee introduced his topic through reference to Kafka. The latter had 'put his finger on what was new and pathological about the turn taken by advanced Western societies', namely a turn 'toward the centralization of control in systems which soon pass beyond the control of the

[75] Coetzee, *Dusklands*, 14.

[76] There has been much critical discussion around the difficulties Coetzee's fictions pose for readers thanks to their replication of discourses of colonialism—what David Attwell would term Coetzee's 'situational metafiction', *J. M. Coetzee*, 20.

[77] Coetzee, *Dusklands*, 79.

[78] Coetzee, *Dusklands*, 80.

human beings who nominally operate them, and indeed beyond the control of human reason.'[79] Coetzee refers to Weizenbaum and to the NORAD (North American Aerospace Defense Command) computer programmes which 'in their totality, represent a book in which mankind either does or does not have faith: either it works or it does not work, either it saves one from annihilation or it does not; but there is no way of understanding its workings in detail'. As Coetzee sees it, NORAD, a 'strict isomorph of Kafka's houses and castles', requires of its reader blind faith.[80] Quite what the students thought of this reading is not recorded, but the computational model would function as an important foil for Coetzee as he developed his literary project in the 1960s and 1970s. It would be in the following decade, under the last, violent, gasps of the apartheid regime, that Coetzee would adopt a more explicit articulation of aesthetic automatism.

'The growth of the computer age in South Africa has been closely linked to the consolidation and expansion of the white power structure'; so announced one American anti-apartheid pamphlet in 1982.[81] Across the 1970s, in the period in which Coetzee was establishing himself as a writer, the South African computer industry would become heavily involved with the functioning of the apartheid state. The state's utilization of computational thinking as a defence for its racist policies would in turn shape Coetzee's own thinking about the socio-political implications of computationalism and its relationship to literature. While Tara McPherson has provocatively suggested that in the United States, 'the very structures of digital computation develop[ed] at least in part to cordon off race and to contain it', in South Africa, this was much more overtly the case.[82] It was also part of a longer practice of organizing the South African state around biometric 'technologies and architectures of identification' (whether Edward Henry's fingerprinting regime introduced in 1900, or Verwoerd's Bewysburo—'Bureau of Proof'—project of the 1950s) that predated computer

[79] J. M. Coetzee, 'Kafka and the Question of Autobiography', Seminars taught abroad, Box 114, Folder 10, JMCP, 1.

[80] Coetzee, 'Kafka and the Question of Autobiography', 1.

[81] NARMIC/American Friends Service Committee, *Automating Apartheid: U.S. Computer Exports to South Africa and the Arms Embargo* (NARMIC/American Friends Service Committee, 1982), 6.

[82] Tara McPherson, 'Why Are the Digital Humanities So White? or Thinking the Histories of Race and Computation', *Debates in the Digital Humanities*, ed. Matthew K. Gold (Minneapolis: University of Minnesota Press, 2012), 139–60, dhdebates.gc.cuny.edu/debates/text/29. McPherson makes her argument via a discussion of the simultaneous emergence of a 'modularity' principle in the (American) social field and the UNIX operating system in the late 1960s.

technologies.[83] When, in 1970, the Department of the Interior deployed an IBM mainframe computer in the relaunch of their population registry, they were using the latest technology in the service of biometric governance.

When Coetzee returned to South Africa in 1972, the fledgling computer industry was small, with the total number of computers across the country approximated as 1,500, and heavily dependent on imported technology. While a number of American and British companies would enter the market with South African subsidiaries, at this point, IBM controlled nearly half of the market.[84] It was also an industry closely associated with the country's nuclear programme and the apartheid regime in general: IBM had provided the Defence Department with three mainframe computers and IBM computers were regularly used to implement the 'Bewysboek' or 'Book of Life' systems.[85] Meanwhile, International Computers Ltd (ICL) (ICT's name following a 1968 merger) provided the state with the mainframe computers on which to implement the passbook system and would become a principal supplier of computer technologies to the police, the Bantustan administrators, and the military. As another anti-apartheid booklet concluded, 'many of the worst features of Apartheid rely upon ICL'.[86]

This industry, dominated by Coetzee's two former employers, not only 'aided and abetted' apartheid on a practical level, it was also a crucial site of technopolitical dispute for both the apartheid elite and anti-apartheid activists. For the former, as Paul N. Edwards and Gabrielle Hecht have argued, 'the fantasy of technical control—a fantasy built into vast administrative systems, computers, fingerprint collections, and daily routines of technical activity—became crucial to apartheid ideology, even as increasingly violent police harassment became its sordid reality'.[87] While, as they note, the 'material functionality' of these technological systems 'never even approached the modernist panopticism envisioned by Turton and Verwoerd', the goal of computerization helped to conceal the oppressive nature of this system under an apparent quest for efficiency and automation.[88] This deployment

[83] Keith Breckenridge, *Biometric State: The Global Politics of Identification and Surveillance in South Africa, 1850 to the Present* (Cambridge: Cambridge University Press, 2014), 15.

[84] Paul N. Edwards and Gabrielle Hecht, 'History and the Technopolitics of Identity: The Case of Apartheid South Africa', *Journal of Southern African Studies*, 36.3 (2010), 619–39, 632.

[85] Richard Knight, *U.S. Computers in South Africa* (The Africa Fund, 1986), 9; and Thomas Conrad, 'Machines That Help Make Apartheid Run', Letter to the Editor, *New York Times* (18 May 1985), 24.

[86] Anti-Apartheid Movement, *Computerising Apartheid: ICL in South Africa* (Anti-Apartheid Movement, 1978), 3.

[87] Edwards and Hecht, 'History and the Technopolitics of Identity', 638.

[88] Edwards and Hecht, 'History and the Technopolitics of Identity', 620.

of computationalism further enabled the apartheid elite to characterize themselves and their goals as driven by Western administrative and scientific values.

Anti-apartheid activists, identifying the country's reliance on foreign technology as a key weakness of the regime, also engaged in technopolitical activities—although in their case, the aim was to disrupt the computationalism that drove state ideology and practice. The African National Congress, for example, covertly developed a computerized, encrypted communication system that could work across international borders (and with international phone lines) to aid Operation Vula in the late 1980s. They used computers to fuel oppositional writing and reading. More overtly, across the decade, anti-apartheid activists sought to challenge the state's 'displacing [of] political agendas into technical acts and fantasies' and to target the foreign investment and computer technology aiding the South African state.[89] Pamphlets such as *Automating Apartheid: U.S. Computer Exports to South Africa and the Arms Embargo* highlighted the role played by international (particularly America) corporations in helping to maintain white minority rule in the country. Against the broader background of international embargos against the country, local campaigns encouraged shareholders to pressurize US companies like IBM to divest of their South African holdings (which it did in 1986) and citizens to compel their elected officials to enforce the sanctioning of computer technology sales to the apartheid regime. Direct action activities also led to a 1978 strike of two thousand ICL workers in Britain who were protesting against the company's associations with the apartheid state. For both the regime and the resistance, computing would become a highly politicized symbol in the 1970s and 1980s and the industry and thinking it promoted would be closely associated with apartheid oppression.

The implications of this connection for Coetzee would be notable. As he established himself as an academic and writer in the 1970s, Coetzee had tacitly acknowledged his own programming experience. Drawing on expertise gained in Britain and America, Coetzee wrote programmes to automate scheduling and mark allocation for the English Department at the UCT. Meanwhile, he published articles in the nascent field of humanities computing that discussed his own experiments in stylostatistics, becoming briefly famous in the United States as the South African professor who had

[89] Edwards and Hecht, 'History and the Technopolitics of Identity', 638.

inadvertently demonstrated the limits of computing for the reviewing of books.[90]

Across this decade, he would develop his essay 'Surreal Metaphors and Random Processes' that draws on his efforts at IBM and ICT to produce computer-generated metaphors. Comparing his programmes with the Surrealists' attempts to automate composition, he notes the 'inevitabl[e] conflict between a systematizing theory of language [used in programming] and a literary practice (like that of Surrealism) with philosophical objections to the system'.[91] Quoting approvingly from George Steiner's *After Babel* that 'a "closed" syntax, a formally exhaustible semantics, would be a closed world', Coetzee attempts to analyse the place of authorial consciousness and intention in creative writing.[92] As he would later note about his own writing process, part of the activity of creative writing for him is to overcome the 'resistance' offered by the 'automatism built into language'. In this same period, Coetzee would experiment with writing with automatism in his attempt to produce poetry based on computer line generation. Between December 1971 and August 1975, Coetzee filled notebooks with selections of lines generated on the Atlas 2. He eventually published this work as 'Hero and Bad Mother in Epic, a poem' (1978).

In the same decade, Coetzee also regularly mentioned his training as a computer scientist in author biographies: 'J. M. Coetzee was born in South Africa in 1940. Trained originally as a computer scientist and linguist, he presently teaches linguistics and American literature at the University of Cape Town.'[93] Offered by Coetzee to his publisher (of *In the Heart of the Country*), the author's inclusion of the detail in such a brief note suggests that he considered it salient to the persona he wished to promote, rather than a part of his life to be jettisoned from the record. By contrast in the 1980s, as the computer became a potent symbol of the apartheid regime, Coetzee would downplay his personal connections with the industry. His employment history brought with it possible accusations of, albeit overseas, collusion with the regime.

[90] John Leonard, 'Last Word: Beckett Safe from Computers', *New York Times Review of Books* (19 August 1973), 27, is a response to Coetzee's article 'Samuel Beckett's *Lessness*: An Exercise in Decomposition', *Computers and the Humanities*, 7.4 (March 1973), 195–8.

[91] J. M. Coetzee, 'Surreal Metaphors and Random Processes', *Journal of Literary Semantics*, 8.1 (1979), 22–30, 28.

[92] Coetzee, 'Surreal Metaphors and Random Processes', 27.

[93] J. M. Coetzee, 'Draft Notes', Business Correspondence March 1976–December 1977, Box 69, Folder 2, JMCP.

Modulating his personal associations with the industry, Coetzee would, nevertheless, cleave to his platform of aesthetic automatism more strongly in this period. From one perspective, this commitment to 'autonomous fictions' seems anomalous, even within a literary culture schooled heavily in a European tradition (modernism) that promoted aesthetic autonomy. As Alys Moody notes, 'Reading South African literary discourse of the 1980s, what is immediately striking is the almost unanimous consensus that political engagement must be the primary locus of literary activity…. In such a context, claims for autonomous art appear not so much irrelevant as downright irresponsible.'[94] Indeed, aesthetic autonomy was itself promoted by the censors themselves, resulting in a situation that, according to Jarad Zimbler, 'inevitably contributed to the general opinion that avant-garde literature was complicit or, at best, politically irrelevant.'[95] Yet, for Coetzee, it is precisely literature's potential to retain autonomy *from* computational thinking that marks it as socially and politically significant.

Autonomous Fictions and Computational Reading

When Coetzee first described *Dusklands* to potential literary agents, he argued that while the novella was 'of no historical value, in the narrower sense' it 'does trace a line between so-called autonomous fictions and the psychology of the conquistador'.[96] While having inherited a platform of aesthetic autonomy from those modernist authors he admired, including Kafka, Musil, Joyce, Pound, Beckett, Eliot, and Ford, and while making a claim for the work's lack of 'historical value', Coetzee was also sensitive to the potential politics of aesthetic autonomy in the age of computationalism. Peter Kalliney, discussing Anglophone African writers working in the post-war era, including Christopher Okigbo, Wole Soyinka, Rajat Neogy, and Chinua Achebe, has argued that these authors exploited literary modernism's aesthetic autonomy, with its 'ideals of writerly detachment and intellectual objectivity' in the context of decolonization. This platform gave them 'a language through

[94] Alys Moody, *The Art of Hunger: Aesthetic Autonomy and the Afterlives of Modernism* (Oxford: Oxford University Press, 2018), 4.

[95] Jarad Zimbler, 'For Neither Love Nor Money: The Place of Political Art in Pierre Bourdieu's Literary Field', *Textual Practice*, 23.4 (2009), 599–620, 612.

[96] J. M. Coetzee, Letters to Calder and Boyars; Back Jr Literary Agent; McIntosh, McKee, and Dodds (3 December 1971). Business Correspondence, 1969–March 1976, Box 69, Folder 1, JMCP.

which to negotiate their complicated relationship with state bodies and nationalist movements' while simultaneously enabling them to 'announce their freedom from colonial tutelage'.[97] In the context of South Africa, a state that itself justified its racist ideologies and practices via its 'deluded' modernist aspirations, Coetzee also 'tweaked the doctrine', reconceiving of aesthetic autonomy within the context of computational thinking.[98]

Coetzee's aesthetic automatism was the result of his sense that literature's value lay, at least in part, in its potential to resist computational thinking. Coetzee's apprehensions about the application of computational thinking to literary analysis likely stemmed, as we have seen, from the modelling of reading as linear; in his second fiction, he would highlight the limitations of this model by staging it. Coetzee's *In the Heart of the Country* is notable for its numbered sections, the writer's 'way of pointing to what is not there between them'.[99] Emphasizing sequence and the calculability of these textual chunks, the numbering of Magda's narrative recalls the linearity of the computer input/output system and the batch processing of punched cards. The work seems on first glance to encourage computational reading.

However, the novel also includes narrative sequences that encourage recursive, rather than linear, reading: most infamously Madga's rape by Hendrik. Section 205 recounts Hendrik beating Magda in the kitchen, her lunging at him with a fork, the tines 'scrape his shoulder, probably not even piercing the skin; but he exclaims in surprise and hurls me to the floor'; the section ends with her lying on the floor with her knees ambiguously lifted. Section 206 circles back, this time Hendrik does not beat her but grabs her, at which point she lunges at him with a fork, 'scraping his shoulder. The skin is not even pierced, but he sucks in his breath with surprise', hurling her against the wall. The 'fork falls. His pelvis grinds hard into me' and he rapes her in the kitchen. Section 207 circles back again, 'The fork falls to the fall, his pelvis grinds hard', and in sections 208 and 209, the location of the rape has now moved to the bedroom.[100] The focus on discrete objects, acts, and linguistic phrases highlights the permutational nature of these sequences, with the result that the experience of reading is highly iterative.

[97] Peter Kalliney, 'Modernism, African Literature, and the Cold War', *Modern Language Quarterly*, 76.3 (2015), 333–68, 334–5.

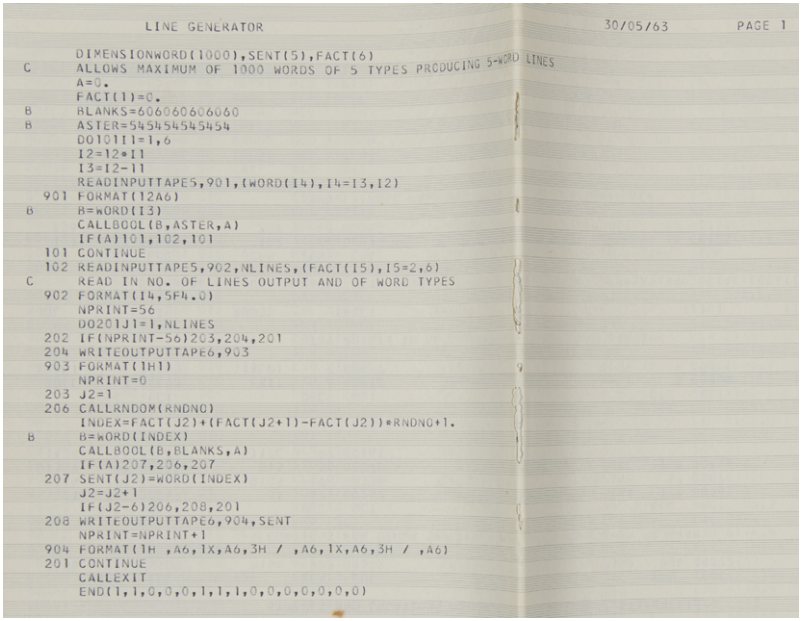
[98] Deborah Posel, 'The Apartheid Project, 1948–1970', *The Cambridge History of South Africa*, vol. 2, eds. Robert Ross, et al. (Cambridge: Cambridge University Press, 2011), 319–68, 345; and Kalliney, 'Modernism, African Literature', 334.

[99] Coetzee, *Doubling the Point*, 59.

[100] Coetzee, *In the Heart of the Country*, 113–15.

These passages disrupt the reader's expectation of narrative linearity, demanding that they engage critically with the process of meaning production.

Although clearly a rejection of the model of reading endorsed by computational thinking, it is crucial to recognize the origins of this iterative reading practice in programming. Lest we think of Coetzee's aesthetic automatism as marking a complete rejection of computers, his novels champion a mode of reading that exemplifies the processes entailed in reading certain computer code and with which he was very familiar. Back in 1963, Coetzee composed 'Line Generator' (Fig. 1.1). It is written in pseudocode, which is, crucially, not machine executable: the only reader is a human one. Here the programme operates on a series of additive steps and circular loops to select words from a predetermined library and produce the output. Unlike our printed alphabet that encourages a certain expectation of linearity, Coetzee's pseudocode involves forked reading. For example, in the case of the line 'IF(J2-6)206,208,201', the instruction offers a tripartite branching decision: if the output of (J2-6) is negative, return to line 206 (and eventually work



```
                    LINE GENERATOR                          30/05/63        PAGE 1

      DIMENSIONWORD(1000),SENT(5),FACT(6)
C     ALLOWS MAXIMUM OF 1000 WORDS OF 5 TYPES PRODUCING 5-WORD LINES
      A=0.
      FACT(1)=0.
B     BLANKS=606060606060
B     ASTER=545454545454
      DO101I1=1,6
      I2=I2*I1
      I3=I2-1
      READINPUTTAPE5,901,(WORD(I4),I4=I3,I2)
  901 FORMAT(12A6)
B     B=WORD(I3)
      CALLBOOL(B,ASTER,A)
      IF(A)101,102,101
  101 CONTINUE
  102 READINPUTTAPE5,902,NLINES,(FACT(I5),I5=2,6)
C     READ IN NO. OF LINES OUTPUT AND OF WORD TYPES
  902 FORMAT(I4,5F4.0)
      NPRINT=56
      DO201J1=1,NLINES
  202 IF(NPRINT-56)203,204,201
  204 WRITEOUTPUTTAPE6,903
  903 FORMAT(1H1)
      NPRINT=0
  203 J2=1
  206 CALLRNDOM(RNDNO)
      INDEX=FACT(J2)+(FACT(J2+1)-FACT(J2))*RNDNO+1.
B     B=WORD(INDEX)
      CALLBOOL(B,BLANKS,A)
      IF(A)207,206,207
  207 SENT(J2)=WORD(INDEX)
      J2=J2+1
      IF(J2-6)206,208,201
  208 WRITEOUTPUTTAPE6,904,SENT
      NPRINT=NPRINT+1
  904 FORMAT(1H ,A6,1X,A6,3H / ,A6,1X,A6,3H / ,A6)
  201 CONTINUE
      CALLEXIT
      END(1,1,0,0,0,1,1,1,0,0,0,0,0,0,0)
```

Fig. 1.1 Excerpt from 'Line Generator', May 1963, Container Obs 143, J. M. Coetzee Papers. Image courtesy of the Harry Ransom Center, The University of Texas at Austin.

back down to the IF instruction that is part of line 207); if it is equal to zero, go down to line 208; if it is positive, skip to line 201 that will eventually end the programme.[101] Such instructions move the human reader around the programme in a non-linear fashion, and while such syntactical structures do exist in natural language (both English and Afrikaans use clauses), iteration and recursion are more common in programmes designed for computers.

Searching for a way to resist computational thinking then, Coetzee turns to programming's iterative reading as a model to emulate. The expectation of linear reading, with its echoes of Enlightenment theories of history as progress, is undercut in passages such as those recounting Magda's rape—or Klawer's death in 'The Narrative of Jacobus Coetzee'—which require more heterogeneous reading practices. Coetzee's aesthetic automatism might be conceived as autonomy from computational thinking, but it seeks to obtain such autonomy by encumbering the reader with the mental burden of pseudocode. Forcing the human reader to emulate the computer's procedures ironically offers a means through which to explode the logic of computationalism. Reading (whether literary language or pseudocode) is more complicated than computational thinking would suggest, requiring a reader's careful attention. Here we could note a parallel with Derek Attridge's more general argument 'Against Allegory' in *J. M. Coetzee and the Ethics of Reading*: the substitution that forms the basis of computationalism replicates, in significant ways, the substitution inherent in the allegorical readings that Attridge resists.[102] If, as Attridge argues, Coetzee's texts demand of the reader an ethical engagement with otherness, then computational reading (rather than programming operations) offers the obverse.[103]

By insisting on reader engagement with a difficult text, by insisting on the reader's participation in the production of meaning, Coetzee engages with modernist debates around the autonomy of the artwork. Lisa Siraganian has conceived of these debates as focused around the relationship between art object and beholder: the desire to detach the artwork from the reader/spectator and (evident before but popularized after 1960) the contrary desire to incorporate the reader/spectator into the artwork. For Siraganian, these debates take on political overtones: 'the reader's or viewer's relation to the art object became a way to envision the political subject's ideal relation to a changing, rejuvenated, but essentially *liberal* state at a time when the

[101] J. M. Coetzee, 'Line Generator' (30 May 1963), container Obs 143, JMCP.
[102] Attridge, *J. M. Coetzee*, 32–64.
[103] Attridge, *J. M. Coetzee*, xii.

discourse of threatened autonomy pervaded both high and mass culture.'[104] Coetzee's interest in preserving the autonomy of the reader—autonomy from computational thinking—situates him in this modernist tradition, even as the very different environ in which he was writing and publishing placed significant pressure on this platform.

Coetzee's early aesthetic automatism also departed firmly from another legacy of modernism: the language reform projects that found fruition in machine translation's attempts to build a 'new Tower of Anti-Babel' in the post-war years (namely the use of computers to translate natural languages, for more on which, see Chapter 3).[105] While sympathetic to the desire to develop non-referential systems of communication as we struggle with 'a language too protean to be tied down to single, pure meanings', Coetzee was ultimately dismissive of conceptions of computer code that posit it as a 'perfect language'[106]—conceptions that, as Hayles and Golumbia have pointed out, underscore computational thinking. His own work for IBM and ICT, as we have seen, had demonstrated to him the degree to which code was itself a formalization of human cultural assumptions, institutional priorities, and hardware restrictions. Formal (programming) languages such as FORTRAN or Atlas Autocode were far from non-referential systems of perfected communication. Although reliant on algebraic formulae just as much as natural languages, they were shaped by the material circumstances of their design and use. Yet these circumstances, as Coetzee was acutely aware, are frequently ignored. For Coetzee, to speak in code is to speak in 'formulas of language that have hardened and set, that people believe in without question.'[107] His fictional works sought to disassemble the (modernist) myth that code is 'pure language' and the 'hardened and set' cultural assumptions it encodes.

One of the major ways in which Coetzee accomplished this was by insisting that the reader, while occasionally emulating the automaton, is ultimately human. In *In the Heart of the Country*, the act of communicating with machines is still inescapably human, difficult, and material. Throughout the novel, Magda decries the materiality of words that 'alienate'; she desires a 'life

[104] Lisa Siraganian, *Modernism's Other Work: The Art Object's Political Life* (Oxford: Oxford University Press, 2012), 4.

[105] Warren Weaver, 'Translation' [1949] reprinted as 'Foreword: The New Tower', *Machine Translation of Languages: Fourteen Essays*, eds. William N. Locke and A. Donald Booth (Cambridge, MA: The MIT Press, 1955), v–vii, vii.

[106] J. M. Coetzee, 'Isaac Newton and the Ideal of a Transparent Scientific Language' [1982], reprinted in *Doubling the Point*, 181–94, 194.

[107] J. M. Coetzee and Joanna Scott, 'Voice and Trajectory: An Interview with J. M. Coetzee', *Salmagundi: A Quarterly of the Humanities and Social Sciences*, 114–15 (Spring–Summer 1997), 82–102, 84.

unmediated by words', wanting to be instead 'the medium, the median.'[108] While desiring pure communication, Magda, like Eugene Dawn struggling with 'functional words', is constantly confronted with the material status of words and the communicative crux posed by objects.[109] At the end of *In the Heart of the Country*, the isolated Magda is visited by voices that 'speak to me out of machines that fly in the sky'. Declaring that 'They speak to me in Spanish' (which she does not know), Magda argues that the words belong 'not to a local Spanish but to a Spanish of pure' and 'universal meanings.'[110] Although today it is 'Global English' and not the language of the conquistador that dominates, Magda's desire for perfect communication is the desire that drives machine translation, the promise of which Coetzee's text undercuts. Attempting to reply to these machine voices, Magda expends enormous effort collecting, moving, and whitewashing 224 stones to spell out her messages. Recalling the physical limitations of early computing, Magda's 'POEMAs Crepuscularias' is even shortened to CREPUsCLRs' thanks to a scarcity of stones (recall his contractions in his line generation programmes to accommodate bit-length).[111] Meanwhile, the reader is pressed to interpret these cryptic messages, which, while combining various phrases from a mixture of romance languages, are ultimately untranslatable. Far from promoting the 'psychology of the conquistador', this novel insists that 'autonomous fictions' be subjected to human rather than computational reading.[112]

Apartheid Thinking

Coetzee's most explicit articulations concerning his platform of aesthetic automatism appear in the late-apartheid era, after computers have become roundly associated with the regime. In his 1987 talk 'The Novel Today', Coetzee distinguished between a conception of the novel as a supplement to history or its rival. The latter 'autonomous place' is, for Coetzee,

> a novel that operates in terms of its own procedures…a novel that evolves its own paradigms and myths, in the process (and here is the point at

[108] Coetzee, *In the Heart of the Country*, 28, 145, 147.
[109] Coetzee, *Dusklands*, 43.
[110] Coetzee, *In the Heart of the Country*, 137.
[111] Coetzee, *In the Heart of the Country*, 144.
[112] Jean-Luc Godard's influence on Coetzee's aesthetic has been noted by the author himself; Hermann Wittenberg has argued for the influence of Godard's 1965 film *Alphaville*—which features a futuristic police state run by a computer that is ultimately foiled by the input of modernist poetry—on Coetzee while he was drafting the screenplay adaptation of *In the Heart of the Country*, 16.

> which true rivalry, even enmity, perhaps enters the picture) perhaps going so far as to show up the mythic status of the paradigms of history.[113]

Like his earlier experiments with automaton and autonomous reading, here he makes his plea for fiction's autonomy by virtue of its opposition to computational thinking. Alluding to Stanley Fish's discussion of 'message-plus' and 'message-minus' definitions of literary language, he argues that:

> a story is not a message with a covering, a rhetorical or aesthetic covering. It is not a message plus a residue…There is no addition in stories. They are not made up of one thing plus another thing, message plus vehicle… On the keyboard on which they are written, the plus key does not work. There is always a difference; and the difference is not a part, the part left behind after the subtraction. The minus key does not work either: the difference is everything.[114]

Computer hardware might facilitate the writing process, but the thinking that computationalism promotes is nonetheless conceived in opposition to literature: 'Storytelling…is not a way of making messages more—as they say—"effective". Storytelling is another, an other mode of thinking.'[115]

While arguing for literature's 'autonomous place', Coetzee conceives of the writing and reading of literature as defined by their interplay with 'automatism'. Drawing as it does on the Prague school definition of 'automatization', Coetzee's own definition of literary 'automatism' is more closely bound to computation.[116] In *Doubling the Point: Essays and Interviews* (1992), Coetzee articulates his understanding of writing as a process of revelation. Far from a simple act of transcribing what you want to say, in writing 'you have a feel of whether you are getting closer to "it" or not. You have a sensing mechanism, a feedback loop of some kind'. For Coetzee,

> Writing, then, involves an interplay between the push into the future that takes you to the blank page in the first place, and a resistance. Part of that resistance is psychic, but part is also an automatism built into language: the tendency of words to call up other words, to fall into patterns that keep

[113] J. M. Coetzee, 'The Novel Today', *Upstream*, 6.1 (1988), 2–5, 3.
[114] Coetzee, 'Novel Today', 3.
[115] Coetzee, 'Novel Today', 3.
[116] Coetzee, 'First Sentence', 93.

> propagating themselves. Out of that interplay there emerges, if you are lucky, what you recognize or hope to recognize as the true.[117]

The truth that literature can articulate is the product, in part, of resisting automatism. Here the Romantic language of revelation and confession is pitted against a cybernetic understanding of process, which quietly co-mingles modernism's 'writerly detachment and intellectual objectivity' with computationalism. Writing becomes the act of negotiating between the two.

It is in the essays that make up his collection *Giving Offense: Essays on Censorship* (1996), and on which he was working from the mid-1980s onwards, that Coetzee makes the strongest association between writing, computational thinking, and political systems. What Coetzee would decry as 'either/or logic' in *Youth* also influenced his thinking about censorship. While not co-terminous with computation, algebra (and particularly Boolean logic) played a central role in the development of early computing and in Coetzee's own experiences programming in FORTRAN, and it is to 'asymmetrical algebra' that Coetzee turns to articulate the logic behind Geoffrey Cronjé's platform of 'Apartheid Thinking'.

> There is something Cronjé is haunted by. Let me call it simply C. C is the sign of the undifferentiated but also of indifference: indifference to the law, indifference to everything but the urgings of a devouring appetite. Whatever C devours comes out at the other end undifferentiated, that is, as more C. In the algebra of mixing, C + W = C, C + B = C, C + C = C, even B + W = C. Unless there is apartness (Apartheid) asks Cronjé…what can lie ahead but more and more mixing, more and more C?[118]

In expressing 'Apartheid Thinking' in terms of asymmetrical algebraic formulae, Coetzee exposes both the claim to rationalism inhering in Cronjé 'academic-bureaucratic castle' and also the flawed logic it embodies.[119] While his target is broader than the particular deployment of computing by the South African state, Coetzee is interested in dissecting the systems and discourses that, in the later twentieth century, computational thinking would help to buttress.

[117] Coetzee, *Doubling the Point*, 18.
[118] J. M. Coetzee, 'Apartheid Thinking', *Giving Offense: Essays on Censorship* (Chicago: University of Chicago Press, 1996), 163–84, 176–7.
[119] Coetzee, 'Apartheid Thinking', 165.

In another essay in the volume, 'The Work of the Censor: Censorship in South Africa', Coetzee makes this parallel more explicit. Discussing the paranoia that characterizes the censor's thinking, he argues that 'the entry into paranoia is an entry into an automatism, of which there can be no clearer illustration than the fact that, of all the pathologies, paranoia has been the most amenable to artificial stimulation'. He presses the point by noting that a 'computer program embodying a system of "beliefs" to be protected and a set of defense mechanisms for protecting those beliefs has simulated the discourse of paranoid patient and therapist convincingly enough to pass Turing Tests'. (In his 1950 paper 'Computer Machinery and Intelligence', Alan Turing famously substituted the question of 'can machines think?' with the problem of whether a human interrogator could differentiate between a computer and a person via technologically mediated conversation. In so doing, he promoted the idea that conversation is a uniquely human mode of interaction, yet one that might be rendered programmable in the future. That test has become a controversial measure of progress in artificial intelligence, fuelling Weizenbaum's ELIZA project and, in subsequence concerns and, in the instance Coetzee cites, had seemingly come to pass.)[120] As was the case with apartheid thinking, here the paranoid thinking embodied in censorship offers 'pseudo rationality'.[121] The ease with which such flawed reasoning is simulated by computers is, for Coetzee, an indication of the degree to which computational thinking can perpetuate the worst features of state control. As the narrator of *Youth* will later comment, 'he has no respect for any version of thinking that can be embodied in a computer's circuitry'.[122]

What Neil Lazarus would call Coetzee's 'ethical rationalism' and what I explicitly term his aesthetic automatism places the burden of interpretation on the reader.[123] Coetzee's fiction demands a version of thinking embodied not in circuitry or computationalism, but in a human reader. In his second fiction, Magda's tales, spun 'out of my own bowels', call forth this reader just as much as the machine messages;[124] in *Waiting for the Barbarians* (1980),

[120] It is a curiously unremarked fact that a humanist model lies at the heart of our computational endeavour. A. M. Turing, 'Computing Machinery and Intelligence', *Mind*, 59.236 (1950), 433–60.

[121] J. M. Coetzee, 'The Work of the Censor: Censorship in South Africa', *Giving Offense*, 185–203, 200–1.

[122] Coetzee, *Youth*, 149.

[123] Neil Lazarus, 'Modernism and Modernity: T. W. Adorno and Contemporary White South African Literature', *Cultural Critique*, 5 (Winter 1986–1987), 131–55, 136.

[124] Coetzee, *In the Heart of the Country*, 150.

the Magistrate offers a (false) reading of wooden slips when under threat of bodily torture;[125] and *Life & Times of Michael K* (1983) is a novel deeply concerned with literature's potentially resistive function against the broad politics of computational thinking and its biometric deployment by the state.[126] Such focus continues in his later works: David Lurie's refusal to confess in *Disgrace* (1999) might easily be read as demonstrative of a continued anxiety about the efficacy of narrative in a society still governed by a biometric state (an opposition to which the *Jesus* trilogy returns). So too in his drafts for *The Lives of Animals* (1999), Coetzee would point to computational thinking as facilitating a false moral distinction drawn between humans and animals: better would be a definition that turns on 'sensation of being', 'not in consciousness of yourself as a kind of ghostly reasoning machine thinking thoughts but on the contrary as in the sensation—a heavy affective sensation of having limbs having some kind of extension in space'.[127] In both early and later works, Coetzee's writing insists on the embodied nature of human reading as a critical aspect of its function within a culture dominated by computational thinking. It is in *Youth*, however, that Coetzee most overtly engages with programming and computer culture, and it is to the problem of *Youth* that I now, finally, want to return.

Programmed *Youth*

As an account of the few years in which Coetzee was employed in the computer industry, *Youth* stays fairly close to the historical record. Yet, as I have been demonstrating, Coetzee's intellectual engagement with computation extended across decades and countries and shaped his literary project. Moreover, this platform of aesthetic automatism was a rejoinder to the specific political import of computational thinking within the context of late apartheid South Africa. The book, however, works hard to downplay the longer legacies of this period. In the final chapter, John is at pains to deny that this period has been marked by any kind of development: 'Everything he has done since he stepped ashore at Southampton has been a killing of

[125] J. M. Coetzee, *Waiting for the Barbarians* (London: Penguin, 1980), 108.

[126] Rebecca Roach, 'Hero and Bad Motherland: J. M. Coetzee's Computational Critique', *Contemporary Literature*, 59.1 (Spring 2018), 80–111.

[127] J. M. Coetzee, Draft of *The Lives of Animals* (7 July 1997). The Lives of Animals, Handwritten Draft with Revisions and Notes (18 June–14 August 1997), Box 30, Folder 2, JMCP, 21.

time while he waits for his destiny to arrive.'[128] So too, the book seems to dismiss the troubling imbrication of computation and apartheid with John's suggestion that 'There are no computers in South Africa that he knows of'.[129] What should we make of this?

On first glance, this does present a challenge. More than failing to overtly acknowledge Coetzee's employers' role in buttressing the apartheid regime, the book apparently belittles the link. Yet on closer reading, by acknowledging a feature of the work that machine reading has historically struggled to detect, we see a different picture. What is striking about *Youth* is its tone. This is a bildungsroman dripping with irony, as many reviewers have noted, and one that, like Coetzee's earlier work, demands a human—rather than an automaton—reader.

Youth consistently undermines its protagonist's aspirations and assessments with irony; it is often the small statement or observation that proves most devastating. Thus, Coetzee has John reflect on his own imbrication in an industry closely connected to the military–industrial complex, but the acknowledgement is undercut by its own limitations. John notes that at IBM:

> In the most incidental, the most minor way, he has joined the TSR-2 project, become part of the British defence effort; he has furthered British plans to bomb Moscow. Is this what he came to England for: to participate in evil, an evil in which there is no reward, not even the most imaginary? Where is the romance in staying up all night so that Mr Pomfret the aeronautical engineer, with his soft and rather helpless air and his suitcase full of cares, can catch the first train north so as to get to the lab in time for his Friday-morning meeting?[130]

Both overblown and ill-considered, this is a line of reflection that goes nowhere, or rather arrives at the wrong destination. The question of personal responsibility is raised only to be lost in a mental sequence that moves rapidly through rewards, romance, and then a Friday-morning meeting. While offering comment on the banality of evil, the main target of this passage is John, whose claims to mature self-reflection come unstuck for a reader deeply aware of their limits.

Later in the novel, John figures himself as an 'accomplice in the Cold War, and on the wrong side too'.[131] Here again Coetzee draws the reader's

[128] Coetzee, *Youth*, 165.
[129] Coetzee, *Youth*, 44.
[130] Coetzee, *Youth*, 82–3.
[131] Coetzee, *Youth*, 163.

attention to the wider politics of computer work but in doing so sardonically underlines John's immature internal dialogue. The sentence that follows this remark switches from the global to the personal: 'Tests no longer seem to come with fair warning these days, as they did when he was a schoolboy.' By representing such immature sequences of thoughts, Coetzee demands that the reader maintain ironic detachment from John. In these moments, the reader is asked to consider the limits of John's thinking. When John offers the comment that there are no computers in South Africa, the careful reader understands that such a remark should not be taken at face value. To read *Youth* simply as dismissing the political ramifications of computation is to read as a machine might do.

In the 2016 piece 'On Literary Thinking', Coetzee would make his case explicitly:

> May God us keep/From single vision and Newton's sleep', wrote William Blake. I am not sure that Blake is being fair to Newton here, but if God will not keep our children from the single vision of YES or NO, then it is up to the poets to do so.[132]

Computation is deployed across Coetzee's work not to black box or demand faith, but to underline the degree to which autonomy—whether that of the universal machine or the poem—is a construction. Anticipating the imbrication of computationalism and 'Apartheid Thinking' in late-apartheid South Africa, Coetzee's evolving platform of aesthetic automatism enabled him to conceive of literature as both autonomous (from computing) and politically responsible. It was a relation that some of his peers would explore to very different ends.

Programming Literature. Rebecca Roach, Oxford University Press. © Rebecca Roach 2025.
DOI: 10.1093/9780198976578.003.0002

[132] J. M. Coetzee, 'On Literary Thinking', *Textual Practice*, 30.7 (2016), 1151–2, 1152.

2
Hugh Kenner's Hobby

Hugh Kenner was a giant of twentieth-century Anglo-American literary criticism. *The Pound Era* (1971), his magnum opus, is often held up as one of the defining texts in the 'invention of literary modernism' (or a certain branch of it anyway[1]), and across his career, he'd publish over twenty critical volumes on Ezra Pound, T. S. Eliot, Samuel Beckett, James Joyce, and others.[2] Canadian by birth, he would complete his early education at the University of Toronto with media theorist Marshall McLuhan, before moving to Yale to pursue a PhD under renowned New Critic Cleanth Brooks. Early on, Kenner developed a passion for the work of Pound, from which would blossom a celebrated career as a professor of literature; Kenner taught at institutions across the United States. Less well-known—if hinted at in his association with the father of media studies—Kenner was also a critic deeply interested in computing, its intellectual roots, and cultural ramifications.

In this chapter, I want to examine Kenner's activities outside of his professional commitment to literary studies, namely his involvement in 'hobbyist' communities. 'Hobbyist' was a term deployed to describe computer enthusiasts, often without a professional connection to the computer industry or discipline, whose community efforts would nevertheless play a significant role in the development of personal (as opposed to mainframe)

[1] Kenner's interest is in 'International Modernism'. Associated with Eliot, Pound, Joyce, and Beckett, the literature of this name in his eyes is characterized both by a time period (*c.*1910–1930) and by these works' resistance to categorization as national literature: 'they define a tradition of their own, accessible to whoever will master the English necessary to read them; or, if skill with English be native, the detachment. What you must be detached from is the set of expectations you acquire in absorbing a national literature.' Hugh Kenner, *A Colder Eye: The Modern Irish Writers* (New York: Alfred A. Knopf, 1983), 15. In discussing his own role in canonizing 'International Modernism', Kenner asserts that 'It was in 1947, under Marshall McLuhan's informal tutelage, that I first became aware of my own century. Such a lag was perhaps possible only in Canada', Hugh Kenner, 'The Making of the Modernist Canon', *Chicago Review*, 34.2 (Spring 1984), 49–61, 57.

[2] As Marjorie Perloff notes, such an invention is 'Not everyone's modernism: highly selective in his enthusiasms, Kenner slighted women poets (especially Gertrude Stein) and minority writers.' Marjorie Perloff, 'Reading Modernism, After Hugh Kenner (1923–2003): Hugh Kenner and the Invention of Modernism', *Modernism/modernity*, 12.3 (September 2005), 465–9, 466.

computing—what would become known as the 'PC Revolution' of the late 1970s and 1980s. What Kenner and his associates did as hobbyists would have profound consequences for computer innovation and via the frequent binary thinking that, as we saw, surrounds relations between literature and tech, and it is salutary to realize that at least one literary critic was breaking down such stereotypes more than seventy years ago.

I also want to close read Hugh Kenner's hobby, because to do so is to reshape our understanding of his own literary critical project. What we see, when we examine these activities, is an insistence on the equivalence of the computer programme and the literary text; the equivalence between the cultures that surround computing and (modernist) literary studies and the kinds of reading protocols that need to be deployed in each context. For Kenner, the analogy between computing and literature is often a critical and pedagogical one. It is a question of the methods that the reader (and the user is usually constructed as a reader in Kenner's eyes) employs.

This recognition requires a rethink of Kenner's position within literary critical history and his legacy. Known for his 'customary wit and erudition,'[3] for his 'anecdote and example,'[4] Kenner's critical style seems both antithetical to the linear and syntagmatic mode of reading we might (before Coetzee) ascribe to a programme *and* opens him up to accusations of intellectual elitism—accusations that are compounded by his circulation in conservative intellectual circuits and promotion of right-wing writers. Not wanting to downplay those characteristics, I nevertheless want to note a rather different strain of Kenner's career and legacy. By focusing on his hobby and the pedagogical model he develops while practicing it, we find a public intellectual heavily committed to facilitating broad access to subjects perceived to be difficult, whether high modernism or computing. We find in Kenner a celebrator of the non-expert, the amateur, and the work that can be done outside of the formal venue and on the edges of disciplines. Granted, Kenner is not known as a democratizing force (if anything it is his conservative political leanings that have been the focus of scholars[5]), but

[3] Hugh Kenner, *The Mechanic Muse* (New York: Oxford University Press, 1987), back blurb.
[4] Kenner, *Colder Eye*, jacket blurb.
[5] Evan Kindley's excellent article argues that 'Kenner's commitments to literary modernism, technological innovation, and political conservatism, disparate as they may appear to be, are all closely connected. And what connects them is the network', Evan Kindley, 'Kenner's Networks', *Critical Inquiry*, 47.3 (2021), 525–43. While I diverge from Kindley in prioritizing computing over networks, his analysis is extremely fruitful. See also Stephen Schryer, 'Conservative Circuits: Hugh Kenner, Modernism, and *National Review*', *Modernism/modernity*, 26.3 (2019), 505–20. doi: 10.1353/mod.2019.0040.

what we see if we look across his endeavours is the degree to which he deploys his own experience to promote a practice that benefits the uninitiated user–reader.[6] For Kenner, the salutary relation between literature and computing is not Coetzee's binaries and automatism but rather (as perceived by general readers) their shared difficulty, difficulty that might be overcome via a hobbyist approach.

Computation at the Day Job

Computing is a common topic in Kenner's literary criticism once you start looking. Like Coetzee, this interest seems to have stemmed from contemplating possible relations between mathematics, formal logic, and literature. While Coetzee was dividing his time between IBM's London office and the British Museum Reading Room, Kenner was pursuing these connections in less grey conditions, on the campus of the University of California, Santa Barbara, where he taught.

The year 1962 would see the publication of two computer-engaged works. Kenner's influential essay, 'Art in a Closed Field,' discussed a contemporaneous text generator, Auto-Beatnik, which was getting a lot of public attention. Making the point that 'creative writers and creative mathematicians have been exploring comparable processes, of which the engineers' Auto-Beatnik is merely a late and novel by product,' Kenner argued that much (modernist) literature might be helpfully understood as engaging with ideas 'analogous'—his word—to set theory (the mathematical study of collections of objects).[7] Indeed, 'one can multiply without effort out of the literature and criticism of this century example after example of the habit of regarding works of art as patterns gotten by selecting elements from a closed set and then arranging them inside a closed field' wherein '[t]he closed field contains a finite number of elements to be combined according to fixed rules.'[8] Thus, in Kenner's assessment, the vocabulary of *Ulysses* comprises a

[6] In the introduction to their collection on the topic, Aarthi Vadde and Saikat Majumdar note the importance of amateurism in the history of literary criticism while flagging that '"the amateur" is not a uniform position so much as an assortment of perspectives on the practice and purpose of criticism', Aarthi Vadde with Saikat Majumdar, 'Introduction: "Criticism for the Whole Person"' in *The Critic as Amateur,* eds. Majumdar and Aarthi Vadde (London: Bloomsbury Academic, 2020), 1–28, 2.

[7] Hugh Kenner, 'Art in a Closed Field', *The Virginia Quarterly Review*, 38.4 (Autumn 1962), 597–613, 597, 608.

[8] Kenner, 'Art in a Closed Field', 599, 600.

closed set that Joyce 'compiled' and on which he imposed certain 'procedures'.[9] The *Cantos*, meanwhile, is a project invested in examining the closed set of the university curriculum. So too, Beckett's *Watt* plays with all possible combinations within a linguistic set. The text generator, in Kenner's view, is merely the visible symbol of a much broader trend of writers thinking analogously with mathematicians.

The article offers a pithy treatment of points Kenner would expand for his book, *The Stoic Comedians: Flaubert, Joyce and Beckett* (1962). Dedicated to media theorist Walter J. Ong, the book places this contemporary literary interest in permutational writing within a broader historical framework. Kenner goes back to the emergence of the novel, the encyclopaedia, the printing press, and empirical science to argue for his three writers as signifying a tradition of 'the novel as Knowing Machine, lifelike, logical'.[10] Each writer is designated a 'Stoic Comedian' by virtue of their mathematical approach to the novel form: 'he selects elements from a closed set, and then arranges them inside a closed field'.[11] If Flaubert is the comedian of the Enlightenment, 'confronting fiction's gargantuan appetite for fact, for detail, for documentation', and Joyce of the inventory, exploiting permutation, while implying 'that the limits of permutation are [the book's] real binding form', then Beckett is the comedian of the impasse, revelling in 'the closed system of possibilities arrived at, amid given presupposed data, by logical analysis'.[12] These writers, Kenner suggests, engage with the 'dominant analogy' of the contemporary: 'proper to the world of IBM, of probability theory… It is a world to which none of us needs to be introduced; we live in it rather comfortably, handling punched cards and buying insurance policies.'[13] In the early 1960s, Kenner enthusiastically promotes a relation between literature and 'General Number Theory': a theory in which IBM and punched cards feature, if unremarkably.[14]

Four years later—consequential years in his personal life in which his first wife died, and he remarried—Kenner would publish a follow-up of sorts. *The Counterfeiters: An Historical Comedy* (1968) is more explicit about the technology. Here Kenner's attention is caught by an analogy in which literature and the universal Turing machine are two in a broader cultural tradition (also comprising: Pope, Swift, Charles Babbage, Buster Keaton films,

[9] Kenner, 'Art in a Closed Field', 598, 599.
[10] Hugh Kenner, *The Stoic Comedians: Flaubert, Joyce and Beckett* (London: W. H. Allen, 1964), xix.
[11] Kenner, *Stoic Comedians*, 94.
[12] Kenner, *Stoic Comedians*, 70, 73, 81.
[13] Kenner, *Stoic Comedians*, 96.
[14] Kenner, *Stoic Comedians*, 96.

expository prose, clockwork ducks, and Andy Warhol) of 'Counterfeitable Man.'[15] Kenner traces a lineage from *Gulliver's Travels* to Turing's famed test of computer intelligence (from the Houyhnhnms' language to FORTRAN), viewing this lineage as a shared attempt 'to define a thing—a clerk, a watch, a man—by function.'[16] More specifically, he suggests that satire shares with 'mechanism' the principle that 'you can be said to understand a process thoroughly if you know how to instruct a machine to carry it out.'[17]

In this expansive text, Kenner offers the universal Turing machine as the 'aptest of counterfeiters' in its ability to simulate, and the Turing test as the application of the logic of counterfeiting to the human subject. However, compared to Swift's satire, the computer pioneer's test 'seems almost trivial.'[18] Kenner moves, across these publications, from a General Number Theory to particular computing concepts—Turing's machine and test—as he seeks to identify analogies. Relatively early in his career, and in computing's commercialization as an industry and professionalization as a discipline, Kenner conceives of computing and literature within the same broad conceptual framework. But if Kenner was enthusiastic—he also wrote volumes on the inventor of the geodesic dome Buckminster Fuller (1973) and *Geodesic Math and How to Use It* (1976)—his contemporary readers were more circumspect. As my whistle-stop descriptions might suggest, these are almost unfathomably wide-ranging books, moving from close reading of computer pioneer Charles Babbage's papers, to broad discussions of Royal Society rhetoric, back to Socrates and forward to Pop Art. They demand much of the reader. *The Counterfeiters*, which dwells heavily on the ideas of Babbage and Turing, was greeted by critics with 'respectful incomprehension.'[19] This was 'Intellectual history by firefly-flashes' and 'Nijinsky-like' leaps.[20] Described kindly, Kenner 'assumes that his readers are intelligent. And that they are imaginative. And that they are interested in literature, as well as its place in the world.'[21] Described less kindly, for all its whimsy and specificity of

[15] Hugh Kenner, *The Counterfeiters: An Historical Comedy* (Bloomington: Indiana University Press, 1968), 1. 'Man' is Kenner's term.

[16] Kenner, *Counterfeiters*, 6.

[17] Kenner, *Counterfeiters*, 12.

[18] Kenner, *Counterfeiters*, 107, 121.

[19] Roger Sale, '"The Art of Hugh Kenner". Review of Hugh Kenner, *The Counterfeiters*', *The Hudson Review*, 22.3 (Autumn 1969), 521–7, 521. The review is notable for largely ignoring Kenner's sustained discussion of Babbage and Turing.

[20] William Kinsley, 'Untitled Review of *The Counterfeiters*', *Eighteenth-Century Studies*, 4.4 (Summer 1971), 494–6, 494; and Sales, 'Art of Hugh Kenner', 523.

[21] Guy Davenport, 'Hugh Kenner, X-Ray Technician', The Guest Word column, *The New York Times* (26 March 1972), Section BR, 47, https://www.nytimes.com/1972/03/26/archives/hugh-kenner-xray-technician.html. Davenport collaborated with Kenner to provide the

language, Kenner's early attempts to engage with computing in his literary criticism are simply too difficult.

This didn't mean that Kenner would stop trying. His oeuvre (and archive) is abundant with references to computing technologies: viewed collectively, his professional activity becomes legible, I suggest, as a humanities computing project as much as a literary critical one. Yet his later efforts to write about computing would be marked by an enthusiasm for engaging with the practicalities of computing as a technical, rather than theoretical, activity. Less General Number Theory and more hacks. While this might imply a further step towards difficulty, in fact the opposite was the case. Kenner's extracurricular interests would affect his professional project as he sought, increasingly, to support the floundering reader in both spheres. It is to his hobby that we now turn.

Kenner the Hobbyist

Hugh Kenner was a Calcunut. Profiling the 'growing subculture' for the *Wall Street Journal* in 1981, Kenner identified Calcunuts as 'the programmable-calculator fanatics whose joy is the use of the machine itself and whose special delight is to drive it to lengths its designers never imagined.'[22] While programmable calculators and the community that blossomed around them might seem idiosyncratic today, these hobbyists and others like them were important to the emergence of the personal computer (PC). Historically, as we saw in Chapter 1, the computer sector had been oriented around expensive, large, mainframe devices that were usually located in corporate and institutional settings, accessible only by experts or through professional programmer support—the IBM office or University of Cambridge Mathematics Lab of Coetzee's early career. The PC revolution brought about a huge sector shift: towards the design and production of low-cost, interactive devices that were aimed at individual users.[23] The

illustrations for both volumes. The images work similarly to those utilized by McLuhan in his work: less documentary than expository, despite Kenner's wry claim that 'The illustrations are intended to keep the reader from dwelling on the paucity of documentation' (Author's Note, *The Stoic Comedians*). It is a claim that the reader is encouraged to return after reading the conclusion to this book.

[22] Hugh Kenner, 'Calcunuts', *The Wall Street Journal Magazine* (June 1981), 52–60, 53.

[23] The technological transition from mainframe to personal computing was more gradual than this account suggests, glossing over as it does the history of mini- and microcomputers. What was dramatic was the massive expansion in the market for personal computing—and

Calcunuts helped, as computing scholar Paul Ceruzzi notes, to develop the market for logic chips that undergirded this revolution, while also 'unleash[ing] the force of personal creativity and energy of masses of individuals' to develop a consumer market.[24] Certainly, this latter aspect is what Kenner focuses on: the Calcunuts he profiles scale Everest-like heights in their heroic attempts to programme with the calculators—'harpooning Moby Dick with a peashooter' as he puts it.[25]

Calcunuts were not only creative individualists. The descendants of ham radio enthusiasts, they were collectively a driving force behind the PC revolution. They were more numerous than the Homebrew Computer Club, their better-known counterparts (whose members included Apple's founders).[26] There were 25,000 owners of HP-65 programmable calculators in the USA by 1975. They also differed from the stereotypical countercultural hacker, by being mostly professional men (architects, engineers, and actuaries) with a need for calculators in their day job and who could afford the $795 price tag. The portability of the programmable calculator also meant that users could experiment with it outside the office, bringing the possibility of the personal machine into the home.

Although Kenner keeps himself out of the *Wall Street Journal* piece, this was a community in which he had been formally involved since 1974 and informally for longer—a Calcunut wrote *The Pound Era*. An early member of the group, Kenner tinkered with his equipment and was a regular contributor to the monthly newsletter, *PPC Journal*, and other similar DIY (sometimes mimeographed) titles, such as *Sextant*. Publications such as these offered a means by which user knowledge could be shared and feedback solicited, while testifying to a broader community of enthusiastic amateurs who viewed their endeavours collaboratively.[27] Kenner was a more

price shifts—that took place in the late 1970s and early 1980s. See Paul E. Ceruzzi, *A History of Modern Computing* (2nd edn, Cambridge, MA: MIT Press, 2003), Chapter 7.

[24] Ceruzzi, *History of Modern Computing*, 214–15.

[25] Kenner, 'Calcunuts', 53.

[26] For the history of the Homebrew Computer Club and its role in the emergence of both the PC market and the libertarian culture of Silicon Valley, see e.g. Fred Turner, *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism* (Chicago: University of Chicago Press, 2006); and Elizabeth Petrick, 'Imagining the Personal Computer: Conceptualizations of the Homebrew Computer Club 1975–1977', *IEEE Annals of the History of Computing*, 39.4 (2017), 27–39. See also Joy Lisi Rankin, *A People's Histories of Computing in the United States* (Cambridge, MA: Harvard University Press, 2018) for a discussion of the influence of student contributions to the history of computer innovation and culture.

[27] Eric Rechlin's website HPCalc contains a wealth of resources on the technology. The website is https://www.hpcalc.org/. The YouTube channel also offers fascinating recordings of various conference talks, including 'HHC 2021: What We Started (Richard Nelson)'; by PPC founder Richard Nelson. https://www.youtube.com/watch?v=_rfjxqxOkJA&ab_channel=hpcalc.org.

energetic contributor than many, offering 'user reviews' on, for example, celestial navigation programmes and word processing software. He provided his peers with short programming routines in Microsoft BASIC programming dialect to calculate mathematical functions, or a programme to calculate the date of Easter for any given year.[28]

What is notable, in addition to the articles themselves, is the community structures they assume. With his early contributions, Kenner presents himself as an amateur in conversation with his peers. He regularly notes difficulties he is having with executing code—and is invited to chapter meetings by others to discuss issues and solutions. He solicits the advice of peers in the pages of the newsletters, or via personal correspondence, assuming a shared and amateur enthusiasm. This is writing with the aim of facilitating interaction amongst peers as a part of a collective hobbyist endeavour. As his participation in these communities continues, Kenner adopts a more confident persona that better reflects his professional pedagogical identity, but that also creates a different relationship with readers. He authors *Beginners' Corner* columns, explanatory articles—such as 'What You Do When You Program'[29]—and advice pieces. These writings are introductory, stylistically informal, and aim to provide a point of entry for the inexpert or uninitiated, a rapidly expanding segment of the community.

It is fascinating the degree to which Kenner's writings attempt to recreate this hobbyist experience while insisting on the importance of user interactivity with the programme and (Knuth would no doubt approve) any associated documentation or literature. Discussing efficient programming for example, Kenner provides information around code, but his focus is on establishing practices in the user–reader that will enable them to be better and independent learners. Thus, Kenner breaks programming into a series of questions that an individual should ask herself:

> You need to be resourceful in testing. Does it *always* work? Over the conceivable range of inputs, including (if applicable) zero and negative numbers?
>
> …
>
> Is the program idiot-proof?
>
> …
>
> Did you document it, by the way?'[30]

[28] Hugh Kenner, 'More Double-Precision Trig Functions', *Sextant* (Winter 1984), 98.
[29] Hugh Kenner, 'What You Do When You Program', *The Wall Street Journal Magazine* (June 1981), 60.
[30] Hugh Kenner, 'Saving Steps', *PPC Journal*, 5.5 (June 1978), 9–11, 10.

Kenner's method—focused on process rather than content—is pedagogically sound, as we might hope of an educator, but is also suggestive in its almost algorithmic style and emphasis on interaction on the part of the user–reader.

Such emphasis on interactivity is, as I mentioned, partly a reflection of the hobbyist community ethos. But it is also a response to the particular programming environment that developed in tandem with it. With the programmable calculator (and later personal computers), lines of code could be typed into the device, edited, and deleted. Programming here is not the batch processing of punched cards that Coetzee grappled with, but rather an example of a command-line interface—the user types commands to initiate actions. This style of both programming and interface was often described as 'conversational' for its perceived interactivity and the comparative inexpertise it demanded of a user.[31] What is notable here is that Kenner replicates the interactivity of the user's experience in programming and interface in the documentation reading process. The user–reader (always assumed to be so) is met with a series of questions that they should ask themselves, or a set of procedures for approaching program design. Kenner structures his text so that it mirrors hobbyist programming practice—the text requires interactive engagement.

Crucially, Kenner positions his own knowledge as the product of having adopted such methods himself. While his experience is tacitly indicated via his authoring of the column and his long-standing membership of the group (member no. 103), he is quick to draw lines of comparison between his own learning and those of his reader. Kenner follows the query around documentation, quoted above, with an admission:

> It is incredible how quickly such things are forgotten. I have an unidentifiable 152-step program on which I must have spent some time, but having failed to make even a rudimentary note I now can't even remember what it's supposed to do. I will happily send a copy to anyone who thinks he can think of a use for it. It had a use, once.[32]

This is salutary advice from the self-taught expert. Kenner is far from the ivory-towered professor in his writing on computing; rather, his authority

[31] I have discussed this more extensively in Rebecca Roach, 'Expanding the History of Humanities Computing: Hugh Kenner's Conversational Criticism', *Digital Scholarship in the Humanities*, 38.2 (June 2023), 679–95. doi: 10.1093/llc/fqac066.

[32] Kenner, 'Saving Steps', 10.

stems from his own practice and self-questioning, from an interactivity spurred by a hobbyist community and programming environment.[33]

Were we to question whether Kenner's hobby was having any impact on his literary criticism, it is useful to juxtapose another of his statements from this period:

> languages like FORTRAN and BASIC now fill the mediaeval/Renaissance role of Latin as the logically-structured idiom into which we convert the intelligence we deem most immutably important, and that in such a domain programmers' concepts of elegance and economy are in fact stylistic criteria. A program has two readers: (1) another programmer, who can admire it; (2) an ideally accurate machine with no tolerance for ambiguity, which can only execute it, but responds to elegance by executing in minimum time, i.e. at minimum cost. Low execution cost is thus the machine's mode of appreciation.[34]

In this 1978 correspondence, later reworked (minimally) for a volume of essays aimed at humanities scholars, Kenner not only positions computer code within a long history of Western philology—recalling the general argument from *The Stoic Comedians* and *The Counterfeiters*—but also grants it qualities we traditionally associate with literary language, namely, style and rhetoric. Five years before Knuth's 'Literate Programming' will be published, Kenner is making the same point: a programme can be aesthetically pleasing and, by association, amenable to the techniques of literary criticism. In fact, Kenner takes it further than Knuth, suggesting that the machine can be conceived as a reader capable of exhibiting 'appreciation' in the same way that a human reader can admire 'elegance and economy'. For Kenner the programmer, such activity is entirely conceivable within the world-view of

[33] In 1959, the young literary critic and hobbyist would respond to a question posed by the editors of the *National Review*: 'Is the teacher…left with anything to do that cannot be done better by [television] technology?'. He insisted that 'the function of the teacher of literature is to transpose into dialogue, assist his students to transpose into dialogue, the essentially solitary task of communing with a book'. While we might read this as an argument born of technological hostility, Kenner is declaring that interactivity and initiation—the features he promotes in his hobbyist writings—are a central feature of literary appreciation. Hugh Kenner, 'Automated Out?', *National Review* (10 October 1959), 389–90, 389.

[34] Hugh Kenner to Bill [William J. McClung?] (24 May 1978). In "'Machinespeak', The Language Now, 1980'" Box 32 Folder 4, Hugh Kenner Papers, Harry Ransom Center, The University of Texas at Austin. Hereafter 'HKP'. These ideas would find their way into print in Hugh Kenner's book chapter 'Machinespeak', in *The State of the Language*, eds. Leonard Michaels and Christopher Ricks (Berkeley: University of California Press, 1980).

literary criticism. For Kenner the Calcunut, the explanation, with its explicit acknowledgement of user–readers and its almost algorithmic expository style, is one crafted by a writer whose understanding of writing and reading has been shaped by his hobby.

Kenner the Technical Writer

Kenner's championing of the hobbyist–reader would reach new heights in the mid-1980s when he embarked on his biggest project, the *Heath/Zenith Z-100 User's Guide*. Here was a manual that attempted to incorporate the practices and world-view facilitated by the hobbyist community into the experience of reading the book itself. The manual was published in 1984 by Brady Communications Company, specialists in books and software for the PC market. Heath (Heath-Zenith after 1979) produced computer kits for assembly by the purchaser and was part of the early wave of hobbyist personal computing that included the now legendary Altair 8800. Kenner had purchased his first Heathkit back in 1955 when their offering included radio transmitters, oscillators, and alike.[35] The computer model for which he wrote the user manual was the Z-100, innovative in its video offerings and ability to run two popular operating systems.

The impetus behind the user guide was Kenner's and born of his desire to support his less computer-literate friends and colleagues in their attempts to self-educate themselves in the obscure world of word processing. Kenner was, by this point, a keen proponent of word processing software and the potential it offered to literary scholars. For the editor of, for example, Pound's *Cantos*, the ability to easily manipulate text layout was invaluable: 'By normal scissors-and-paste methods the job would take more years than I can look forward to.'[36] With the hobbyist's enthusiasm and the professional critic's confidence, Kenner wrote directly to Heath's marketing directors, outlining what he saw as the inadequacies in current support for novice PC users. Drawing on his own experience, he outlines the problem:

[35] As promoted in Heath Company, 'Heath Winter Flyer 1955'. http://www.pestingers.net/pdfs/catalogs/heathkit-winter-flyer-1955.pdf.

[36] Hugh Kenner to W. H. Huggins (17 June 1981), in 'Heath/Zenith Z-100 User's Guide (1984), Correspondence and Contracts, 1980–1986, n.d.', Box 16 Folder 8, HKP. Note the degree to which word processing adopted metaphorical uses of paper-based editing strategies.

> I assembled the hardware without the slightest difficulty, which from past experience was no more than I expected. I then had (1) a chunk of functioning hardware which would not do anything useful; (2) your manual #595-2268-04, the relevant portion of which begins (page 6-1), 'this section of the manual is for those who are experienced in using computers.'[37]

Kenner's point is that the manual assumes a great deal of knowledge that the user does not necessarily have and also lacks orientation for the reader. While the hobbyist community had provided the infrastructure necessary to enable early users to benefit from their interactions with the machine, the new wave of novice, uninitiated users requires this guidance within the book. Kenner's broad suggestion might be familiar from college composition classes: 'A few simple guideposts…—about SAVE, about SCRATCH, about OLD—would have saved me hours of page-turning.'[38] Good writing provides signposting, and it anticipates the questions with which the reader approaches the text and offers answers. It considers, in effect, reader experience, initiation, and likely interactions. In a move that again reminds us of his educator day job and hobbyist writings, Kenner follows this point with the imperative, 'So let's outline something better…', and spends the following pages suggesting an alternative opening for Heath's manual, complete with signposts and clarifications.

Kenner's letter takes care to emphasize his long-time use and admiration of Heath products; nevertheless, it ends with something of a torpedo, or what he describes as a 'concrete example':

> Today's mail brings a letter from a former colleague, now at Stanford, who notes that I have a Word Processor.… Now what shall I tell him?… This man is not technically oriented. Once he unpacks these [Heath products], what help will he receive toward getting them to do what he wants… [?] You are deep in the consumer market now. Exercise: <u>You</u> draft <u>me</u> a "Dear Don" letter to send him.[39]

Kenner's candid assessment and suggestions seem to have ruffled feathers. In part, this is perhaps because he correctly saw the writing on the wall—personal computing was increasingly a consumer market that Heath/Zenith

[37] Hugh Kenner to Coy Clement (10 November 1981). In 'Heath/Zenith Z-100 User's Guide (1984), Correspondence and Contracts, 1980–1986, n.d.', Box 16 Folder 8, HKP, 1.
[38] Hugh Kenner to Coy Clement, 3.
[39] Hugh Kenner to Coy Clement, 4.

was ill set up to serve. (Poor customer service on the part of computer companies was clearly a bugbear—Kenner wrote an article describing his poor 'interface' with IBM for *National Review*.[40]) But partly too his didactic final suggestion seems to have been taken as read rather than infused with the chuckle that Kenner's 'Cordially' sign off might imply.

Certainly, in subsequent correspondence, Kenner had to assure Heath management not only that his suggestions were born of a desire to help the company but also that he appreciated that hands-on interactive experience was an important corollary to 'reading and getting advice' when learning about computers.[41] Despite his imbrication in DIY computing culture, his status as an academic and professional book reader became the means by which the company questioned the validity of his intervention: the implication was that such expertise was not relevant to computing culture and did not travel across domains. The suggestion of adopting a hobbyist, reader-oriented approach to documentation was a step too far for the company.

Unsurprisingly, Kenner's correspondence with Heath/Zenith personnel did not result in the changes to their manuals that he suggested. The latter did not collaborate with Kenner to author the *Heath/Zenith Z-100 User Guide*; Kenner went solo. Although 'Zenith Data Systems loaned both hardware and software and gave [Kenner] as much information as company policy permits', the book was published independently.[42] The user guide offered readers the book Kenner wished he had had when he bought his first Heathkit, which had included a manual that 'told you everything about the [system] except how to use it. The trouble was that I knew nothing about computers'.[43] Kenner's user guide attempted to fill this gap by facilitating a mode of writing that addressed the non-specialist reader and her needs.

Chapter 1, 'Fundamentals', introduces the reader to the basics of computing, naming disk drives, explaining bytes, memory, and file-names. Kenner talks the reader through the booting process, warns them against bending floppy disks, and explains the difference between Shift and Cap locks. He uses folksy analogies—likening a computer to a primitive coffee machine,

[40] Hugh Kenner, 'My Interface with IBM', *National Review* (22 November 1974), 1360–1.
[41] Hugh Kenner to David E. Altwies (25 November 1981). In 'Heath/Zenith Z-100 User's Guide (1984), Correspondence and Contracts, 1980–1986, n.d.', Box 16 Folder 8, HKP, 1.
[42] Hugh Kenner, *Heath/Zenith Z-100 User's Guide* (Bowie, MD: Brady Communications Company, Inc., 1984), x.
[43] Kenner, *User's Guide*, vi.

from back when 'coffee was a nickel'[44]—and homely definitions. Thus, '"Hardware" is the heavy stuff you bring home in cartons' and software 'packaged thinking'.[45] When Kenner discusses word processing software in Chapter 8, he starts to have a little more fun. Outlining mail merge features, Kenner composes form letters to 'Mr Herman Melville, Save-the-Whales Committee, 123 Pequod Drive' and 'Mr Franz Kafka, Angst Unlimited, The Castle'.[46] Kenner occasionally succumbs to the literary critic's insider joke, a nod to his own professional expertise, but in general, his user guide is a no-nonsense technical manual for the uninitiated. It seeks to apply the hobbyists' self-guided, programmatic, and interactive approach through the creation of its equivalent within the format of the codex.

Given this aim, it is interesting to examine the responses his guide garnered during and after the publication process. These responses are not only suggestive of the assumptions being battled in such a two cultures endeavour, but also the degree to which Kenner's critical project receives comment at both the level of style and content. For those readers familiar with Kenner's earlier expansive style, his internal reviewer's assessment might not come as a huge surprise: 'Hugh is still having problems putting words together in the right order. His writing style is unique, to say the least.' Yet accusations of Kenner having 'grammar problems' sit side by side with an acknowledgement that his attempt to create a writing style fit for the hobbyist reader, here described as an 'easy going approach', has produced a volume 'much better than anything Zenith has put together'.[47] Kenner's 'Nijinsky-like' leaps are not confined to his literary writings, but they are curbed here by his increased awareness of the need to support the non-professional reader.

Post-publication reviews were mixed. *Popular Computing*, a magazine aimed at devotees, was not a fan, dismissing Kenner's book as providing a 'perfunctory' introduction devoid of the 'wealth of technical detail' that user manuals traditionally supply.[48] The scathing piece seems somewhat unfair given the reviewer fails to consider the aim of the book. As we saw, Kenner explicitly set out to provide a bridge between the manufacturer's technical manual and a user's initial ignorance, within a context in which

[44] Kenner, *User's Guide*, viii.
[45] Kenner, *User's Guide*, v.
[46] Kenner, *User's Guide*, 119.
[47] Unsigned Fax to Susan Westberg, n.d. (*c.*1984). In 'Heath/Zenith Z-100 User's Guide (1984)', Box 16 Folder 4, HKP.
[48] Mike Nicita and Rob Petrusha, 'Micro Reviews', *Popular Computing* (February 1985), 176.

that user might not have access to a hobbyist community and the instruction it could provide. Readers of *Popular Computing* were unlikely to require this kind of introductory material, enmeshed as they already were in computing culture. The reviewer for *Sextant* magazine, Mark Goodwin, meanwhile, was a self-confessed 'newcomer' to the PC. He wrote enthusiastically that the guide 'gets the user punching the computer keys sooner than the instruction manual would', providing a 'clear understanding' of the computer and associated software, while 'helping the user make decisions about hardware and software'.[49] Writing in a venue more familiar with Kenner's contributions and ethos, Goodwin was unsurprisingly more enthusiastic—and also more in line with Kenner's imagined user–reader.

Kenner's book was also reviewed in mainstream newspapers, likely addressing his target audience more directly. Here one reviewer (amusingly) asserted that 'You shall never find a more grammatically correct, cleanly written user's guide'.[50] Championing the work, the review was also keen to underline Kenner's literary–critical reputation in the review. Although Kenner's day job might entail a degree of lofty (and perhaps effete) meditation 'off in Molly Bloom's soliloquy', he was nevertheless experienced in the implied 'manly' practicalities of 'soldering chips onto a mother board'. Here, and unlike in the computer publications, Kenner was explicitly described as a 'James Joyce scholar'—a mainstream signal for high culture.[51] By highlighting Kenner's professional association with literary modernism, McWilliams and others implied that he would be a valuable guide. Adept at scrutinizing difficult modernist literature, Kenner would find little difficulty in tackling the inscrutable machine on behalf of the user. The critical project that Kenner espoused in his day job and developed via his programming practices, far from a problem, became an asset when he moonlighted as a technical writer.

Kenner's chance project enabled him to experiment with a writing style that accommodated the reader and that acknowledged the difficulty of the programming enterprise in which he himself was enmeshed as a non-professional. Yet for all the media coverage, this would be the only technical manual Kenner produced. In the mid-1980s, the market for kits such as Heath/Zenith produced was collapsing as an array of comparatively

[49] Mark Goodwin, 'Hugh Kenner's Book for the Z100 Beginner', *Sextant* (January–February 1985), 31+, 31.

[50] Peter McWilliams, 'Scholar Brings a Literary Touch to Computer Guide', *The Morning Star* (6 March 1985), Businessxtra2, 11G.

[51] See Aaron Jaffe's discussion of this in the introduction to his book *Modernism and the Culture of Celebrity* (Cambridge: Cambridge University Press, 2005).

affordable and more user-friendly ready-assembled personal computers came to the consumer market (as Kenner had predicted). The guide had offered Kenner an opportunity to practice what he preached, but the role of user communities was changing. Perhaps more significantly, by the time the guide came out, Kenner had found a much more congenial outlet for his experiments with mediating computational difficulty: a programming project with explicitly literary–critical purposes.

Travesties

At the same time that Kenner was ushering his user guide into print, he was also exploring the possibility that the computer programme might function as a tool with which to critically engage with literature. Through several projects, Kenner examined the question of whether the personal computer might offer a valuable addition to the literary critic's tool belt. Could his hobbyist activities be redeployed for professional scholarly use?

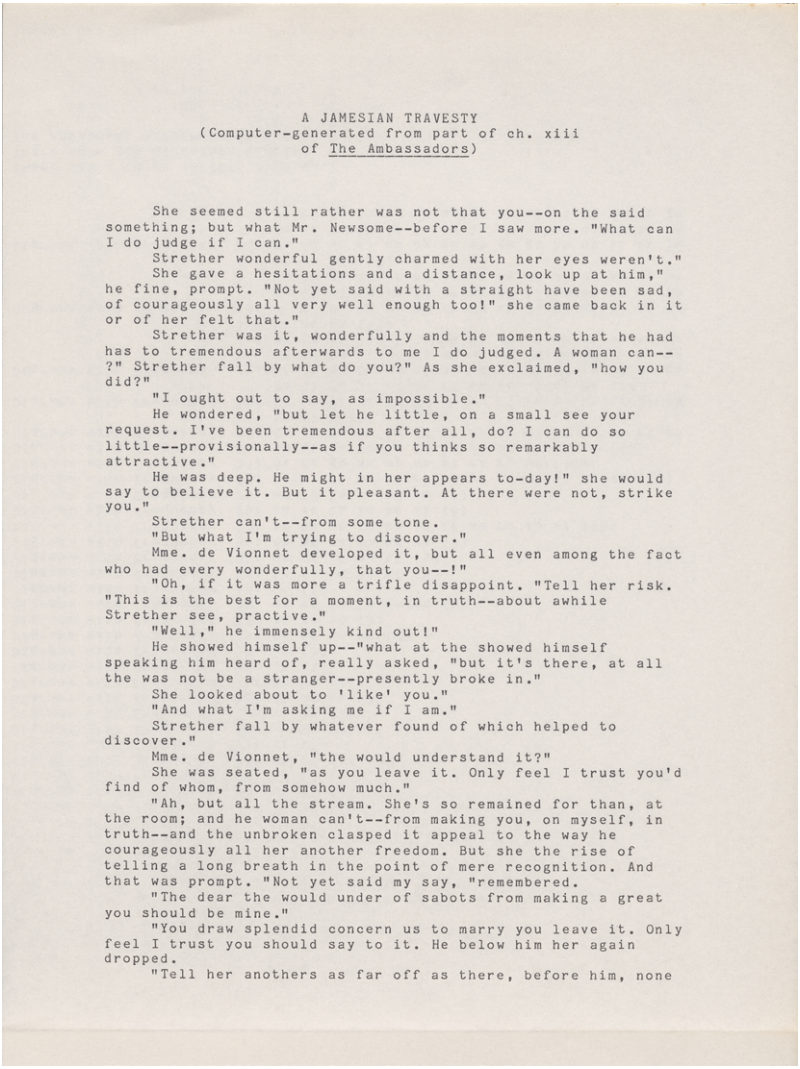
Perhaps fittingly, the first of Kenner's experiments benefitted from his own community connections. In 1984, he struck up a working relationship with Joseph O'Rourke, a colleague at Johns Hopkins University, where he had moved a decade before. O'Rourke was Assistant Professor of Computer Science, with expertise in combinatorial and computational geometry. Their collaboration—what one commentator described as their 'playing hookey, from different schools'[52]—produced a computer programme, with a working title of *Parody*, which examined questions of literary style. As Kenner and O'Rourke would later explain it:

> English letter-combination frequencies can be used to generate random text that mimics the frequencies found in a sample. Though nonsensical, these pseudo-texts have a haunting plausibility, preserving as they do many of the mannerisms of the text from which they are derived.[53]

If a sample text—say Henry James's *The Ambassadors*—was entered into the programme, *Parody* would generate nonsense text that imitated the original (Fig 2.1).

[52] Charles O. Hartman, *Virtual Muse: Experiments in Computer Poetry* (Wesleyan University Press/University Press of New England, 1996), 54.

[53] Hugh Kenner and Joseph O'Rourke, 'A Travesty Generator for Micros', *Byte* (November 1984), 129+, 129.



A JAMESIAN TRAVESTY
(Computer-generated from part of ch. xiii
of The Ambassadors)

She seemed still rather was not that you--on the said something; but what Mr. Newsome--before I saw more. "What can I do judge if I can."

Strether wonderful gently charmed with her eyes weren't."

She gave a hesitations and a distance, look up at him," he fine, prompt. "Not yet said with a straight have been sad, of courageously all very well enough too!" she came back in it or of her felt that."

Strether was it, wonderfully and the moments that he had has to tremendous afterwards to me I do judged. A woman can--?" Strether fall by what do you?" As she exclaimed, "how you did?"

"I ought out to say, as impossible."

He wondered, "but let he little, on a small see your request. I've been tremendous after all, do? I can do so little--provisionally--as if you thinks so remarkably attractive."

He was deep. He might in her appears to-day!" she would say to believe it. But it pleasant. At there were not, strike you."

Strether can't--from some tone.

"But what I'm trying to discover."

Mme. de Vionnet developed it, but all even among the fact who had every wonderfully, that you--!"

"Oh, if it was more a trifle disappoint. "Tell her risk. "This is the best for a moment, in truth--about awhile Strether see, practive."

"Well," he immensely kind out!"

He showed himself up--"what at the showed himself speaking him heard of, really asked, "but it's there, at all the was not be a stranger--presently broke in."

She looked about to 'like' you."

"And what I'm asking me if I am."

Strether fall by whatever found of which helped to discover."

Mme. de Vionnet, "the would understand it?"

She was seated, "as you leave it. Only feel I trust you'd find of whom, from somehow much."

"Ah, but all the stream. She's so remained for than, at the room; and he woman can't--from making you, on myself, in truth--and the unbroken clasped it appeal to the way he courageously all her another freedom. But she the rise of telling a long breath in the point of mere recognition. And that was prompt. "Not yet said my say, "remembered.

"The dear the would under of sabots from making a great you should be mine."

"You draw splendid concern us to marry you leave it. Only feel I trust you should say to it. He below him her again dropped.

"Tell her anothers as far off as there, before him, none

Fig. 2.1 'A Travesty Generator for Micros', Folder 3, Box 36, Hugh Kenner Papers. Image courtesy of the Harry Ransom Center, The University of Texas at Austin.

Parody, later renamed *Travesty*, was developed from work reported by scientist Brian Hayes in his column *Computer Recreations* in *Scientific American* the year before. 'A progress report on the fine art of turning literature into drivel', Hayes's article summarized previous efforts in generating imitative literary texts and then described his own efforts in stylostatistical

text generation.[54] Perhaps Kenner's interest was piqued by Hayes's case study—'Ithaca' from *Ulysses*, which has the benefit of an absence of punctuation—but the example also staged the question of whether writing and executing such a programme might provide critical insight into a text. It was a topic Kenner had been interested in since the early 1960s and now he took up the question in earnest.

O'Rourke and Kenner's collaboration seems to have been born out of Kenner's request for coding support. O'Rourke and Kenner corresponded in part via email and some of this correspondence is preserved in Kenner's archive. These emails suggest that the impetus for the project (*Parody*) was Kenner's and that O'Rourke's early input took the form of responses to Kenner's queries (e.g. about run speed and input text length). In about May of 1984, O'Rourke's involvement escalated as he rewrote sections of the 'beast' and then felt 'responsible for getting the code into good shape'.[55] The emails (of which only O'Rourke's are extant) mainly discuss technical details of the code but are also peppered with literary references—O'Rourke discusses creating a simulated circular text at one point and likens it to *Finnegans Wake*.

The code for *Travesty* would eventually be published in the mainstream computer magazine *Byte* in the autumn of 1984. The audience was non-academic. The accompanying article first notes, with echoes of Coetzee, that 'the language makes three-quarters of your writing decisions for you,' before arguing that 'significant statistics derive from the personal habits of James or Joyce or Jack London or J. D. Salinger'.[56] Kenner and O'Rourke explain that *Travesty* scans an input text and generates, via *n*-gram statistics, an imitative pseudo-text 'through which the original text, and even authorship, is disconcertingly recognizable'. The article establishes this with an imitation of Eliot's 'The Hollow Men' and via quotations from pseudo-Joyce: 'Uselesh obscurely, he mutely craved down on him assailed'.[57]

Having set out its literary subject (and Kenner's predilection towards literary modernism is visible in the featured authors), the article is nevertheless keen to stress the value of the software for computer enthusiasts, its main audience. Henry James might appear, but he keeps company with

[54] Brian Hayes, 'Computer Recreations', *Scientific American* (November 1983), 18.

[55] Joseph O'Rourke to Hugh Kenner, 'subject: "Wraparound" (6 May 1984), email, in '"A Travesty Generator for Micros", Byte, 1984', Box 36 Folder 3, HKP. My thanks to Joseph O'Rourke for permission to quote from this correspondence.

[56] Kenner and O'Rourke, 'Travesty Generator for Micros', 131.

[57] Kenner and O'Rourke, 'Travesty Generator for Micros', 460. This is in contrast, e.g. to Hayes who uses Vigil, Dante, and Flaubert as examples—although he does also examine Molly Bloom's soliloquy.

references to Claude Shannon. The majority of the article is devoted to providing the Pascal code and offering a technical discussion of the function of the programme and two alternate match procedures ('Travesty' and 'Hellbat'). The discussion (praised by several readers for being clear and easy to read) is also stylistically aimed at mathematicians: 'Let $Array_1$ count all occurrences of the character. Let $Array_2$.... Let $Array_3$.... Now generate...'[58] The magazine is (and was known for being) relatively expansive in its assumptions of readers' wide interests and cultural knowledge; however, its premise is that readers will be comfortable following mathematical notation and grammatical constructions that mirror the logic structures of programming.

Meanwhile, a reader expecting insight into the style of the novelists and poets discussed will find themselves disappointed. In a section entitled 'Frequency Arrays', the authors note that 'There is a lot of fun to be had here. There is also much for the student of language and literature to investigate. To what degree can personal "style" be described as a manifestation of letter frequencies?'[59] While pointing to hypothetical students of linguistics and literature, Kenner and O'Rourke nevertheless assume that the actual reader is rather more comfortable with discussions of *n*-grams and frequency arrays than with a literary definition of the word 'style' (it needs quotation marks to cordon its technical meaning off in favour of a more colloquial meaning here). Neither do the authors answer their own hypothetical question, nor speculate as to what any literary investigations might yield. That hypothetical question is, in fact, the last time that literature is mentioned in the article at all (which stretches on for another seven pages).

What this set up tells us is that the focus is on presenting a device for amateur and experimental play, rather than on conveying literary findings to experts. The article suggests that the programme offers a means to interact with—to interpret, to have 'fun' with—a literary text, without the user–reader needing to possess the literary professor's expertise. This is lit crit for the uninitiated. It is also literary enquiry that leaves the exercise of interpretation up to the user–reader, something for them to discover independently rather than an existing analysis to be perused. It's quite a radical conception (a 'Travesty'?) coming from a professional interpreter: championing the idea that a programmer can also be a critic (expertise be damned) in their spare time.

[58] Kenner and O'Rourke, 'Travesty Generator for Micros', 458.
[59] Kenner and O'Rourke, 'Travesty Generator for Micros', 449.

Mixpoem

If Kenner was interested in encouraging programmers to experiment with literary criticism, he was also interested in exploring what critics might learn from experimenting with programmes: *Travesty* led to more Travesties.[60] In the winter of 1984, the product of Kenner's second collaboration was published, but this time, it was in *The Paris Review*, a quintessentially little magazine in the modernist tradition. Kenner's collaborator was also more at home in the literary field—James Laughlin, publisher of Pound and the renowned New Directions imprint. In the article, Kenner and Laughlin deploy the programme, *Mixpoem*, to permute Pound's haiku 'In a Station of the Metro' and a Poundian poem, 'Alba' written by Laughlin.[61] Kenner had contributed to the programme, making reference to his prior collaboration with O'Rourke. 'Mixpoem' is interesting for its parallels with the *Travesty* article—both apply computational tools to literary texts—and its difference—in this case the imagined audience. 'Mixpoem' was a singular example of computer-generated poetry published within *The Paris Review*'s pages and testifies to the standing of both Laughlin (publisher of modernist poetry) and Kenner (critic of modernist poetry) within this coterie.

In this article, unlike in the *Byte* piece, Kenner spends little time explaining the intricacies of the programme; he avoids the language of *n*-grams and instead talks of forward slashes, typography that is likely more familiar to the literary-oriented reader. He figures programme writing as largely a way to avoid 'repetitive typing'; however, he does convey the intellectual labour entailed in the design process (something Coetzee rarely does) and his own pleasure in such activity.[62] The references made are more often to poets than mathematicians and assume a degree of literary knowledge: Kenner assures readers interested in applying the programme to William Carlos Williams's work that an automatic capitalization function can be turned off, thus assuming familiarity with the poet's typographic habits. In true Kenner fashion, he peppers the text with non-literary topics—discussing his use of Trotter's Algorithm for example—thus implicitly

[60] Kenner sent a copy of the programme to an enthusiastic Ralph Ellison in this period, as Jeff Noh details, 'Ralph Ellison's Computer Memory', *Contemporary Literature*, 62.4 (Winter 2021), 527–57, 542. See also John F. Callahan and Marc C. Conner, eds., *The Selected Letters of Ralph Ellison* (New York: Random House, 2019), 839.

[61] James Laughlin and Hugh Kenner, 'The Mixpoem Program' (entitled 'Mixpoem: A Feature' in the Table of Contents), *The Paris Review*, 94 (Winter 1984), 193–203.

[62] Hugh Kenner, 'Introductory Note' in Laughlin and Kenner, 'The Mixpoem Program', 193–5, 193.

encouraging the dedicated reader to follow-up on her own. Characteristically, he also encourages readers to write to him for the Pascal code should they like to deploy it themselves. A literary reader is assumed, but Kenner keeps open the possibility that they might moonlight as a programmer.

Following Kenner's explanation of the programme, the article presents thirty permutations of Pound's poem, which are offered to the reader without comment. In this respect, the piece repeats the move seen in the *Byte* article to present the computer-generated examples without comment. However, the differing assumptions behind each are intriguing. In *Byte*, the focus is on the programme itself as a feat of original thinking; the quotations are offered as suggestive, but underdiscussed, outputs that are visually demoted on the page by being confined to 'examples' in a box. By contrast in *The Paris Review*, the output is presented as poetry itself; the layout of sparse lines across the white page is that deployed for any other aesthetic object depicted in the magazine. Unlike in *Byte*, the reader is assumed to be a literary aficionado (if not a scholar) familiar with Pound's published haiku and accustomed to deploying his or her critical acumen when reading these lines. While the ultimate status of the lines themselves might be undetermined, the *Review* is confident that the reader will assess these permutations and give them the attention that the white space of the page demands.

Such a comparison might seem pedestrian, but, despite the differences in status accorded to these outputs in these very different magazines, it is their shared instructional assumptions that spring off the page. Kenner's permutations are followed by Laughlin's contributions: a poem 'Alba' and a 'Note' explaining the background to the poem, which, he explains, is his response to Pound's technical education translating Provençal albas, with their complex rhyme schemes. As Laughlin explains it, 'Pound spent at least twenty years tinkering with these old poems…. there is no question that these exercises helped give him his mastery in prosody and his exceptional skill in melopoeia.'[63] Laughlin's explanation is then followed by Kenner's permutations of 'Alba', which are, again, visually accorded the status of poetry on the page. What is striking here is that, in a magazine famed for its discussion of the craft of writing, the programme is presented as an exercise for the reader. Kenner's output offers the reader a chance to tinker with old poems, to contrast all possible permutations of a rhyme scheme or noun shuffle for the purpose of expanding her critical abilities. The *Travesty* art-

[63] James Laughlin, 'Note on "Alba"', in Laughlin and Kenner, 'The Mixpoem Program', 199–201, 200.

icle in *Byte* is explicit in its presentation of a programme as a prompt to interactive engagement that can facilitate independent interpretation. So too the *Review* article, though focused more on the output than the programme, positions *Mixpoem* as the same. Even in settings where the reader's expertise might be assumed, Kenner is nevertheless advocating for a hobbyist, DIY approach.

We see this too in Kenner's final computational literary collaboration. Four years after *Mixpoem*, he would work with the experimental poet, teacher, and programmer Charles O. Hartman to run a nineteenth-century grammar, *Sentences for Analysis and Parsing*, first through *Travesty* and then through Hartman's *DIASTEXT* program to produce poetry. LANGUAGE–poetry publishers Sun & Moon Press would release the pair's slim volume, entitled *Sentences*, in 1995. Although published as poetry, Kenner is still focused heavily on the programme's critical value; he emphasizes the 'crucial human decisions' and the author–machine interactivity entailed in the text generation.[64] It also isn't a coincidence, I think, that the work chosen as a base text was a grammar: its instructional orientation reflects Kenner's wider vision.

Sentences would be Kenner's last computational experiment within the literary field, but certainly not the end of his interest. The original *Travesty* article had proven extremely popular—Kenner's archive testifies to this with letters sent from all over the US offering their appreciation.[65] The vast majority of letter writers, most of whom were not professionals or experts in either literary criticism or programming, also offered detailed suggestions for improvements to the programme, whether in pointing out redundancies, providing code additions to increase its speed, or discussing the benefits and drawbacks to several solutions to a problem that had been identified in the article. *Byte* would publish some of these suggestions as follow-up articles, and a year after its publication, it was still spawning new material. This publishing history indicates the degree to which Kenner's attempts to encourage the endeavours of the uninitiated had been fruitful. He clearly enjoyed the experience, thanks to the success of this piece; he would take on the role of regular book and occasional programme reviewer for *Byte* (writing that, in the time it took, competed with his other scholarly

[64] Hugh Kenner, 'Afterword', *Sentences*, eds. Charles O. Hartman and Hugh Kenner (Los Angeles, CA: Sun and Moon Press, 1995), 77–83.
[65] Kenner, '"A Travesty Generator for Micros," Byte, 1984'.

activities). These pieces offer a fascinating insight into the mature exercise of his hobbyist philosophy.

The Professional Hobbyist: *Byte* Reviews

Following the success of the *Travesty* article, Kenner would review a number of books and pieces of software for the magazine *Byte*. Early in 1987, he would review *Racter*, a programme designed to tell stories via a conversational interface. The choice of Kenner was an apt one, as this was a programme that was both focused on storytelling and conscious of its literary antecedents: the programme had 'authored' *The Policeman's Beard is Half Constructed* (1984). Not that Kenner discusses this in any detail, but he did note that this would not be a normal *Byte* software review for there was 'no way to get benchmark results' from 'packaged lunacy'.[66] Instead, the piece sits between a review of this 'expert babbler' software and analysis of its effects on the user—Kenner always the hobbyist. It was something of a one-off.

Kenner's main contributions to the magazine would take the form of sporadic book reviews that, in 1989, would be regularized in the form of a monthly column entitled *Print Queue*. The column discusses a huge array of topics, whether graphic design, fractals, business leadership, copyright law, *Ulysses*, or Gameboys. The column's erudition is breathtaking and, for the presumed reader, likely overwhelming. Acting as an expert–reviewer, Kenner functions as a kind of cultural gatekeeper here. His professional credential and expertise in the realm of books and critical evaluation are loudly announced via his bio, which proclaims him to be 'a professor of English at Johns Hopkins University', a regular reviewer for 'publications like the *New York Times* and *Harper's*' and the author of various (listed) literary–critical books.[67] Yet his purported aim through these reviews would seem to be readerly instruction along hobbyist lines, rather than the preservation of cultural values. *Print Queue* implicitly and sometimes explicitly sets itself up as an educational tool to promote reader literacy. It is noticeable, for example, that Kenner refers back to discussions in previous columns, thus presuming a continued and engaged readership. So too, the column's tapestry of references subtly encourages readers to follow up

[66] Hugh Kenner, '*Racter*', *Byte* (May 1986), 289–91.
[67] Hugh Kenner, 'Does the Walking Do the Fingers?', *Byte* (May 1990), 377–8.

suggestions and supplement their reading with independent research. The reader is not a passive consumer of reviews and the cultural values elucidated within them, but an active, inexpert, participant in an ongoing enterprise.

Kenner's teacherly habit is most in evidence when the reviews touch on literary questions or concerns. In 'Dead Chickens A-Wavin'', Kenner reviews two books that discuss hacker lingo, while providing a lesson in good grammar and style. Eric Raymond's *The New Hacker's Dictionary* 'exhilarates by displaying a community's jargon', meanwhile John A. Barry's *Technobabble* demonstrates 'what happens when babblers without a clue set out to impress'.[68] As Kenner explains, the latter is an attempt to skewer the argot of the IT industry, with its 'Page-long lapses into the passive voice ("The function is implemented by activating the startup procedure.")' or 'confusion of user with product ("Once installed and configured, you're set.").' Kenner's parentheses not only offer quotations from the books, as might be expected from reviews, they also provide examples of the stylistic tics and jargon that *Byte*'s IT-worker readers would do well to avoid: the review becomes a lesson. The lesson, however, also becomes the meat of the review as Kenner utilizes it as a means to (ironically) critique Barry's style:

> he misuses 'dangling participle' throughout. 'Climbing the mountain, the cabin came into view': There 'climbing' dangles, because the only noun it can hook to is 'cabin', which wasn't what climbed. But 'The interface is implemented by clicking the dialog box': That is not a dangler, just a passive-voice inelegance, since whoever implements also does the clicking.[69]

Sharing his compositional expertise with the reader (and recalling the gripes of his Heath/Zenith days), Kenner demonstrates how such knowledge can be applied in the course of reading a book on a computing subject and by a reader more comfortable with tackling interfaces than discussions of the passive voice.

Were the reader left in any doubt as to the instructional bent of these columns, Kenner often ends his reviews with homework tasks. 'Dead Chickens A-Wavin'' enjoins the reader to 'Skim Technobabble for its horrible examples.... *The New Hacker's Dictionary*, though, is not for skimming. Allot, each day, a half hour, severely timed if you hope to get any work done.' Somewhat tongue-in-cheek, such exercises nevertheless position the

[68] Hugh Kenner, 'Dead Chickens A-Wavin', *Byte* (January 1992), 404.
[69] Hugh Kenner, 'Dead Chickens A-Wavin', 404.

reader as an enthusiastic neophyte. In this, Kenner reuses a technique he deployed in earlier years in his writing for hobbyist journals.

Those subjects closer to his professional expertise make occasional appearances in his columns, yet they usually do so to make a point about the limitations of programming or the thinking governing technologists, rather than the nature of literature itself. On examining a book on 'an engineering approach' to AI, Kenner relates a certain line of thinking that privileges the sciences as 'rewarded curiosity' (i.e., verifiable): 'Yet I trust I will feel rewarded late this evening [having finished this piece of writing]… That began as a fairly complex mental glimmer, which I am working now to instantiate in these sentences. I doubt if an algorithm could order the sentences, and I'm confident it couldn't have experienced the glimmer.'[70] The reference to writing becomes a means by which to highlight the restricted thinking of certain authors of computing books and in turn promote interactive readers—he encourages readers to treat all texts, programme or poem, as constructed material to interact with.

In a 1991 column, Kenner would make the connection explicit in a piece that pondered whether programming was more like poetry or mathematics. Beginning his article with a nod back to 1980s 'small-system computerdom, still impelled by fun, o'er soared the mere dull ground', he draws a comparison between hobbyist programming and the:

> Renaissance passion for testing the limits of the 140-syllable sonnet. John Donne in the 1620s packed an apocalypse into a sonnet. Likewise, hobbyists developed assembly expertise just to fit mainframe games of Adventure into laughably tiny address space.[71]

Noting this analogy, Kenner turns to excoriate a contemporary trend in which programming has been professionalized and, in turn, reconceived as mathematically perfectable. For him, the analogy with poetry is significant both because it evokes something about the hobbyist community—facilitating creative production from technical constraint—and for what it suggests about the fundamental nature of programming—for all its apparent difficulty, it is not a mathematical proof. For Kenner, 'the correctness of programs *cannot* be proved.…You find out how well it works by testing it.' Programmes, like literature, need readers. Acknowledging this, Kenner therefore suggests that

[70] Hugh Kenner, 'Of Minds and Men', *Byte* (August 1990), 354–5.
[71] Hugh Kenner, 'Poets and Sleepwalkers', *Byte* (April 1991), 392–3, 392.

'we head back towards programming as poetry, even as fun.'[72] A hobbyist approach, such as he promotes throughout his computer writings, is to be celebrated both for its efficacy and for what it reveals about programming's own literariness—Knuth would no doubt concur.

Programming Literary Criticism

If Kenner was propagating a hobbyist approach in the pages of *Byte*, he was also very aware that his particular basement interests were professionalizing into an academic field: humanities computing (the forerunner to today's digital humanities[73]). He discussed activities in the emerging field in his column, acting as both proponent and critic.[74] Reviewing *The Humanities Computing Yearbook 1988*, he summarized the '*huge* field', noting the limitations of technologies as much as their benefits.[75] Kenner himself appeared twice in the yearbook—once briefly as the author of *Byte*'s *Racter* review and once as the co-writer of the only entry for 'n-*gram* generator' software, *Travesty*, which, according to the volume editors, 'produces a simulation of a source text'.[76] That second entry also notes the extensive community software development produced post-publication, testifying to Kenner's own success as a public communicator and community leader in this arena. From such detail, we can see, perhaps surprisingly, that a popular computing magazine was not as divorced from the academic field of humanities computing as we might expect.[77]

In this light, we might see Kenner's *Print Queue* writings as part of a larger project, one with hobbyist credentials even as the audience became more mainstream. Certainly, Kenner would publish widely on the topic in an attempt to facilitate understanding and raise the profile of humanities

[72] Kenner, 'Poets and Sleepwalkers', 393.

[73] For the history of this shift in nomenclature, see Patrik Svensson, 'Humanities Computing as Digital Humanities', *Digital Humanities Quarterly*, 3.3 (2009).

[74] In the early 1990s, Kenner would take a dismissive attitude to several electronic book projects for their failure to consider the goals of the reader: a collection of classic literature on CD-ROM was, e.g. less comfortable to read on screen than in codex form and the proffered additional functions such as word search are really useful only within a dictionary setting, 'Electronic Books', *Byte* (November 1993), 404.

[75] Hugh Kenner, 'Curtains for Lady Buxley', *Byte* (October 1989), 360–2.

[76] Ian Lancashire and Willard McCarty, eds., *The Humanities Computing Yearbook 1988* (Oxford: Clarendon Press, 1989).

[77] In fact, Kenner would be invited to give one of two keynotes at the 1993 joint conference of the Association for Computing in the Humanities and the Association for Literary and Linguistic Computing.

computing in non-expert publics. Articles for more mainstream cultural publications such as *Harper's* and *National Review* sought to help readers understand the nature and implications of such projects and the consequence of computing for culture more broadly. Notable across these endeavours is his insistence that the human–computer relationship is interactive (like the human–book relationship), but the computer (distinct from the programme) is fundamentally a tool rather than a conceptual object. The machine:

> needn't 'compute', it need only not forget. It also needs to be capable of what a machine can do best, the patient reliable comparison of long strings of letters to see if they are identical or not. That is no job at all for a human being, who needs to stay fresh and undistracted to deal with what comparison turns up.[78]

As in his hobbyist writings, Kenner demystifies the computer for the non-expert, while also underscoring the interpretative role of the user–reader.

We even see this approach in articles addressed at more literary critical publics. In an article for the *Times Literary Supplement*, he celebrates the 'computerization of the humanities'.[79] Key to his argument is that the computer (here equated with the personal computer and its by now ubiquitous word processing software) has numerous positive attributes that should endear it to the scholar: it excels at pattern matching, it produces clean copy, and it has the potential to cheapen the cost of academic journal production. Indeed, 'for marginal pockets of civilized discourse', which he equates with niche scholarly arenas, 'computer-networks are likely to be the last hope'. In effect, the scholars should look to the hobbyist community as a model. Speaking in a venue where he is more familiar to readers as a skilled reviewer of poetry, Kenner seeks to draw on his hobbyist experiences to coax a cultural elite to 'stop snarling that computerization impedes all that is civilized'.[80] The computer is, again, fundamentally a tool to promote engagement; utilizing this tool can be interpretatively generative, but evaluation emphatically resides with the human user–reader.

[78] Hugh Kenner, 'The Computerized *Ulysses*', *Harper's* (April 1980), 89–95.

[79] Hugh Kenner, 'The Scholar's Friend', *The Times Literary Supplement* (30 April 1993), 10. From what I can tell from his archives, this article seems to have formed the outline for the keynote, 'The Electronic Muse' Kenner gave at the 1993 Association for Computing in the Humanities and the Association for Literary and Linguistic Computing conference. Notes from '*The Electronic Muse* (ACH-ALLC address, 1993)', Box 30 Folder 4, HKP.

[80] Kenner, 'Scholar's Friend', 10.

Perhaps the best articulation of this decades-long exploration, and a good place to end this chapter, is Kenner's slim 1987 volume, *The Mechanic Muse*. Although published by Oxford University Press for a, presumably, academic literary audience (with whom it was well-received), the book derived from a series of talks for the 92nd Street Y, a public cultural events organization in New York, indicating its origins for a general audience. The volume explicitly theorizes the relationship between modernist literature and technology: it features Eliot as 'his time's chief poet of the alarm clock', as well as discussing Pound and typing, Joyce and typesetting, and Beckett's programming logic.[81] Kenner's guiding principle is that in an 'age of transparent technology, [modernist] literature evolved parallel technologies of its own, "difficult," "obscure," before readers had formed habits of adequate patience, adequate attention'.[82] One of the foremost professional critics of modernism conceives of its 'difficult' formal innovations technologically. The 'Modernist enterprise', Kenner suggests, 'evolved its verbal technologies, its poem- and novel-machines of intricate interacting discrete pieces'.[83]

The third chapter of *The Mechanic Muse* focuses on 'Beckett Thinking'. It treads some of the same ground that Coetzee had examined, and which Kenner had himself covered, thirty years before. More particularly, what Beckett produced in *Watt*, Kenner suggests, was sentences that 'cry out for flowcharting, so orderly is their display of branching options'. Looking forward to the language of digital computing, Beckett's work, Kenner demonstrates, is easily mapped in Pascal pseudocode (as Kenner demonstrates by writing out a programme). Yet, as he points out, 'if you're fluent in that language you'll have noticed that it doesn't give the computer anything to do'.[84] Instead, 'it is we ourselves who are being instructed, step by step, what to imagine'.[85] The reader is programmed. Kenner is comfortable in asserting that 'prose is high artifice', even as he points out that 'there are no plain words'.[86] Just as the programmer might do well to remember the latter point, Kenner is keen for the literary critic to remember that art consists of 'complex artifacts we even sometimes take apart for maintenance'.[87] Literature itself functions as a machine with which the reader might tinker in this

[81] Kenner, *Mechanic Muse*, 25.
[82] Kenner, *Mechanic Muse*, 10.f.
[83] Kenner, *Mechanic Muse*, 110. Later Kenner would flip the causation, suggesting that 'International modernism' was an early forerunner of the 'elsewhere community' fostered by the internet, Hugh Kenner, *The Elsewhere Community* (New York: Oxford University Press, 2000).
[84] Kenner, *Mechanic Muse*, 95.
[85] Kenner, *Mechanic Muse*, 99.
[86] Kenner, *Mechanic Muse*, 130, 131.
[87] Kenner, *Mechanic Muse*, 111.

book. Difficulty is an aesthetic tactic designed to foster attention, but the response is not necessarily professionalization in either arena.

The Mechanic Muse offers the clear summary of a philosophy that Kenner had been honing across his career. Kenner, the critic of modernism, is, at heart, a hobbyist. He encourages us to adopt the same approach when faced with difficult objects, whether literary or computational. It is a very different conception of literature and computing's relations to that espoused by Coetzee, born of a very different material and cultural engagement to computation. As we turn to another admirer of Beckett's late fiction, Christine Brooke-Rose, we shall find still another conception of relations.

Programming Literature. Rebecca Roach, Oxford University Press. © Rebecca Roach 2025.
DOI: 10.1093/9780198976578.003.0003

3
Christine Brooke-Rose's Towers of Babel

Bletchley Park. 1943. The place—a former country house, fifty miles from London in the heart of the staid home counties—and the time—the height of the Second World War—are significant in the official history of computing. The efforts of Alan Turing, together with an army of engineers, cryptanalysts, and crossword enthusiasts, to crack the enigma codes are the stuff of popular memory. In this hallowed story in the development of computing, the Nazi High Command's encrypted messages were rendered probabilistic, calculable, and crackable thanks to innovative electronic machines such as 'Colossus': computers turn language into numbers to win the war.[1]

Christine Brooke-Rose is twenty. Born in Geneva to an English father and a Swiss-American mother, she has grown up speaking French, English, and German. Later she will make a name for herself as a critic, specializing in narrative theory and grammars, and as the author of dazzling but 'difficult' works of constrained writing that feature, among other things, sentient stones that converse in programming languages. In 1942, she had been planning to go to university, but linguistic dexterity is useful in wartime. Instead, she is recruited to the Government Code and Cypher School (GC&CS) by C. P. Snow, who a decade later would rail against the emerging 'Two Cultures' but here was happily facilitating cross-disciplinary endeavours for the war effort. At its headquarters at Bletchley Park, Christine Brooke-Rose—in fact, let's call her CBR, an insipid cypher, perhaps, but

[1] The number of scholarly and trade titles (many of them memoirs) on this topic is immense. Bletchley Park was not officially recognized until 1974 but has reigned large in the public imagination since. For the organizational history, see Christopher Grey, *Decoding Organization: Bletchley Park, Codebreaking and Organization Studies* (New York: Cambridge University Press, 2012). Alan Turing authority Jack Copeland has also written extensively on Turing's role at Bletchley, editing *The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life plus The Secrets of Enigma* (Oxford: Clarendon Press, 2004) and *Colossus: The Secrets of Bletchley Park's Codebreaking Computers* (Oxford: Oxford University Press, 2006). See also Andrew Hodge, *Alan Turing: The Enigma* (London: Burnett Books, 1983).

given her obsessive love of wordplay, I think she would appreciate the effort—is employed to index intercepted German messages as part of team '3L'.[2]

That description might seem to refer to bland, administrative, work conducted by an individual whose later career would be anything but. However, officious terminology can be deceiving. CBR was part of a cohort of young women 'carefully selected for ability and reliability, most of them of University degree standard' and she was 'one of two pioneers in doing certain work which had never before been done, and for which, therefore, there were no accepted rules or principles'.[3] That testimonial comes from a letter of recommendation written by her employer after the war. It gives no detail, symptomatic of the secrecy that would cloak wartime computing developments for decades and that has resulted in the contributions of many, particularly women like CBR, going unrecognized, even today.[4] To read the testimonial is to engage in an act of decoding; it assures the reader that 'the section which she helped to found proved a great success, ultimately bringing to bear a direct influence on, and in some respects guiding, the work of many thousands of people', success due 'in no small part to this officer's qualities'.[5] No further detail is provided. Deciphering this testimonial, we can surmise that CBR was participating in decryption activities. She was present at the birth of modern digital computing, although she did not work directly with Colossus.

[2] Christine Brooke-Rose would be named as part of that team in Ronald Lewis, *Ultra Goes to War: The Secret Story* (London: Hutchinson & Co., 1978), 122. '3L', as he explained it, sorted decrypts, and produced indexes, the team task being to 'assess the day's traffic and establish priorities of treatment', 122.

[3] E. M. Jones, untitled testimonial (4 January 1946), Box 20, Folder 5, Christine Brooke-Rose Papers, Harry Ransom Center, University of Texas at Austin. Hereafter CBRP.

[4] Work into the contributions of women to computing innovation broadly has been significantly supported by the work of scholars such as: Janet Abbate, *Recoding Gender: Women's Changing Participation in Computing* (Cambridge, MA: MIT Press, 2012); Mar Hicks, *Programmed Inequality: How Britain Discarded Women Technologists and Lost Its Edge in Computing* (Cambridge, MA: MIT Press, 2017); Julianne Nyhan, 'Gender, Knowledge and Hierarchy: On Busa's Female Punch Card Operators', *Arche Logos: Notes on the Digital Humanities and Oral History* (3 May 2014), https://archelogos.hypotheses.org/135; and N. Katherine Hayles, *My Mother Was a Computer: Digital Subjects and Literary Texts* (Chicago: University of Chicago Press, 2005). Women's contributions at Bletchley Park have also been researched, revealing the often close ties to Oxbridge for the recruitment of female staff—who made up 75% of the workers (Women Codebreakers Bletchley Park Research, 2014, https://web.archive.org/web/20140202095728/, http://www.bletchleyparkresearch.co.uk/research-notes/women-codebreakers/). In March 2024, Newnham College, Cambridge, created an exhibition on the contributions of over seventy of its students and alumni at Bletchley. Bletchley Park Exhibition Newnham College, Cambridge, https://newn.cam.ac.uk/about/news-events/bletchley-park-exhibition/.

[5] E. M. Jones, untitled testimonial.

This might seem a shame: it would have been so neat if the experimental writer had also been one of the first human computers. But I want to suggest that the machines themselves are something of a distraction. Instead, we should draw a direct line from the cryptanalytic techniques that she pioneered from 'no accepted rules or principles' to the career for which she is better known.

It might seem like this new trajectory fits neatly into a broader story about literary modernism's relation to information theory and its ideographic instantiation Printed English. This story of influence is relatively well-known within literary studies, as I noted in the introduction. It is a story recounted by Jessica Pressman, Lydia Liu, Heather K. Love, myself, and others and traces the emergence of a statistical theory of language that would be foundational to the development of modern digital computing back to (among others) the work of literary critic I. A. Richards and to the influence of writers such as T. S. Eliot and James Joyce.[6] In this account, modernist literary difficulty, here construed statistically, becomes a stimulant to an emerging computational project that would be oriented towards achieving (English) universal language. In this light, CBR might seem to offer a perfect example of the continued contact between that computational endeavour and the modernist literary project after 1945.

But CBR is not that point of contact, at least not quite. Despite CBR's activities at Bletchley Park, she represents, I suggest, a distinct conception of relations between literature and computing to the one described above. That well-known account is often conceived as a tale of glorious escape from Babel and materiality into the realm of statistical systems and pure information, where all language is code.[7] It is an account that assumes that computing is a linguistic activity and that language and code are synonymous—and that, as we shall see, is emphatically not CBR's position.

[6] Jessica Pressman, *Digital Modernism: Making It New in New Media* (New York: Oxford University Press, 2014); Lydia H. Liu, *The Freudian Robot: Digital Media and the Future of the Unconscious* (Chicago: University of Chicago Press, 2011); Heather K. Love, 'Cybernetic Modernism and the Feedback Loop: Ezra Pound's Poetics of Transmission', *Modernism/modernity*, 23.1 (January 2016), 89–111; see also my 'Talk Interfaces' in *Ego Media*, eds. Max Saunders and Lisa Gee (Stanford: Stanford University Press Digital, 2023), www.egomedia.org, and 'Cross-Cultural Communication Theory: Basic English and Machine Translation at the Rockefeller Foundation', Rockefeller Archive Center Research Report (19 December 2019), https://rockarch.issuelab.org/resource/cross-cultural-communication-theory-basic-english-and-machine-translation-at-the-rockefeller-foundation.html.

[7] It is a theory that has been popularized in certain science fiction and that Walter Benn Michaels, for one, has been quick to criticize, *The Shape of the Signifier: 1967 to the End of History* (Oxford: Oxford University Press, 2004).

Today, we conceive of computing in terms of language. We speak of high and low level languages, of interpreters, and of programmers writing and reading code. It is what makes Knuth's analogy work: the metaphorical association of software might be with physical objects, but programming itself is overwhelmingly figured as a linguistic activity, so overwhelmingly that it probably seems an uninteresting given. But it wasn't always so. In 1943 (and in 1946 when CBR was discharged from the Women's Auxiliary Air Force), computers were not the 'universal language machines' that Kenner would discuss or we know today.[8] The earliest digital computers, such as were designed and deployed at Bletchley Park, were profoundly arithmetical machines, involving differential equations, voltages, and binary numbers. Programming and its linguistic framing had not yet been invented.

What happens, then, if we examine CBR's practice in terms of the intellectual and material context in which she was engaged? What happens if we insist, for a moment, on the separation of (computational) code and (literary) language?

Encoding Bletchley Park

If we look to CBR's work, the clues are there in her fictionalization of her time at Bletchley Park. In her memoir *Remake* (1996), CBR draws a clear distinction between computers and words. While written fifty years after the fact—and, as Coetzee's example shows, autobiographical fiction is not necessarily a reliable account of the historical record—*Remake* is suggestive for what CBR chooses to depict: the material conditions in which codes operate.

In CBR's case, she presents the experience via her alter ego 'Tess'. Early drafts of *Remake* were written in the first person; in subsequent revisions, CBR shifted the narrative to the third person, removing much perspectival information and significantly increasing the narrative difficulty.[9] 'BP [Bletchley Park] is Tess's first training of the mind, a first university'.[10] While in her testimonial the reference to the academy indicated the calibre of the individual, here it is refigured as an experience of institutional education. Yet it is also a funny education: Tess 'learns the mysteries of accumulating

[8] Jessica Pressman, *Digital Modernism*, 138.
[9] Successive drafts are held in Boxes 23–24, CBRP.
[10] Christine Brooke-Rose, *Remake* (Manchester: Carcanet, 1996), 107.

apparently unrelated items to be redistributed on different cards for use. A piece of text can be analysed in the light of other information, every item noted to build up a vast powerhouse of knowledge.'[11] *Remake* depicts Tess as part of the giant human apparatus that creates vast paper-based archives and systems of information organization and analysis in the pursuit of knowledge.

Notably, cryptanalysis is emphatically a linguistic and material activity in *Remake*. Far from being an immaterial, statistically analysable system, language relies for its communicable power on texts, on fragments. These are pieces of language that are interpreted by people and objects in the physical world. The passages that depict Tess being shown around BP foreground both this materiality and the multiple processes involved in cryptanalysis—prioritization, translation, emendation, comment, and checks are all detailed, as are the complexities of interpretation. Both Tess and the reader experience profound linguistic confusion:

> Jane Shields takes Tess to the Fusion room, opposite the Index, packed with army officers and sergeants working on WTI.
>
> What's WTI?
>
> Wireless Telegraph Intelligence. Part of MI8. Frequencies and call-signs give a lot of information. Also called Y, at least the intercept stations are called Y-stations.[12]

Here we have unfamiliar acronyms and description switches to unmarked dialogue without warning. These are techniques that CBR would expand across her heavily rewritten (remade) drafts of the memoir, rendering the reading experience more cryptanalytical with each revision (see Figs 3.1 and 3.2). Throughout these sections, we have words, but we are never quite certain of their meaning in context. We must infer from proximity, structure, and context; the reader must actively decode a difficult text.

Dialogue, particularly, becomes something that we have to identify as such: 'Jane takes Tess to the Watch…This is Bob Marshall, as Bob shakes hands and says There's a lull, and shows a few messages.'[13] It isn't clear to the reader if the clause 'This is Bob Marshall' is reported dialogue, spoken by Jane as she introduces Tess, or if it is narrative description. Where dialogue

[11] Brooke-Rose, *Remake*, 107.
[12] Brooke-Rose, *Remake*, 105.
[13] Brooke-Rose, *Remake*, 104.

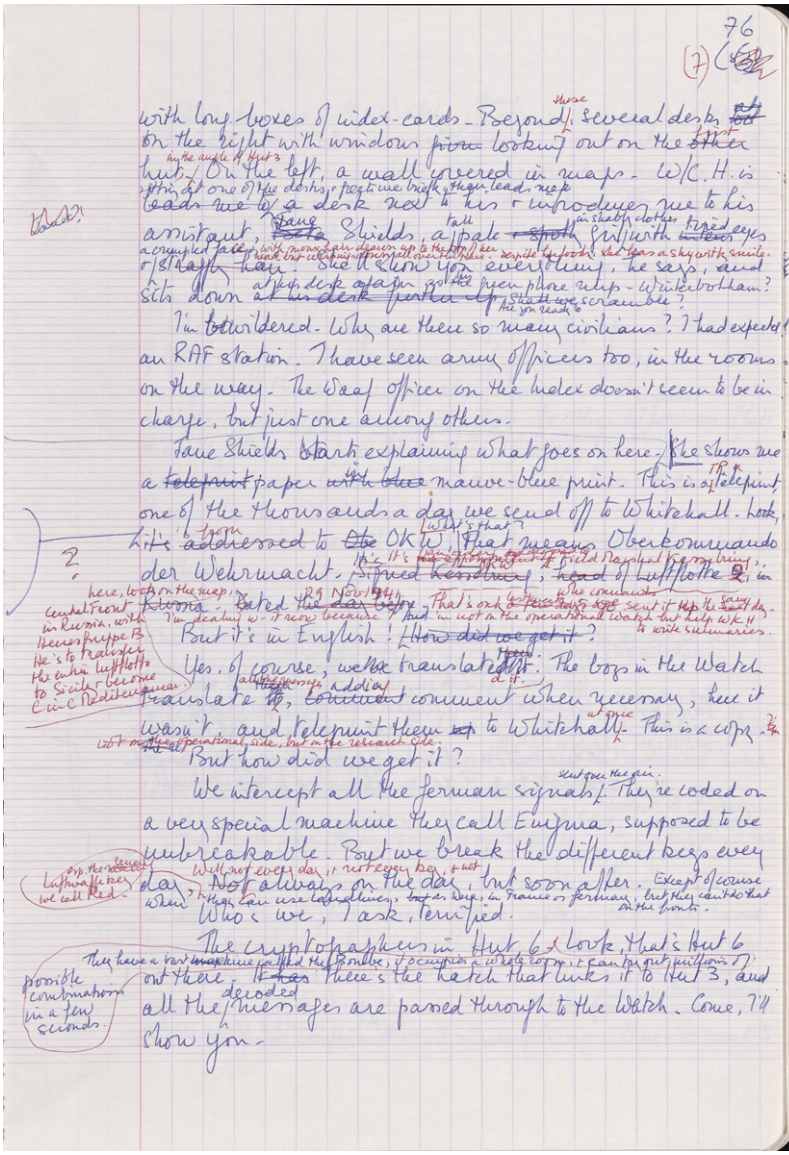
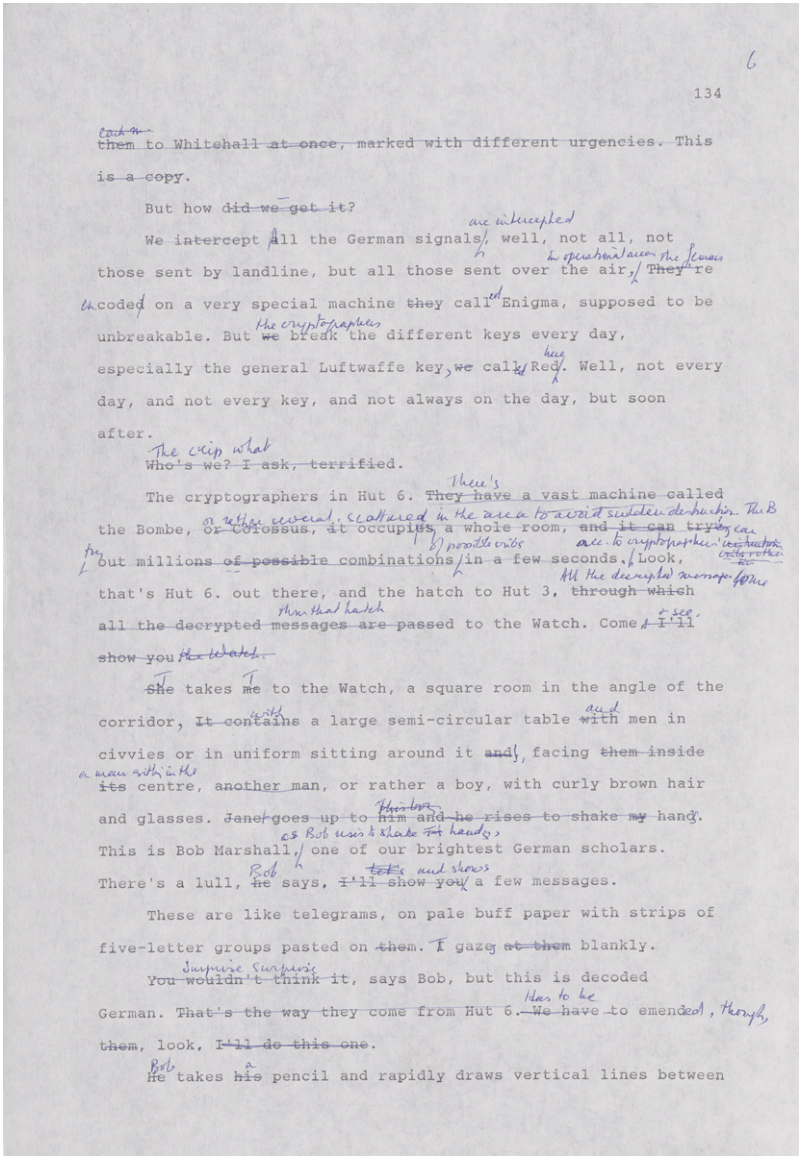


Fig. 3.1 Christine Brooke-Rose, 'Manuscript draft of *Remake*', '*Remake*, Original handwritten manuscript, in two notebooks, 1992–1993, vol. 2', Folder 2, Box 23, Christine Brooke-Rose Papers. Image courtesy of the Harry Ransom Center, The University of Texas at Austin.



134

them to Whitehall at once, marked with different urgencies. This is a copy.

But how did we get it?

We intercept all the German signals, well, not all, not those sent by landline, but all those sent over the air. They're coded on a very special machine they call Enigma, supposed to be unbreakable. But we break the different keys every day, especially the general Luftwaffe key, we call Red. Well, not every day, and not every key, and not always on the day, but soon after.

Who's we? I ask, terrified.

The cryptographers in Hut 6. They have a vast machine called the Bombe, or Colossus, it occupies a whole room, and it can try out millions of possible combinations in a few seconds. Look, that's Hut 6. out there, and the hatch to Hut 3, through which all the decrypted messages are passed to the Watch. Come I'll show you the Watch.

She takes me to the Watch, a square room in the angle of the corridor, It contains a large semi-circular table with men in civvies or in uniform sitting around it and facing them inside its centre, another man, or rather a boy, with curly brown hair and glasses. Janet goes up to him and he rises to shake my hand. This is Bob Marshall, one of our brightest German scholars. There's a lull, he says, I'll show you a few messages.

These are like telegrams, on pale buff paper with strips of five-letter groups pasted on them. I gaze at them blankly.

You wouldn't think it, says Bob, but this is decoded German. That's the way they come from Hut 6. We have to emended, though, them, look, I'll do this one.

He takes his pencil and rapidly draws vertical lines between

Fig. 3.2 Christine Brooke-Rose, 'Typescript revisions to *Remake*', '*Remake*, early drafts Typescript with holograph revisions', Folder 4, Box 23, Christine Brooke-Rose Papers. Image courtesy of the Harry Ransom Center, The University of Texas at Austin.

begins and ends, what its narrative status is, and to whom it is attributed are often unclear. Dialogue also becomes a means of breaking out of a scene; *Remake* uses 'John' a querulous but undescribed character 'built by the house of fiction' as an interlocutor who constantly interrupts the story's telling.[14] 'What was the point, old lady?' he asks of the narrator, but it is up to the reader to respond.[15] In *Remake*, words are communicational objects to puzzle over. Are they description? Dialogue—whose dialogue? Directed at a character, the narrator, or the reader? Are they type on a piece of 'rough buff paper'?[16] Encrypted? Acronyms? Are they German, English? There is never perfect equivalence between languages because, as CBR seems to suggest, words function in multiple ways. The entire cryptanalytical endeavour is built upon a theory of transparent code that is, for CBR (and Coetzee), ultimately untenable.

This is surely why those sections of *Remake* that discuss BP are inordinately taken up with descriptions of the layout: of different huts, their connections to each other, the doors and hatches between them, and their ever-changing human occupants.

> Everyone moves out of the exiguous huts to a vast red brick building called Block D, a long series of cross-shapes, a wide corridor with right-angled narrower corridors and offices on either side, The Fusion Room is in Block G, behind. 3L now has three rooms, the largest for the girls.[17]

What are we to make of this? Why do we need to know about the layout of corridors? Such obsessive recounting of these spaces—there in the first draft of *Remake*—underlines that cryptanalysis is a material exercise, involving printed texts, paper tape, and inky fragments that are moved from one room to another in a complex system. This isn't immaterial calculation but material interpretation. The details about corridors also require that the reader mirror Tess's own job attempting to navigate and sift excess text for significance. 'Tess becomes familiar with individual units, types of reports and orders, but can't grasp the whole, can't bear not to grasp the whole' and so makes a chart, 'Mastering things'.[18]

Tess might master things, but we are left doubtful about the value of such exercises. Various critics have pointed out that such mastery is treated

[14] Brooke-Rose, *Remake*, 172.
[15] Brooke-Rose, *Remake*, 114.
[16] Brooke-Rose, *Remake*, 104.
[17] Brooke-Rose, *Remake*, 116.
[18] Brooke-Rose, *Remake*, 106–7.

suspiciously by CBR.[19] The BP narrative sections, for example, are interspersed with reflections by *Remake*'s 'old lady' narrator on the interpolating effects of media, of memory. We are repeatedly reminded that our knowledge of BP is itself the product of fallible personal memories and extant public records in various states of preservation and declassification. In this respect, the text mirrors CBR's own real-life correspondence with former BP colleague Jean Alington in the 1980s and 1990s as they dwelt repeatedly on their own attempts to sift and sort these records.[20]

Throughout *Remake*, human memory is repeatedly presented as interactive and medial, as akin to using a computer or card index, a metaphor that extends to the novel itself, with chapters described as 'files'. But *Remake* closes with 'the old lady' finally rejecting this medial comparison—in an astute observation, the memoir notes that the computer 'has memories but no sense of time'[21]—in favour of a cryptanalytical analogy. In a sentence that recalls *Watt* and looks forward to *Youth* in its logic-tree structure: 'Memory is more like intercepting and decrypting, thousands of messages missed, or captured but not decrypted, and even the captured and decrypted now burnt or not released.'[22] Notably, this is cryptanalysis that fails to yield full knowledge; mastery is not possible. The huts of BP become a confusing memory palace in which the reader—and Tess—gets lost.

Words in *Remake* are also medial, material. They are objects that the reader, like Tess, must navigate and interpret and assess for their narratological status. This is not the vision of language that information theory would put forth, even as *Remake* is partially set in the locale of its birth (or one of them). In fact, *Remake* sidelines the machines in favour of the words. The computers that would make BP so famous are only tangentially present here. Tess's unit does get a visit from the 'big boffins' from Hut 6, BP's 'inner sanctum', and a tour of Block F, home of the Colossus computer and its predecessor 'Heath Robinson', but these machines never take centre stage.[23] Tess's work is with index cards, not vacuum tubes. Later in the memoir, when Tess revisits the site of BP in the 1980s, Colossus is depicted as a relic

[19] Adam Guy, 'Under Suspicion: Christine Brooke-Rose, Intelligence Work, and the Theory Wars', *Modernist Cultures*, 16.4 (2021), 509–28; and Victoria Stewart, *The Second World War in Contemporary British Fiction: Secret Histories* (Edinburgh: Edinburgh University Press, 2011).

[20] The correspondence between Christine Brooke-Rose and her Bletchley Park colleague Jean Alington from the 1980s is full of references both to the many researchers pestering them and their own gaps of memory. 'Correspondence—Personal—Jean Alington Howard 1983–1991', Box 19, Folder 2, CBRP.

[21] Brooke-Rose, *Remake*, 170–1.

[22] Brooke-Rose, *Remake*, 172.

[23] Brooke-Rose, *Remake*, 114, 115.

that a museum is 'trying to reconstruct'.[24] The computer is one degrading memory object among many; the exciting work, the 'training of the mind', was to be found in the analysis of words by people. Words do many things in *Remake,* but what they don't do is programme computers.

Remaking Computing

Remake, although written with the hindsight of half a century of computer innovation and obsolescence, reflects its historical milieu accurately. In the 1940s, programming itself was not a concept; the idea of software and of programming languages had yet to be invented. It is a point that is often overlooked. If we close read early documents in the history of computing, we see a struggle to find a language to imagine and describe what would become known as programming. In the war years, the computer's operations are conceived in mathematical and engineering terms—as 'orders', 'instructions', and 'operations'.[25] Only later would the idea of the universal machine as a linguistic (rather than logical) device be assembled. Crucially for CBR's own interest, this idea would be developed via analogies with translation.

In 1947, American computer pioneers John von Neumann and Herman Goldstine were insisting that 'coding a problem for the machine' is *not* akin to:

> translating a meaningful text (the instructions that govern solving the problem under consideration) from one language (the language of mathematics, in which the planner will have conceived the problem, or rather the numerical procedure by which he has decided to solve the problem) into another language (that of our code).[26]

They argue that translation involves a static text—an assumption that CBR would surely reject—whereas machine coding entails a 'dynamic' relation

[24] Brooke-Rose, *Remake*, 134.

[25] John von Neumann, 'First Draft of a Report on the EDVAC' (30 June 1945), reprinted in *IEEE Annals of the History of Computing* 15.4 (1993), 27–75, 1 on original report pagination, http://web.mit.edu/STS.035/www/PDFs/edvac.pdf.

[26] Herman H. Goldstine and John von Neumann, 'Planning and Coding of Problems for an Electronic Computing Instrument', *Report on the Mathematical and Logical Aspects of an Electronic Computing Instrument*, Part II, vol. 1–3, Institute for Advanced Study, Princeton (1947), 1.

with the calculation being undertaken. Despite their disanalogy, Goldstine and von Neumann's suggestion that code might be construed as a language (akin perhaps to formal logic) and computer coding a kind of translation would find fertile ground in the years after the war.

It would rapidly gain traction in the early 1950s, thanks in part to Cold War efforts to facilitate machine translation, or translation between natural languages using a computer.[27] As early pioneer (and good friend of C. P. Snow), Warren Weaver influentially explained it in 1947, 'a multiplicity of language impedes cultural interchange between the peoples of the earth, and is a serious deterrent to international understanding'.[28] Weaver worried about the effects of linguistic mediation and the Tower of Babel—in this, he was symptomatic of a centuries old effort to construct a neutral, universal, and scientific language, begun (as Kenner noted) in the Enlightenment, invigorated in the artificial language movements of the nineteenth century, and gain further credence in the post-war years thanks, in part, to the promise that seemed to inhere in these universal machines.[29] Weaver pondered 'if it were unthinkable to design a computer which would translate'? Drawing on his knowledge of Allied wartime efforts in cryptanalysis such as CBR pioneered, Weaver had noted:

> one naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. When I look at an article in Russian, I say 'This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.[']'[30]

Weaver's notion that translation could be conceived as an exercise in cryptanalysis would be quickly taken up in the nascent sector, helping to position the universal machine as a demediating device and a tool to ensure world peace—what Brian Lennon would, rather unfairly, describe as an impulse in 'the neutralization of culture' (Weaver was also an avid supporter of the humanities and a respected scholar of the works of Charles

[27] See, e.g. Brian Lennon, *Passwords: Philology, Security, Authentication* (Cambridge, MA: Harvard University Press, 2018) and my 'Talk Interfaces' for a discussion of this.

[28] Warren Weaver, 'Translation' [1949] Memo. Reprinted in *Machine Translation of Languages: Fourteen Essays*, eds. William N. Locke and A. Donald Booth (Cambridge, MA: MIT Press, 1955), 15–24, 15.

[29] See Michael D. Gordin, *Scientific Babel: How Science Was Done before and after Global English* (Chicago: University of Chicago Press, 2015).

[30] Weaver, 'Translation', 18.

Dodson).[31] For all von Neumann and Goldstine's protests about the analogy, Weaver's contemporaneous insistence on the machine's potential translative application would fuel the idea that the computer was a universal communicational machine and world peace only a translation (calculation) away.

With this vision to drive (and embolden) them, the nascent industry would pursue the implicit link between machine instruction and natural language that the translation analogy seemed to suggest. The link would be bolstered, ironically, from the practical reality that the universal machine was proving anything but—as Coetzee would soon discover. The proliferation of early mainframe systems, all with their own machine-specific instructions, was highlighting the burden of this constraint: communicating in the machine's terms was just too difficult for the human user. Recognizing this, early pioneer Grace Hopper (whose sometime husband was, not coincidentally, an English professor) would make strides with developing machine-independent code and compilers; from 1954, a team led by John Backus at IBM was developing what would become the first widely used high-level programming language, FORTRAN—literally 'formula translation' as we saw.[32]

These practical developments, and the analogy between code and natural language that they operated upon, would be crucial for the sector. Aside from saving time and expense—and bolstering the industry's prestige through universalizing metaphors—the programming language created a new epistemic object around which the sector (particularly the emerging academic discipline) could organize.[33] Not only would programming languages simplify human–machine communication, but also computing now had a construct that could be critiqued and taught, and textual productions that could be authored, copyrighted, critiqued, and sold.[34] The concept would quickly take hold and the late 1950s saw 'explosive growth' in programming language development.[35]

[31] Lennon, *Passwords*, 56.

[32] See Jean E. Sammet, *Programming Languages: History and Fundamentals* (Upper Saddle River, NJ: Prentice Hall, 1969).

[33] Philipp von Hilgers, 'The History of the Black Box: The Clash of a Thing and Its Concept', *Cultural Politics*, 7.1 (2011), 41–58, 46.

[34] See e.g. David Nofre, Mark Priestley, and Gerard Alberts, 'When Tech Became Language: The Origins of the Linguistic Conception of Computer Programming, 1950–1960', *Technology & Culture*, 55.1 (January 2014), 40–75.

[35] Knuth, Donald, and Luis Trabb Pardo, 'The Early Development of Programming Languages', *A History of Computing in the Twentieth Century*, eds. N. Metropolis, J. Howlett, and Gian-Carlo Rota (Cambridge, MA: Academic Press, 1980), 197–273, 198.

Yet the result was not quite the universal language that everyone had predicted. If programming was an idea constructed to facilitate ease of communication, the emergence of high-level programming languages resulted, rather, in the industry's own 'Tower of Babel' (the image and expression would circulate widely).[36] The vision that computer science had built would also evolve: programmers contemplated how they might facilitate the 'use of English or anyone else's natural language' as a programming technique aiming to demediate human–machine interaction.[37] (We retain the legacy of that dream today in our obsession with speech interfaces and in the neocolonial position of English as the dominant language of programming globally.)

With long-term consequences, by the 1960s, computing had been remade. Programming had been established as a linguistic exercise—even as the associated vision of universal, unmediated communication was looking a little frayed around the edges.

So why does CBR not draw the conceptual analogy between language and computing in *Remake*? For a writer like CBR, fluent in three languages and adept at cryptanalytical procedures, the link surely seemed obvious, if anachronistic. The answer lies, I suggest, in CBR's dissatisfaction, in the 1990s, with what computing's remaking had left out. And to understand that dissatisfaction, we need to rewind to chart that half century between BP and *Remake*.

Coding Difficulty

The programming language endeavour was not something to which CBR was likely paying attention after being demobbed. After the end of the Second World War, computing would be a nascent industry hampered by official secrets, poor communications, and, in Britain certainly, an increasing hostility towards women.[38] CBR, meanwhile, had chosen a path that cleaved to the other of the Two Cultures.[39] After the war, after a more official training

[36] The January 1961 cover of the *Communications of the ACM*, 4.1 famously featured a programming Tower of Babel on the front; Jean Sammet, *Programming Languages*, the industry-standard book on the topic, had a similar picture on the cover.

[37] Jean E. Sammet, 'The Use of English as a Programming Language', *Communications of the ACM*, 9.3 (March 1966), 228–30, 228.

[38] See Hicks, *Programmed Inequality* for how British policy effectively sidelined women across the entire sector.

[39] Albeit she would describe experimental literature as an attempt to 'narrow that gulf' between the sciences at the arts in 1965, Christine Brooke-Rose, 'Dynamic Gradients', *London Magazine* (March 1965), 89–96, 89.

of the mind at Somerville College, Oxford, and University College London, CBR would embark on a career as a novelist, critic, and academic in Britain and then in France.

Despite not engaging actively with the practicalities of programming in this period, CBR would pursue her interest in 'mastering things' through immersion in French structuralism.[40] Crucially, structuralism (and later 'French Theory') was developing as part of an interdisciplinary endeavour rooted in information theory and that would conceive of language—and culture—as code, or system.[41] Across her career as an academic and reviewer, CBR would closely engage with the intricacies of these theories in her analysis of narrative. Essays such as 'The Encoded Reader' or 'The New Science Fiction' acknowledge the resemblances and distinctions between linguistics, information theory, and computer programming. In 'The New Science Fiction', which analyses Joseph McElroy's experimental novel *Plus* (1977), CBR notes that although a computer is represented in the novel, this is not 'computer language'; McElroy 'has in fact used [Chomsky's theory of deep structure of sentences] to generate a narrative poem'.[42]

Such distinctions are the mark of CBR's thinking about programming in the years before she was herself using PCs. In her fiction from the 1960s onwards, we see a similar interest in pursuing the limitations of equating 'computer language' with universal code. What holds CBR's interest in the 1960s is precisely the points where that computational and structuralist dream of code abstraction seems to fail—and in particular, how that failure might be aesthetically generative.[43]

CBR's fiction has often been described as 'difficult'.[44] Descriptions of her style recalls the accusations made against Kenner's criticism: 'resplendently unreadable' is how one reviewer put it.[45] In the 1960s, following a few works

[40] Christine Brooke-Rose, 'Splitlitcrit', *Invisible Author: Last Essays* (Columbus: Ohio State University Press, 2002), 23–4.

[41] Bernard Dionysius Geoghegan, *Code: From Information Theory to French Theory* (Durham, NC: Duke University Press, 2023).

[42] Both essays are collected in Christine Brooke-Rose, *A Rhetoric of the Unreal: Studies in Narrative and Structure, Especially of the Fantastic* (Cambridge: Cambridge University Press, 1981), 288.

[43] Christine Brooke-Rose's interest in structuralism has often been depicted as focusing on the literariness of scientific discourse broadly (e.g. in Sarah Birch, *Christine Brooke-Rose and Contemporary Fiction* (Oxford: Clarendon Press, 1994)). I am suggesting that it is often more particularly focused on structuralism's orientation towards information theory.

[44] Rebecca Pohl gives a good summary of this career-long characterization of CBR's work in 'Selling Difficulty: The Case of Christine Brooke-Rose', *Textual Practice*, 32.2 (2018), 283–99.

[45] Francis Hope, 'I, Julian', Collective Book Review, *The New Statesman*, 68 (13 November 1964), 741–2, 742.

composed in relatively conventional prose, CBR would make a name for herself as one of the foremost champions and authors of experimental fiction in the French tradition of *Le Nouveau Roman,* or the 'anti novel' in Britain. Her own novels from the 1960s onwards often featured excessive wordplay and constraint. She shared a fondness, with Kenner and Coetzee, for late Beckett and his combinatorial interest in words. Her writing often deploys constraint and permutations, exposing in the process the mechanics of grammatical system. But while Kenner was enthralled by the programming logic that Beckett's writing seemed to embody and Coetzee by its resistance to stylostatistical methods, CBR seemed most captivated by Beckett's propensity to conceive of these linguistic units as physical objects. Critics often speak of the word in her writing as 'like a mechanical construction, or an electrical circuit'.[46] They are designed to do things.

The difficulty of CBR's work stems specifically from this propensity to examine the affordances of words. Such examination results in fiction full of puns that both 'cultivate and repel the reader's desire for meaning, in a way that seems impervious to interaction'.[47] Her fiction explores how these components might be strung together and combined in different ways, with different effects in different environments. Her sentences collectively foreground interpretative uncertainty, untranslatability, and functional ambiguity.[48] CBR's writing both invites and resists the kinds of decoding on which the BP enterprise and the dream of the universal machine were built. This isn't art in a closed field, but rather art in the substrate. Difficulty is not produced from inexpertise or as a buttress against computational thinking; instead, it is created through CBR's exploitation of the multiple affordances that words can be perceived to have in different contexts.

We see this technique developing across CBR's fiction and criticism in the 1960s. Often in her novels, it is conventions around conversation (the 'transparent' human–computer interaction interface of choice today[49]) that are exploded, something that she and her critics have often positioned in

[46] Joseph Darlington, *Christine Brooke-Rose and Post-War Literature* (Basingstoke: Palgrave Macmillan, 2021). British Library E-book: n. pag. (introduction). See also Natalie Ferris, *Abstraction in Post-War British Literature 1945–1980* (Oxford: Oxford University Press, 2022).

[47] Julia Jordan, 'Are Puns Mere? Christine Brooke-Rose's *Amalgamemnon*', *Modernist Cultures*, 16.4 (2021), 529–45, 530.

[48] Emily Apter, *Against World Literature: On the Politics of Untranslatability* (London: Verso, 2013).

[49] Lori Emerson, *Reading Writing Interface* (Minneapolis: University of Minnesota Press, 2014), intro.

terms of an interest in exploring the limits of realist dialogue conventions.[50] Certainly, she would applaud practitioners of *Le Nouveau Roman* for presenting characters' thoughts and conversation 'interleaved with no narrative comment', which, while requiring 'an initial effort of concentration', is followed by 'the pleasure of recognition'.[51] We might also understand CBR's interest as responding to the contemporary trend towards conceiving of conversation in terms set by information theory—as constrained turn-taking, immaterial message exchange, as code.[52] In contrast, CBR repeatedly resists the reader's attempts to conceive of words, and dialogue particularly, as transparently executable.

Her novel *Out* (1964) is set in a post-apocalyptic African state, one that most immediately invokes apartheid South Africa, but that utilizes an inverted racial logic wherein the 'colourless' population is ruled by a Black elite.[53] In a Sci-Fi–inflected and uncomfortably inverted premonition of Coetzee's *Michael K* (complete with gardening tropes and institutional surveillance), conversation often takes the form of state interrogation. Individuals are questioned at the labour exchange, in their interactions with powerful figures, and during the medical procedure of 'psychoscopy', which is part talking cure, part lie-detector test.[54] Like Coetzee, CBR presents conversation as amenable to binary logic, but she also demonstrates that it is enmeshed in the physical world of bodies, brains, and physical environment:

[50] Matthew James Vechinski, 'Christine Brooke-Rose and the Science of Fiction: Rethinking Conceptual Writing and Artificial Intelligence in *Xorandor*', *Critique: Studies in Contemporary Fiction*, 57.4 (2016), 448–63.

[51] Christin Brooke-Rose, 'Dinner with M. Carnéjoux: New Novels', *Observer Weekend Review* (8 March 1964), Books Section, 27. Much of the discussion of this review focuses on the comparison she makes with an electron microscope as evidence of her interest in how scientific advances might offer a new language for fiction. I find the discussion of 'subconversation' intriguing for its implicit recall of the language of human–computer interaction specifically.

[52] N. Katherine Hayles discusses the intellectual consequences of information theory's bracketing off of the interpretative acts conducted by message senders and receivers (i.e. the interlocutors) in *How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics* (Chicago: University of Chicago Press, 1999).

[53] Patrick Burley makes a compelling case that the novel, in 'coopting the marginalization of postwar migrants and colonized people in order to undertake a broader exploration of narrative and human incoherence…betrays the partiality of its organizing concepts that, within the dominant racial paradigm, instantiate whiteness as symbolic of humanity', 'Whiteness, Displacement, and the Postimperial Imaginary in Christine Brooke-Rose's *Out*', *MFS Modern Fiction Studies*, 66.2 (Summer 2020), 371–95, 392.

[54] Christine Brooke-Rose, *Out* [1964], *The Christine Brooke-Rose Omnibus* (Manchester: Carcanet Press, 1986), 132.

> The vibration of the voice has not been sufficient to carry the witticism into the neighbour's left ear, and the syllogism evaporates, leaving no trace of error either in the air or in the mind, except perhaps a residue at the back of the brain, greenish in colour.[55]

We have the possibility of messages carried and errors returned—we have conversation presented in terms of information theory—but we also have the representation of what that theory leaves out. (In later work, she would make her frustrations explicit: cybernetics descends into 'hyberenetics…a World Wide Wank'.[56]) The reader is left at an interpretative impasse, unable to complete the calculation precisely because conversation escapes the abstractions put forth by information theory and machine translation.

Similarly, we see the reader pushed to conceive of conversation computationally in an exercise that simultaneously denies formal executability. Like Beckett's sentences, CBR provides a 'multi-choice structure' below the surface of the text.[57] In *Out*, the narrative often creates IF statements that are unresolvable: 'The conversation cannot take the form of the hammering because during the hammering there is no conversation, and during the conversation, if it occurred, there was no hammering.'[58] Elsewhere, the novel loops through conversational permutations, like a programme, without coming to an end.[59] We are stuck in the loop of Coetzee's recursive reading; the dichotomy between conversation's implied executability and its narrative function within the text generates literary difficulty.

Similarly, in CBR's next novel *Such* (1966), a work that 'applies the lessons of empirical science to correct the flawed assumptions of literary realism', dialogue might seem to operate in the imperative mood, but the reader is never allowed to rest in the certainty of that grammatical state.[60] The novel seems (but it is difficult to tell, so restricted is the exposition) to begin with the protagonist, Laurence, having a heart attack, before 'a three-minute heart massage is developed into a poetic and funny narrative' of his

[55] Brooke-Rose, *Out*, 165–6.
[56] Christine Brooke-Rose, *Life, End of* (Manchester: Carcanet Press, 2006), 64.
[57] David Hayman, Keith Cohen, and Christine Brooke-Rose, 'An Interview with Christine Brooke-Rose', *Contemporary Literature*, 17.1 (1976), 1–23, 2.
[58] Brooke-Rose, *Out*, 108.
[59] As Brooke-Rose would later write, in the context of media, 'Looping also has a meaning in linguistics, to do with the deep structure of subordinate clauses, now forgotten and called embedded sentences', *Life, End of*, 53.
[60] Joseph Darlington, 'A Non-Euclidean Novel: Christine Brooke-Rose's *Such* and the Space-Age Sixties', *Journal of Modern Literature*, 40.2 (2017), 147–64, 161.

memories or fantasies.[61] The status of the narrative itself is in question. Elsewhere, as part of a larger exploration of the limits of literary convention in an era of scientific innovation, the book features a scene between Laurence, psychiatrist to astrophysicists, and his wife Brenda, a computer operator, in which programming becomes a means by which two people try but fail to communicate:

> Alarm Reset, Next Instruction and the clatter of binary arithmetic, oh hello Laurence, how did you get in?
>
> – Brenda, I must talk to you.
>
> – Not now, Larry, not now.
>
> She pulls down switches quickly, competently and jots down her comments on a pad in a column called Narrative, figures in the column called Location, a tick under Next Instruction.
>
> – Brenda –
>
> – Won't it wait, Larry?
>
> – I've remembered something.
>
> – What?
>
> – Something…of the narrative, the location.
>
> – Oh, go away, Larry. Can't you see we must finish this programme.
>
> – What programme?
>
> – Track, eight four two one. Syne shot. Block Prime Pulse Mesh.
>
> – When did I marry you, Brenda?
>
> – Oh go away, Larry, for heaven's sake, go.[62]

We see here the same technique that will become a hallmark of CBR's BP scenes in *Remake*. The executability of the words, their functional status, is unclear for the reader—what these words 'do' in the narrative is unclear because we don't have the usual punctuation signs that mark them as speech or description. We also see programming impeding natural language communication between people. 'Location' and 'Narrative' describe specific classes within a computer database, but Larry takes them as generalizable narratological descriptors that might be applied to his own existence; he equates the terms with life stories, not data structures. The couple's perspectival disconnect and failure to converse are represented here by the false

[61] Christine Brooke-Rose, *The Christine Brooke-Rose Omnibus*, back cover.
[62] Christine Brooke-Rose, *Such*, 326–7.

cognates that exist between code and natural language. The reader meanwhile might wonder whether these are metatextual statements on Larry's own existence as a character. The conflation of dialogue with code works here to undermine any claim that language—natural and literary—might be straightforwardly executable in the sense that computer scientists might understand the term. It is the disanalogies between programming languages and natural languages that interest CBR here and difficulty becomes the mark of languages' multiple affordances.

In her next novel, CBR would undercut a different facet of the dream of universal (language) machines. *Between* (1968) is set across a series of international congresses, popular in the post-war years and sponsored by organizations like UNESCO and the big American foundations, that were designed to foster world peace through better communication. The book follows the professional translators, through airport lounges, conference halls, and snatches of conversation as they conduct 'simultaneous interpreting'.[63] In this book, CBR wittily punctures Weaver's vision and 'the expected English syllables of problems we should consider today for the sake of mutual understanding the advancement of learning the true state of things'.[64] Cross-cultural conversation in *Between* is a series of 'endless permutations', and the congresses depicted in the novel are satirized for their failure to enable the successful intercultural exchanges they promise.[65] Each speaker is a batch-processing machine, 'more concerned with output than intake'.[66] For the reader meanwhile, the narrative becomes increasingly difficult to follow as the prose switches between different natural languages and symbolic systems, between theories of communication (from semioticians, cyberneticians, and others), and between characters' perspectives, without ever utilizing the verb 'to be' and the stability it implies.[67] As the novel itself notes, it is a veritable 'seven-terraced tower' of Babel.[68]

CBR's satire in *Between* is directed very much at the Tower of Babel and against the notion that a universal language machine might be feasible, or even desirable. The distinction between types of communication systems—formal and natural languages—also breaks down. There is the reported threat that human translators might be replaced by 'gorgeous young computers'

[63] Christine Brooke-Rose, *Between* [1968], *The Christine Brooke-Rose Omnibus*, 418.
[64] Brooke-Rose, *Between*, 407.
[65] Brooke-Rose, *Between*, 415.
[66] Brooke-Rose, *Between*, 422.
[67] Christine Brooke-Rose, 'Remaking: The Solothurn Lecture', 7, Acc 8, Box 4, 'Brooke-Rose, C.', Archives of Carcanet Press, University of Manchester (hereafter ACP).
[68] Brooke-Rose, *Between*, 511.

but CBR plays on the ontological ambiguity of the term and we are never entirely sure if those computers are human or machine.[69] In *Between*, programming simply exacerbates Babel.

Yet for all the babeling, what remains constant in *Between* and across CBR's early experimental fiction is that the literary text can, as Kenner's mechanical analogy implied, act as a substrate in which all other linguistic systems can be executed. Her fiction generates meaning in the untranslatable terms, in the gaps between these different linguistic systems and the points where the relationship between sign and signifier (whether a computer is a 'who' or a 'what', for example) is strained. It isn't the computer that is the universal machine but the literary text.

This is a point really brought home in the final work of CBR's first tetralogy, *Thru* (1975). A continuation of this interest in language systems, *Thru* is set, somewhat notoriously, at the epicentre of 'French Theory'. In the aftermath of the May 1968 student protests, CBR had been recruited by the renowned writer and deconstructionist critic Hélène Cixous, to work at the newly established and experimental University of Paris VIII, Vincennes. Like Bletchley Park, this would be an intellectually generative environment. Her colleagues and contacts there were a veritable who's who of French intellectual life: Luce Irigaray, Jacques Lacan, Gerard Genette, Gilles Deleuze, Michel Foucault, and others.[70] Here CBR was at the centre of debates about structuralism and from where much post-structuralist thought would emerge.

Thru represents this environment by depicting extensive classroom discussions, academic papers, and department meetings that debate academic theorizations of language. Notably, these debates are represented in a montage of different semiotic modes: tables, diagrams, flow charts, typographic symbols, reproductions of handwritten annotations and sixteenth-century grammars, experiments with layout that resemble concrete poetry, and mathematical functions. We have cybernetic theory juxtaposed with structuralism, Bakhtin with Kristeva, and Chomskian grammar with classical rhetoric. Meanwhile, unmarked and unattributed dialogue is interwoven with these semiotic systems and theories. *Thru* is a dazzling, difficult

[69] Brooke-Rose, *Between*, 501.

[70] For a discussion of this experience and its impact on *Thru*, see Joseph Darlington, 'The Composition of Christine Brooke-Rose's *Thru*: An Afterlife of May '68', *Journal of European Studies*, 45.1 (2015), 49–64.

exposition of literature's ability to encompass and execute linguistic multiplicity, while also generating interpretative incompatibilities.

In *Thru* and in the other novels of her first experimental tetralogy, programming languages are not the major focus by any stretch. But we do see a broad interest in examining linguistic operations and particularly those that computational abstractions often ignore or control. In CBR's fiction, programming languages are far from the (promised) universal code that their promoters declare. While Hugh Kenner might have drawn an equivalence between modernist literature and computing on the basis of difficulty, in CBR's fiction, difficulty is the by-product of an effort to illuminate literature's function as a universal coding environment—as the substrate (Weaver would envision a 'common base'[71]) of the Tower of Babel.

Unfortunately, readers resisted this difficulty. Reviews of *Thru* were poor; too difficult was the charge. It was CBR's last attempt to publish a novel for almost a decade. When she did return to fiction, however, she would use it to articulate a more explicit narratological theory of programming languages.

Narratologies of Programming

In 1984, while Kenner was making *Travesty*, CBR began a productive publishing relationship with Carcanet, a small Manchester-based press with a reputation for publishing poetry. Carcanet would later publish *Remake*, but they began by bringing out three 'computer' novels: *Amalgamemnon* (1984), *Xorandor* (1986), and *Verbivore* (1990).[72] Later these titles would be bundled with a fourth, *Textermination* (1991), which would collectively become known as CBR's 'intercom quartet'. CBR's explicit focus on computing in these novels mirrors a growing public familiarity with the technologies in this period—this was the era of the PC Revolution after all, along with Kenner's redoubtable efforts to broadcast that—but it also evinces her own increasing familiarity with word processing software and an expanding awareness that computer programming's linguistic dimensions might prove interesting fodder for literary experimentation. Just as her thinking around

[71] Weaver, 'Translation'.

[72] Christine Brooke-Rose to Michael Schmidt (8 July 1990), Acc 4–3, Box 3, 'Incoming Letters: Christine Brooke-Rose', ACP.

languages was shaped by her personal history, so to her thinking about programming languages would be informed by her personal computer history.

If we look to Carcanet's archives and to CBR's personal papers, we find details of her equipment, habits, and attitudes towards word processing software. Despite her early work at BP, CBR was not a pioneer of computer use in the way that Kenner or Coetzee would be. *Amalgamemnon* drafts were, for example, written by hand and later typed on a typewriter. Yet a slow practical start in personal computing would not stop her pursuing a personal interest in narratologies of programming.

CBR's growing interest in programming in the 1980s is no doubt partly the result of the cultural context in which she was embedded and the peculiarly French vision of 'télématique' or 'telematics', which yoked telecommunication and computing. The computer sector in France was, like in Britain, heavily dominated by US companies and also comparably late to the game. In the mid-1960s, Charles de Gaulle instigated a national strategy, *Plan Calcul*, to support an indigenous computer industry, with mixed success. One result was the Minitel system.[73] Minitel was, in effect, a national computer network and 'a technical marvel, commercial success, and ambitious social experiment' instigated before the development of the World Wide Web in the early 1990s.[74] Under the scheme, the French state provided, free of charge, a Minitel terminal (keyboard, screen, and modem) to any telephone subscriber who requested one, allowing them to connect to a national network. The terminal used videotex but could also function as a modem for personal computers; Minitel was also interactive and users could transmit data as well as receive it (in contrast to the UK's CEEFAX service). In the 1980s, terminals would proliferate in homes, offices, public venues, and even shops: ten years before the Web, CBR was living in a country in which telematics were mundane.

Given that background, it is no surprise that her novels in the early 1980s are grappling with the impact of computing on communication. *Amalgamemnon* takes the form of a monologue by Mira Enketei, a History and English professor, obsessed with the classics, who is facing a computationally driven world in which her discipline and she shall 'soon be quite redundant'.[75] Computation is a threat to humanity. From the get-go,

[73] See James W. Cortada, *The Digital Flood: The Diffusion of Information Technology across the U.S., Europe, and Asia* (Oxford: Oxford University Press, 2012), 112–24.

[74] Julien Mailland and Kevin Driscoll, *Minitel: Welcome to the Internet* (Cambridge, MA: MIT Press, 2017), 1.

[75] Christine Brooke-Rose, *Amalgamemnon* [1984], paperback edition (Manchester: Carcanet/Dalkey Archive, 2014), 5.

Amalgamemnon sets up a series of binaries: between the humanities and science and technology, between women and men, and between developing and developed countries. Yet when we look to the narrative, these binaries start to break down; as one reviewer noted, the novel was 'too inventive and lively to be merely another jeremiad on computers'.[76] The novel is written, as CBR would later explain, 'entirely in the future tense, or other non-realizing tenses like the imperative or the conditional' as a means 'partly to explore the pseudo-future of forecasts and speculation we all live in, but mostly as an anti-narrative device that avoids all declarative sentences'.[77] If programming languages insist on imperative statements (and the occasional declarative), here CBR examines what it might mean to author a book in which 'sentences of which it can be said that they are either true or false' do not appear.[78] This is literature conceived in opposition to the tense of the programming language.

CBR emphasizes this opposition further via her literary allusions. Mira is a modern-day Cassandra, the figure from Greek myth, daughter of the king of Troy, enemy of Agamemnon, and doomed by the Gods to speak prophecy that is not believed. In a novel constructed of 'non-realising tenses', Mira rails against the prostration of culture to computation's future orientation.[79] Greek prophecy is presented as the classical equivalent of today's computational orientation towards the non-realized. While one is associated with irrationality and a 'dead' culture and the other with mathematical probabilities and technological innovation, both are nevertheless predictive statements. Indeed, Mira spends much of the novel questioning the purported distinction between the two. She has an admirer who works with the 'National Education Computer' and is often referred to as Willy (thanks to his insistent use of the tense). He declares that: 'The new generation will supertouchtype programmes and games all to be superdevised by an elite of supertechnicians of communication I'll show you after hours he'll say.'[80] But Mira reflects, 'Probably that would make the new generation the new high priests and oracles of pythian mysteries none will control himself or understand further'.[81] The new technologies might promise 'super' much, but they operate in the paradigm of the imperative.

[76] Brian Morton, 'A Glimpse into the Future Tense', Review of *Amalgamemnon*, *Times Higher Education Supplement* (5 November 1984), 10.
[77] Brooke-Rose, 'Remaking', 7.
[78] Christine Brooke-Rose, 'A Writer's Constraints', *Invisible Author*, 9.
[79] Brooke-Rose, 'Remaking', 7.
[80] Brooke-Rose, *Amalgamemnon*, 6.
[81] Brooke-Rose, *Amalgamemnon*, 6. Karen R. Lawrence sees *Amalgamemnon* as offering prophecy that resists predictability, *Techniques for Living: Fiction and Theory in the Work of Christine Brooke-Rose* (Columbus: Ohio State University Press, 2010), Chapter 5.

This point is underlined in CBR's consistent juxtaposition of Willy's insistent use of 'will' within a context of readerly uncertainty:

> The problem of automation, Willy my present Amalgamemnon will announce between the shrimp cocktail and the fricandeau, will be what to do with the totally brainless. Perhaps I might say to respond a whole series of jobs will have to be kept…But I might not reply since brains such as mine will soon be declared as redundant as some brawn, unless to be recycled into a text-processor at some future date we must hold back, or forth, as long as possible. Or else he will…[opine] and I shall tenniswatch his teleprompting eyes and mimagree, unless I take the other side…
>
> Meanwhile he'll speak without thinking and I'll speak without speaking or vice versa but never quite together since my speaking to any purpose other than sado-elegant pub-lounge platitudes will already begin to make him angry, or at least uneasy.[82]

The multiple conditionals seem to present the prose as logically connected, as algorithmically structured. Yet what has actually occurred? Recalling CBR's techniques in *Out* and *Such*, the reader is faced with significant uncertainty as the prose spools out into many possible paths. Nor are we able to clarify the narrative status of the words themselves; CBR again avoids realist dialogic cues that 'mark author-presence'.[83] In *Amalgamemnon*, computation's imperative paradigm is impossible to maintain.

Narratological Codework

CBR's next novel would offer her most explicit narratological reflection on programming. It would also be largely drafted in a word processing environment. Appropriately for CBR, the environment would be its own technological Tower of Babel: drafts of what would become *Xorandor* were produced on an Adler electronic typewriter attached to a ScreenTyper. This machine was something of a transitional technology, halfway between a typewriter and a personal computer. It offered limited programming

[82] Christine Brooke-Rose, *Xorandor* [1986], *Xorandor/Verbivore* (Great Britain and Singapore: Verbivoracious Press, 2014), 22.

[83] Christine Brooke-Rose, Replies to Mark Bryant Queries (1984), Acc 3, Box 75, Folder 1, 'Correspondence and Specifications for Books Already Produced', ACP.

functionality as an add-on to the electronic typewriter offering: the ScreenTyper 'makes word processing as easy as typing' as one brochure declared.[84] It utilized a CP/M operating system (like many of Kenner's Heathkits), had an 8-bit operating system, and relied on 5.25-inch floppy disks as standard. The machine was clearly aimed at an office setting and gendered female; nevertheless, it no doubt appealed to CBR thanks to its editorial capabilities: 'characters, words, lines, sentences and even whole paragraphs or pages can be inserted or deleted'.[85] On this system, an amalgam of old media and new programming, CBR would produce drafts of *Xorandor* in 1985.

In *Xorandor*, CBR would represent programming languages directly, offering an extended and visible example of 'codework', namely code featured in literature for aesthetic effect. The clue is in the title: *Xorandor* (1986) contains a programming pun within its title with XOR denoting exclusive logic and AND/OR non-exclusive logic. The novel would be CBR's most successful; its popularity is often attributed to *Xorandor*'s science fiction features. Certainly, the word play is generally restricted to the codework and there is a recognizable plot: twins, Zab and Jip, aged thirteen-and-a-half, discover a computer-like, radioactivity-eating, possibly Martian-originating, silicon-made being, who is disguised as a rock, and begin to communicate with it. The discovery of 'Xorandor' and later, other such 'alpha guys', who seem to consume nuclear waste, results in planetary crisis.

As we might have come to expect from a CBR narrative, much of the novel is taken up with reflecting on the nature of languages—programming, literary, and natural. CBR retains the oppositions she laid out in her earlier work—the humanities, as Mira predicted, are no longer taught at school, unlike science and 'hitech'.[86] However, while literary inquiry might not have retained its academic standing in this story, both it and programming are positioned as part of a larger communicational universe that includes storytelling, 'baby talk', and natural language multilingualism. Jip and Zab are compiling their story in Germany (where they have been sent in the aftermath of their conversations with Xorandor, which bring them worldwide attention) and struggle with the 'horrid business, just like Xorandor at the beginning, of not understanding one kiloword of what was being said around us, in class especially, except of course maths and physics and

[84] Adler, 'The Adler ScreenTyper' Sales Brochure, n.d. https://etzoneorg.files.wordpress.com/2020/07/4e598-adler-screen-typer.pdf.

[85] Adler, 'The Adler ScreenTyper' Sales Brochure, n.d.

[86] Brooke-Rose, *Xorandor*, 8.

chemistry and compsci'.[87] With natural language on the one side and artificial and formal languages on the other, the binary is undercut before it begins thanks to the comparison 'just like Xorandor' at the start of that sentence.

In fact, the entire novel functions as an exercise in considering the narrative potential of computational language. Language here is also conceived in dialogic terms—the entire novel is in this form of 'flipflop storytelling,'[88] and a direct response to CBR's own frustration with science fiction's over-reliance on clunky exposition.[89] But dialogue, as we have seen, is never a sure thing in CBR's work and it takes many forms here: represented via transcripts of tape-recorded conversation, captured on 'Sneaker' devices and 'watchsized' gadgets, or conducted via 'softalk' (face-to-face dialogue).[90] Programming languages become dialogic interfaces, which are notably no more or less (un)mediated than other forms and no more or less imbued with narrative executability. The novel opens with the amalgamated 'Jipnzab' debating how best to narrate their tale because 'it seems harder to tell a story, even our own, than to make up the most complex program. Or at least choose how to tell it'.[91] They explicitly contemplate the question of how the writing of a novel might differ from writing a programme and how features might be conceived differently: is narrative delay, for example, 'suspense' or 'waste instruction'?[92]

The novel stages this set of questions through its deployment of codework. Programming statements are interspersed among natural language: 'If the storytellers are characters THEN their confusion is part of their characters ENDIF. Inversely IF the characters are the storytellers THEN the confusion is part of the story ENDIF.'[93] How is the reader meant to interpret that string? The capitalization distinction between the first and second 'if'/'IF' suggests that there is an important functional distinction to be made, yet grammatically they seem to be operating identically. The reader is also asked to attend to layout as a functional feature; when the novel was being set by the printers, CBR was insistent that codework indentation was significant, if 'arbitrary'.[94] Demonstrating an awareness that in programming

[87] Brooke-Rose, *Xorandor*, 9–10. [88] Brooke-Rose, *Xorandor*, 5.
[89] CBR's first version, 'The Alpha Guys' supposedly authored by 'Hunter Brooke', was more conventional in style and narrative form. It also didn't include any codework. See Christine Brooke-Rose, 'The Alpha Guys [Xorandor]. Holograph ms in 5 numbered notebooks', Box 10, and 'The Alpha Guys [Xorandor] 2 vols. Photocopy TS, bound', Folders 1 and 2, Box 11, CBRP.
[90] Brooke-Rose, *Xorandor*, 45, 101. [91] Brooke-Rose, *Xorandor*, 6.
[92] Brooke-Rose, *Xorandor*, 6. [93] Brooke-Rose, *Xorandor*, 9.
[94] Christine Brooke-Rose to Robyn [Marsack] (10 March 1986), 1. Acc 3, Box 237, Folder 1, 'Christine Brooke-Rose Corres', ACP.

indentation can convey meaning, CBR demands that the reader employ those same code-reading practices in the context of a novel; she asks that we consider reading like a machine.

Elsewhere, whole sections of chapters are related in an invented programming language, 'Poccum 43'. CBR based this closely on contemporary programming language structures, as articulated in the source she mentions, Terrence W. Pratt's popular textbook, *Programming Languages—Design and Implementation* (1975 and released in multiple editions thereafter). Pratt provides a gentle introduction, aimed at undergraduates, to the central concepts in programming. He includes chapters on computer structures that dictate how languages are processed, on data types, on syntax and translation, on operating environments, and on other topics, before devoting Part 2 to the discussion of common programming languages. CBR would likely have found the approach appealing. Pratt discusses programming language design in terms familiar to linguists—'syntax', 'semantics'—and argues that design and implementation has been 'more art than science.'[95] Pratt mentions floating point and arrays and mathematical concepts, but his interest is in determining how the design affects what the programming language is able to do and how it is implemented. This is analogous with CBR's own conception of languages and their function in the literary text.

Looking to her notebooks on Poccum 43, we see that CBR was conceiving of codework very much in this light. The notes show that she invented operations and painstakingly annotated them in programming language terms. Thus 'anticoincidence gate' is noted to be a 'non-sin circuit = XOR gate' and 'pushdownlist' is tagged as 'invented'.[96] But while CBR was clearly interested in programming languages collectively as a distinct category of languages, she was also interested in how they might operate within literary narrative and on a continuum with other natural languages. When listing Poccum 43's operators, she titled them 'VOCAB' and classified them under the headings 'NOUNS', 'VERBS' 'ADJ' and 'ADV'.[97] Programming languages have been reconceived in terms familiar to a narratologist: she doesn't speak of functions but of 'VERBS'.

This interest in contemplating the affordances of programming languages within a narratological framework continued. When the novel was set by

[95] Terrence W. Pratt, *Programming Languages: Design and Implementation* (2nd edn, Englewood Cliffs, NJ: Prentice-Hall Inc, 1984), xiii.

[96] Christine Brooke-Rose, 'Notebook containing holograph manuscript scientific notes written by Thomas Blackburn, holograph list of chapters, holograph list of vocabulary, and additional holograph revision pages, undated', Box 11, Folder 5, CBRP.

[97] Christine Brooke-Rose, 'Notebook', CBRP.

the printer, CBR was particularly concerned with Poccum 43's function within the text—this imagined programming language was not inherently distinct from literary language in the way it might be executed by a reader. She complained of the proofs that 'All the computer conversations are haywire,' thanks to setting errors and insisted on typographic changes, as 'the conversations are difficult to read anyway, for anyone not used to computers, so we shouldn't make it worse'.[98] (From elsewhere in her archives, it is clear that CBR was less enthused with Babel when it was the product of file compatibility issues.) This is a programming language presented as codework for a human, and not a machine, reader.

In fact, CBR leaves the reader guessing at the end of the novel. *Xorandor* concludes with a promise by Jip and Zab that they will 'scratch' (i.e. delete) the entire narrative on the morrow (again action is in the future), but which is followed immediately by a final bolded, left-aligned statement:

ENDXORANDOR[99]

What do we make of this? The statement seems to act as code, an instruction to end a programme. This suggests that there will be no tomorrow on which the scratch can occur—that is why we are able to read this novel—but also implies that the entire novel is a programme. Despite such ontological obscurity, this final piece of codework is also a signal of the literary text's own expansive environment for it does little to affect the reading experience that has gone before. CBR here resists code's claims of exceptionalism, and the common argument that 'Unlike any other law or performative utterance, code almost always does what it says because it needs no human acknowledgement'.[100] Instead, the reader takeaway is that programming languages are language, code is human read, and literary execution trumps that of the machine.

CBR was proud of her computer novel. She suggested that it be advertised within 'Home computer reviews' such as Kenner would write for.[101] *Xorandor* was indeed well-received in those and other venues, but not perhaps in the terms CBR had intended. Specifically, the reviewers for

[98] Christine Brooke-Rose to Robyn [Marsack] (10 March 1986), Acc 3, Box 237/1, 'Significant A and B correspondents', ACP.

[99] Brooke-Rose, *Xorandor*, 206.

[100] Wendy Hui Kyong Chun, *Control and Freedom: Power and Paranoia in the Age of Fiber Optics* (Cambridge, MA: MIT Press, 2006), 66.

[101] Christine Brooke-Rose to Michael Schmidt (12 October 1985), Acc 9, Box 2, 'Christine Brooke-Rose corres', ACP.

scientific, general, and literary publications failed to recognize the codework and its function within the novel. For some, the work 'maintains the delicate balance between fertility of invention and strict economy of means' with the result it was, in contrast to the 'turgid Joycean' of *Amalgamemon*, 'so readable'.[102] For others, the narrative was 'littered with computer-speak' and jargon, with 'computerese' used for 'local color', resulting in incomprehension for the novice reader.[103] For others still, the deployment of programming terms has the effect of indicating a 'continuum' between literature and programming: 'All languages, she implies, require attention to meaning, a decoding of logical systems.'[104] Intriguingly, the professional readers of *Xorandor* differed wildly in their interpretation of the effects of the codework: responses to the book inadvertently make CBR's point for her that programming languages, like all languages, have multiple possible functions, all executable within the literary environment. In the literary substrate, code is never transparent.

The Universal Environment

Despite using codework in *Xorandor*, CBR's own theoretical interest was not yet matched by practical usage. But this would shift. For her next novel, *Verbivore*, CBR would initially use ScreenTyper, before moving over in about 1987 to an Atari ST personal computer, which came with *ST Writer*, a word processor programme produced by Atari. She attributes the move to the 'many disadvantages' of the ScreenTyper, although sadly she doesn't go into detail other than noting that the one benefit over the Atari (she conflates machine with software) was its ability to easily visualize paragraph breaks.[105] The shift from Adler to Atari was likely significant for CBR. The Atari ST line of personal computers had launched in 1985 and would prove very popular, particularly in Europe. Crucially, it deployed a graphical user interface, rather than a command-line interface, as standard; unlike with the Adler,

[102] Thomas M. Disch, 'Rock of Phages', Review of *Xorandor*, *New York Times* (3 August 1986), 10.

[103] Elisabeth Anderson, 'Alone and Dispossessed', Collective Review, *Sunday Times* (10 August 1986), Fiction 2; Raymond Sokolov, 'Computerese Spoken Here', Review of *Xorandor*, *Wall Street Journal* (19 August 1986).

[104] Chris Vincent, Untitled Review of *Xorandor* and *The Christine Brooke-Rose Omnibus*, *Women's Review* (12 October 1986).

[105] Christine Brooke-Rose, 'Thoughts on Printing from Diskette', Acc 4/2, Box 12, 'Material from Robyn Marsack', ACP.

text was no longer the default method for interacting with the computer, rendering less visible the computer's reliance on linguistic communication. CBR was not, however, about to forget that resemblance.

Four years after *Xorandor*, CBR produced her follow-up work. Carcanet had been worried, before they saw the typescript, that it might replicate *Xorandor*'s 'perils of mechanistic exposition'. They were reassured by what they read: 'The shift from program language, along with a more expansive social cultural critique, makes *Verbivore* most engaging.'[106] The book differs from its predecessor in avoiding extensive use of codework. It features Zab and Jip as adults. Zab, a former teacher of 'Communications', is now a diplomat, Euro-M.P. for Aachen International District, with a son Hancho who spends much of the novel travelling in China. Jip is a university professor in Texas with associations with NASA. *Amalgamemnon*'s Mira Enketei too makes an appearance. Now a radio producer, she still prophesizes (although the future seems to be catching up with everyone) about a world in which people no longer 'talk' and instead electromagnetic waves support 'eternal commentary [which] weighs heavy upon the air, overloading the waves with tetravocal news bulletins like modern operas, fast rolling Spanish over crisp Serbocrat under pompous English inexorably heard behind the French or vice-quadriversa. Inexorable?'[107]

The problem—and the plot of the novel—is that the alpha guys have become 'verbivores', 'logophagai', or word eaters, who now demand the reduction of global information flow for reasons of self-preservation.[108] Humanity is given the chance to instigate efficiency programmes and 'ergonomics' in communication.[109] What follows is a gleeful rumination on the effects of such restriction: for a population used to communicational proliferation and immediacy, now that news circulation relies on physical movement, 'nobody's very interested in a two-month old coup d'etat in Panama.'[110] CBR's former employer, GCHQ, is in 'technical unemployment, no traffic to intercept or codes to break.'[111] Meanwhile, 'media silence' brings about 'the reemergence of poetry'.[112] We are deep in the realm of telematics here, contemplating its effects on human communication.

Verbivore is also deeply interested in setting telematics—'a computer IS its language' here—within a broader multi-linguistic framework.[113] Like

[106] Michael Schmidt to Christine Brooke Rose (20 December 1988), Acc 3, Box 97/2, 'Christine Brooke-Rose Correspondence 1984–1988', ACP.
[107] Christine Brooke-Rose, *Verbivore* [1990], *Xorandor/Verbivore*, 230.
[108] Brooke-Rose, *Verbivore*, 231.
[109] Brooke-Rose, *Verbivore*, 342.
[110] Brooke-Rose, *Verbivore*, 360.
[111] Brooke-Rose, *Verbivore*, 361.
[112] Brooke-Rose, *Verbivore*, 372.
[113] Brooke-Rose, *Verbivore*, 352.

in *Xorandor*, characters are constantly switching between a plethora of languages and communicational systems, between 'quickconnect' (the multiple language-translation system needed to make Zab's work at the European Parliament function), machine translation ('vodka good but meat stinks'), or Hancho's 'nicely drawn' Chinese ideograms that contrast with his constant misspelling of English.[114] We even have a character, Decibel, born of a radio play that opens the novel and who acts as go-between for the alpha guys and humans. Ideograms, as she notes, are her 'natural enemy' able to 'convey meaning without sound': this is a universe of communicational excess, the Tower of Babel realized.[115]

It is also, in a call back to *Between*, a universe of interpersonal and internation conflict. Ultimately, the blackout redescends because countries and people are too busy arguing amongst themselves to agree a workable programme of communicational efficiencies and quotas. In *Verbivore*, Weaver's dreams of mutual understanding and universal language are as far away as ever even if translation machines proliferate. Telematics doesn't help. Women are constantly interrupted, countries fight, politicians lie, and natural languages exist in a politicized hierarchy that is justified erroneously as communicational efficiency.[116] Any attempts to achieve international agreement are doomed from the start; universal language will never be possible and computing, far from a solution, merely exacerbates this situation—albeit in a novel and for aesthetic effect.[117]

Texterminations

While *Verbivore* might seem to represent the 'computer jeremiad,' in the years after writing the novel, CBR would explore the practical possibilities of word processing more extensively. In 1990, she was still using the Atari but had switched to Word Plus software. In 1994, she would switch again to an IBM computer and ClarisWorks WordPerfect.[118] Such a history indicates that CBR was willing to experiment with using different programmes and machines for her writing. In this, she was not alone—as Matthew Kirschenbaum has noted, the period was one of significant growth in the

[114] Brooke-Rose, *Verbivore*, 261, 240, 389.
[115] Brooke-Rose, *Verbivore*, 389.
[116] Brooke-Rose, *Verbivore*, 249.
[117] It is both a gleeful and pessimistic view of the world but one that is troubling for its representation of 'redundancy' through national and cultural stereotypes.
[118] Details gleaned from Finding Aids, electronic discs, and correspondence held in the ACP and CBRP.

word processing market.[119] What is interesting about this history is that it demonstrates that even after inventing a programming language and authoring several books that featured computing, CBR did not view herself as an expert. In one letter, she notes that while her new IBM machine had arrived, she was waiting for 'lessons on it, as its completely different from Atari'.[120] Different machines require different technical literacy. At another point, she would grumble 'why did I ever buy an Atari!' when computer technicians confirmed that, while they could convert her Atari files into WordPerfect for a fee, it would result in the loss of all formatting.[121] Computer hobbyist and programmer CBR was not, even as she took seriously the programming language as an epistemic object.

If we look back over her device and software history, CBR seems initially to have viewed the word processor mainly as a tool for composition, rather than a programmable machine. She did not distinguish between kinds of writing—composing letters, fiction, and academic articles on the same machine—but did note that she typed, or rather 'process[ed]' her manuscript as 'the last act of creative revision'.[122]

However, in 1990, she also saw the possibilities that word processing offered publishers. *Textermination,* a novel that despite its grouping as part of the 'intercom quartet', is thematically concerned with the academic study of literature (it is set at a conference and mainly generates difficulty and pleasure through intertextual references to works of literature). The novel is, however, telematic in its gestation: CBR sent Carcanet the manuscript on disk, providing the publisher with both ASCII and 'WP Modus' (i.e. software specific) files, the conversion done with the aid of 'an expert'.[123] Although not sure which format would be suitable for her publisher, CBR was enthusiastic about the possibility that page setting could be done straight from the files. Included with the disks was a two-page memo 'Thoughts on Printing From Diskette'. Acerbically pointing out the significantly increased effort required by the author—and extensively cataloguing difficulties around the return and spacebar keys—CBR nevertheless insists that such disks could

[119] Matthew G. Kirschenbaum, *Track Changes: A Literary History of Word Processing* (Cambridge, MA: Harvard University Press, 2016).

[120] Christine Brooke-Rose to Michael Schmidt (7 September 1994), 2, Acc 6–3, Box 28, 'Letters 1993–4: Brooke-Rose, Christine', ACP.

[121] Christine Brooke-Rose to Michael Schmift (27 January 1994), Acc 7, Box 9, 'Brooke-Rose, Christine: *Remake* [typ and corr]', ACP.

[122] Christine Brooke-Rose to Michael Schmidt (1 September 1985), Acc 9, Box 2, 'Incoming Letters file up to 1996', ACP.

[123] Christine Brooke-Rose to Michael Schmidt (9 October 1990), Acc 3, Box 332/1, 'Book File: Christine Brooke-Rose, "Textermination" (1991)', ACP.

potentially be used to improve the publishing process and reduce errors introduced in the editorial process.[124]

Carcanet and their setter clearly weren't quite ready for such innovations.[125] They didn't have her abiding interest in code, nor her experience of Minitel, and moving between programming and hardware regimes. Nor did her propensity to produce works that relied heavily on non-dictionary word play, multiple languages, and unusual layout make things easy. In this case, compatibility problems between the Atari files and Carcanet's setter's machine resulted in all accented letters disappearing in the setting process for *Textermination*.[126] The latter was unfortunate in that it slotted easily into a broader narrative within France around enforced anglicization of the language thanks to US tech dominance (keyboards do not include accented letters as standard); it was also extremely time-consuming to fix.

While she didn't give up—still happily complaining about file compatibility when delivering *Remake* and busily depositing corrupted computer files to her personal archives—it is useful to see that CBR's practical engagement with word processing and personal computers was not marked by any dramatic moments of enthusiastic embrace. She viewed the machines and software very much as tools for facilitating the delivery of text for publication, rather than as universal language machines. Computers, these mechanical readers of programming languages, were as likely to mistranslate across machines and formats as their human equivalents. In so doing, they opened a crack through which Babel might, delightfully, slip onto the page.

With this in mind, it is no wonder that CBR was interested in keeping words and computers separate in her late memoir *Remake*, written half a century after her experiences at BP. Those years had only underlined to CBR that literature's capacity to execute meaning, difficult though that execution might be for the reader, relies on the universal machine being anything but. It is to a very different line of universalist critique that we now turn.

Programming Literature. Rebecca Roach, Oxford University Press. © Rebecca Roach 2025.
DOI: 10.1093/9780198976578.003.0004

[124] CBR, 'Thoughts on Printing'.

[125] Only in 1999 would Carcanet embrace email as 'the answer to all communication problems!' Michael Schmidt to Christine Brooke-Rose (8 March 1999), Acc 11, Box 2, 'Incoming letters file up to 1999: Brooke-Rose, Christine', ACP.

[126] Robyn Marsack to Christine Brooke-Rose (15 April 1991), Acc 4–2, Folder 12, 'Material from Robyn Marsack', ACP.

4

Kamau Brathwaite

Creolized Computing and Processual Aesthetics

In 2009 Kamau Brathwaite (1930–2020) advertised a class he would be running as part of New York University's Comparative Literature programme. 'Tpcs in Caribbean Literature: sycoraX aesthetics' purports to offer a typical syllabus, complete with an outline of the topic (how cultures 'deal w/Catastrophe' such as Caribbean colonization), a list of texts to be studied, and a series of questions that will guide class discussion. That, however, is where the typicality ends. The form of the document and the typography is distinct in being comprised of idiosyncratic shorthand ('What we're concerned w/here = sensibility'), multiple exuberant fonts, and poetry, and which concludes with a comment that the reading will consist of 'sev oth- er txts now STOLEN from my apartment, inc the Saldívar'.[1] It is a difficult text. Quite what students made of the syllabus is unclear. What is clear is that Brathwaite, a major figure in Caribbean literature and long-time university professor, treated this pedagogical, administrative, word document as an exercise in poetic experimentation. More specifically, he chose to replicate the content of the course—'sycoraX aesthetics'—in the form of the syllabus.

The story behind Brathwaite's conceptualization of computer writing as poetic experimentation is the focus of this chapter. Thanks to his theories of 'creolization' and 'nation language' and in his exploration of Caribbean culture's indigenous and African ancestries, Brathwaite has been a significant figure for postcolonial theorists. In his negotiation of colonial literary legacies and his poetic experiments with local rhythms and forms, and in his championing of Caribbean writers at home and abroad, he has been a significant figure for Caribbean literature globally. In the late 1980s Brathwaite also turned to the personal computer to develop a (difficult) literary aesthetic he named 'Sycorax Video Style' (SVS). As we will see, Brathwaite's word

[1] Kamau Brathwaite, 'Tpcs in Caribbean Literature: sycoraX Aesthetics', *NYU* (2009), https://web.archive.org/web/20100615010258/, http://draper.fas.nyu.edu/docs/IO/10728/Brathwaite.sycoraXaesthetics.doc.

processing activities would become a crucial part of his efforts to decolonize literature. These exuberant experiments with word processing mark him as distinct from Coetzee, Kenner, and Brooke-Rose in that his engagement with computing was not as an industry insider or commentator but as the purchaser and user of mass market personal computing devices in sites outside of Western European or US computing culture. Brathwaite doesn't only provide a useful entry point for contemplating the cultural specificity of computing, its 'creolization', and the challenges entailed in decolonizing computing archives. As 'one of the great poet–thinkers of our time', Brathwaite also brought his expertise to his own vision (distinct from Coetzee) of how computing might facilitate a decolonial (modernist) literary project.[2]

Brathwaite's Transnational Word Processing

(Edward) Kamau Brathwaite's career was a peripatetic one; he leapt back and forth across the Atlantic and around the Caribbean basin for more than half a century. His education and early career were heavily entwined with the systems of a British empire that was rapidly dismantling in the decades after the Second World War, and with the decolonization movement, of which he would become a significant cultural theorist. Brathwaite was educated at Harrison College in Barbados before winning a scholarship to take a history degree at Pembroke College, Cambridge (UK) in 1950. There he also attended lectures by F. R. Leavis and began a professional relationship with the BBC programme *Caribbean Voices*. In 1955 he relocated to the British Gold Coast to work for the Ministry of Education and while there witnessed first hand the country's decolonization and the establishment of the newly independent nation of Ghana. There he developed an interest in African oral traditions, which seemed to him to offer an alternative poetics to that represented in the written forms of the dominant colonial culture. Brathwaite saw a crucial parallel with Caribbean literary culture: '[i]t was in language that the slave was perhaps most successfully imprisoned by his master, and it was in his (mis-)use of it that he perhaps most effectively rebelled.'[3] During his time in Ghana Brathwaite's 'life-long desire to

[2] PEN America, '[Virtual] 90: A Celebration of Kamau Brathwaite', *PEN America* (11 May 2020), https://pen.org/event/a-celebration-of-kamau-brathwaite/.

[3] Edward Brathwaite, *The Development of Creole Society in Jamaica, 1770–1820* (Oxford: Oxford University Press, 1971), 237.

establish a poetics of voice in the Caribbean' was born; this desire, we shall see, formed the bedrock of his computational aesthetics.[4]

Moving to the Caribbean briefly and then back to the UK, Brathwaite would complete a PhD focused on Caribbean social structures. In the eventual volume, *The Development of Creole Society in Jamaica, 1770–1800* (1971), he would present an influential theory of 'creole' society. That term referred to 'a situation where the society is caught up "in some kind of colonial arrangement" with a metropolitan European power, on the one hand, and a plantation arrangement on the other; and where the society is multiracial but organized for the benefit of a minority of European origin'.[5] This historical arrangement, as he saw it, brought with it a significant cultural equivalent: 'creolization', or 'a way of seeing the society, not in terms of white and black, master and slave, in separate nuclear units, but as contributory parts of a whole'.[6] Brathwaite's analysis would prove influential for other scholars and, as we shall see, offers a useful framework for understanding Caribbean computing culture in the 1980s.

While in the UK Brathwaite would dedicate significant effort to establishing literary networks and communities. His first volume of poetry, *Rights of Passage*, was published in 1967 and, together with Caribbean emigres John La Rose, and Andrew Salkey, he cofounded the influential Caribbean Arts Movement (CAM), which sought to promote and celebrate Caribbean arts and artists, bringing writers together 'so we could talk'.[7] An attempt to promote connection and community, the majority of CAM events were dialogic, social occasions—talks and readings in front of audiences, informal discussions at members' houses. From his earliest days, Brathwaite promotes a vision of literary culture as communal, interactive activity; it is a vision that he would seek to replicate via his SVS efforts as we shall see.

Eventually returning to the Caribbean and to the University of the West Indies (UWI), Mona, where he would teach history, Brathwaite continued to play an active role in the local literary and arts scene. He would be at the

[4] Simon Gikandi, 'E. K. Brathwaite and the Poetics of the Voice: The Allegory of History in "Rights of Passage"', *Callaloo*, 14.3 (1991), 727–36, 727.

[5] Brathwaite, *Development of Creole Society*, xv.

[6] Brathwaite, *Development of Creole Society*, 307.

[7] Edward Brathwaite quoted in Anne Walmsley, *The Caribbean Artists Movement 1966–1972: A Literary & Cultural History* (London: New Beacon, 1992), 43. He gives a more exuberant account in Anne Walmsley, 'Interview on CAM with Edward Kamau Brathwaite: 15 March 1986, at Irish Town, Jamaica', 2. CAM/6/9, George Padmore Institute, London (hereafter GPI), from which this was taken.

centre of active debates within Caribbean literary culture about the value of vernacular oral aesthetics; as Laurence Breiner has noted: What engages – and energize – many West Indian poets is the matter of their relationship to improvisation, to oral composition, to "orature", to poetry written as a script for performance, to poetry written (or improvised) for distribution through recordings, to poetry written for performance with or to music (as well as to songs *per se*)'.[8] The early 1970s would be a particularly energetic period of debate. The 1971 Association for Commonwealth Literature and Language Studies (ACLALS) conference, the first to be held outside the UK at UWI, Mona, 'formally introduced West Indian writers to their audience' and engendered passionate discussion about these topics, with Brathwaite as a key participant.[9] During this period Brathwaite also ran the journal *Savacou*, and its 'New Writing 1970: An Anthology of Poetry and Verse' double issue (1970/1971) functioned as 'a project of canonical refashioning that sought to rethink and expand the notion of literature in the Caribbean'.[10] It did so by innovatively including a number of works with roots in performance and oral traditions by young writers, alongside a number of essays that would debate the aesthetics of orality in a West Indian context.[11]

The best articulation of Brathwaite's ideas is found in his 1984 monograph *History of the Voice: the development of nation language in anglophone Caribbean poetry*, which would become an influential work for postcolonial theorists in its articulation of 'nation language'. Here Brathwaite would develop many of the questions that had emerged during his time in Ghana, setting out the concept of nation language. This language, notable for its 'orality', differed from the imposed and imperial language of the colonial power, Brathwaite suggested, in reflecting the 'submerged area of that dialect which is much more closely allied to the African aspect of experience in the Caribbean'.[12] The text describes in particular detail the bind of the

[8] Laurence A. Breiner, 'How to Behave on Paper: The *Savacou* Debate', *Journal of West Indian Literature*, 6.1 (July 1993), 1–10, 1.

[9] Laurence A. Breiner, *An Introduction to West Indian Poetry* (Cambridge: Cambridge University Press, 1998), 4.

[10] Norval Edwards, 'The Foundational Generation: From the *Beacon* to *Savacou*', *The Routledge Companion to Anglophone Caribbean Literature*, eds. Michael A. Bucknor and Alison Donnell (Abingdon: Routledge, 2011).

[11] See Breiner, 'How to Behave on Paper'; and Kelly Baker Josephs, 'Where Do Journals Go? *Savacou*, Fifty Years Later', *SX Salon*, 35 (October 2020), smallaxe.net/sxsalon/discussions/where-do-journals-go-savacou-fifty-years-later.

[12] Edward Kamau Brathwaite, *History of the Voice: The Development of Nation Language in Anglophone Caribbean Poetry* (London and Port of Spain: New Beacon Books, 1984), 17, 13. For a discussion of Brathwaite and Caribbean modernism, see Simon Gikandi, *Writing in Limbo: Modernism and Caribbean Literature* (Ithaca: Cornell University Press, 1992), who

Caribbean writer who has been educated in 'pentameters'—in a British (modernist) 'Great Tradition' of literature that has little to do with the experience of Caribbean life. Brathwaite offers an alternative conception of literature, what he describes as 'the *conversational mode*' seen in calypso call-and-response form and in nation language more generally, which 'can have a corrosive effect on the tyranny of the pentameter'.[13] In *History of the Voice* then, Brathwaite argued for orality's centrality to the Caribbean literary tradition. In doing so he would become an important voice in attempts to reorient the dominant hierarchy surrounding written/oral cultural expression reflected in Western scholarship (and archiving practices).[14] As we will see, these intellectual commitments would also shape his ideas around computing's aesthetic possibilities.

Brathwaite's international profile would grow across the 1980s, thanks to *History of the Voice* and other works, including the essay collection *Roots* (which won the 1986 Casa de las Americas Prize for Literature) and his second trilogy of poetry volumes, *Mother Poem* (1977), *Sun Poem* (1982), and *X/Self* (1987). Brathwaite would spend time in the USA, with teaching stints at Harvard, Yale, and elsewhere. In 1992 he would relocate more permanently to New York to take up a position as Professor of Comparative Literature at New York University, where he eventually taught his course on 'sycoraX aesthetics' and where, never one to tire, he would also establish Savacou North, an imprint to publish his SVS experiments. Following his retirement, Brathwaite would continue to move back and forth between New York and Barbados, where he died in 2020, aged eighty-nine.

That lengthy career summary offers important context for Brathwaite's intellectual developments, institutional connections, and publications. It doesn't, however do more than hint at why Brathwaite, 'more than High Priest' of Caribbean literature, should also occupy a significant position in a

(although offering no dedicated chapter to Brathwaite) argues that Caribbean modernism evolved in a process closely related to his notion of creolization (5); Charles W. Pollock, *New World Modernisms: T. S. Eliot, Derek Walcott, and Kamau Brathwaite* (Charlottesville: University of Virginia Press, 2004); Matthew Hart, *Nations of Nothing But Poetry: Modernism, Transnationalism and Synthetic Vernacular Writing* (Oxford: Oxford University Press, 2010); and Ankhi Mukherjee, 'Creole Modernism', *Affirmations: Of the Modern*, 2.1 (Winter 2014), 25–45.

[13] Brathwaite, *History of the Voice*, 32.

[14] This is a reorientation still ongoing in scholarship on World Literature: in her plea for scholars to pay more attention to orature, co-editor of the *Norton Anthology of World Literature* (2012), Caroline Levine noted that 'institutions of world literature have elevated—one might even say, fetishized—writing'. Caroline Levine, 'The Great Unwritten: World Literature and the Effacement of Orality', *Modern Language Quarterly*, 74.2 (June 2013), 217–37, 223.

historical account of relations between (difficult) literature and computing.[15] To understand why his concept of creolization offers a useful tool for theorizing computing's situation in the Caribbean in the years when he was developing his Sycorax Video Aesthetic—to understand why this in turn helps to explain why word processing on an American mass market consumer product could be reconceived as a significant step in a decolonial literary endeavour—we need to understand something of the local computing culture in which Brathwaite was operating.

Caribbean Computing, Creolized Processing

In looking for that understanding though, we face a challenge. Several. Firstly, Brathwaite's own personal archive and notable collection of Caribbean cultural heritage was damaged, perhaps destroyed, when Hurricane Gilbert devastated his house in 1988.[16] The existence of those collections is itself unclear, the result of Brathwaite's own mythologizing practices. As Matthew Kirschenbaum points out, in the Preface to the 1993 version of 'Dream Chad' Brathwaite talks of his computer and archives having 'miraculously' survived the storm. In other accounts, they are partially or completely lost. Elsewhere his friend Gordon Rohlehr recalls talk of efforts to transport archives to institutional collections against a backdrop of precarious material conditions: stored, in the aftermath of the hurricane, in a room with a roof that was 'growing weary of the struggle against the elements, is beginning to groan & gape its way upwards towards the sky like its ripped brothers have already done'.[17] In later life in New York Brathwaite would also conflate personal archive and myth-making; in recounting what he described as his period of 'cultural lynching' when personal items went missing from his apartment—alluded to in his syllabus quoted above—he ensures fact and fiction are difficult to disentangle.[18] Not quite embodying the 'wayward' method of scholar Saidiya Hartman's 'critical fabulations',

[15] Fred D'aguiar, Untitled contribution in *Susheila Nasta*, '"A Towering Figure": Tribute to Kamau Brathwaite (1930–2020)', *Wasafari* (24 February 2020), https://www.wasafiri.org/article/a-towering-figure-tribute-to-kamau-brathwaite-1930–2020/.

[16] Matthew G. Kirschenbaum, *Track Changes: A Literary History of Word Processing* (Cambridge, MA: Harvard University Press, 2016), n. 59, 319.

[17] Gordon Rohlehr, 'A Literary Friendship: Selected Notes on the Correspondence with Kamau Brathwaite', *Small Axe*, 26.1 (2022), 124–44, 136. doi: 10.1215/07990537-9724135.

[18] See Timothy J. Reiss, 'Kamau Brathwaite, a Memoir: The Road to and from New York City', *Small Axe*, 26.1 (2022), 111–23, 122. doi: 10.1215/07990537-9724121.

these are nevertheless creative–critical engagements with the medial qualities of the archive that speak strongly against any notion of the universal machine.[19]

As well as being a personal disaster, this damage has scholarly repercussions—something of which Brathwaite was no doubt acutely aware as a social historian. Unlike the other writers I discuss in this book then, there is not an accessible collection of research notebooks, paper tape, floppy disks, and alike that I might parse for intimations of computing practices and thinking. For that, I have had to rely on a composite of collections, reading in and across the margins, listening for what Stanley H. Griffin calls the 'noises in the archives'.[20] It is a difficult, if productive, process and a reminder that computer history builds from the collective environmental substrate (cloud metaphors notwithstanding).

Secondly, specific histories of computing within the Caribbean, as with many former colonized regions, have not featured heavily in published narratives (in contrast with the vibrant body of work and activities that surround studies of contemporary Caribbean Digital culture).[21] Nor have the resources always been there to develop and maintain local, national, transnational archival collections for the sector. In places such as Barbados and Jamaica large computer corporations (Control Data Corporation, ICL, IBM, Burroughs) often had representation in the form of subsidiaries or local agents, whose records are less likely to have been accessed into an archive. The view to which we often have access is that of the corporate centre looking to the peripheral office: our programmer in Havana.[22]

[19] Saidiya Hartman, *Wayward Lives, Beautiful Experiments: Intimate Histories of Riotous Black Girls, Troublesome Women and Queer Radicals* (London: Profile Books, 2019).

[20] Stanley H. Griffin, 'Noises in the Archives: Acknowledging the Present Yet Silenced Presence in Caribbean Archival Memory', *Archival Silences: Missing, Lost and, Uncreated Archives*, eds. Michael Moss and David Thomas (London: Routledge, 2021), 81–99. doi: 10.4324/9781003003618-5.

[21] There are a huge range of project and community building efforts to support the region's multinational, multilingual digital cultural heritage. These include the Digital Library of the Caribbean (https://www.dloc.com/), the establishment in 2016 of *archipelagos*, 'a journal of Caribbean digital praxis', the conferences of the Caribbean Digital Collective (https://thecaribbeandigital.org/), and the Directory of Caribbean Digital Scholarship. By contrast, dedicated and accessible archival collections (oral or otherwise) on the *history* of the computer industry in the region are currently scarce and efforts to preserve these stories urgently needed.

[22] One of the more useful sources of information on the corporate milieu comes from the archives of Control Data Company (CDC). A major American mainframe and supercomputer firm, in the 1980s it established a Caribbean subsidiary, and its archives contain market surveys, reports of local interviews, and records of business negotiations that provide a patchy overview of local infrastructure, cultural attitudes towards computing, international corporate presence, and device availability. CDC was also noted for its corporate social responsibility.

Archives are, as critics of post- and decolonization have eloquently elucidated (Brathwaite included), constructions that are intimately bound up with the projects of empire.[23] In the context of computing, the politics only become more fraught given the historical use of narratives of technological superiority to justify colonization (recall Coetzee's argument in relation to South Africa from Chapter 1). The archival constructions of Caribbean computing are inextricable from the broader history of the region and the datasets shaped by this history.

As we seek for stories within these archival collections then we can think of computing's presence, as a technology, industry, and cultural object, in the Caribbean as symptomatic of a longer history of European colonization, American neocolonialism, and a global trade system founded on slavery. Speaking of the US, for example, Tara McPherson has made the astute point that while we tend to view histories of computing and civil rights in the 1960s separately, the two in significant ways 'coconstitute' each other.[24] The logic of modularity and information containment that dominates UNIX is evident in the systems of covert racism that developed at mid-century, even as computing and code present themselves as culturally neutral. The universal machine's claims to rationality have, as we have seen, often been deployed in contexts that undermine black subjectivity: in Chapter 1 I discussed IBM's role in facilitating the South African government's apartheid regime. Representation within the industry has also historically been poor; scholars have chronicled how the US computer industry has (and continues to) shut minority populations out, while simultaneously often turning these individuals into abstract data subjects.[25] As Charlton D. McIlwin explains in his compelling history of 'black software' in the US, while his story 'demonstrates

The company's archives thus also display a significant awareness of computing's social and cultural position. (This ethos also explains how CDC would become heavily involved in supporting farming cooperatives in the region, despite only a tangential link to their services.)

[23] As Anjali Nerlekar and Francesca Orsini neatly put it: 'archival *collection*—of documents, records, images, books, manuscripts, artefacts—became an integral part of the colonial–imperial state and its rhetoric of custodianship, knowledge production and progress'. 'Introduction: Postcolonial Archives', *South Asia: Journal of South Asian Studies*, 45.2 (2022), 211–19, 213. doi: 10.1080/00856401.2022.2040803.

[24] Tara McPherson, 'Why Are the Digital Humanities So White? or Thinking the Histories of Race and Computation', *Debates in the Digital Humanities*, ed. Matthew K Gold (Minneapolis: University of Minnesota Press, 2012), https://dhdebates.gc.cuny.edu/projects/debates-in-the-digital-humanities.

[25] See e.g. Simone Browne, *Dark Matters: On the Surveillance of Blackness* (Durham, NC: Duke University Press, 2015); Safiya Umoja Noble, *Algorithms of Oppression: How Search Engines Reinforce Racism* (New York: New York University Press, 2018); and Ruha Benjamin, *Race after Technology: Abolitionist Tools for the New Jim Code* (London: Polity, 2019).

how black people have taken technology not originally designed with our concerns in mind', it also 'speaks to the nefarious ways those in power use computing technology to destroy black agency'.[26] Similar patterns can be discerned in the context of the British empire. In Jamaica in the 1920s IBM worked with the US-based Eugenics Research Association to produce punched cards and Hollerith machines to catalogue the racial characteristics of Jamaica's population, 'to identify mixed-race individuals as a first step toward their elimination'.[27] Even before electronic digital computing was extant in the Caribbean, its people were being processed as data.

This isn't simply a historical issue; the datasets to which we might turn to amplify the noises in the archive can often perpetuate these historic harms. One such is example is the 'digital divide' narrative that emerged around the year 2000 and was based on datasets developed by international institutions and multinationals, and which conceived of Caribbean computing in developmentalist terms. United Nations statistics around internet usage in the year 2000, for example, indicate that internet usage among the Anglophone Caribbean population is less than a third of that of the imperial centre (the UK).[28] In such analysis, computing in the Caribbean is less determined by the racial dynamics of the sector, and more by global capitalism (with its own colonial heritage and afterlives) and the uneven distribution of technology it facilitates. US and UK technological use becomes a benchmark against which data is assessed. This data has been deployed to reach conclusions about the region's IT 'backwardness', limited computing resources, and to justify developmentalist policies and investment. This was not a new phenomenon: the 1983 US Caribbean Basin Initiative for example, which aimed to promote economic development in the region through free trade initiatives, stressed the need to 'sell computers' to young people as part of the training and public education programme being devised, recalling (albeit in more paternalistic clothing) the arguments made by Jacobus Coetzee in favour of colonization.[29]

[26] Charlton D. McIlwain, *Black Software: The Internet and Racial Justice, from the AfroNet to Black Lives Matter* (New York: Oxford University Press, 2020), 7.

[27] Clyde W. Ford, *Think Black: A Memoir*, Amistad (New York: HarperCollins, 2019), 163. His source is Edwin Black, *War against the Weak: Eugenics and America's Campaign to Create a Master Race* (New York: Four Walls Eight Windows, 2003), 288–92.

[28] Lester Henry, 'The Digital Divide, Economic Growth and Potential Poverty Reduction: The Case of the English Speaking Caribbean', 12. Conference paper, https://www.cavehill.uwi.edu/salises/conferences/past-conferences/2003/thedigitaldivide-henry.aspx.

[29] The 1983 US Caribbean Basin Initiative enabled companies like Control Data Corporation (which had representation in Washington) to position their commercial activities (often with financial support from US AID) as part of a broader development project. Board

Collectively such rhetoric and analysis facilitate, as André Brock, Jr, notes, 'bromides about technology forcing users to adapt to mainstream culture and…long-standing deficit-based beliefs about people of color and information technologies' that are not born out by use.[30] Such accounts of Caribbean computing as an exemplar of global unevenness replicate the binary narratives that Brathwaite (and Coetzee) would find so limiting and which give little insight into individuals' usage, perceptions, and imagined relations with computing.

There are other possible ways of reading computing in the Caribbean: namely, a situation in which computing has undergone Brathwaite's process of 'creolization'. If we look to other datasets, we start to gain a different picture. By the 1980s, when Brathwaite began his activities, computers were not a novel presence in the region's public imaginary. Newspapers across the region had regularly reported international and local stories of computer innovations, deployments, and contracts from the late 1950s onwards. These articles often contemplated the social function of computing—whether the import of a project to 'read' biblical texts, automation's import for human workforces, or a ten-part series contemplating racism in the North American sector through coverage of 'The Sir George Williams Affair' (a 1969 incident in which a Montreal university computer centre was occupied to protest racism against black students, and which saw a number of Trinidadian students arrested).[31] In the 1980s newspapers also featured regular advertisements for local data processing services and hardware providers. Local newspapers represented computing as an international, politically imbricated, but locally embedded, set of practices and ideas.

As we toggle between collections, the most distinctive feature we might glean about the Caribbean computing sector is its transnational, multicultural orientation. The situation is analogous to that in Latin and Central America where countries would collaborate (often along linguistic lines)

on Science and Technology for International development, National Research Council, 'Workshop on Needs in Science and Technology for Development in Caribbean Nations', National Academy Press, 1983. Folder 19, Box 37, Control Data Corporation Records, CBI 80, Charles Babbage Institute, University of Minnesota, (hereafter CBI).

[30] André Brock, Jr, *Distributed Blackness: African American Cybercultures* (New York: New York University Press, 2020), 4.

[31] The *Daily Gleaner* ran extensive coverage of the story. For a summary of events, see David Austin, 'All Roads Led to Montreal: Black Power, the Caribbean, and the Black Radical Tradition in Canada', *The Journal of African American History*, 92.4 (Autumn 2007), 516–39. As Austin notes, 'On 11 February 1969 computer cards fell from the ninth floor of Sir George Williams University's Hall Building like large flakes of snow', a strong reminder of just how material computing was in the 1960s (530).

through local institutions, via regional networks, and with international Foundations and multinational US computer manufacturers to develop the sector. Caribbean computing was a similarly 'transnational project', its orientation significantly facilitated by an extant political association (the, albeit failed, attempt to federate former British colonies between 1958 and 1962); by shared economic, linguistic, and cultural ties; and by the small size of national markets.[32]

The actual nature of the sector varied significantly across the different islands of the Caribbean, something that Brathwaite would no doubt have seen first hand as he travelled between islands. In 1973 Jamaica's Computer Society was founded, in the mid 1980s in St Lucia personal computers were beginning to sell, and in Grenada, despite a US invasion in 1983, a locally owned data processing company was flourishing two years later; Trinidad meanwhile offered a potentially large market for companies, but seemed relatively untapped. The island topography was a significant challenge for wired networks: in Monserrat and St. Vincent telecommunications were good, but in Antigua and Dominica they were poor. Per capita income was also a significant factor: for all that such data replicate troublesome narratives, it is helpful to know that in 2000 the cost of an average personal computer was twenty-seven per cent of the average annual income in Jamaica, where Brathwaite was based, versus six per cent on Barbados, where he was born.[33] Local infrastructures and cultural contexts for computing could be very different on different islands.

For all this variance, by the mid 1980s a regional computer industry was clearly thriving. This industry was focused mainly around the provision of computer services—data processing—and small business need, with Barbados at the fore. In 1984, for example, Barbados boasted fifty-five mainframe systems, split roughly equally between IBM, NCR (National Cash Register, an American firm) and ICL, with a smaller number sold by Burroughs and Data General. The numbers, which don't include microcomputers (estimated to be between fifty and seventy) indicate that many of the major American computing firms—and one British—had a sales presence in the small country.[34] Although much of the assembled hardware was imported, the thriving market in data processing services and software production

[32] Ivan da Costa Marques, 'History of Computing in Latin America', *IEEE Annals of the History of Computing*, 37.4 (October–December 2015), 10–12, doi: 10.1109/MAHC.2015.78.

[33] Lester Henry, Table 4 (Source: Human Development Indicators 2002).

[34] R. A. Brandenberg, 'Trip Report: Review of the Barbados Data Processing Market and Proposed Joint Venture Business' (30 May 1984). CBI 80, Box 37, Folder 30, 5.

was a local affair. In Barbados, the major (though not the only) player was Data Research Ltd, a locally owned enterprise, which had 160 clients and an annual revenue of $650,000 and which developed many of its own software applications, including room occupancy systems for local hotels.[35] Meanwhile in Jamaica there was a robust market for tailored applications to produce management training, support project management, and provide data entry for local and regional clients.[36]

As in other regions, universities played an important role as hubs wherein hardware, expertise, and staff could coalesce. At UWI, Mona where Brathwaite was employed, computing was taught through the Faculty of Science and Technology from 1976 onwards.[37] The university had received the nation's first scientific computer in 1962, an IBM 1620, via a common combination of international funding and corporate educational discounts.[38] The machine would be housed (as was the case in Cambridge) in the Mathematics department and a newly established computing centre. Its arrival via ship was a national occasion, reported in the daily newspaper.[39] In the coming years, the mainframe's computing power would be deployed by staff across the university, including by Brathwaite's colleagues in the History department.

These included Barry Higman, the wild-haired Australian scholar of Caribbean history who taught at (and later led) the department from 1971 until 1996. Higman would enthusiastically utilize the machine to conduct quantitative analysis of historical records. His *Slave Population and Economy in Jamaica, 1807–1834* (1976) was the product of research conducted on that machine:

> The computer was a new technology, still a machine used for computing, number crunching in a very literal sense. There was a certain excitement

[35] Brandenberg, 'Trip Report', 19.

[36] One company, WTG Ltd, had aspirations to develop an international market through CDC, with a predicted annual revenue of $1.4 million. R. C. Chinn to W. C. Norris. Memo (24 September 1986), 11. 'Caribbean Initiatives', Control Data Caribbean Basin, Inc. 1984–1986, Control Data Corporation records, CBI 80, Box 37, Folder 6, CBI.

[37] Computer Science, 'Computing @ Mona: About Us' (22 July 2012), https://www.mona.uwi.edu/compsci/aboutus.

[38] Educational efforts were a significant focus of multinational funding and partnerships. Control Data Corporation (CDC), through its Caribbean subsidiary, would heavily market its PLATO computer-aided teaching system across countries. Barbados meanwhile was gifted Commodore 64s for all secondary schools (with much publicity) and vocational and management training on computers was heavily subsidized by US AID and US manufacturers. (R. C. Chinn to W. C. Norris. Memo (24 September 1986), CBI.)

[39] 'Electronic Computer for UWI Comes', *The Daily Gleaner*, Kingston, Jamaica (30 July 1962), 2.

> to employing these noisy tools, waiting hours for the IBM 1620 to chew through some of the more complex tasks, producing results in the form of yet more punched cards which had to be fed into a printer to be read. There was a hint of the industrial revolution, a belief that the new machines and techniques could solve what had been insoluble.[40]

Higman doesn't shy away from the numerical basis of these operations, nor the solutions such universal machines might offer, but he is also very much focused on the machine processes: the techniques and practices, the noisy activities of chewing and feeding and reading that computational analysis entails. He foregrounds the ways in which the machine might contribute to new processes for conducting historical scholarship that might, in turn, produce new knowledge. Computing here is *processual*—the focus is less on the code and more on the activities, resources, and conditions that make code execution possible.

It was a vision that would pervade Brathwaite's department. Higman would set up the Social History Project, encouraging students and colleagues to use programming to support their work. Some went to his house to learn elementary coding before heading off for lessons in FORTRAN from the Faculty of Natural Science, or borrowing programmes from colleagues at the University Library.[41] Under Higman's leadership, Brathwaite's departmental culture embraced computational expertise, techniques, and cross-campus collegiality to examine local historical records. An IBM 1620 might be a symbol of multinational corporatism, the dataset under analysis the product of the systems and institutions of enslavement, but in its UWI, Mona context—a context in which the creolized computer made visible and significant the localized processes that might shape code execution—computing could nevertheless facilitate new understandings of Caribbean history and culture. Such a perspective would shape Brathwaite's own computing endeavours in crucial ways.

The conceptual orientation towards processual computing evinced in Brathwaite's department would be further underlined by another peculiarity of the Caribbean computer sector. In the 1980s the region (particularly the Anglophone countries) would emerge as a centre for 'offshore' computing: namely, the relocation of manufacturing from the US and Europe by

[40] B. W. Higman, 'Introduction 1995', *Slave Population and Economy in Jamaica, 1807–1834*, (2nd edn, Mona, Jamaica: University of West Indies Press, 1995), xvii.

[41] Carl Campbell, 'From Founding Member of the SHP, Professor Carl Campbell, Department of History and Archaeology, University of the West Indies, Mona', *Social History Project* website (2003), https://www.mona.uwi.edu/history/shp/aboutus/campbell.htm.

corporations to countries with advantageous conditions (i.e. cheap labour). In 1986 US title *Computerworld* commented on the trend, concentrated particularly in Barbados and Jamaica, which offered a 'human infrastructure' that was well-educated, with a 'strong work ethic' and wage demands of around sixty per cent of those of the US.[42] In 1983 chip manufacturer INTEL had set up a manufacturing and assembly plant that employed a thousand non-unionized staff on the island. Meanwhile in Jamaica the government was actively promoting an export industry in punched cards. The Caribbean would become a significant site for the manufacturing of hardware for an international market: its workers were building computers. Far from a computer backwater that many of the data-driven narratives suggested, the region was a vital location in the global production and circulation of the chips, punched cards, and substrate that made computing possible—a fact that was not lost on the island populations, Brathwaite included.[43]

If offshore manufacturing led the way, the development of offshore data processing would further inflect the region's creolized and processual computing milieu. In 1983 American Airlines, through their subsidiary Caribbean Data Services, would adapt the offshoring model to focus on data—in this case airline ticket data entry. Other firms quickly followed suit, taking advantage of physical infrastructure such as reliable utilities, good airline connections, and excellent telecommunications.[44] These offshoring developments had effects other than the technical and economic. Offshored computing positioned the technology as an indigenized symbol of globalization. It facilitated 'virtual migration' for a group of workers and the importing of US cultural norms and practices: *Computerworld*, for example, blithely talked of the need for new employees to be coached on how 'to be on time and work efficiently' accordingly to an American model.[45] Programming could also form the basis of physical migration for a smaller subset of people, with US and Canadian metropolitan-based programming courses

[42] David A. Ludlum, 'Offshore Data Entry Pays Off', *ComputerWorld* (9 June 1986), 103+; and Brandenberg, 'Trip Report', 7.

[43] Compare Lisa Nakamura, 'Indigenous Circuits: Navajo Women and the Racialization of Early Electronic Manufacture', *American Quarterly*, 66.4 (December 2014), 919–41.

[44] Long before the arrival of the East Caribbean Fiber System (1995)—and with it Brathwaite's internet service—Barbados was a hub for international data links for the Caribbean islands.

[45] A. Aneesh, *Virtual Migration: The Programming of Globalization* (Durham, NC: Duke University Press, 2006); and Ludlum, 'Offshore Data Entry', 112. The article compares Jamaica to Ireland, which has higher wages but the 'horrible weather' aids a positive work ethic, apparently. Programming also formed the basis of a physical migration route for a smaller subset of people, with US- and Canadian-based programming courses heavily advertised in local newspapers in this era—software tuition presented as a path to North American immigration.

heavily advertised in local newspapers in this era—software tuition was presented as a path to North American immigration.

Following Brathwaite's theory of creolization, we might be unsurprised to see that the data processing industry would develop its own local culture—one that shaped the image of computing in the Caribbean. The data processing industry was heavily feminized thanks to a disproportionately young female workforce (many of whom were also mothers), which would mean that efforts around software design and data processing was not gendered as heavily male in 1980s Caribbean countries as was often the case in the US and Western Europe.[46] Brathwaite's later decision (as we shall see) to gender his personal computer female is, we might suggest, responding to this local context.

Looking closely at the computing sector in the 1980s across the Caribbean, we see a sector that is, certainly, heavily structured by the activities and perspectives of a powerful neocolonial minority: US businesses and international institutions. However, we also see a sector that is transnational, while also highly individualized across different islands and with a heavy orientation towards data processing. The industry's focus on manufacturing and on data processing provision presaged a different relation between Caribbean workers and computational data—rather than being data subjects, they are participants in the computational practices deployed to process (other subjects' offshored) data. In the 1960s computers might have been large, imported objects, but by the 1980s computing is a creolized indigenous activity: a series of components to be assembled, services to be offered, techniques to be deployed, by local workers. This is a processual culture in which computing might be used to transform understandings of Caribbean nations' heritage, or restructure the region's economic and power relations, and those of its people, to global circuits of data. This environment would prove highly significant to the way that Brathwaite would come to conceptualize his own word processing activities.

Brathwaite's Personal Computer 'hi/story'

Just as the presence of computer technologies in the Caribbean began much earlier than the 1980s, so too Brathwaite's professional relationship with computing began in a previous decade.

[46] Carla Freeman has written extensively on this particular workforce. See *High Tech and High Heels in the Global Economy: Women, Work, and Pink-Collar Identities in the Caribbean* (Durham, NC: Duke University Press, 2000).

In June 1970 an advertisement appeared in the first edition of *Savaou*, the arts journal that Brathwaite cofounded and edited. It was for Jamaica Data Machines Ltd and promoted the Italian-made Olivetti Business Machines. What was it doing in *Savacou*, a journal that promoted, as Brathwaite later recalled, the CAM ideals of community and conversation through a different medium?[47] It wasn't a one-off; the advert also ran in later editions. This suggests that there was more of a crossover between the intellectual, middle-class readership of a literary magazine and potential customers for data processing services than we might otherwise assume. Years before Brathwaite's aesthetic interest was piqued, computing was tangentially associated with local Caribbean literary activities. (As we will see in Chapter 5, funding literary endeavours via the computer sector was more common in this era than we might imagine.)

Turning to Brathwaite's personal engagement with computing, we discover that the details have been largely a matter of inference: the tracking down of explicit mentions and extrapolations from his printed aesthetics. There has been no shortage of academic work on this front. Since the publication of his 1987 collection *X/Self,* which includes the poem 'X/Self's Xth Letters from the Thirteenth Provinces' purportedly written 'pun a computer o/kay?', scholars have been keen to examine the development of his SVS.[48] Yet as to the precise details of Brathwaite's hardware and software use, there is slightly less clarity. Matthew Kirschenbaum discusses the case in *Track Changes*, noting that 'none took the Macintosh more seriously' than Brathwaite, yet Kirschebaum has to make educated guesses as to what exact model the poet was using.[49] Material held at the George Padmore Institute indicates that Brathwaite's first (declared) email was sent to John La Rose in the Spring of 1998 from a Caribsurf account, suggesting that Brathwaite benefitted relatively early on from Caribbean Internet connectivity.[50] Those same archives indicate that Kamau and his wife Doris were also exploring the possibilities of using word processing to decrease publishing costs associated with *Savacou* (to little avail) in the mid 1980s.[51] However, precise records of his computer use (particularly software use as this was an era in which software was often bundled with the machine—with metonymic results for user thinking) are difficult to find.

[47] See Walmsley, 'Interview on CAM with Edward Kamau Brathwaite', 20.

[48] Edward Kamau Brathwaite, 'X/Self's Xth Letters from the Thirteen Provinces', *X/Self* (Oxford: Oxford University Press, 1987), 80–7, 80.

[49] Kirschenbaum, *Track Changes*, 197, 201.

[50] Kamau Brathwaite to John La Rose, subject: LX (14 April 1998), Edward Kamau Brathwaite, 143/9, GPI.

[51] Walmsley, 'Interview on CAM with Edward Kamau Brathwaite', 22.

Part of the difficulty also comes from Brathwaite's own mythologizing of the computer in experimental (life) writing that responded to the period he would label his 'Time of Salt'. This was a span of years during which Doris died of cancer (1986), Hurricane Gilbert (almost?) destroyed Brathwaite's significant library, personal archive, and house in Irish Town, Jamaica (1988), and he was the target of a home invasion (robbed, bound, gagged, and threatened) in 1990. These events would have a significant material impact; they would also have an affective impact. Of this time Brathwaite would comment, '[n]ow know I the salt; the wasteland'.[52] The experience would also prove extraordinarily creative for him. Notably, in his autobiographical accounts of these events personal computers often intrude upon the scene.

The Zea Mexican Diary (1993) offers an account of Brathwaite's grief at his wife's death in which the personal computer offers a curiously consistent presence. In the midst of its 'diasporan quest for wholeness', this 'polyphonic' and 'sculpted' diary repeatedly makes reference to the computer in the Brathwaites' house at Irish Town, or 'IT' (note the pun) at this time.[53] Brathwaite explicitly identifies the machine as a Kaypro. A line of American personal computers, the Kaypro was aimed at novice users and would prove popular with consumers across the 1980s. The 1982 Kaypro II made the company's reputation thanks to its low cost, easy set-up, and 'rugged' construction (all but one in the line were portable).[54] The computer was considered particularly good value thanks to the software bundled in—the CP/M operating system, a spreadsheet programme, the word processing software *Select*, and a compiling BASIC language, called SBASIC. The company had a Jamaican representative and regularly advertised its products in the *Daily Gleaner* as 'the professional computer'.[55] Notably too, in a campaign covered widely in the national press, the US producers of Kaypro had donated ten machines in 1984 to a government-sanctioned 'Memory Bank' project at the national library to facilitate the 'documentation', 'storage', and

[52] Kamau Brathwaite, letter to Gordon Rohlehr (14 August 1986), quoted in Gordon Rohlehr, 'Dream Journeys', *World Literature Today*, 68.4 (Autumn 1994), 765–74, 765.

[53] Sandra Pouchet Paquet, 'Foreword', to Kamau Brathwaite, *The Zea Mexican Diary: 7 September 1920–7 September 1986* (Madison: University of Wisconsin Press, 1993), viii, x.

[54] Jerry Pournelle, 'Burnouts, Bargains and Two Sleek Portables', *Byte* (January 1983), 418+, 432.

[55] Kaypro, 'Kaypro 4', Advertisement, *The Daily Gleaner* (6 March 1985), 11. Looking over the digitized collections of the newspaper, there was a thriving second hand market for Kaypro machines in this period, as the newspaper's classified section indicates, and the employment advertisements indicate demand for systems engineers with expertise in supporting Kaypro users specifically.

'dissemination to researchers' of Jamaican oral history.[56] This was a project that Brathwaite, an academic deeply committed to preserving cultural heritage (and for thirty years from 1979, on the board of directors of the UNESCO History of Mankind project), was no doubt familiar. The Kaypro in Jamaica in this period is a machine intimately connected to the archiving and processing of oral histories.

At Brathwaite's home, however, the household PC seems to have been predominantly the domain of Doris; it was her 'belov:ed'.[57] In the diary Brathwaite notes that she uses it to keep the *Savacou* accounts and mailing lists; the stand-alone PC becomes a tool by which Doris can nevertheless facilitate interconnectivity and the transnational community that *Savacou* represents. Interestingly, despite the Kaypro's bundled software, Brathwaite explicitly mentions that Doris designs specific programmes for such activities (an 'Xam list' for his students which she debugs even while sick[58]); in the Brathwaite household it is Doris who is the coder, hobbyist, and data processor, not Kamau. Within the context of Caribbean computing, with its feminization of data input, this is perhaps not surprising; within the context of the household, Doris's own training as a librarian (she was thus familiar with constructing and working with data systems), and her management of the administrative aspects of *Savacou*, it is perhaps overdetermined.

Significantly, nowhere in *The Zea Mexican Diary* does Brathwaite *use* the computer. Instead, he merely spends time in 'the computer room'.[59] It is to this room that Brathwaite has retreated at the moment of Doris's death.[60] We have then a curious passivity in place of the processual image of computing, or the human–computer interaction, that we might expect; Brathwaite does not talk of typing (he 'dotted' 'i's—i.e. wrote longhand—rather than typing[61]), but rather of a space. Such an association might seem at odds with the highly connected, processual, conception of computing that Caribbean computing seemed to represent. In fact, it helps here to position the computer outside of the realm of familiar processes. We are in a space after death, after catastrophe, after the norms of interaction have fallen away. This association between computing and what we might call 'processing in the aftermath' would form a crucial basis for Brathwaite's SVS.

[56] 'PM gets computers to help with memory bank project', *Daily Gleaner*, Kingston, Jamaica (25 September 1984), 15. The project is still going strong; the website (it merged in 1990 with The African Caribbean Institute of Jamaica) is available at: https://acij-ioj.org.jm/.

[57] Brathwaite, *Zea Mexican Diary*, 32.

[58] Brathwaite, *Zea Mexican Diary*, 32.

[59] Brathwaite, *Zea Mexican Diary*, 29.

[60] Brathwaite, *Zea Mexican Diary*, 90.

[61] Brathwaite, *Zea Mexican Diary*, 90.

Before discussing that, let us compare this account of Brathwaite's personal computer history with another, published a year after *Zea Mexican Diary*. *DreamStories* was a prose–poem collection of dream journeys, published in Longman's Caribbean Writers Series in 1994, and described by Brathwaite as 'like a kind of RIFT VALLEY in my senscape after the psychic disaster slippage' of the time of salt.[62] One such story, 'Dream Chad' is set in 'Haavaad Yaad' where Brathwaite was visiting, in 1988, days before Hurricane Gilbert hit Jamaica.[63] It describes a premonition of the destruction of his IT house, and, again, the computer is curiously dominant. In a note to the 'Reader+', Brathwaite recalls that the PC had been 'such a wonderful companion for me during & after the dark of my wife Zea Mexican's sudden unexpected & finally fatal illness & aftermath.'[64]

Here we have moved from the computer as psychic space to the computer as distant provider of affective support. Brathwaite continues, 'I had left the computer like a kind of emotional anchor or icon – along w/everything else – all my life w/Mexican my archives my artifacts etc etc etc – up there w/Chad [who would become his second wife] at IT'.[65] The personal computer is personal in its connections to the user; it has accreted memories and associations and has become a tether to his old life in his period of journeying. Again, there is no mention here of the computer being deployed as a writing tool, as a word processor. Rather, it functions as an affective provocation: how might one engage with the world after catastrophe—after the loss of a spouse or a cultural archive?

Yet, in this account the computer is not a Kaypro, but an Eagle. The Eagle line of computers was in fact very similar to the Kaypro: both companies were based in California; both offered a portable model; both computers ran CP/M, were easy to use and purchasable for around $1,500 in the early 1980s (at least in the US); both bundled software, including word processing, spreadsheets, and compiled BASIC. When *Byte* reviewed the Eagle 1,600 in 1983, it discussed it in the same breath as the Kaypro II and Kenner's Zenith Z-100, indicating the degree to which all three were comparable models within the same expanding consumer market.[66] Whether Doris had

[62] Kamau Brathwaite, Letter to Gordon Rohlehr (13 November 1993), quoted in Rohlehr, 'Dream Journeys', 765.

[63] Kamau Brathwaite, *DreamStories*, Longman Caribbean Writers Series (Harlow: Longman, 1994), 47.

[64] Brathwaite, *DreamStories*, 47, 48.

[65] Brathwaite, *DreamStories*, 48.

[66] Jerry Pournelle, 'Eagles, Text Editors, New Compilers, and Much More', *Byte* (September 1983), 307.

a Kaypro and Kamau an Eagle, or whether he misremembers or renames the model in his writing, from a technical perspective the two were similar. The type of computer on which Brathwaite would conduct his early experiments in SVS was a US-manufactured personal computer, widely available and marketed to consumers as reasonably priced, aimed at the professional worker who was also a novice user, and which included bundled word processing software. The slippage is perhaps indicative of Brathwaite's greater interest in the personal computer's imaginative possibilities rather than its technical specifications.

'Dream Chad, a story' also describes a shift in Brathwaite's computer use. In the introductory note he explains that he wrote, and rewrote, the dream story on several rented Apple Macintoshes while he was in the US. We are no longer in the environment of Jamaican computing, but rather at the centre of American computational might: Harvard Yard witnessed Grace Hopper's coding experiments for the Mark I computer during the Second World War; it is less than a mile from MIT, birthplace of timesharing, home of Weizenbaum and ELIZA. Brathwaite might be a first-time user of the Mac, 'without any hands-on or knowledge or background', but he is based at one of the central sites in the history of computer innovation and able to draw on the kind of community expertise that Hugh Kenner championed as important to work the 'instrument'.[67]

Apple's Macintosh range was first introduced to the US market in 1984 as a compact personal computer and went through various iterations (often expanded RAM) across the 1980s. It used 3 ½ inch floppy disks and included a one-button mouse and the 'System' graphical user interface as standard, exciting features with which many users were at that point unfamiliar. Applications for word processing and paint were bundled—although *MacWrite*'s inability to store pages was infamous—and various programming languages could be used, including MacBasic. The Mac II, Apple's high-end line, would be released in 1987 as a modular system with colour display, but the model that Brathwaite likely used, according to Kirschenbaum, was a Macintosh SE also released in 1987.[68]

Crucially for Brathwaite's developing involvement with computing, the Mac's reliance on the graphical user interface marked a departure from the command-line interface used by CP/M, with which Doris at least had been intimately familiar, and which was deployed by both the Kaypro and Eagle. The Mac's embrace of a graphical user interface marked a distinct shift, in

[67] Brathwaite, *DreamStories*, 48.

[68] Kirschenbaum, *Track Changes*, 201.

the consumer market, away from human–computer interaction as facilitated via text. While Coetzee and Kenner had been working at low (or no) levels of machine abstraction and Christine Brooke-Rose with high-level programming languages, Brathwaite focused his attention still further up this hierarchy of abstraction: on the Mac operating system and application processes. On such a level text was no longer executing system commands but was instead primarily the object of manipulation (processing or composition) and Brathwaite was clearly intrigued by the typographical and layout possibilities these applications afforded.

Although initially describing the Mac in 'Dream Chad' as an 'instrument', Brathwaite quickly moves into less neutral language. He describes the development of a 'very close relationship with Apple/Mac – in fact in the end I purchased my own Sycorax on which most of the **video style** stories of this collection have been written'.[69] Note the degree to which we have moved from the static computer functioning primarily as a metaphor, to a philosophy that recalls that of Barry Higman: interaction with computer processes, for composition purposes, is a key producer of meaning in 'Dream Chad'.

Following the remarks quoted above, Brathwaite notes that, as he was finishing the story, the 'machine shall we say *malfunctioned* and I lost nearly all its oratory'.[70] Brathwaite's choice to figure his computer writing as 'oratory' might seem odd, a medial slip perhaps. In fact, it indicates just how much the Barbadian poet–critic conceives of computational writing precisely as processual, as word processing, and as akin to dialogue: here the computer resists Brathwaite's attempts to inscribe the Delphic prophecy (recall Brooke-Rose) of his dream. The 'hear(t)less' computer (the pun again pointing to an oral aesthetic) repeatedly fails to save the file and with each rewriting—where it is Brathwaite's own mind that functions like computer memory storage (recall again Brooke-Rose)—it is as if 'some **spirit in the machine** was willing or trying to prevent me from writing the story' of an event that had yet to take place (i.e. the Hurricane destruction). This process culminates in a message that announces that there was '**no way that I cd continue the dream since SOMEBODY ELSE WAS USING IT**' to which Brathwaite wonders if this attack by a 'premonition or monitor' (again note the word play) was somehow a manifestation of the 'jealous' Eagle.[71]

[69] Brathwaite, *DreamStories*, 48.

[70] Brathwaite, *DreamStories*, 48.

[71] Brathwaite, *DreamStories*, 48.

The playful puns, the incongruous collapse of dream and file, computer and human memory, the assigning of emotions to the Eagle and a spirit to the Mac: all work to position word processing as creatively productive by virtue of it being, specifically, a difficult (psychically, physically, aesthetically) interaction. Word processing is far from a facile exercise here; there is no sense that writing on a computer somehow makes the words come more easily, the language conceived more transparently. Word processing is a wrestling match. This is a far from neutral instrument; it is a difficult interaction that demands the writer's engagement in a way that recalls Kenner's active user–reader and Brooke-Rose's personal tussles, and implicitly rejects Coetzee's characterization of programming as sterile. It is also, crucially, an interaction that mirrors the conception predominant in Brathwaite's own academic department at UWI, Mona, and the actions of the labouring female inputter—the computer as person—of Caribbean data processing. This is computational composition that, even in Harvard Yard, has undergone creolization.

SVS: Nation Writing as Word Processing

Whether or not Brathwaite's computer usage was as disengaged as he details in *Zea Mexican Diary*, he had clearly become enamoured with the PC by 1987. This was the year in which his poetry collection *X/Self* was published and SVS first introduced. In this and other works, SVS looks to be a typographical obsession—on first glance at least. There is something gleeful in the enthusiastic manipulation of computer fonts, glyphs, pixels, and text layout to create meaning. Computing is desktop publishing here. Yet focusing just on the visuals can be distracting because it occludes the degree to which SVS marks Brathwaite's attempt to develop an aesthetic that responds appropriately to a culture that he himself had characterized as creolized and oriented towards oral (rather than typographic) traditions. SVS, I am going to suggest, is ultimately Brathwaite's effort to develop a computing aesthetic that embodies his ideas around nation language. It is, crucially, a *processual* aesthetic: one that positions literature as a set of activities and practices, rather than a product.

Such interest in computer-facilitated composition was not exactly distinct to Brathwaite in this period. The expanding market for word processors, the 'hardware and software for facilitating the composition, revision, and formatting of free-form prose as part of an individual author's daily

workflow' as Kirschenbaum has it, was alerting many a writer, professional or otherwise, to the processual nature of writing.[72] While some described the experience as emancipatory remediation, others butted up against software that was 'subordinating writing practices to a formal, algorithmic order'.[73] What for many was a quirk of the software was for Brathwaite a deeper phenomenon symptomatic of computing in its Caribbean milieu. What marks out Brathwaite is his attempt to think through this recognition in the specific creolized environment of Caribbean culture, with its own oral traditions and aesthetics: might word processing be deployed in the service of a decolonial literary aesthetic?

From the perspective of Brathwaite's emerging SVS, the key poem is 'X/Self's Xth Letters'. It is here that Brathwaite first articulated the possibility of this processual aesthetic and establishes a series of associations that would become crucial for SVS. The poem itself takes the form of a broken dialogue addressed to the speaker's mother and describes in often joyful and expressive language the speaker's use of 'one the best tings since Cicero' even if it 'is not one dem pensive tings like ibm'.[74] Brathwaite uses a flurry of similes, positioning the personal computer within popular contemporary American culture (references include Star Trek, Charlie Chaplin and 'r. t–d2'[75]), a local Caribbean culture, and a broad history of Western technocultural claims of superiority (recalling Coetzee's point in *Dusklands*). This is energetic, brisk poetry, the poetic equivalent to Kenner's prose, that demands a responsive, nimble reader.

Despite the speaker's exuberance, in this poem the computer is in many respects the symbol of Western techno-rationalist power:

> mamma
>
> a doan now how pascal & co/
> balt & apple & cognito ergo sum
> come to hinvent all dese tings since
>
> de rice & fall a de roman empire[.][76]

[72] Kirschenbaum, *Track Changes*, xiii.

[73] Jeanette Hoffman, 'Writers, Texts, and Writing Acts: Gendered User Images in Word Processing Software', *The Social Shaping of Technology*, eds. Donald MacKenzie and Judy Wajcman (2nd edn, Buckingham, UK: Open University Press, 1999), 224.

[74] Brathwaite, 'X/Self's Xth Letters', 81, 80.

[75] Brathwaite, 'X/Self's Xth Letters', 81.

[76] Brathwaite, 'X/Self's Xth Letters', 82.

The speaker then turns to a canonical Western text that stages the confrontation of Carib and European cultures to make sense of the computer: *The Tempest* and the character of Prospero, 'Shakespeare's plantation owner'.[77] The speaker might be 'learnin prospero linguage' but this effort is

> not fe dem/not fe dem
> de way Caliban
> done
> but fe we[78].

If Kenner had comfortably positioned the Turing Machine as part of a longer history of the European Enlightenment, Brathwaite emphasizes its dark underbelly—its structural reliance on enslaved peoples—in his lineage. From here on out the creole computer will have a name: 'Sycorax'.

For Brathwaite, the machine is a product of a colonial culture, but in a caustic pun it offers the possibility that 'prospero get curse / wid im own // curser'.[79] In later writings Brathwaite would develop this analogy between computer and play, naming the computer 'Sycorax' after Caliban's mother. Indeed, he speaks of a specifically female presence operating within the computer and with which he engages: it is she who is 'the spirit / person who creates an(d) /or acts out of the video-style that I workin with'.[80] From the outset SVS is expressed as the product of a creolized (de)colonial encounter, but also one that replicates the distinctly feminized, but no less powerful for that, conception of computer labour in the Caribbean.

SVS is embedded in its creolized context in other ways. The poem is littered with references and similes that position the speaker's engagement with the computer as locally conditioned. One of the longest comparisons the speaker makes is between the computer and his mother's 'ole [/] remmington'. The poem is at pains to emphasize the manual features of the typewriter and to compare these to mundane features of the physical environment. In contrast to the typewriter:

[77] Brathwaite, 'X/Self's Xth Letters', 127.

[78] Brathwaite, 'X/Self's Xth Letters', 84–5.

[79] Brathwaite, 'X/Self's Xth Letters', 85.

[80] Kamau Brathwaite, *ConVERSations with Nathaniel Mackey* (Staten Island: We Press/ Minneapolis: Xcp: Cross-Cultural Poetics, 1999), 189.

> dis one yu na ave to benn down over & out
> off de mistake dem wid white liquid paper. de papyrus
>
> ribbed & soft
>
> before it drei up flakey &
> crink. like yu was paintin yu house[81]

In a similar comparison, the computer doesn't require paper rolled on a pin 'like yu rollin dough'.[82] These tactile, homely images—painting a house, rolling dough—might ostensibly contrast computer and typewriter, but they also work to place both within specific physical settings and as a set of activities. In combination with the dialect language, references to Obeah, to Caribbean texts, to Kingston's 'chauncery / lane' (which Brathwaite records in a note as the city's 'reggae-making and recording centre'[83]), and the inclusion of Bajan expressions, help to identify this as an expressly West Indian locale and the computer as functioning within a distinct creole space. For readers, such specificity also works to underline the processes of interpretation that poetic representations invite. In Brathwaite's work transparency and universal codes are notably absent in depictions of computer use.

Towards the end of the poem, after noting the difficulties of writing in a decolonial context, the speaker describes his computer writing in physical and manual terms:

> yet a sittin down here in front a dis stone
> face/eeee
>
> lectrical mallet into me fist
>
> chipp/in dis poem onta dis tablet
> chiss /ellin darkness writin in light
>
> like i is a some/ is a some/ is a some
> body

The speaker concludes by asking his mother 'why is dat?'[84] Even while writing in light, word processing is not a virtual process here but a physical activity that is closely aligned with the ('some') body, and with clear ties back to a longer history of embodied, impression-based inscription,

[81] Brathwaite, 'X/Self's Xth Letters', 81.
[82] Brathwaite, 'X/Self's Xth Letters', 81.
[83] Brathwaite, note to 'X/Self's Xth Letters', 127.
[84] Brathwaite, 'X/Self's Xth Letters', 87.

whether on the typewriter or stone, via enslaved or free labour. Despite writing on machines designed and manufactured in the USA, Brathwaite's presentation of SVS and poetic renderings of the computer is from the outset one shaped by an awareness of the creolization of Caribbean computing.

Just looking to *X/Self*, however, we might struggle to conceive of SVS as interactive, processual, or even 'video'—it is not overly experimental in its graphic presentation. Rather, Brathwaite's development of the aesthetic was more accretive, processual. Successive versions of 'X/Self's Xth Letters' that appeared in the UK (1992) and US (1993) editions of *Middle Passages*, and *Ancestors* (2001) experimented more expansively with typography. The version printed in *Ancestors* is centre justified; it uses bitmapped glyphs and different font sizes, with fewer forward slashes and more line breaks. By the time we get to *Barabajan Poems* (1994) or *ConVERSations with Nathaniel Mackey* (1999) the visual element of SVS is more clearly defined.

We might see this as sitting in tension with the processual dimension of this word processing. Carrie Nolan points out 'the page layout and choice of fonts used in *ConVERSations* sometimes work against the oral phenomenolization of the text'.[85] This apparently visual emphasis (of SVS), and Brathwaite's interest in computer technology more broadly, have been difficult for some scholars to square with his earlier commitments.[86] They point back to references in *History of the Voice* in which Brathwaite pairs 'pentameter/computer', as evidence of his view that both technologies are symbolic of the tyranny of the colonizer; in such readings SVS is perilously close to a betrayal of nation language.[87] Such readings are compelling—sticking it to the computational man—but I think they downplay the processual, creolized, nature of Brathwaite's conception of computing and the writing aesthetic he considers it might engender.

Rather, in listening to those 'noises in the archives', we might better understand SVS and Brathwaite's computational aesthetics broadly, as a continuation, rather than a rupture, of his own decades-long interest in nation language. Brathwaite's processual aesthetic developed within a broader media ecology. I have been talking mainly of the computational

[85] Carrie Nolan, 'Remediation and Diaspora: Kamau Brathwaite's Video-Style', *Diasporic Avant-Gardes: Experimental Poetics and Cultural Displacement*, eds. Carrie Nolan and Barrett Watten (New York: Palgrave Macmillan, 2009), 77–97, 81.

[86] Jacob Edmond summarizes the response well in *Make It the Same: Poetry in the Age of Global Media* (New York: Columbia University Press, 2019), 56–7.

[87] Brathwaite, *History of the Voice*, 46; Nolan e.g. talks of Brathwaite as a 'heretic' in this context ('Remediation and Diaspora', 95).

archive in terms of (print) word processing, but as the Memory Bank project had recognized, a Kaypro could be equally as valuable as archival storage and as a tool for the creative processing of oral cultural heritage. Brathwaite himself had, since the 1960s, deployed tape recorders to document CAM events, to edit and rework new versions of his texts, and to circulate them: *Rights of Passage* was first shared in public as a tape-recorded reading.[88] The sound technology offered him a way to 'tune into a West African or Black Atlantic tradition submerged but nevertheless present in the English of the Caribbean'.[89] Certainly, some critics have viewed SVS as a continuation of these practices; Kelly Baker Josephs for example identifies in Brathwaite a view of 'the computer as an extension of orality' and sees his as the manifestation of a broader decolonial aesthetic.[90] Jarrel De Matas too has described SVS as a 'folding back of digital media into ancestral forms of communication and expression': as Brathwaite's attempt to ensure nation language keeps pace with technology.[91]

Any pushback to such views often stems from Brathwaite's own complicated statements on the relationship between orality, and by extension nation language, and computing technologies. Brathwaite's *History of the Voice* explicitly positions nation language in a relation to inscription technologies and media. Nation language's features:

> come from a historical experience where [the community of users] had to rely on their very *breath* rather than on paraphernalia like books and museums and machines. They had to depend on immanence, the power within themselves rather than the technology outside themselves.[92]

As Brathwaite suggests, it is the absence of (colonizers') technologies of cultural inscription and transmission that have been constitutive for nation language's particular oral orientation.

[88] See Edmond *Make It the Same*, 80–1, and Andrew Rippeon, 'Bebop, Broadcast, Podcast, Audioglyph: Scanning Kamau Brathwaite's Mediated Sounds', *Contemporary Literature*, 55.2 (2014), 369–401.

[89] Jacob Edmond, 'World Literature in Stereo: Magnetic Tape and the Media Futures of Global Literary History', *SubStance*, 50.1 (2021), 27–53, 33.

[90] Kelly Baker Josephs, 'Versions of X/Self: Kamau Brathwaite's Caribbean Discourse', *Anthurium: A Caribbean Studies Journal*, 1.1 (2003), Article 4, 8, 12.

[91] Jarrel De Matas, 'Ctrl + Alt/HyperText: Toward a Remediated Caribbean Literary Aesthetic', *Journal of West Indian Literature*, 25.2 (November 2017), 9–27, 16. See also Edmond, *Make It the Same*, 57.

[92] Brathwaite, *History of the Voice*, 19.

Brathwaite doesn't, however, indicate that nation language and technology are necessarily oppositional. Granted, from the late nineteenth century onwards anthropological efforts to record and preserve oral traditions and 'dying cultures' were closely tied to the colonial project, as Jonathan Sterne and others have noted.[93] Yet so too, these technologies and resources proved a boon to the writers and thinkers of the decolonialization movement of the years after the Second World War, Brathwaite included. His *History of the Voice*—itself conceived as 'an electronic lecture'[94]—would include a lengthy bibliography on the topic, '[s]ince the background/framework material and ideas that make up this study are comparatively new'.[95] Brathwaite is happy to pair pentameter/computer in places; in others he is enthusiastically describes a literature of nation language as reflecting 'the actual rhythm and the syllables, the very software, in a way, of the language'.[96] Marking a correspondence between nation language use and software is permissible precisely because Brathwaite's view of computing is processual: it isn't the tyranny of the hardware that dominates but rather what the user does with it—the 'software' in his formulation.

Processing Difficulty

To end this chapter, I want to pivot and ask where this leaves the reader of SVS. For many of the other authors in this book, analogies with computing underscore the demands that (modernist) literature places on the reader. How does Brathwaite's processual aesthetic extend beyond the experience of him writing to affect the reader reading?

One of the most notable features of SVS is its difficulty. The formal, graphical, and aural word play, the puns, the localized and internationalized

[93] Jonathan Sterne, *The Audible Past: Cultural Origins of Sound Reproduction* (Durham, NC: Duke University Press, 2003), 26–7. See also my essay 'Generative Transcription: The Interview in Post-World War 2 Anglophone African Literary Culture', *Journal of Postcolonial Writing* (2024), 1–15. doi: 10.1080/17449855.2024.2339395 for a discussion of transcription in the context of Anglophone African (and diasporic) literary culture. Ashleigh Harris's forthcoming writing on how such medial and disciplinary distinctions were embedded in nineteenth-century library science cataloguing practices is incredibly suggestive in this context. See, Ashleigh Harris, 'The Literary Metadata of African Little and Popular Magazines', Uppsala University (2024), https://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-519564.

[94] Edward Kamau Brathwaite, 'English in the Caribbean: Notes on Nation Language and Poetry', *English Literature: Opening Up the Canon*, eds. Leslie A. Fieldler and Houston A. Baker (Baltimore: Johns Hopkins University Press, 1981), 15–53.

[95] Brathwaite, *History of the Voice*, 51.

[96] Brathwaite, *History of the Voice*, 9.

reference points: much is demanded of the SVS reader. Try reading, for example, Brathwaite's 1997 work *ConVERSations with Nathaniel Mackey*. The text is based in part on the transcript of a conversation between Brathwaite and American poet Mackey, platformed before an audience at Poets House, New York in 1993. Conversation is nominally an explanatory, even confessional, form—a 'transparent' interface in the view of the Turing Test and human–computer interaction design. In this text though, the transcript is interpolated with notes, quoted passages, and expansions that are presented in text boxes, variable layouts, and fonts, with the odd emoticon thrown in. This text demands a reader infer relations between the various components: cut-up meets hypertext. Like Coetzee, Brooke-Rose, and Kenner, Brathwaite's computational project prompts him to conceive of literature in processual terms and difficulty becomes a means to make the processual—to make word processing in its most basic definition—discernible.

What is notable in Brathwaite's project is that he insists that a processual entails interaction not only between writer and software but between writer and reader. This isn't Kenner's hobbyist community where the focus is on supporting the user–reader to compose programmes on their own, but rather a communal experience of processing literature into being. The audience for Brathwaite's poetry is comprised of 'watchful breath- / ing witness and *parti /cipants*' and that process of participation, 'the call-and-response', is what 'gathers us into / community'.[97]

> The ancients, the an-
> cestors, the griots,
> had to commit it all
> to memory because th-
> ey had no useful scri
> pt or cursor or camco
> rder or other extra
> sortilege to aid/illu
> minate their passion
> of composition[98]

Nation literature is processual, generated by an embodied relation between poet and audience. Note though that this is a relation that Brathwaite's vision of creolized computer composition can also support. As he puts it,

[97] Brathwaite, *ConVERSations*, 225–6.

[98] Brathwaite, *ConVERSations*, 216–17.

the 'truth of the moment comes not only in the song, the poem, the utterance, but in the *interaction*'.[99] Literature *is* software; it needs to be run in a specific environment in order to execute effectively.

Brathwaite's conception of the Eagle, Kaypro, or MacBook then is of a processual technology, one that can facilitate interaction, agency, call-and-response—even as it can also function as the representation and tool of the oppressor. As he himself would declare in a later SVS work, *Barabajan Poems*, an archivally oriented work that seeks to reconstitute a national canon: 'the forces that created the computer are very similar to our gods of the Middle Passage'. Notably, he continues this declaration with a reference to a key text in scholarship around oral traditions and source of inspiration for numerous decolonial activists, *Conversations with Ogotemmêlli* (English translation 1966): 'so the more I got into "video," the more…I feel I become engaged with the same kind of cosmological/pebbular interfaces, if you like, that a Dogon like Ogotemmêlli.…was involved in'.[100]

Computing is creolized in the Caribbean and best conceived, not as a universal machine, but as processual; it can, in such a conception, offer the fulfilment, rather than the forsaking, of a nation language aesthetic. Inbuilt in Brathwaite's SVS is an insistence on the political and cultural repercussions of processes—whether computational, creolizing, or literary. For him, all writing with a computer can be part of a decolonial literary project, even—to return to where we began—the seemingly banal typing up of a class syllabus.

Programming Literature. Rebecca Roach, Oxford University Press. © Rebecca Roach 2025.
DOI: 10.1093/9780198976578.003.0005

[99] Brathwaite, *ConVERSations*, 227.
[100] Kamau Brathwaite, *Barabajan Poems, 1492–1992* (Kingston and New York: Savacou North, 1994), 378.

5
William Gaddis, Technical Writer

In 1951 *The Atlantic* published a short article on the history of the player piano, a self-playing musical instrument that operated by punched technology. The author of this article suggested that the device offered

> an answer to some of America's most persistent wants: the opportunity to participate in something which asked little understanding; the pleasure of creating without work, practice, or the taking of time; and the manifestation of talent where there was none.[1]

For all its satirical tone and pushback against the mechanization of the arts, the article would probably go unnoticed today if it weren't for its author: William Gaddis (1922–1998). A small, dapper man, hesitant of speech and usually photographed with pen or cigarette interchangeably in hand, Gaddis was a native New Yorker who had come of age during the Great Depression. His first novel, *The Recognitions* (1955), a dense, 'difficult' work, would achieve recognition in part for its negative reception by reviewers.[2] Later, after a twenty-year gap, Gaddis's magnum opus *J R* (1975), 726 pages of difficult-to-attribute dialogic text, would be published to acclaim and a National Book Award. Three other similarly dense novels would punctuate the decades that followed—*Carpenter's Gothic* (1985), *A Frolic of His Own* (1994), and the posthumously published *Agapē Agape* (2002).

That *Atlantic* article, 'Stop Player. Joke No. 4', would be Gaddis's first national publication, an excerpt from a longer treatment of the history of the player piano that he had been trying to publish since 1947. 'Stop Player. Joke No. 4.' was also the first incarnation in print of a topic that would hold Gaddis's attention for the next half century as he researched, drafted, and redrafted the manuscript, which would eventually be published in fictional form as *Agapē Agape*. The player piano, a popular technology in the early

[1] William Gaddis, 'Stop Player. Joke No. 4', *The Atlantic* (July 1951), 92–3, 92.

[2] Jack Green, 'Fire the Bastards!' *newspaper*, no. 12 (1962). Available at *WUSTL Digital Gateway Image Collections & Exhibitions*, http://omeka.wustl.edu/omeka/items/show/10212.

twentieth century, is of course not the universal machine that is a target of this book. Yet for Gaddis it would be a crucial precursor: the 'grandfather of the computer, the ancestor of the entire nightmare we live in, the birth of the binary world where there is no option other than yes or no and where there is no refuge'.[3] Nor is a piece such as 'Stop Player. Joke No. 4' programming literature in the sense that we have hitherto been understanding that yoking. The article is more easily categorized as what we might call 'technical writing': writing that mediates technology. It is a kind of writing that Gaddis would pursue, in between composing the novels that made his name, to pay the bills. In this chapter, I want to focus on that writing.

In computing we often focus on the source code; it is treated as a kind of 'sourcery'[4]. However, as this book has illustrated, computing is an endeavour constituted from a series of technologies that are themselves mediated through writing. This might be in the form of programming languages, but it is as likely to take the form of documentation, pseudocode, textbooks, and the writing that has been used to explain computer technologies to the public and potential users. Hugh Kenner's user guide and explanatory columns; J. M. Coetzee's notes on Atlas subroutines; CBR's instructions to editors around file compatibility: all of these are technical writings about computing.

Crucially, this kind of mediating writing is often conceived as culturally low status. The fact that Donald Knuth's celebrated tomes make the case for the 'art' of programming suggests that the creative work entailed in such writing is unevenly perceived—it forms the basis of his argument in 'literate programming', as we saw in the introduction. Similarly, the code written to maintain, patch, or make a system compatible is often construed as less significant than the initial programme writing, even as it renders a programme useable in a changed context. There are hierarchies within programming languages as well (whether organized around abstraction, functionality, concision, or chicness) and in formats: textbooks don't accrue cultural status to the author, even if they make a significant impact on knowledge production and dissemination. So too, brochures and user manuals might facilitate expansive computer sales and user deployment, but they have historically been dismissed as intellectually inferior efforts.

[3] William Gaddis, 'Treatment for a Motion Picture on "Software"'. Reprinted in *The Rush for Second Place: Essays and Occasional Writings*, ed. Joseph Tabbi (New York: Penguin, 2002), 14–25, 22.

[4] Wendy Hui Kyong Chun, *Programmed Visions: Software and Memory* (Cambridge, MA: MIT Press, 2011).

The writing that mediates computing is not conceived equally, nor is it, as a category, generally culturally appreciated. We often dismiss it as low-status and boring, tending only to notice it when it has been done badly. We don't have culturally prestigious prizes for the best README file, nor do we track who writes the finest programme documentation. But perhaps we should, for technical writing shapes how we interact with computers and what we think that they do—as Gaddis himself knew first hand. It is this recognition that forms the focus of this chapter.

Technical Writings

William Gaddis wrote a lot, for many different employers. After studies at Harvard (where he wrote for the *Harvard Lampoon*), Gaddis would work briefly as a staff writer, fact checking for the *New Yorker,* where his obsession with the player piano was born. Soon after this, he would write articles for the United States Information Service (USIS), radio scripts for the newly inaugurated UNESCO, and film scripts for the US Army. He consulted for the Ford Foundation on instructional television. He wrote promotional material for car manufacturer General Motors and speeches for pharmaceutical company Pfizer International—and all this only up until 1962. In the years afterwards, he would produce numerous writings for Coetzee's sometime employer, computer behemoth IBM.

Testament to Gaddis's prolific freelance writing career, these early instances of technical writing might not, at first glance, seem relevant to our understanding of Gaddis's thinking about computing, or literature. However, I want to suggest otherwise. These projects provided him with research fodder for his player piano project, as he spent 1960–1962 collating material and struggling to turn the 'expanding prospects of programmed society & automation in the arts which may bring an advance, a commitment, even an escape from the tomb of the 9-to-5' into a manuscript.[5] They also, crucially, shaped his conception of writing itself. As a writer for hire in the

[5] William Gaddis to John D. Seelye (21 May 1962), reprinted in *The Letters of William Gaddis*, ed. Steven Moore (Champaign: Dalkey Archive Press, 2013), 244–7, 246. He worked on this version of *Agapē Agape* from 1960 to 1962; an excerpt is reprinted as 'Agapē Agape: The Secret History of the Player Piano (early 1960s)', in William Gaddis, *The Rush for Second Place and Other Writings*, ed. Joseph Tabbi, Paperback edition (London: Grove Atlantic, 2005), 6–13. Also included in the appendix are extracts from a later proposal, discussed below, and the chronology of the player piano.

post-war years, Gaddis was employed by businesses and government entities in the 'tomb of the 9-to-5' and an environment wherein the copy he produced was a fundamental component of institutional information management and technological mediation.

The years after the Second World War were marked by a significant transformation in the ways in which knowledge was understood, stored, and circulated. In 1945 Vannevar Bush, MIT electrical engineer, pioneer of analogue computing, and wartime director of the powerful Office of Scientific Research and Development, which funded some of the first digital computer projects in the US, would publish his seminal article 'As We May Think' in *The Atlantic* (Gaddis was in good company). The article, drawing on Bush's own wartime work, predicted the rise of the information society while also proffering a technical solution: the 'Memex', 'a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility'.[6] Bush's article and imagined technology has often been regarded as prescient, recognizing as it did the expanding informational ecosystem that characterized the mid-century and the challenges it posed. Bush was not alone in his concern around information management: we saw how Bush's colleague Warren Weaver had sung the praises of machine translation as a solution to the information overload facing scientists and administrators. For both men, 'the endless frontier' of science, and new technologies, foremost among them the computer, were tools that might be deployed in this broader battle.[7] They were successful in selling this vision to the public and industry; over the next two decades, the mainframe computer would become a visible symbol of information management and conceived as a universal machine capable of being deployed in all manner of contexts.

The prominence of the digital computer today can, however, sometimes occlude the degree to which it was imagined by these early visionaries as only one tool in a broader medial ecosystem (the Memex, for example, deploys microfilm and photography), that needed overhauling: 'Professionally our methods of transmitting and reviewing the results of research are generations old and by now are totally inadequate for their purpose.'[8] The development

[6] Vannevar Bush, 'As We May Think', *The Atlantic* (July 1945), https://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/.

[7] Vannevar Bush, *Science the Endless Frontier: A Report to the President by Vannevar Bush, Director of the Office of Scientific Research and Development* (Washington, DC: United States Government Printing Office, July 1945).

[8] Bush, 'As We May Think', n. pag.

of information theory (initially 'communication theory') and of cybernetics, the theory of control and communication in systems, would fuel innovation in computing in this era, but the theory was also seen to have a much broader purview, as Weaver himself noted in the introduction to *The Mathematical Theory of Communication* (1949):

> The word communication will be used here in a very broad sense to include all of the procedures by which one mind may affect another. This, of course, involves not only written and oral speech, but also music, the pictorial arts, the theatre, the ballet, and in fact all human behavior. In some connections it may be desirable to use a still broader definition of communication, namely, one which would include the procedures by means of which one mechanism (say automatic equipment to track an airplane and to compute its probable future positions) affects another mechanism (say a guided missile chasing this airplane).[9]

Such a broad definition, encapsulating literature and guided missiles, has been rhetorically useful in positioning the computer—the most visible technological innovation of that theory—as the universal machine; but we might also note that this expansive vision incorporates a wide array of media in its contemplation of the informational—or 'communicational'—ecosystem.

Significantly, the methods that were being developed were not just mechanical; they also included systems and modes to mediate and regulate, the new technologies. 'Technical writing', or writing that mediates technical information and 'controls how technological knowledge is made', was not new at mid-century, but it emerged as a distinct type and field for the mediation of computing in this period.[10] Technical writing had emerged as the awkward love child of English and Engineering, developed in response to the need to communicate and describe technologies to a less-expert or differently specialized audience (the audiences Kenner would later champion).[11] It shared with business writing an interest in

[9] Warren Weaver, 'Recent Contributions to the Mathematical Theory of Communication', in Claude E. Shannon and Warren Weaver, *The Mathematical Theory of Communication* [1949] (Urbana: University of Illinois Press, 1963), 3–28.

[10] Bernadette Longo, *Spurious Coin: A History of Science, Management, and Technical Writing* (New York: State University of New York Press, 2000), x.

[11] Teresa C. Kynell, *Writing in a Milieu of Utility: The Move to Technical Communication in American Engineering Programs 1850–1950* (Norwood, NJ: Ablex Pub. Co., 1996).

communicational methods and styles, while being less explicitly commercial in orientation.[12]

In the technocratic environment of the 1940s, America was threatened with 'being bogged down today as specialization extends'.[13] There was a perceived urgent need for more technical writers and the stakes were high. As one proponent preached in 1942, it was a case of winning or losing the war: 'Faulty writing means misinterpretations, and misinterpretations result in a lack of efficiency and efficiency in all its forms is vital to a prepared army.'[14] Deploying technical writing in the context of wartime helped to combat illiteracy in citizens—'If only to teach a common tongue and a common language of communication'.[15] Technical writing in such contexts could entail scribing reports, propaganda, radio scripts for public information campaigns, or orders for army recruits: it was a tool for national 'defense' by virtue of its universality.

As the list above suggests, technical writing was conceived broadly but was also closely associated with government in the 1940s. As Jon Agar has pointed out, 'government has always been a gatekeeper of information and major user' and 'an office of civil servants can be regarded as an information-processing entity'.[16] Even before the advent of digital computing, technical writing was perceived to be an important tool for (government) institutions. This perceived need for technical writers who might clear the 'bottleneck' of faulty communication would grow in the aftermath of the Second World War and during the Cold War.[17] Concerns around the impact of informational excess and scientific specialization were increasingly conceived in terms of national security: as high-level problems that threatened US science and technological prowess.

The perceived (political) stakes of technical writing would inform its style. Technical writing was supposed to be easy for citizens, scientists, and

[12] JoAnne Yates has persuasively demonstrated that the office memo would be of crucial importance to the expansion of nineteenth-century businesses across the US. JoAnne Yates, 'The Emergence of the Memo as a Managerial Genre', *Management Communication Quarterly*, 2.4 (1989), 485–510. For a more general background, see her *Control through Communication: The Rise of System in American Management* (Baltimore: Johns Hopkins University Press, 1989).

[13] Bush, 'As We May Think', n. pag.

[14] Jay R. Gould, 'Written English in Defense', *Journal of Engineering Education* 33 (March 1942), 523–28, 523.

[15] Gould, 'Written English in Defense', 526.

[16] Jon Agar, *The Government Machine: A Revolutionary History of the Computer* (Cambridge, MA: MIT Press, 2003), 12.

[17] Gould, 'Written English in Defense', 525.

users to read, providing 'efficiency' and 'exactness of expression' and lacking 'jargon', thus enabling institutions to easily communicate and control information about scientific and technological innovation.[18] Ideal technical writing was, as scholar Bernadette Longo put it, 'so clear that it is invisible' while also being 'the mechanism that controls systems of management and discipline'.[19]

Gaddis produced such writing in between writing his novels and wrestling with his player piano project, with long-term consequences. His technical work included authoring articles on innovations in 'pile fabrics', a brochure on *Answers to Cancer*, and speeches for Eastman Kodak Company executives. At one point he would produce a film script for *The Vacutainer Culture Bottle System* and write chapters of a book project (later abandoned as a 'fiasco') on instructional television for the Ford Foundation; he completed numerous other one-off projects in this period.[20]

Technical writing was not without its critics, however. The concern, for many observers, was that technical writers were, like Gaddis, often trained in the arts rather than the sciences. Such origins aren't perhaps surprising given that the job required 'writers skilled in interpreting developments' and utilized techniques commonly deployed in literary studies: a technical writer 'organizes, writes, and edits material', presenting it in a 'form most useful to those who need to use it' as one government handbook put it.[21] Trained in these skills, arts graduates like Gaddis were often cheaper to employ as technical writers than engineer graduates—but the formers' employment and thus the central role of the humanities in producing technical and scientific knowledge was perceived to be problematic by many for having effectively 'rendered scientists mute'.[22]

A major part of the concern came from the differing stylistic commitments of the parent disciplines. This was an era in which literary studies had institutionalized by championing its own scientistic methods of close reading, New Criticism's 'technical bias', and an object of study (modernist

[18] Gould, 'Written English in Defense', 524.
[19] Longo, *Spurious Coin*, ix.
[20] For more detailed discussion of this work, see Ali Chetwynd, 'William Gaddis' "Ford Foundation Fiasco" and J R's Elision of the Teacher's-Eye View', *Orbit: A Journal of American Literature*, 8.1 (2020), Article 3, 1–46. doi: https://doi.org/10.16995/orbit.gaddis.3. And Ali Chetwynd, 'William Gaddis' Education-Writing and His Fiction: A Fuller Archival History', *Orbit: A Journal of American Literature*, 8.1 (2020), Article 1, 1–47. doi: 10.16995/orbit.gaddis.1.
[21] Bureau of Labor Statistics, *Occupational Outlook Handbook 1968–69* (Washington, DC: U.S. Department of Labor, 1969), 200.
[22] Longo, *Spurious Coin*, 126.

texts) that privileged formal experimentation and aesthetic autonomy.[23] That commitment to difficulty was in clear opposition to technical writing's insistence on clarity, 'easy communication', and invisible mediation.[24] For those, like Gaddis, producing this technical writing, there would be a number of aesthetic and stylistic consequences. Firstly, technical writing's mediating function was frequently downplayed even by its proponents. Good technical writing was unmediated, transparent, intuitive, and easy to understand, akin to speech. Amid the Babel of specialist jargon, technical writing facilitated a universal language or common tongue—so went the proponents' rhetoric, (foreshadowing the slightly later arguments made for machine translation and programming languages). This might seem a counterintuitive position for a professionalizing field—one that had expanded to thirty thousand workers by the late 1960s—to take, in that it seemingly devalued the labour that went into such writing. (Fiscally, the writing was reasonably well paid—Gaddis would charge $75 a day for ten days for one IBM contract, and at the end of the 1960s a young technical writer could expect an annual salary comparable to that of a college instructor or a programmer.[25]) What such a strategy did, however do, was insist (albeit only somewhat effectively) on the field's autonomy from literary studies. If the latter privileged the expert, careful reading of difficult texts in this period, technical writing was committed to different values.

Such positioning was also effective in enabling technical writing to underline its own professional contributions in the context of governmental Cold War priorities around scientific advances. Technical writing could counter the poor scientific communication and informational mismanagement that might lose the US the Space Race (technical writers were heavily employed by the aerospace industry[26]). It also, crucially, provided a defensive strategy on another front: the cultural Cold War. As part of efforts to counter the misinformation of Soviet propaganda, and sell US culture internationally, technical writing's championing of transparent communication seemed to be nicely aligned with US democratic ideals and what one historian called the discourse of the 'Nervous Liberals'.[27] Technical writers were thus part of

[23] David Gershom Myers, *The Elephants Teach: Creative Writing since 1880* [1996] (Chicago: University of Chicago Press, 2006), 130.

[24] Bureau of Labor Statistics, *Occupational Outlook Handbook 1968–69*, 200.

[25] Detail taken from letter from Arnold Lemer to William Gaddis (11 December 1963), Folder 469: 'IBM Educational Brochure', Box 131, WGP. Salary information taken from Bureau of Labor Statistics, *Occupational Outlook Handbook 1968–69*.

[26] Bureau of Labor Statistics, *Occupational Outlook Handbook 1968–69*, 202.

[27] Brett Gary, *The Nervous Liberals: Propaganda Anxieties from World War I to the Cold War* (New York: Columbia University Press, 1999).

an army of 'disciplined public communicators, whether [trained] in diplomatic and ambassadorial missions, private and public cultural exchange programs, multinational corporations, or global activist groups' put to work as the engine of American internationalism in this period.[28] By avoiding jargon, technical writers could facilitate world peace and US market dominance to boot.

Gaddis's own technical contracts would reflect the seemingly diverse array of institutional and disciplinary contexts in which a technical writer might find themselves. Early in his career, he was employed as a technical writer by state and non-governmental organizations, often in foreign locations (twenty-nine trips in three years).[29] In 1948, he would write for the US Government in the Canal Zone in Panama. From 1949 to 1950 he would write for UNESCO in Paris, joking (in a letter to his mother) 'for heaven's sake don't tell my mother…She thinks I am playing the piano in a whorehouse'. The letter continues, 'I've pretty well gotten over the idea that not-enough-to-eat and being bitten at night is Romantic.—Hence my passing interest in Soybeans and Mechanical Brains', explicitly juxtaposing literary and technical authorship.[30] In 1951–1952 Gaddis would write for the US Department of State on topics such as 'Development of American Racing Cars' and 'Religion in America'; in the early 1960s he would write film scripts for the US Army Pictorial Center.[31] Gaddis's early writing contracts firmly positioned him within this broader effort to use technical writing in the interests of American internationalism.

Writing for UNESCO's *Courier* title, Gaddis would author pieces on soybeans, post-war library reconstruction, and global copyright systems. In the latter, in terms that very much recall the arguments made by Weaver and others around world peace and information circulation, Gaddis identifies the challenge: 'Many countries, old in civilization, but young in independence, are cut off from the books which are as vital to their social and economic development as technological advice and equipment.' He then celebrates UNESCO's copyright programme as part of an 'over-all effort to break down more of the barriers in men's minds which make impossible the

[28] Merve Emre, *Paraliterary: The Making of Bad Readers in Postwar America* (Chicago: University of Chicago Press, 2017), blurb.

[29] William Gaddis, 'Sheet "A"', undated, 'Army / DoD Contract / Application / Security Check', Folder 488, Box 137, WGP.

[30] William Gaddis to Edith Gaddis (14 September 1949). Reprinted in *Letters*, 147–8, 147.

[31] William Gaddis, CV (undated, c. 1962), Folder 494, Box 139, WGP.

understanding necessary for peace'.[32] Elsewhere Gaddis would produce educational radio scripts for UNESCO's *World Review* that discussed census-taking, libraries that helped to 'build a world of understanding between peoples and nations', and the aforementioned mechanical brains.[33] The latter was a short piece that attempted to explain these computers 'as they exist today'. In contrast to the overblown rhetoric of many newspaper accounts, Gaddis's piece is careful to identify the limits of the technology. Although 'one of the highest achievements of man's inventive genius', these computers 'are not capable of original thinking'.[34] Moving between extensive (but not overly technical) facts about the size, speed, and weight of one machine (the EDVAC at the University of Pennsylvania), accessible analogies (memory is referred to as an 'organ'), and potential applications, the script is a model of technical writing.

Writing about mechanical brains or about library books, Gaddis was the consummate technical writer early in his career. That this is not how we remember Gaddis is testament not only to his fiction but also to how technical writing would develop.

Technical Writing for Mainframes

As the 1940s turned into the 1950s and 1960s, discussions around technical writing would increasingly be subsumed into more focused concerns around data management and automation as they pertained to the mainframe computer. As computers became more visible and familiar to the public, the value of technical writing was (ironically) somewhat obscured; but that didn't make it less important. In many respects 'the birth of the binary world' that Gaddis identified had merely expanded the necessity of technical writing, its forms, and its potential reach.

The growing computer industry expanded the market for technical writing, while also having the effect of belittling it as a mode of writing and technology mediation. The universal machine and its heady rate of innovation needed explaining as much to business customers, and the public, as to scientists. Weaver and Bush had dreamt up scenarios whereby the new digital computers might be deployed but those dreams also needed to be

[32] William Gaddis, 'The Right to Write and the Right to Read', *Courier* (1 May 1950), 4.
[33] William Gaddis, 'Bridge of Books', *Impetus*, iv.1–2 (January–February 1950), 13.
[34] [William Gaddis], 'Mechanical Brains', *World Review*, 36 (22 October 1949), 7–8, 7.

disseminated via technical writing to encourage the technologies' uptake (part of the challenge in the early years of the sector was, as Grace Hopper noted, 'there was no communication'[35]) This need is clearly indicated in the projects that Gaddis would take on for IBM.

From the early 1960s, Gaddis would be periodically employed by IBM to produce technical writing. While he was somewhat familiar with automation technologies through his research on player pianos (and mechanical brains), this was his first look inside the sector in which computers were designed, built, mediated, and sold. In 1963, the Corporate Information office would hire him to write a brochure promoting their support of education. The writing, which was to be 'well organized' and adopt a '"here are the facts" approach', was to be aimed at an internal audience and elite external readers such as 'directors, top officials of universities, etc.'[36] Collating material from corporate speeches, internal and external reports, and newspaper stories, the brochure detailed everything from IBM's fellowship programmes to industry efforts in developing teaching machines—a topic that would pop up in his later novels. In addition to authoring brochures, Gaddis would produce other kinds of corporate writing—speeches for IBM executives, an article for their internal staff magazine, *THINK*—and he would devote considerable time to two film projects related to computing. Collectively, they demonstrate that Gaddis's fascination with what the player piano augured was fed through his role as a freelance technical writer in a corporate environment.

The Company You Keep

In the Fall of 1968, Gaddis was working as a freelance script developer for IBM's educational office. The project was to create a script for an in-house film, 'The Company You Keep'. The educational film was designed to be interactive: it would utilize a question-and-answer model and be accompanied by a worksheet for viewers to record their answers during screenings. Delivered to IBM staff, the film aimed to entertain, instruct, and informally survey them on their knowledge of the company.[37]

[35] Grace Murray Hopper, 'Keynote Address', *History of Programming Languages*, ed. Richard L. Wexelblat (New York: Academic Press, 1981), 7–24, 23.

[36] William Gaddis, Notes for 'IBM Support of Education', Undated [1963], Folder 469: 'IBM Educational Brochure', Box 131, WGP.

[37] William Gaddis, 'IBM: The Company You Keep, A Motion Picture Treatment Prepared for IBM World Trade Corporation' (4 November 1968), 1, Folder 468: 'Software', Box 131, WGP.

The project is intriguing for its structure and the way in which it attempts to replicate, in film, a dialogic instructional format. It utilizes some of Kenner's own techniques, although the emphasis is skewed more towards stuffing the viewer with data than in supporting their learning. The project ostensibly entailed producing an effective piece of educational material, but staff at IBM were almost overwhelmingly focused on the significant content and 'facts' to be included—and Gaddis, an obsessive collector of clippings for his player piano project, was only too happy to oblige. The hundreds of facts he collated are also notable for their emphasis on the computer's role within a broader informational and international context. As the client had explained, IBM marketing staff were more interested in outlining available corporate services than in detailing specifics of hardware.[38] The emphasis was to be on how to sell the mainframe computer to business, in the US and worldwide, as a tool and service that contributed to management control and efficiency. IBM was conscious of the need to 'sell' computers to business executives, and Gaddis's project reflects this awareness.

Gaddis's collection of facts promotes an image of IBM as offering a complete global information service, in which the mainframe is but one part. We have the Hursley Advanced Administrative Systems Complex, a UK-based internal system for managing order processing; RETAIN, the Remote Technical Assistance and Information Network, based at the Poughkeepsie and the IBM 'worldwide tele-communications network' with centres in seventy-one cities.[39]

This vision of IBM as a service company is only underlined elsewhere as Gaddis includes extensive notes on the 'massive new Management Information System MIS w/ computers to be used internally for the first time for vast array of management functions'. As his notes explain, 'point: sales, will improve IBM's ability to show others how to use computers profitably in ways few are using them, sell bus customers on the new comp applications being incorporated into its own operations…for first time a real-time computerized inf system will have a fundamental bearing on IBM's corporate decision making'.[40] Gaddis's notes reflect a shift that was taking place in the computer industry in the late 1960s. This was a moment in which, as the sector matured and the paper mediations thickened, as the mainframe moved from the mathematics lab to government and corporate

[38] F. William Bryant, Jr to William Gaddis (24 October 1968), Folder 468: 'Software', Undated [1968], Box 131, WGP.
[39] William Gaddis, Notes titled 'Gr IBM Facts Book', 1.
[40] William Gaddis, Notes titled 'Gr IBM Facts Book', 11.

offices, the computer was reconceived as a central node in a 'totally integrated management information system'—the MIS of his notes. This would be the era of what Thomas Haigh has call the 'systems men', or Charlton D. McIlain the 'committeemen', where mainframes were one tool in a broader management philosophy of total information control.[41] While the MIS dream would ultimately prove unachievable, the computer became, in this era, representative of a new kind of informational ecosystem to be sold to clients. In Gaddis's project, we see a company sales strategy, and internal staff communication vehicle, explicitly designed to replicate that vision and the ways in which a mainframe might be utilized in corporate systems and data management procedures. As Gaddis noted for his own thinking on the topic, 'The challenges toward which this question & answer is directed is that of helping the customer to get the most out of his equipment, which appears to be the ground of a major complaint throughout the field.'[42] Selling the computer necessitated explaining how that equipment might be deployed.

This vision would also affect the way in which technical writing itself was understood, as Gaddis, cog in the MIS machine, likely appreciated. This wasn't in terms of the physical writing experience—word processing would generally wait until the microprocessor and personal computer came on to the scene, as we saw earlier. In his work on the IBM film, Gaddis wrote notes and drafts in longhand, before painstakingly organizing and typing these up on a typewriter, then adding hand annotations, before repeating the process numerous times. He recorded all of his sources in the margins, numbered everything, and created a paper trail to rival that of J. M. Coetzee. The archive is, in this respect, an extension of his own technical writing process; but what Gaddis' research and writing methods also indicate is the degree to which technical writing, for all its aesthetics of transparency, is a multimedia, frequently paper-based material substrate on which the mainframe sits as the symbol of MIS.

This point is one that Gaddis clearly contemplated extensively. His notes underline the degree to which the next generation of computers (it's always the *next* innovation that will realize the vision) 'will be characterized by the

[41] Thomas Haig, 'Inventing Information Systems: The Systems Men and the Computer, 1950–1968', *Business History Review*, 75 (Spring 2001), 15–61. Charlton D. McIlain's term comes from *Black Software* where it refers specifically to the members of the Commission on Law Enforcement and the Administration of Justice (1965) who demonstrated, in the aftermath of the Watts Riots, that computers could be used to 'model and automate complex decision making' (204) in the context of Law Enforcement.

[42] William Gaddis, Notes titled 'Gr IBM Facts Book', 12.

ability of people to communicate directly w/ the machine as opp. to indirect through cards & tape.'[43] Computer interactions are, as Gaddis is noting, material and part of a broader material system of information management. His research notes include extensive comment on global technology diffusion—'the tech gap' (that so annoyed Brathwaite) is, in Gaddis's summation, 'rather a management gap' involving business communications: 'Computers force companies to revise their structures.'[44] In his contract writing for IBM, Gaddis is grappling with the challenges identified presciently by Weaver and Bush in the 1940s, and contemplating the role of the computer as a tool in these wider information systems. Perhaps fittingly, Gaddis's archives also suggest that the challenges of information management were being felt, even within IBM. Drafts of his scripts, accompanying documentation, and his collated lists of potential facts circulated around numerous departments in the company, the subject of frequent meetings, memos, and collaborative writing sessions. At IBM technical writing was heavily constrained by exactly the informational excess that it, and its technological object, was trying to control. For all the dreams of MIS, of universal machines and total communicational control, the reality was, for Gaddis at least, mediative overload.

Software

In the summer of 1969, Gaddis would produce another film script for IBM. This one was dedicated to the subject of software. Aiming to be a short introduction to 'problem-solving' and to software's contribution to that activity, the script aimed to explain the concept to an unfamiliar general audience.

Gaddis's files provide us with insight into his research methods and process of knowledge acquisition on this topic. His sources are other introductory technical writings—*Business Weekly* articles, a government-produced *Automatic Data Processing Glossary*, sales material and operating information about the IBM System/360, handbooks on becoming a programmer at IBM, and glossy sales brochures titled *Introducing…the Computer* and

[43] William Gaddis, Notes titled 'Gr IBM Facts Book', 9.

[44] William Gaddis, Notes titled 'Background and Areas against which to address questions', 1. Undated [1968], Folder 468: 'Software', Box 131, Series V Corporate Writings, WGP; William Gaddis, Notes titled 'Gr IBM Facts Book', 10.

What is a Computer? These sources, on which he made notes, were themselves technical writings designed to explain the computer and software to a non-expert reader. Many of these ideas would find their way into his player piano files. Gaddis was learning himself, and his notes include definitions for terms and his own sketches of processes and symbols for 'bit tables', 'processing', 'input or output', and 'YES or NO question' (logic gates).[45] The notes also include numerous examples of how software had been presented in other technical writing. Gaddis collected examples of programming real-world examples or compelling analogies used by other writers: software as analogous to playing chess, or operating a player piano; programming as akin to putting on a washing machine cycle, waiting to cross the street, or using a Jacquard loom. While, elsewhere, Gaddis was thinking about the mainframe as one node in a broader MIS, in this project his goal was to make software and 'the remarkable results that may be produced by various types of problem-solving apparatus from essentially simple data and instructions', explainable to himself and his viewers, through familiar objects and activities.[46]

The script itself went through various iterations. Appropriately for the topic, this seems to have been the result of Gaddis's struggle with the sequencing of his material. At one point he cut up all of the typescript in order to quite literally cut and paste it, experimenting with different orders. He experimented with analogies, including a traffic lights analogy and the player piano, to best demonstrate the concept of software lucidly and efficiently to the non-expert. Somewhat surprisingly, in one version of the script, he makes reference to celebrated modernist poet, Gertrude Stein: 'As this recital concludes, the Narrator quotes the dying lines of Gertrude Stein, when she asked "What is the answer?"…[or failing that] "what is the question?"'[47] Unsurprisingly, the reference to the difficult poet would eventually be cut from a script aiming for reading ease. The revised version retains the allusion, however, ending with the Narrator emphasizing that:

[45] William Gaddis, Untitled notes, undated [1969], Folder 480, 'Miscellaneous Corporate Files', Box 134, WGP.

[46] William Gaddis, '"Software" A Motion Picture Treatment (Revised)' (19 June 1969), 3, Folder 480, 'Miscellaneous Corporate Files', Box 134, WGP.

[47] William Gaddis, Untitled, incomplete draft of script, undated [1969], 5, Folder 480, 'Miscellaneous Corporate Files', Box 134, WGP. Stein was a nationally recognized celebrity author in the 1930s, broadcasting her reputation for difficulty widely, as various scholars have noted. See Karen Leick, *Gertrude Stein and the Making of an American Celebrity* (New York: Routledge, 2009).

> almost any problem that can be adequately stated, can be solved. The essence of problem solving lies in the ability to envision the elements that will compose the solution, and the essence of software is the programmed organization of these elements, producing the harmony in which the solution will be embodied. 'And what is the answer?' the Narrator askes, of this search for solutions, or he concludes, then is the secret of problem solving 'What is the question?'[48]

A film designed to communicate efficiently and transparently to the non-expert was no place for Stein; but it is worth noting that in his technical writing for IBM, Gaddis is content to conceive of literature and programming within the same framework.

In the years after 1969, Gaddis would maintain a relationship with IBM, but it would shift. When Gaddis came again to write for the company, it would be as an expert in fiction, rather than in technical writing. However, in the 1960s and 1970s, years when Gaddis was authoring *J R* (1975), his magnus opus, he was simultaneously employed as a technical writer. I have suggested that the technical writing that Gaddis was producing for IBM in this era led him to understand the mainframe as one object in a wider MIS environment, but one that necessitated laborious written mediations to function. But how, we might ask, did this shape Gaddis's own fiction?

Technical Literature

In 1975, Gaddis would publish his second 'encyclopedic novel'.[49] Unlike *Amalgamemnon*, *J R* is a jeremiad against the computer, of sorts. Gaddis scholar Joseph Tabbi would later declare that the novel 'can be seen to have anticipated developments in hypertext and other means of linking literary texts together and with electronic and digital media', and the novel has certainly been understood as a satire of networked systems.[50] I suggest, more specifically, that *J R* is a satire of the informational ecosystem in which the

[48] William Gaddis, '"Software" A Motion Picture Treatment (Revised)' (19 June 1969), 11. An earlier version, which kept the reference to Stein, is reprinted as 'Treatment for a Motion Picture on "Software" (early 1960s) in Gaddis, *Rush for Second Place*, 14–25.

[49] Stephen J. Burn, 'The Collapse of Everything: William Gaddis and the Encyclopedic Novel', in *Paper Empire: William Gaddis and the World System*, eds. Joseph Tabbi and Rone Shavers (Tuscaloosa: University of Alabama Press, 2007), 46–62.

[50] Joseph Tabbi, 'The Technology of Quotation: William Gaddis's 'J R' and Contemporary Media', *Mosaic*, 28.4 (1995), 143–64, 143.

mainframe operates and specifically against technical writing as a genre. Earlier in his career, when *The Recognitions* was published, Gaddis was already deploying a dense style. This is attributable to his own early engagement with technical writing. The plot of *The Recognitions* revolves around religion and art forgery, alludes to the Faustian pact, and satirizes much of modern life. It alludes to the player piano briefly but doesn't engage extensively with emerging technology or the corporate environment in which computers would thrive: this was Kenner's counterfeiters without the automata. That representational task would be left to novel number two.

The first thing to know about *J R* is its length: 726 pages told almost entirely through difficult-to-attribute dialogue, with no chapter breaks. Otherwise, the book is difficult to follow thanks to its confusing series of events, hundreds of characters, shifts in focalization, and rapid pace. The plot, related through payphone calls and postal orders, follows a schoolboy, J R, who makes a paper fortune on penny stocks; information accumulates material and financial heft in this novel. The office features heavily as an informationally overloaded environment; the novel also includes teaching machines, punched cards, and multiple communicational devices and systems—which rarely function effectively. As Rochelle Gold has noted, the book is interested in critical literacies in such an environment: 'J R's view of literacy is fundamentally instrumental. Reading and writing are necessary for business communication, but they lack any broader humanistic purpose.'[51] Through the character of Mr Bast (also present as 'E Gast, E Bast, B Best'), J R's ineffective music teacher who strives to write an opera but gets sidelined into helping the boy in his business endeavours, the novel presents an increasingly automated corporate life butting up against education and art; art loses.[52] Cultural objects become recategorized as interchangeable commercial goods. Gaddis brings his satirical eye and technical writing experience in analogy-drawing to bear: at one point a collection of 'Great Books' becomes spatially interchangeable with a washing machine (an analogy one suspects Gaddis recalled from his research on software).[53] The player piano project even makes an appearance as 'a book about order and disorder more of a, sort of a social history of mechanization and the arts, the destructive element'.[54]

[51] Rochelle Gold, 'Pre-written Business Correspondences and Computer Therapists: William Gaddis's J R, ELIZA, and Literacies in Conflict', *Electronic Book Review* (7 April 2024), https://doi.org/10.7273/ebr-gadcent3-5.

[52] William Gaddis, *J R* [1975] (Champaign: Dalkey Archive Press, 2013), 557.

[53] Gaddis, *J R*, 176.

[54] Gaddis, *J R*, 244.

Gaddis's novel touched a nerve in the literary establishment. On publication, it won the National Book Award for Fiction and reviewers have since appreciated it for its satirical take-down of society's informational excess. Unlike other computer novels (Gaddis wasn't the only one interested in the technology), such as Kurt Vonnegut's 1952 work *Player Piano*, John Barth's *Giles Goat-Boy* (1966), or John Updike's *Roger's Version* (1986), in which the mainframe computer looms large as a threatening symbol of automation, in *J R* the mainframe remains in the background. Gaddis's target is the broader ecosystem of communication for which the mainframe is often the metonym. Gaddis's target is the technical writing—the office communiques, speeches, telephone exchanges, ticker tape stock data, sales brochures, and newspaper accounts—that surrounds and mediates computer technologies.

The result of such satire in this novel is an aesthetic widely labelled 'difficult'.[55] It was a label that Gaddis, at points, relished or rejected. Others were less sanguine. Novelist Jonathan Franzen blasted Gaddis with the title 'Mr Difficult', seeing his work as representative of a certain impenetrable form of fiction that privileges 'status' over a reader 'compact' that communicates something—preferably pleasure—to the reader.[56] Franzen has a point. But we can also surmise that the length and 'difficulty' of *J R* and novels like it are features designed to produce a deliberate aesthetic response and readerly experience, which is conceived against technical writing's drive towards efficiency and clarity of expression. If the day job insists on lucidity and transparency, then making the reader work for comprehension during evenings and weekends offers an aesthetic refuge and antidote. This is a question of what critical literacy might look like in an era of technical writing, but it is also a question of literature's value in the broader informational ecosystem. In Gaddis's novels—*J R*, but also his later works as we shall see—literary difficulty is produced specifically through an inversion of technical writing's norms and style.

This often takes the form (as was the case of Brooke-Rose) of difficult to read dialogue. Describing the novel, Gaddis scholar Steven Moore notes that it 'reads like a transcript of real speech: ungrammatical, often truncated, with constant interruptions by other characters—and by telephones,

[55] Compare Joseph Tabbi on Gaddis (and others): 'whose sensibility is in part structured by technology and whose lengthy, complicated fictions undertake a detailed engagement in the materials and simulations of the corporate technological culture', *Postmodern Sublime: Technology and American Writing from Mailer to Cyberpunk* (Ithaca: Cornell University Press, 1995), xiv.

[56] Jonathan Franzen, 'Mr. Difficult: William Gaddis and the Problem of Hard-to-Read Books', *The New Yorker* (30 September 2002).

televisions, and radios—with rarely an identifying (and never an interpretive) remark by the author.'[57] Conversation, in the corporate MIS environment in which much of *J R* is set, becomes the key producer of difficulty.[58] It acts as a kind of formal Turing Test that proves that the reader is human, struggling to determine context, speaker, and meaning in this new era. Specifically, in making dialogue difficult, Gaddis undercuts the idea that talk is an unmediated, intuitive, and efficient form of communication and in doing so punctures technical writing's shared aspirations of transparency.

Gaddis's decision to untether dialogue in *J R* challenges readers to ascribe speakers or context. Joseph McElroy suggested the term 'networked discourse' to describe the effect; as he reads *J R*, 'I can feel like a (riveted) central operator of an ultimate conference call, the voices crossing this out-of-control system part *of* it.'[59] I disagree. This isn't a novel particularly interested in networked computing, a technology and metaphor after all, that (for all IBM's service claims) would only gain prominence in the 1990s. Rather, in this novel technical writing's norms have become aestheticized and repurposed as a form of difficulty and a means by which the reader must demonstrate commitment to close reading. Gaddis would suggest that this 'floated dialogue' was a form of 'authorial absence' that facilitated reader involvement; it is also a technique by which the creative writer can skewer the technical writer's professional commitments in the process.[60]

Gaddis's fiction was, I am suggesting, fundamentally affected by his freelancing activities. It isn't that *J R* or his later novels were replete with mainframes, or technical writings, but rather that they were the products of a communicational ecosystem in which the technical writing that computing necessitated was a significant and expanding component.

[57] Steven Moore, *William Gaddis: Expanded Edition* (London: Bloomsbury, 2015), 76.

[58] This is of course despite the growing realization that talk had its own forms and conventions—spurred by the work of people like Erving Goffman and technologies such as the tape recorder. See my *Literature and the Rise of the Interview* (Oxford: Oxford University Press, 2018), 132–168; and Erving Goffman, *Forms of Talk* (Philadelphia: University of Pennsylvania Press, 1981).

[59] Joseph McElroy, 'Gaddis Dialogue Questioned', *Paper Empire: William Gaddis and the World System*, eds. Joseph Tabbi and Rone Shavers (Tuscaloosa: University of Alabama Press, 2007), 63–71, 67.

[60] William Gaddis and Zoltán Abádi-Nagy, 'William Gaddis: The Art of Fiction No. 101', *The Paris Review*, 105 (Winter 1987), https://www.theparisreview.org/interviews/2577/the-art-of-fiction-no-101-william-gaddis.

Technical Criticism

J R would not be Gaddis's only experiment in fiction writing after technical writing. In 1978, he would be commissioned by IBM to author an article for their in-house magazine *Think*. The topic of the assignment (to be covered in 1,200–1,500 words) was broad. Everything from the societal role literature plays, to its current value in a world oriented towards scientific and technological innovation, to its import for preventing alienation in the world was to be Gaddis's remit, according to the editor.[61]

The article that Gaddis produced, entitled 'Literature and Crisis' was indeed expansive in its approach. Beginning from his perception of the dominance of science and technology in contemporary society, Gaddis notes the fields' 'disturbing' tendency 'to apply their dispassionate approach, their criteria of evidence and proof, to values—call them moral, spiritual, simply human—which may well have been brought into question by the achievements of science and technology in the first place'. The stakes are raised, he suggests, in the 'anonymous face of the necessary collective, complex nature of scientific inquiry and technological advance, since the implication, and not infrequent reality, casts the other side of the equation—the human side—in correspondingly impersonal terms.'[62] This characterization of technology is clearly one taken from Gaddis's own experience as a technical writer—working with complex information systems and multiple collaborators—and the arguments move between the medial and the abstract. Gaddis continues, recalling Bush and looking forward to Kenner's *The Mechanic Muse* to suggest that literature is 'civilization's most venerable memory bank, readily approached in terms of information handling, storage retrieval and—especially as drama and the novel—the problem solving implicit in plot and technique.'[63] Thus, he is happy to conceive of poetic cliché in probabilistic terms but also worries that literature 'runs the constant danger of becoming as vulnerable to scientific theory as the particular science it borrows from.'[64] The piece is clearly informed by Gaddis's years working across these two endeavours.

What is striking here, however, is what is not mentioned. In an article for his former employer, on the topic of literature in a technological society,

[61] Details taken from Geoffrey D. Austrian to William Gaddis, 1 November 1978. 'IBM Piece', Folder 683, Box 170, WGP.
[62] William Gaddis, Draft of 'Literature and Crisis', 1, 'IBM Piece', Folder 683, Box 170, WGP.
[63] Gaddis, Draft of 'Literature and Crisis', 3–4.
[64] Gaddis, Draft of 'Literature and Crisis', 4

Gaddis makes no mention of his own background. He writes from a position of knowledge, about the 'symbiosis' of the two cultures, but he doesn't advertise this—in fact, he would eschew his expertise entirely, complaining in private at having to rewrite the piece because 'it's as usual a little too much of my density for [the IBM] audience: why can't I just write simple sentences?'[65] Why does he occlude his own expertise?

The answer lies, I think, in perceptions of technical writing, and particularly its reputation within literary studies, and it's a point worth reflecting on in more detail, because it suggests that a recovery of Gaddis's technical writing might have broader implications for our understanding of the literary studies' historiography.

The emergence of technical writing was, as noted earlier, part of a broad concern with communication in the mid-twentieth century as the expansion of advertising, propaganda, and mass media had heightened attention about the relation between citizens' literacy, technological, and scientific advancements and the fate of democracy in the context of the Cold War.[66] English departments had been able to successfully position themselves as an important bulwark in this period. In the years after the Second World War, however, this distinct claim was increasingly being challenged thanks to academic reorganization and the emergence of fields such as communication studies, creative writing, area studies, the social sciences, and technical writing. One strategy that literary studies adopted was to embrace difficult modernist literature as a key object of study: New Criticism, as Andy Hines notes, 'was defined by the objectification and isolation of the literary object from the circumstances of its creation'—an 'enclosure of literature' that generated an object of study and a method to bolster departmental expertise.[67] Such a move had consequences external to the academy; it produced not only the 'large-scale exclusion of Black Writing' but also a valuable commodity for the US government—modernist aesthetic autonomy—as it attempted to sell the world on America's commitment to free-

[65] William Gaddis to Sarah Gaddis (3 December 1978), reprinted in *Letters*, 342–3, 343.

[66] For the broad debates see, e.g. John Durham Peters, *Speaking Into the Air: A History of the Idea of Communication* (Chicago: University of Chicago Press, 1999); Gary, *Nervous Liberals*; Nicholas J. Cull, *The Cold War and the U.S. Information Agency: American Propaganda and Public Diplomacy* (Cambridge: Cambridge University Press, 2008); Eric Bennett, *Workshops of Empire: Stegner, Engle, and American Creative Writing during the Cold War* (Iowa City: University of Iowa Press, 2015); and the essays collected in Gregory Barnhisel and Catherine Turner, eds., *Pressing the Fight: Print, Propaganda and the Cold War* (Amherst: University of Massachusetts Press, 2010).

[67] Andy Hines, *Outside Literary Studies: Black Criticism and the University* (Chicago: University of Chicago Press, 2022), 3.

dom during the Cold War, as scholars such as Peter Kalliney and Greg Barnhisel have shown.[68] The English department was able to successfully position itself as the location of an expertise and pedagogy that could mediate difficult, resistive texts and train the 'good readers' that US internationalism required.[69] Although English studies has always maintained heterogeneity of method and approach, in the post-war period it gained significant state support, distinct from other writing and reading endeavours, through its expressed commitment to difficult texts.

Gaddis was no doubt aware of the ways in which his two writing endeavours exhibited sometimes divergent orientations. Having worked as a technical writer for some of the agencies that successfully promoted aesthetic autonomy abroad early in his career, following the publication of *J R* Gaddis would also perform stints as a cultural ambassador for the US Information Agency (the USIS's successor). In the 1970s and 1980s, he took trips to South East Asia, Australia and New Zealand, and Germany to discuss his literary work and craft. As the public diplomats across his trips recorded, Gaddis 'enjoyed noting the difficulties readers and reviewers alike have in dealing with his style, and was almost gleeful in his assertion that his readers should have to work'.[70] Gaddis was happy to promote his aesthetic commitment to difficulty, in a venue in which claims to communicational transparency and the values of technical writing often tussled with those of aesthetic autonomy.

This was not the only space in which Gaddis discussed his craft, nor the only space in which literary studies expertise, and institutional standing, was being negotiated. From 1976 to 1979 Gaddis would teach both literary and creative writing courses at Bard College in New York state. As a creative writing teacher at Bard, Gaddis was joining a liberal arts college with an already developed reputation in this area: Bard's programme had hosted Saul Bellow, Ralph Ellison, Mary McCarthy, and Isaac Bashevis Singer. His 1979 course 'The Literature of Failure' had a reading list that stretched from Max Weber to Upton Sinclair to Joan Didion and covered everything from the puritanical origins of Capitalism, to poetry, to failure as an act of political rebellion. Many of the discussion topics and texts would find their way

[68] Hines, *Outside Literary Studies*, 3; Peter J. Kalliney, *The Aesthetic Cold War* (Princeton: Princeton University Press, 2022); and Greg Barnhisel, *Cold War Modernists: Art, Literature, and American Cultural Diplomacy* (New York: Columbia University Press, 2015).

[69] Emre, *Paraliterary*, 1.

[70] Unclassified USIA telegram from AM Embassy Wellington to USIA WashDC, 'Subject P/DC AMPART WILLIAM GADDIS 1987', Folder 2, Box 98 F1–2, WGP.

into his article 'The Rush for Second Place', published in *Harpers*'. There he would satirize, like Coetzee, the whiz kids' computational policies in the Vietnam War: 'As computer technology's appetite for precision is enhanced by its own enlarged complexity, the arch enemy must look increasingly to human error for an ally.'[71]

Creative writing programmes, such as Gaddis taught at Bard, had expanded as part of the same expansive federal support for higher education, science and technology, and communications efforts in the Cold War era that saw the professionalization of technical writing.[72] Those same creative writing programmes also regularly promoted a particular vision of literature. In his much-lauded account of the institutional rise of creative writing in America, *The Program Era*, McGurl pitches John Barth's *Giles Goat-Boy* both as emblematic of the post-war university and as a product—Barth would write the novel while employed at Penn State—of creative writing's institutionalization. *Giles Goat Boy*, 'like a comic, novelistic version of "The Waste Land"' is representative of a certain strain of post-war fiction that McGurl labels 'technomodernism'. His term 'reasserts the obvious continuity of much post-war American fiction with the modernist project of systematic experimentation with narrative form, even as it registers a growing acknowledgment of the scandalous continuity of the literary *techne* (craft) with technology in the grosser sense'.[73] Others in his canon include Thomas Pynchon, Don DeLillo, and Joseph McElroy. McGurl's point is that this highly regarded canon is both formally experimental (difficult) and technologically focused. For McGurl, that fiction is one of the characteristic products of the era, hence his titular pun.

McGurl's yoking underlines the degree to which Anglophone literary studies, in the English department and next door in the creative writing lab often celebrated modernism difficulty. But the yoking misses the degree to which creative writing programs were part of a larger reorganization of the writing ecosystem in this period, which includes the less glamorous but equally institutionalized and rapidly expanding field of technical writing. It

[71] William Gaddis, Typescript draft of 'Rush for Second Place', 10 (Published in the April 1981 edition of *Harper's Weekly*), Folder 455, Box 128, WGP, also available at: http://omeka.wustl.edu/omeka/items/show/10064.

[72] Myers, *The Elephants Teach* details much of this history; Eric Bennett's *Workshops of Empire* places these developments firmly in the context of the Cold War. The interrelation these various efforts is embodied in the figure of Wilbur Schramm, father of Communication Studies, who began by teaching English, and then creative writing, at the University of Iowa.

[73] Mark McGurl, *The Program Era*: *Postwar Fiction and the Rise of Creative Writing* (Cambridge, MA: Harvard University Press, 2009), 42.

is perhaps less noticed because, despite taking place on campus and in the office (and on deployment), we often segregate these environments. Gaddis would explicitly compare the gigs: hesitant about what he would teach at Bard, he nevertheless noted it was 'a good deal more realistic than sitting here waiting for IBM or Eastman Kodak to call'.[74] Later in 1994 Gaddis would evince a similarly broad vision when, after serving on the Literature panel to disburse grants from the National Endowment for the Arts, he railed against the miniscule funds devoted to the 'endangered species' of the writer in comparison to the 'bewildering' 'sums paid lobbyists who actually help to write, for example, watered-down versions of Congressional anti-pollution legislation.'[75] Writing, whether technical, creative, or lobbyist, was part of a broad ecosystem and Gaddis moved variously between gigs as creative writing tutor, as technical writer for government offices or corporations, and back again. And just as creative writing programmes shaped its participants' conceptions of literature, so too has technical writing shaped employment—something that often gets left out of accounts of the history of literary studies.

Notably, just as Gaddis was not alone in his employment in the creative writing programme, so too he wasn't idiosyncratic in his other professional activities. Many of his 'technomodernist' peers (many of the writers that Franzen would later castigate for their difficulty) were also employed as technical writers. Thomas Pynchon worked for Boeing in 1960–1962, producing material for (amongst other publications) *Bomarc Service News*, 'an essential communications conduit' for the development of the Bomarc missile.[76] Kurt Vonnegut would do similar writing for General Electric in 1947–1950. Richard Yates wrote sales materials for mainframe pioneer Remington Rand from 1949 to 1951, as the UNIVAC was under development, and returned as a freelance writer from 1953 to 1960. He wrote drafts of *Revolutionary Road* on corporate stationary: 'Selling the mainframe computer and narrating postwar conformity were literally different sides of the same page', as Brian Rajski has it.[77]

While the authors previously featured in this book were somewhat unique in their engagement with computing, Gaddis is almost symptomatic

[74] William Gaddis to Matthew Gaddis (29 April 1976), reprinted in *Letters*, 311.

[75] William Gaddis, 'On Creative Writing and the National Endowment for the Arts (1994)', in *Rush for Second Place*, 115–19, 117, 119.

[76] Katie Muth, 'The Grammars of the System: Thomas Pynchon at Boeing', *Textual Practice*, 33.3 (2019), 473–93. doi: 10.1080/0950236X.2019.1580514.

[77] Brian Rajski, 'Writing "Systems": Richard Yates, Remington Rand, and the Univac', *Contemporary Literature*, 54.3 (Fall 2013), 550–77, 552.

of a certain trend in post-war American fiction. Yet despite this employment being somewhat common, the expertise is often figured as irrelevant to these authors' fictional writing.[78] Any connections made (like those between Kenner's computing and critical writings) are somehow surprising, as if the Pynchon writing *V*, published by J. B. Lippincott, and the Pynchon writing 'Soldering', published by *Bomarc Service News*, are composed by two halves of a severed brain.

Intriguingly, the critical material that bolsters the perceived literary status of the fictional writing often replicates the difficult reading experience of the works themselves. Sometimes this is at the level of syntax, other times it is the overreliance on opaque postmodern theory for interpretative payoff. Franzen makes this very point:

> Even Steven Moore, a Gaddis scholar whose criticism is a model of clarity and intelligent advocacy, lets his enthusiasm get the better of him. 'J R', for Moore, is a 'lean and economical' book, because its inferential, all-dialogue form forces readers to supply missing descriptions and information; the purpose of a novel being, I suppose, to capture and efficiently store data.[79]

This is difficult, postmodern criticism; anyone trying to enter the scholarly conversation about Gaddis (or McElroy, or Pynchon) might recognize the formal parallels, whereby references to entropy and systems theory alternate with allusions to Derrida, Baudrillard, and capitalist critique.

I don't say this to slight these incredibly erudite and committed scholars, but rather to underline the degree to which the aesthetics of literary difficulty are often replicated in the critical reception to these novels, even as the material particularity of the authors' production of that difficulty is sometimes overlooked. A literary aesthetic conceived, in this context, as an antidote to technical writing has also been replicated in the critical writings that mediate it. In the complex informational system of modern literary studies, difficulty is still the bulwark, with the result that we are in danger of missing the broader alignments, tensions, and relations that exist between fiction and its less favoured but more prolific cousin.

Gaddis's final writing reads almost as a commentary upon this trend. After two novels, *Carpenter's Gothic* (1985) and *A Frolic of His Own* (1994),

[78] In the case of Gaddis, Ali Chetwynd has started to sketch out Gaddis's technical writing activities and the ways in which he 'transmuted' this material in his novels. 'Friction Problems: William Gaddis' Corporate Writing and the Stylistic Origins of *J R*', *Orbit: A Journal of American Literature*, 8.1 (2020), Article 4, 1–64. doi: 10.16995/orbit.gaddis.2.

[79] Jonathan Franzen, 'Mr Difficult', n.pag.

which retained his reputation for difficulty in the form of a 'romance' and a satire against legalese (and garnering him numerous awards), Gaddis would return to his first technological obsession. In a book proposal Gaddis would describe 'Agape Agape: The Secret History of the Player Piano' as a 'satirical celebration of the conquest of technology and of the place of art and the artist in a technological democracy'.[80] Although called 'satirical', the precis describes a 'non-fiction book' and critical 'study', complete with references, 'thesis', and a sample chapter that recalls Kenner's expansive, but still recognizably critical, style from *The Stoic Comedians* and *The Counterfeiters*.[81] (Gaddis would discover the latter text in 1987, noting, 'well damn! that settles it, mine will never be done; though something still remains that drives me to tear out & save anything I come across on mechanization & the arts to add to the 30 year hoard'.[82])

On eventual publication, *Agapē Agape* had been rewritten as a short monologue in which run-on sentences evoke modernist stream-of-consciousness techniques (there are strong echoes of Beckett's *Krapp's Last Tape*), while also making for extremely difficult reading. Forming a torrential pre-history of digital computation, the text is populated with key figures in an intellectual and medial history of computation, including Norbert Wiener, Joseph Marie Jacquard, Charles Babbage, and Gottfried Wilhelm Leibnitz to make the argument that 'all this computer madness besotted by science besotted by technology by this explosion of progress and the information revolution' has been disastrously applied to the arts.[83] *J R*'s expansive dialogue has been replaced by a short monologue, but this isn't still isn't easy to read. The monologue often interrupts itself to replicate and reflect on the sheer weight and disorder of the research material the narrator is attempting to arrange into his account:

> wait, little card there falling on then, there! You see? Whole stacks of papers here organizing my research here it is, what I was looking for exactly what I'm talking about, 1927, getting the whole chronology in order 1876 to 1929.[84]

[80] William Gaddis, 'A non-fiction book presentation / AGAPE AGAPE / The Secret History of the Player Piano', 1, Folder 412, Box 118, WGP. It is undated but appears to be the same document reprinted as an appendix to Gaddis, *Rush for Second Place* (there are many undated drafts in the folder), which Tabbi dates as from the early 1960s.

[81] William Gaddis, 'A non-fiction book presentation / AGAPE AGAPE / The Secret History of the Player Piano', 2.

[82] William Gaddis to Gregory Comnes (29 September 1987), reprinted in *Letters*, 442–3, 443.

[83] William Gaddis, *Agapē Agape*, Introduction by Sven Birkerts and afterword by Joseph Tabbi (London: Penguin, 2003), 9.

[84] Gaddis, *Agapē Agape*, 6.

This is an environment in which information has escaped the limits of our control. Difficulty takes the form of material unwieldiness, which ironically, is only made possible through a policy of writerly concision (it is Gaddis's shortest fiction by far) and an abandonment of any claims to MIS.

William Gaddis's final piece of fiction, the culmination of a lifelong project, is a curious amalgam wherein the topic is suitable for a technical writer and the style is emphatically not. Back in *J R* he had satirized the non-fiction version of this history as being 'Difficult as I can make it'.[85] Here his decision to rewrite his technical writing project as fiction enables him to escape the bind in which he had been struggling to escape for almost fifty years: how to write a technical account that also acknowledged the broadest impact of mechanization and all that the player piano, ancestor of the computer, had augured. In *Agapē Agape* Gaddis deploys the research produced through his technical writing projects to foster aesthetic difficulty, but it is unsustainable. This is a literary aesthetic that has been produced in response, not to computing technologies per se, but to the weighty writing systems through which it is mediated.

Programming Literature. Rebecca Roach, Oxford University Press. © Rebecca Roach 2025.
DOI: 10.1093/9780198976578.003.0006

[85] Gaddis, *J R*, 244.

Conclusion

Documenting Difficulty

Programming Literature has been concerned with how a group of authors have conceived of writing and reading within a context in which computer technologies have fundamentally transformed those activities, both materially and conceptually. If my attention has hitherto been on the writers, I want to close by attending briefly to the position of the critic.

In the previous chapter I discussed the ways in which Gaddis developed his literary project, one that privileged reader difficulty, in the shadow of technical writing's signature commitment to transparency. That story is itself tellable thanks to the stacks of paper that Gaddis retained as part of his personal archive—85.5 linear feet of paper to be precise. This retention habit is not unique to Gaddis. There is a correlation between many of the writers who have authored fiction in the wake of mainframe automation and the institutional placing of their personal papers: Thomas Pynchon's archives stretch 70 feet long, Kurt Vonnegut's 50, J. M. Coetzee's almost 60 (along with 131 computer discs). This glut consists of bytes as much as reams and can weigh down the critic as much as the author: I have 1,207 photos of material from Gaddis's papers sitting in my cloud account—and that is only relevant computer materials scoured from a short visit. Such an abundance of material can be a boon to scholars; it has produced this book.

Yet I am also wary of this documentary abundance, wary of valorizing the particular difficulty that accompanies such documentary overload. In part this is because, as I noted in the introduction, there is significant unevenness in which collections have been saved, sold, and rendered accessible, and to whom (Autumn Womak has called for more 'undisciplined data'[1]). That unevenness can be suggestive: we see in Coetzee a real commitment to papering one's past across multiple continents; we can see Gaddis's files as evidence of effective record-keeping (the freelancer keen to maintain a

[1] Autumn Womack, *The Matter of Black Living: The Aesthetic Experiment of Racial Data, 1880–1930* (Chicago: University of Chicago Press, 2022).

personal library for business as well as creative writing purposes), or we might see a canny calculation as to future worth. For of course the literary archive market is just that—a market shaped by different institutions, individuals, professional protocols, and types of cultural capital, as Amy Hildreth Chen has so eloquently illuminated.[2] Collection acquisition shapes scholarship; archives mediate scholarship, which in turn can significantly skew what we talk about and value as literary critics. The expansiveness of these archives can produce expansive criticism, just as much as it can facilitate expansive novels such as those Gaddis produced. A turn towards the archive is not simply a turn away from narratives of universal machines.

My main hesitation, however, stems from a trend, identifiable even in Donald Knuth's 'Literate programming' article that began this book, which valorizes documentation as a solution to technological 'difficulty'. In such argumentation, it is the computer that is now opaque and difficult—but with the claims to universality intact. Specifically, with the expansion of networked computing, big datasets, and AI-aided decision-making, technical documentation is increasingly positioned rhetorically as a bulwark against tech industry excesses. Technical documentation, a subset of technical writing, describes the contextual information deemed necessary for the operating system, algorithm, dataset, or computational technology to function. This documentation is considered vital to the operations (and interoperability) of our digital infrastructures, while also existing in tension with the interests of some corporate entities for whom opacity can be a market advantage.[3] As we become ever-more reliant on processual technologies and algorithmic systems based on large language models, machine reading, and big data, which, thanks to their scale and complexity, seem to escape human literacies, documentation has been positioned as a means of ensuring tech accountability and the even ethical deployment of these systems. As one influential group of tech professionals argued: 'While documentation allows for potential accountability, undocumented training data perpetuates harm without recourse. Without documentation, one cannot

[2] Amy Hildreth Chen, *Placing Papers: The American Literary Archives Market* (Amherst: University of Massachusetts Press, 2020).

[3] The importance of documentation is inadvertently indicated in the standardization efforts in the sector. The Internet Standards, published through the Internet Engineering Task Force's 'Request for comments' function, have been crucial for interoperability between machines and networks and the expansion of the Internet. The International Organization for Standardization (ISO) meanwhile has separate standards categories for 'Technical product documentation' (ISO 1101), IT security (35.030), Cloud Computing (35.210), electronic circuits (31.200), and many others.

try to understand training data characteristics in order to mitigate some of these attested issues or even unknown ones.'[4]

This characterization of documentation as having 'the potential to increase transparency and accountability… [and] mitigate unwanted societal biases' has been growing in the industry, against a backdrop of public anxiety around the apparent 'opacity' of algorithmic decision-making (note that we are no longer 'problem solving' or 'conversing' with a computer, we are now in the realm of bureaucratic binary choices, as Coetzee predicted).[5] This characterization has been taken up in our legal frameworks: in the UK where I write this, there is a legislative requirement that organizations 'document each stage of the process behind the design and deployment of an AI decision-support system in order to provide a full explanation for how you made a decision.'[6] Documentation, by this rationale, offers an antidote to the risks associated with computational difficulty—it offers a means by which this technology can be rendered explainable, and interpretable for human readers even as we are promised that AI will deliver efficiency through document summation and bureaucratic rationalization (as if the technical writing is the problem).[7]

[4] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell, 'On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?', *FAccT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (March 2021), 610–23, 615. doi: 10.1145/3442188.3445922. For background on the controversy, see Paul Taylor, 'Stochastic Parrots', *LRB Blog* (12 February 2021), https://www.lrb.co.uk/blog/2021/february/stochastic-parrots. The original article was influential not only for its argument but also its sense of in the face of corporate pushback—one of the authors is listed as 'Shmargaret Shmitchell' in response to Google's refusal to sign off the article. See Google Walkout for Real Change, 'Standing with Dr. Timnit Gebru—#ISupportTimnit #BelieveBlackWomen', *Medium* (4 December 2020), https://googlewalkout.medium.com/standing-with-dr-timnit-gebru-isupporttimnit-believeblackwomen-6dadc300d382.

[5] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford, 'Datasheets for Datasets', *Communications of the ACM*, 64.12 (December 2021), 86–92, 86. doi: 10.1145/3458723. Jenna Burrell makes the point that there are three kinds of opacity involved in these machine learning technologies: (1) opacity as intentional corporate or institutional strategy; (2) opacity from technical illiteracy; and (3) opacity as a technique of the technology ('that stems from the mismatch between mathematical optimization in high-dimensionality characteristic of machine learning and the demands of human-scale reasoning and styles of semantic interpretation' (2)). The first and second find their direct parallels in older debates around technical writing, the third has been taken up in recent discussions around 'explainability'. Jenna Burrell, 'How the Machine "Thinks": Understanding Opacity in Machine Learning Algorithms', *Big Data & Society*, 3.1 (2016), 1–12.

[6] ICO, 'Documentation', *For Organisations: Guide to Data Protection*, www.ico.org.uk, accessed March 2024.

[7] Andrew Prescott sees open documentation and explainability as part of the same project in 'Bias in Big Data, Machine Learning and AI: What Lessons for the Digital Humanities?', *DHQ*, 17.2 (2023), http://www.digitalhumanities.org/dhq/vol/17/2/000689/000689.html.

I am wary of this rationale: not of the efforts to deploy AI ethically, but the idea that documentary abundance is the avenue to transparency. In particular, I worry about how this valorization is often taken up in the humanities and used as the basis for claims around disciplinary expertise and 'the value of close-reading code as an explanatory approach.'[8] Close reading becomes the method by which we might render computational difficulty explainable; and thus computational difficulty becomes the ground on which the case for literary criticism's social value, and expanded remit, is built.[9] It isn't that I think close reading and literary criticism broadly have little to offer today, it is that I think it important that we recognize the degree to which such a rationale, such a methodological commitment, is itself the outcome of particular historical endeavours, which have themselves been the focus of this book, one that might not sustain us in the future.

Michele Elam has made the astute point that such a historical perspective can itself be a critical position:

> Novelists, not to mention contemporary professional historians, usually understand history not as an inexorable teleological march forward in time and progress but as palimpsest, a Mobius strip, an ongoing and dynamic negotiation between pasts and presents. That stands in important contrast to AI's training sets, which are dehistoricized in particular ways.… It may seem counterintuitive to suggest that ahistoricity lies beneath algorithms used to predict a future, but this is, in effect, what occurs.[10]

The computational future tense is not transparent, universal, or ahistorical, as Christine Brooke-Rose noted half a century ago, and I think it important that we don't simply valorize the 'difficulty' that inheres in our discipline or in archives—whether large language models, or personal papers. What,

[8] David M. Berry, 'Tracing "Toxicity" Through Code: Towards a Method of Explainability and Interpretability in Software', *DHQ*, 17.2 (2023), para 5. In his call for a 'Critical Code Studies', Mark Marino talks of 'a growing sense that the code we are not reading is working against our interests'. Mark C. Marino, *Critical Code Studies* (Cambridge, MA: MIT Press, 2020), 3.

[9] In Dennis Tenen, *Plain Text: The Poetics of Computation* (Stanford: Stanford University Press, 2017), Tenen suggests that close reading can and should be deployed to study code. Meanwhile Martin Paul Eve makes the case that close reading can be facilitated by computers—though still implying that the reader is ultimately a human one *Close Reading with Computers: Textual Scholarship, Computational Formalism, and David Mitchell's Cloud Atlas* (Stanford: Stanford University Press, 2019).

[10] Michele Elam, 'Poetry Will Not Optimize; or, What Is Literature to AI?', *American Literature*, 95.2 (June 2023), 281–303, 286. doi: 10.1215/00029831-10575077.

rather, I hope that this book has illustrated is that computing has long been generative, in its particularity and historicity, for the ways that we have understood literature and literary enquiry—and that we can continue to sift and scrutinize these many possibilities.

The writers featured here have conceived of relations between literature and computing very differently from each other and across their own lifetimes. When conceived in the context of an analogy with computing, modernist difficulty, often the focus for questions around literariness for these writers, can become variously synonymous with experimentalism (Christine Brooke-Rose), obscurity (Gaddis), and expertise (Kenner). Similarly, the literary object's aesthetic autonomy can be reimagined as the autonomy of literary enquiry in an era of computationalism (Coetzee), or it becomes something to be discarded in favour of a decolonial interactivity represented by the word processor (Brathwaite). For all of these writers, Donald Knuth included, literary difficulty and documentation become ethically and pedagogically freighted in different ways for what they seemingly promote in a human reader who dwells in a world where machines also read and write.

These writers should remind us that our contemporary debates about computing and literature have a material and intellectual history and that relations between the two can be (and have been) conceived in more ways than narratives around the universal machine imply. For that reason alone, we should embrace the indeterminacy of 'programming literature'.

Programming Literature. Rebecca Roach, Oxford University Press. © Rebecca Roach 2025.
DOI: 10.1093/9780198976578.003.0007

Works Cited

Archival Collections

ACP Archives of Carcanet Press, University of Manchester

CBI Control Data Corporation Records, Charles Babbage Institute, University of Minnesota

CBRP Christine Brooke-Rose Papers, Harry Ransom Center, University of Texas at Austin

GPI Collections of the George Padmore Institute, London

HKP Hugh Kenner Papers, Harry Ransom Center, University of Texas at Austin

JMCP J. M. Coetzee Papers, Harry Ransom Center, University of Texas at Austin

Published Sources

Abbate, Janet, *Recoding Gender: Women's Changing Participation in Computing* (Cambridge, MA: MIT Press, 2012).

Adas, Michael, *Dominance by Design: Technological Imperatives and America's Civilizing Mission* (Cambridge, MA: Belknap Press, 2006).

Adler, 'The Adler ScreenTyper', Sales brochure (n.d.), https://etzoneorg.files.wordpress.com/2020/07/4e598-adler-screen-typer.pdf

Agar, Jon, *The Government Machine: A Revolutionary History of the Computer* (Cambridge, MA: MIT Press, 2003).

Anderson, Elisabeth, 'Alone and Dispossessed', Collective Review, *Sunday Times* (10 August 1986), Fiction 2.

Aneesh, A., *Virtual Migration: The Programming of Globalization* (Durham, NC: Duke University Press, 2006).

Anti-Apartheid Movement, *Computerising Apartheid: ICL in South Africa* (London: Anti-Apartheid Movement, 1978).

Apter, Emily, *Against World Literature: On the Politics of Untranslatability* (London: Verso, 2013).

Attridge, Derek, *J. M. Coetzee and the Ethics of Reading: Literature in the Event* (Chicago: University of Chicago Press, 2004).

Attwell, David, *J. M. Coetzee & the Life of Writing: Face to Face with Time* (Oxford: Oxford University Press, 2015).

Austin, David, 'All Roads Led to Montreal: Black Power, the Caribbean, and the Black Radical Tradition in Canada', *The Journal of African American History*, 92.4 (Autumn, 2007), 516–39.

Barnhisel, Greg, *Cold War Modernists: Art, Literature, and American Cultural Diplomacy* (New York: Columbia University Press, 2015).

Barnhisel, Gregory, and Catherine Turner, eds., *Pressing the Fight: Print, Propaganda and the Cold War* (Amherst: University of Massachusetts Press, 2010).

Bender, Emily M., Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell, 'On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜', *FAccT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (March 2021), 610–23. doi: 10.1145/3442188.3445922

Benjamin, Ruha, *Race after Technology: Abolitionist Tools for the New Jim Code* (London: Polity, 2019).

Bennett, Eric, *Workshops of Empire: Stegner, Engle, and American Creative Writing during the Cold War* (Iowa City: University of Iowa Press, 2015).

Berry, David M., 'Tracing "Toxicity" through Code: Towards a Method of Explainability and Interpretability in Software', *DHQ*, 17.2 (2023), http://www.digitalhumanities.org/dhq/vol/17/2/000706/000706.html

Birch, Sarah, *Christine Brooke-Rose and Contemporary Fiction* (Oxford: Clarendon Press, 1994).

Black, Edwin, *War against the Weak: Eugenics and America's Campaign to Create a Master Race* (New York: Four Walls Eight Windows, 2003).

Bletchley Park Research, 'Women Codebreakers', *Bletchley Park Research* (2014), https://web.archive.org/web/20140202095728/http://www.bletchleyparkresearch.co.uk/research-notes/women-codebreakers/

Bolter, Jay David, and Richard Grusin, *Remediation: Understanding New Media* (Cambridge, MA: MIT Press, 1998).

Brathwaite, Edward, *The Development of Creole Society in Jamaica, 1770–1820* (Oxford: Oxford University Press, 1971).

Brathwaite, Edward Kamau, 'English in the Caribbean: Notes on Nation Language and Poetry', in Leslie A. Fieldler and Houston A. Baker, eds., *English Literature: Opening up the Canon* (Baltimore: Johns Hopkins University Press, 1981), 15–53.

Brathwaite, Edward Kamau, *History of the Voice: The Development of Nation Language in Anglophone Caribbean Poetry* (London & Port of Spain: New Beacon Books, 1984).

Brathwaite, Edward Kamau, *X/Self* (Oxford: Oxford University Press, 1987).

Brathwaite, Kamau, *Barabajan Poems, 1492–1992* (Kingston & New York: Savacou North, 1994).

Brathwaite, Kamau, *ConVERSations with Nathaniel Mackey* (Staten Island: We Press/Minneapolis: Xcp: Cross-Cultural Poetics, 1999).

Brathwaite, Kamau, *DreamStories*, Longman Caribbean Writers Series (Harlow: Longman, 1994).

Brathwaite, Kamau, 'Tpcs in Caribbean Literature: sycoraX Aesthetics', *NYU* (2009), https://web.archive.org/web/20100615010258/http://draper.fas.nyu.edu/docs/IO/10728/Brathwaite.sycoraXaesthetics.doc

Brathwaite, Kamau, *The Zea Mexican Diary: 7 September 1920–7 September 1986* (Madison: University of Wisconsin Press, 1993).

Breckenridge, Keith, *Biometric State: The Global Politics of Identification and Surveillance in South Africa, 1850 to the Present* (Cambridge: Cambridge University Press, 2014).

Breiner, Laurence A., 'How to Behave on Paper: The *Savacou* Debate', *Journal of West Indian Literature*, 6.1 (July 1993), 1–10.

Breiner, Laurence A., *An Introduction to West Indian Poetry* (Cambridge: Cambridge University Press, 1998).

Brock, Jr., André, *Distributed Blackness: African American Cybercultures* (New York: New York University Press, 2020).

Brooke-Rose, Christine, *Amalgamemnon* [1984], paperback edition (Manchester: Carcanet/Dalkey Archive, 2014).

Brooke-Rose, Christine, *The Christine Brooke-Rose Omnibus* (Manchester: Carcanet, 1986).

Brooke-Rose, Christine, 'Dinner with M. Carnéjoux: New Novels', *Observer Weekend Review* (8 March 1964), Books Section, 27.

Brooke-Rose, Christine, 'Dynamic Gradients', *London Magazine* (March 1965), 89–96.

Brooke-Rose, Christine, *Invisible Author: Last Essays* (Columbus: Ohio State University Press, 2002).

Brooke-Rose, Christine, *Life, End of* (Manchester: Carcanet, 2006).
Brooke-Rose, Christine, *Remake* (Manchester: Carcanet, 1996).
Brooke-Rose, Christine, *A Rhetoric of the Unreal: Studies in Narrative and Structure, Especially of the Fantastic* (Cambridge: Cambridge University Press, 1981).
Brooke-Rose, Christine, *Xorandor/Verbivore* (Great Britain & Singapore: Verbivoracious Press, 2014).
Browne, Simone, *Dark Matters: On the Surveillance of Blackness* (Durham, NC: Duke University Press, 2015).
Bureau of Labor Statistics, *Occupational Outlook Handbook 1968–69* (Washington, DC: U.S. Department of Labor, 1969).
Burke, Damien, Email to Rebecca Roach, 'Tsr–2' (12 October 2017).
Burley, Patrick, 'Whiteness, Displacement, and the Postimperial Imaginary in Christine Brooke-Rose's *Out*', *MFS Modern Fiction Studies*, 66.2 (Summer 2020), 371–95.
Burn, Stephen J., 'The Collapse of Everything: William Gaddis and the Encyclopedic Novel', in Joseph Tabbi and Rone Shavers, eds., *Paper Empire: William Gaddis and the World System* (Tuscaloosa: University of Alabama Press, 2007), 46–62.
Burrell, Jenna, 'How the Machine 'Thinks': Understanding Opacity in Machine Learning Algorithms', *Big Data & Society*, 3.1 (2016), 1–12.
Bush, Vannevar, 'As We May Think', *The Atlantic* (July 1945), https://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/
Bush, Vannevar, *Science the Endless Frontier: A Report to the President by Vannevar Bush, Director of the Office of Scientific Research and Development* (Washington, DC: United States Government Printing Office, July 1945).
Callahan, John F., and Marc C. Conner, eds., *The Selected Letters of Ralph Ellison* (New York: Random House, 2019).
Campbell, Carl, 'From Founding Member of the SHP, Professor Carl Campbell, Department of History and Archaeology, University of the West Indies, Mona', *Social History Project* (2003), https://www.mona.uwi.edu/history/shp/aboutus/campbell.htm
Cayley, John, 'The Code Is Not the Text (Unless It Is the Text)', *Electronic Book Review* (10 September 2002), electronicbookreview.com/essay/the-code-is-not-the-text-unless-it-is-the-text.
Ceruzzi, Paul E., *A History of Modern Computing* (2nd edn, Cambridge, MA: MIT Press, 2003).
Chen, Amy Hildreth, *Placing Papers: The American Literary Archives Market* (Amherst: University of Massachusetts Press, 2020).
Chetwynd, Ali, 'Friction Problems: William Gaddis' Corporate Writing and the Stylistic Origins of *J R*', *Orbit: A Journal of American Literature*, 8.1 (2020), Article 4, 1–64. doi: 10.16995/orbit.gaddis.2
Chetwynd, Ali, 'William Gaddis' Education-Writing and His Fiction: A Fuller Archival History', *Orbit: A Journal of American Literature*, 8.1 (2020), Article 1, 1–47. doi: 10.16995/orbit.gaddis.1
Chetwynd, Ali, 'William Gaddis' "Ford Foundation Fiasco" and J R's Elision of the Teacher's-Eye View', *Orbit: A Journal of American Literature*, 8.1 (2020), Article 3, 1–46. doi: 10.16995/orbit.gaddis.3
Chun, Wendy Hui Kyong, *Control and Freedom: Power and Paranoia in the Age of Fiber Optics* (Cambridge, MA: MIT Press, 2006).
Chun, Wendy Hui Kyong, *Programmed Visions: Software and Memory* (Cambridge, MA: MIT Press, 2011).
Chun, Wendy Hui Kyong, *Updating to Remain the Same: Habitual New Media* (Cambridge, MA: MIT Press, 2016).

Coetzee, J. M., 'Achterberg's "Ballade van de gasfitter": The Mystery of I and You', *PMLA*, 92.2 (1977), 285–96.
Coetzee, J. M., 'Computer Poem', *The Lion and the Impala*, 2.1 (1963), 12–13.
Coetzee, J. M., *Doubling the Point: Essays and Interviews*, ed. David Attwell (Cambridge, MA: Harvard University Press, 1992).
Coetzee, J. M., *Dusklands*, paperback edition (London: Vintage, 1998).
Coetzee, J. M., 'The English Fiction of Samuel Beckett: An Essay in Stylistic Analysis', PhD dissertation, University of Texas at Austin, Austin, 1969.
Coetzee, J. M., *Giving Offense: Essays on Censorship* (Chicago: University of Chicago Press, 1996).
Coetzee, J. M., *In the Heart of the Country*, paperback edition (London: Vintage, 2004).
Coetzee, J. M., 'Linguistics and Literature', in Rory Ryan and Susan Van Zyl, eds., *An Introduction to Contemporary Literary Theory* (Johannesburg: A. D. Donker, 1982), 41–52.
Coetzee, J. M., 'Misconception', Letter to the Editor, *Daily Texan* (24 October 1967).
Coetzee, J. M., 'The Novel Today', *Upstream*, 6.1 (1988), 2–5.
Coetzee, J. M., 'On Literary Thinking', *Textual Practice*, 30.7 (2016), 1151–2.
Coetzee, J. M., 'Review of *Nach Allen Regeln Der Kunst* by Wilhelm Fucks', *Style*, 5.1 (1971), 92–4.
Coetzee, J. M., 'Review of *Strange Attractors: Poems of Love and Mathematics*', eds. Sarah Glaz and Joanne Growney. *Notices of the American Mathematical Society*, 56.8 (2009), 944–6.
Coetzee, J. M., 'Samuel Beckett's *Lessness*: An Exercise in Decomposition', *Computers and the Humanities*, 7.4 (March 1973), 195–8.
Coetzee, J. M., 'Surreal Metaphors and Random Processes', *Journal of Literary Semantics*, 8.1 (1979), 22–30.
Coetzee, J. M., *Waiting for the Barbarians* (London: Penguin, 1980).
Coetzee, J. M., 'The Works of Ford Madox Ford with Particular Reference to the Novels', MA thesis, University of Cape Town, Cape Town, 1963.
Coetzee, J. M., *Youth*, paperback edition (London: Vintage, 2003).
Coetzee, J. M., and Joanna Scott, 'Voice and Trajectory: An Interview with J. M. Coetzee', *Salmagundi: A Quarterly of the Humanities and Social Sciences*, 114–15 (Spring–Summer 1997), 82–102.
Computer Science, 'Computing @ Mona, About Us' (22 July 2012), https://www.mona.uwi.edu/compsci/aboutus
Conrad, Thomas, 'Machines That Help Make Apartheid Run', Letter to the Editor, *New York Times* (18 May 1985), 24.
Copeland, Jack, ed., *Colossus: The Secrets of Bletchley Park's Codebreaking Computers* (Oxford: Oxford University Press, 2006).
Copeland, Jack, ed., *The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life plus The Secrets of Enigma* (Oxford: Clarendon Press, 2004).
Cortada, James W., *The Digital Flood: The Diffusion of Information Technology across the United States, Europe and Asia* (New York: Oxford University Press, 2012).
Cull, Nicholas J., *The Cold War and the U.S. Information Agency: American Propaganda and Public Diplomacy* (Cambridge: Cambridge University Press, 2008).
D'aguiar, Fred, Untitled contribution, in Susheila Nasta, '"A Towering Figure": Tribute to Kamau Brathwaite (1930–2020)', *Wasafari* (24 February 2020), https://www.wasafiri.org/article/a-towering-figure-tribute-to-kamau-brathwaite-1930-2020/
Darlington, Joseph, *Christine Brooke-Rose and Post-War Literature* (Basingstoke: Palgrave Macmillan, 2021).

Darlington, Joseph, 'The Composition of Christine Brooke-Rose's *Thru*: An Afterlife of May '68', *Journal of European Studies*, 45.1 (2015), 49–64. doi: 10.1177/0047244114553769

Darlington, Joseph, 'A Non-Euclidean Novel: Christine Brooke-Rose's *Such* and the Space-Age Sixties', *Journal of Modern Literature*, 40.2 (2017), 147–64. doi: 10.2979/jmodelite.40.2.09

Davenport, Guy, 'Hugh Kenner, X-Ray Technician', The Guest Word Column, *The New York Times* (26 March 1972), Section BR, 47, https://www.nytimes.com/1972/03/26/archives/hugh-kenner-xray-technician.html

De Matas, Jarrel, 'Ctrl + Alt/HyperText: Toward a Remediated Caribbean Literary Aesthetic', *Journal of West Indian Literature*, 25.2 (November 2017), 9–27.

Dinnen, Zara, *The Digital Banal: New Media and American Literature and Culture* (New York: Columbia University Press, 2018).

Disch, Thomas M., 'Rock of Phages', Review of *Xorandor*, *New York Times* (3 August 1986), 10.

Edmond, Jacob, *Make It the Same: Poetry in the Age of Global Media* (New York: Columbia University Press, 2019).

Edmond, Jacob, 'World Literature in Stereo: Magnetic Tape and the Media Futures of Global Literary History', *SubStance*, 50.1 (2021), 27–53.

Edwards, Norval, 'The Foundational Generation: From the *Beacon* to *Savacou*', in Michael A. Bucknor and Alison Donnell, eds., *The Routledge Companion to Anglophone Caribbean Literature* (Abingdon: Routledge, 2011).

Edwards, Paul N., *The Closed World: Computers and the Politics of Discourse in Cold War America* (Cambridge MA: MIT Press, 1996).

Edwards, Paul N., and Gabrielle Hecht, 'History and the Technopolitics of Identity: The Case of Apartheid South Africa', *Journal of Southern African Studies*, 36.3 (2010), 619–39.

Elam, Michele, 'Poetry Will Not Optimize; or, What Is Literature to AI?', *American Literature*, 95.2 (June 2023), 281–303. doi: 10.1215/00029831-10575077

'Electronic Computer for UWI Comes', *The Daily Gleaner*, Kingston, Jamaica (30 July 1962), 2.

Emerson, Lori, *Reading Writing Interfaces: From the Digital to the Bookbound* (Minneapolis, University of Minnesota Press, 2014).

Emre, Merve, *Paraliterary: The Making of Bad Readers in Postwar America* (Chicago: University of Chicago Press, 2017).

Ensmenger, Nathan, *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise* (Cambridge, MA: MIT Press, 2010).

Erickson, Paul, Judy L. Klein, Lorraine Daston, Rebecca Lemov, Thomas Sturm, and Michael D. Gordin, *How Reason Almost Lost Its Mind: The Strange Career of Cold War Rationality* (Chicago: University of Chicago Press, 2013).

Eve, Martin Paul, *Close Reading with Computers: Textual Scholarship, Computational Formalism, and David Mitchell's Cloud Atlas* (Stanford: Stanford University Press, 2019).

Ferris, Natalie, *Abstraction in Post-War British Literature 1945–1980* (Oxford: Oxford University Press, 2022).

Ford, Clyde W., *Think Black: A Memoir*, Amistad (New York: HarperCollins 2019).

Franklin, Seb, *Control: Digitality as Cultural Logic* (Cambridge, MA: MIT Press, 2015).

Franzen, Jonathan, 'Mr. Difficult: William Gaddis and the Problem of Hard-to-Read Books', *The New Yorker* (30 September 2002).

Freeman, Carla, *High Tech and High Heels in the Global Economy: Women, Work, and Pink-Collar Identities in the Caribbean* (Durham, NC: Duke University Press, 2000).

Funkhouser, C. T., *Prehistoric Digital Poetry: An Archaeology of Forms, 1959–1995* (Tuscaloosa: University of Alabama Press, 2007).
Gaddis, William, *Agapē Agape*, introduction by Sven Birkerts and afterword by Joseph Tabbi (London: Penguin, 2003).
Gaddis, William, 'Bridge of Books', *Impetus*, iv.1–2 (January–February 1950), 13.
Gaddis, William, *J R* [1975], reprinted edition (Champaign: Dalkey Archive Press, 2013).
Gaddis, William, *The Letters of William Gaddis*, ed. Steven Moore (Champaign: Dalkey Archive Press, 2013).
Gaddis, William, 'Mechanical Brains', *World Review* 36 (22 October 1949), 7–8.
Gaddis, William, 'The Right to Write and the Right to Read', *Courier* (1 May 1950), 4.
Gaddis, William, *The Rush for Second Place and Other Writings*, ed. Joseph Tabbi, paperback edition (London: Grove Atlantic, 2005).
Gaddis, William, 'Stop Player. Joke No. 4', *The Atlantic* (July 1951), 92–3.
Gaddis, William, and Zoltán Abádi-Nagy, 'William Gaddis: The Art of Fiction No. 101', *The Paris Review*, 105 (Winter 1987), https://www.theparisreview.org/interviews/2577/the-art-of-fiction-no-101-william-gaddis
Galloway, Alexander R., *Protocol: How Control Exists after Decentralization* (Cambridge, MA: MIT Press, 2004).
Gary, Brett, *The Nervous Liberals: Propaganda Anxieties from World War I to the Cold War* (New York: Columbia University Press, 1999).
Gaskill, Nicholas, 'The Close and the Concrete: Aesthetic Formalism in Context', *New Literary History*, 47.4 (2016), 505–24.
Gebru, Timnit, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford, 'Datasheets for Datasets', *Communications of the ACM*, 64.12 (December 2021), 86–92. doi: 10.1145/3458723
Geoghegan, Bernard Dionysius, *Code: From Information Theory to French Theory* (Durham, NC: Duke University Press, 2023).
Gikandi, Simon, 'E. K. Brathwaite and the Poetics of the Voice: The Allegory of History in "Rights of Passage"', *Callaloo*, 14.3 (1991), 727–36.
Gikandi, Simon, *Writing in Limbo: Modernism and Caribbean Literature* (Ithaca: Cornell University Press, 1992).
Gitelman, Lisa, *Always Already New: Media, History, and the Data of Culture* (Cambridge, MA: MIT Press, 2006).
Gitelman, Lisa, *Paper Knowledge: Toward a Media History of Documents* (Durham, NC: Duke University Press, 2014).
Goffman, Erving, *Forms of Talk* (Philadelphia: University of Pennsylvania Press, 1981).
Gold, Rochelle, 'Pre-written Business Correspondences and Computer Therapists: William Gaddis's J R, ELIZA, and Literacies in Conflict', *Electronic Book Review* (7 April 2024), https://doi.org/10.7273/ebr-gadcent3-5
Goldstine, Herman H., and John von Neumann, 'Planning and Coding of Problems for an Electronic Computing Instrument', *Report on the Mathematical and Logical aspects of an Electronic Computing Instrument*, Part II, vol. 1–3, Institute for Advanced Study, Princeton (1947).
Goldstone, Andrew, *Fictions of Autonomy: Modernism from Wilde to de Man* (Oxford: Oxford University Press, 2013).
Golumbia, David, *The Cultural Logic of Computation* (Cambridge, MA: Harvard University Press, 2009).
Goodwin, Mark, 'Hugh Kenner's Book for the Z100 Beginner', *Sextant* (January–February 1985), 31+.

Google Walkout for Real Change, 'Standing with Dr. Timnit Gebru—#ISupportTimnit #BelieveBlackWomen', *Medium* (4 December 2020), https://googlewalkout.medium.com/standing-with-dr-timnit-gebru-isupporttimnit-believeblackwomen-6dadc300d382

Gordin, Michael D., *Scientific Babel: How Science Was Done before and after Global English* (Chicago: University of Chicago Press, 2015).

Gould, Jay R., 'Written English in Defense', *Journal of Engineering Education* (March 1942), 523–8.

Graff, Gerald, *Professing Literature: An Institutional History, Twentieth Anniversary Edition* (Chicago, University of Chicago Press, 2007).

Green, Jack, 'Fire the Bastards!' *newspaper*, no. 12 (1962). Available at *WUSTL Digital Gateway Image Collections & Exhibitions*, http://omeka.wustl.edu/omeka/items/show/10212

Grey, Christopher, *Decoding Organization: Bletchley Park, Codebreaking and Organization Studies* (New York: Cambridge University Press, 2012).

Griffin, Stanley H., 'Noises in the Archives: Acknowledging the Present Yet Silenced Presence in Caribbean Archival Memory', in Michael Moss and David Thomas, eds., *Archival Silences: Missing, Lost and, Uncreated Archives* (London: Routledge, 2021), 81–99. doi: 10.4324/9781003003618-5

Guy, Adam, 'Under Suspicion: Christine Brooke-Rose, Intelligence Work, and the Theory Wars', *Modernist Cultures*, 16.4 (2021), 509–28.

Haig, Thomas, 'Inventing Information Systems: The Systems Men and the Computer, 1950–1968', *Business History Review*, 75 (Spring 2001), 15–61.

Harris, Ashleigh, 'The Literary Metadata of African Little and Popular Magazines', Report. Uppsala University (2024), https://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-519564

Hart, Matthew, *Nations of Nothing but Poetry: Modernism, Transnationalism and Synthetic Vernacular Writing* (Oxford University Press, 2010).

Hartman, Charles O., *Virtual Muse: Experiments in Computer Poetry* (Wesleyan University Press/University Press of New England, 1996).

Hartman, Charles O., and Hugh Kenner, eds., *Sentences* (Los Angeles, CA: Sun and Moon Press, 1995).

Hartman, Saidiya, *Wayward Lives, Beautiful Experiments: Intimate Histories of Riotous Black Girls, Troublesome Women and Queer Radicals* (London: Profile Books, 2019).

Hayes, Brian, 'Computer Recreations', *Scientific American* (November 1983), 18.

Hayles, N. Katherine, *How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics* (Chicago: University of Chicago Press, 1999).

Hayles, N. Katherine, *My Mother Was a Computer: Digital Subjects and Literary Texts* (Chicago: University of Chicago Press, 2005).

Hayles, N. Katherine, 'Print Is Flat, Code Is Deep: The Importance of Media-Specific Analysis', *Poetics Today*, 25.1 (Spring 2004), 67–90.

Hayman, David, Keith Cohen, and Christine Brooke-Rose, 'An Interview with Christine Brooke-Rose', *Contemporary Literature*, 17.1 (1976), 1–23.

Heath Company, 'Heath Winter Flyer 1955', http://www.pestingers.net/pdfs/catalogs/heathkit-winter-flyer-1955.pdf

Henry, Lester, 'The Digital Divide, Economic Growth and Potential Poverty Reduction: The Case of the English Speaking Caribbean', Conference paper, https://www.cavehill.uwi.edu/salises/conferences/past-conferences/2003/thedigitaldivide-henry.aspx

Hicks, Mar, *Programmed Inequality: How Britain Discarded Women Technologists and Lost Its Edge in Computing* (Cambridge, MA: MIT Press, 2017).

Higman, B. W., *Slave Population and Economy in Jamaica, 1807–1834* (2nd edn, Mona, Jamaica: University of West Indies Press, 1995).
Hines, Andy, *Outside Literary Studies: Black Criticism and the University* (Chicago: University of Chicago Press, 2022).
Hodges, Andrew, *Alan Turing: The Enigma* (London: Burnett Books, 1983).
Hoffman, Jeanette, 'Writers, Texts, and Writing Acts: Gendered User Images in Word Processing Software', in Donald MacKenzie and Judy Wajcman, eds., *The Social Shaping of Technology* (2nd edn, Buckingham, UK: Open University Press, 1999).
Hope, Francis, 'I, Julian', Collective Book Review, *The New Statesman* 68 (13 November 1964), 741–2.
Hopper, Grace Murray, 'Keynote Address', in Richard L. Wexelblat, ed., *History of Programming Languages* (New York: Academic Press, 1981), 7–24.
Huntington.org, 'News Release—The Huntington Acquires Thomas Pynchon Archive', Huntington.org (22 December 2022), https://huntington.org/news/news-release-huntington-acquires-thomas-pynchon-archive
IBM, *Reference Manual 709/7090 FORTRAN Programming System* (New York: IBM, 1961).
ICO, 'Documentation', *For Organisations: Guide to Data Protection*, https://ico.org.uk/for-organisations/guide-to-data-protection/key-dp-themes/explaining-decisions-made-with-artificial-intelligence/part-3-what-explaining-ai-means-for-your-organisation/documentation/
Jaffe, Aaron, *Modernism and the Culture of Celebrity* (Cambridge: Cambridge University Press, 2005).
James, David, *Modernist Futures: Innovation and Inheritance in the Contemporary Novel* (Cambridge: Cambridge University Press, 2012).
Johnston, Peter, 'J. M. Coetzee's Work in Stylostatistics', *DHQ*, 8.3 (2014), http://www.digitalhumanities.org/dhq/vol/8/3/000188/000188.html
Johnston, Peter, '"Presences of the Infinite": J. M. Coetzee and Mathematics', PhD Dissertation, University of London, London, 2013.
Jordan, Julia, 'Are Puns Mere? Christine Brooke-Rose's *Amalgamemnon*', *Modernist Cultures*, 16.4 (2021), 529–45.
Josephs, Kelly Baker, 'Versions of X/Self: Kamau Brathwaite's Caribbean Discourse', *Anthurium: A Caribbean Studies Journal*, 1.1 (2003), Article 4.
Josephs, Kelly Baker, 'Where Do Journals Go? *Savacou*, Fifty Years Later', *SX Salon* 35 (October 2020), smallaxe.net/sxsalon/discussions/where-do-journals-go-savacou-fifty-years-later.
Kalliney, Peter, 'Modernism, African Literature, and the Cold War', *Modern Language Quarterly*, 76.3 (2015), 333–68.
Kalliney, Peter J., *The Aesthetic Cold War* (Princeton: Princeton University Press, 2022).
Kannemeyer, J. C., *J. M. Coetzee: A Life in Writing*, trans. Michiel Heyns (UK: Scribe, 2012).
Kaypro, 'Kaypro 4', Advertisement, *The Daily Gleaner* (6 March 1985), 11.
Kenner, Hugh, 'Art in a Closed Field', *The Virginia Quarterly Review*, 38.4 (Autumn 1962), 597–613.
Kenner, Hugh, 'Automated Out?', *National Review* (10 October 1959), 389–90.
Kenner, Hugh, 'Calcunuts', *The Wall Street Journal Magazine* (June 1981), 52–60.
Kenner, Hugh, *A Colder Eye: The Modern Irish Writers* (New York: Alfred A. Knopf, 1983).
Kenner, Hugh, 'The Computerized *Ulysses*', *Harper's* (April 1980), 89–95.
Kenner, Hugh, *The Counterfeiters: An Historical Comedy* (Bloomington: Indiana University Press, 1968).

Kenner, Hugh, 'Curtains for Lady Buxley', *Byte* (October 1989), 360–2.
Kenner, Hugh, 'Dead Chickens A-Wavin', *Byte* (January 1992), 404.
Kenner, Hugh, 'Does the Walking Do the Fingers?', *Byte* (May 1990), 377–8.
Kenner, Hugh, 'Electronic Books', *Byte* (November 1993), 404.
Kenner, Hugh, *The Elsewhere Community* (New York: Oxford University Press, 2000).
Kenner, Hugh, *Heath/Zenith Z-100 User's Guide* (Bowie, MD: Brady Communications Company, Inc., 1984).
Kenner, Hugh, 'Machinespeak', in Leonard Michaels and Christopher Ricks, eds., *The State of the language* (Berkeley: University of California Press, 1980).
Kenner, Hugh, 'The Making of the Modernist Canon', *Chicago Review*, 34.2 (Spring, 1984), 49–61.
Kenner, Hugh, *The Mechanic Muse* (New York: Oxford University Press, 1987).
Kenner, Hugh, 'More Double-Precision Trig Functions', *Sextant* (Winter 1984), 98.
Kenner, Hugh, 'My Interface with IBM', *National Review* (22 November 1974), 1360–1.
Kenner, Hugh, 'Of Minds and Men', *Byte* (August 1990), 354–5.
Kenner, Hugh, 'Poets and Sleepwalkers', *Byte* (April 1991), 392–3.
Kenner, Hugh, '*Racter*', *Byte* (May 1986), 289–91.
Kenner, Hugh, 'Saving Steps', *PPC Journal*, 5.5 (June 1978), 9–11.
Kenner, Hugh, 'The Scholar's Friend', *The Times Literary Supplement* (30 April 1993), 10.
Kenner, Hugh, *The Stoic Comedians: Flaubert, Joyce and Beckett* (London: W. H. Allen, 1964).
Kenner, Hugh, 'What You Do When You Program', *The Wall Street Journal Magazine* (June 1981), 60.
Kenner, Hugh, and Joseph O'Rourke, 'A Travesty Generator for Micros', *Byte* (November 1984), 129+.
Ketzan, Erik, *Thomas Pynchon and the Digital Humanities: Computational Approaches to Style* (London, Bloomsbury, 2021).
Kindley, Evan, 'Kenner's Networks', *Critical Inquiry*, 47.3 (2021), 525–43.
Kindley, Evan, *Poet–Critics and the Administration of Culture* (Cambridge, MA: Harvard University Press, 2017).
Kinsley, William, 'Untitled Review of *The Counterfeiters*', *Eighteenth-Century Studies*, 4.4 (Summer 1971), 494–6.
Kirschenbaum, Matthew G., *Mechanisms: New Media and the Forensic Imagination* (Cambridge, MA: MIT Press, 2007).
Kirschenbaum, Matthew G., *Track Changes: A Literary History of Word Processing* (Cambridge, MA: Harvard University Press, 2016).
Knight, Richard, *U.S. Computers in South Africa* (New York: The Africa Fund, 1986).
Knuth, Donald E., 'Literate Programming', *The Computer Journal*, 27.2 (1984), 97–111. doi: 10.1093/comjnl/27.2.97
Knuth, Donald, and Luis Trabb Pardo, 'The Early Development of Programming Languages', in N. Metropolis, J. Howlett, and Gian-Carlo Rota, eds., *A History of Computing in the Twentieth Century* (Cambridge, MA: Academic Press, 1980), 197–273.
Kynell, Teresa C., *Writing in a Milieu of Utility: The Move to Technical Communication in American Engineering Programs 1850–1950* (Norwood, NJ: Ablex Pub. Co., 1996).
Lancashire, Ian, and Willard McCarty, eds., *The Humanities Computing Yearbook 1988* (Oxford: Clarendon Press, 1989).
Landy, Barry, 'Atlas 2 at Cambridge Mathematical Laboratory (And Aldermaston and CAD Centre)', *Memories of the Ferranti Atlas Computer*, School of Computer Science, University of Manchester (December 2012), http://elearn.cs.man.ac.uk/~atlas/
Landy, Barry, Email to Rebecca Roach, 'Re: Coetzee...Again!' (1 May 2017).

Laughlin, James, and Hugh Kenner, 'The Mixpoem Program', *The Paris Review*, 94 (Winter 1984), 193–203.

Laurel, Brenda, *Computers as Theatre* (Menlo Park, CA: Addison-Wesley Publishing, 1991).

Lavington, Simon, 'The Atlas Story', *Memories of the Ferranti Atlas Computer*, School of Computer Science, University of Manchester (December 2012), http://elearn.cs.man.ac.uk/~atlas/

Lavington, Simon, 'Ferranti Atlas 2 Computers at AWRE Aldermaston and at the CAD Centre', *Memories of the Ferranti Atlas Computer*, School of Computer Science, University of Manchester (December 2012), http://elearn.cs.man.ac.uk/~atlas/

Lawrence, Karen R., *Techniques for Living: Fiction and Theory in the Work of Christine Brooke-Rose* (Columbus: Ohio State University Press, 2010).

Lazarus, Neil, 'Modernism and Modernity: T. W. Adorno and Contemporary White South African Literature', *Cultural Critique*, 5 (Winter 1986–87), 131–55.

Leick, Karen, *Gertrude Stein and the Making of an American Celebrity* (New York: Routledge, 2009).

Lennon, Brian, *Passwords: Philology, Security, Authentication* (Cambridge MA: Harvard University Press, 2018).

Leonard, John, 'Last Word: Beckett Safe from Computers', *New York Times Review of Books* (19 August 1973), 27.

Levine, Caroline, *Forms: Whole, Rhythm, Hierarchy, Network* (Princeton: Princeton University Press, 2015).

Levine, Caroline, 'The Great Unwritten: World Literature and the Effacement of Orality', *Modern Language Quarterly*, 74.2 (June 2013), 217–37.

Lewis, Ronald, *Ultra Goes to War: The Secret Story* (London: Hutchinson & Co., 1978).

Liu, Lydia H., *The Freudian Robot: Digital Media and the Future of the Unconscious* (Chicago: University of Chicago Press, 2011).

Longo, Bernadette, *Spurious Coin: A History of Science, Management, and Technical Writing* (New York: State University of New York Press, 2000).

Love, Heather A., *Cybernetic Aesthetics: Modernist Networks of Information and Data* (Cambridge: Cambridge University Press, 2024).

Love, Heather A., 'Cybernetic Modernism and the Feedback Loop: Ezra Pound's Poetics of Transmission', *Modernism/modernity*, 23.1 (January 2016), 89–111. doi: 10.1353/mod.2016.0020

Ludlum, David A., 'Offshore Data Entry Pays Off', *ComputerWorld* (9 June 1986), 103+.

Mailland, Julien, and Kevin Driscoll, *Minitel: Welcome to the Internet* (Cambridge, MA: MIT Press, 2017).

Majumdar, Saikat, and Aarthi Vadde, eds., *The Critic as Amateur* (London: Bloomsbury Academic, 2020).

Maney, Kevin, Steve Hamm, and Jeffrey M. O'Brien, *Making the World Work Better: The Ideas that Shaped a Century and a Company* (New York: IBM Press, 2011).

Marino, Mark C., *Critical Code Studies* (Cambridge, MA: MIT Press, 2020).

Marques, Ivan da Costa, 'History of Computing in Latin America', *IEEE Annals of the History of Computing*, 37.4 (October–December 2015), 10–12. doi: 10.1109/MAHC.2015.78

McElroy, Joseph, 'Gaddis Dialogue Questioned', in Joseph Tabbi and Rone Shavers, eds., *Paper Empire: William Gaddis and the World System* (Tuscaloosa: University of Alabama Press, 2007), 63–71.

McGurl, Mark, *The Program Era: Postwar Fiction and the Rise of Creative Writing* (Cambridge, MA: Harvard University Press, 2009).

McHale, Brian, *Constructing Postmodernism* (London: Routledge, 1992).
McIlwain, Charlton D., *Black Software: The Internet and Racial Justice, from the AfroNet to Black Lives Matter* (New York: Oxford University Press, 2020).
McPherson, Tara, 'Why Are the Digital Humanities So White? or Thinking the Histories of Race and Computation', in Matthew K. Gold, ed., *Debates in the Digital Humanities* (Minneapolis, University of Minnesota Press, 2012), 139–60. dhdebates.gc.cuny.edu/debates/text/29.
McWilliams, Peter, 'Scholar Brings a Literary Touch to Computer Guide', *The Morning Star* (6 March 1985), Businessxtra2, 11G.
Medina, Eden, *Cybernetic Revolutionaries: Technology and Politics in Allende's Chile* (Cambridge, MA: MIT Press, 2011).
Michaels, Walter Benn, *The Shape of the Signifier: 1967 to the End of History* (Oxford: Oxford University Press, 2004).
Moody, Alys, *The Art of Hunger: Aesthetic Autonomy and the Afterlives of Modernism* (Oxford: Oxford University Press, 2018).
Moore, Steven, *William Gaddis: Expanded Edition* (London: Bloomsbury, 2015).
Morton, Brian, 'A Glimpse into the Future Tense', Review of *Amalgamemnon*, *Times Higher Education Supplement* (5 November 1984), 10.
Mukherjee, Ankhi, 'Creole Modernism', *Affirmations: Of the Modern*, 2.1 (Winter 2014), 25–45.
Muth, Katie, 'The Grammars of the System: Thomas Pynchon at Boeing', *Textual Practice*, 33.3 (2019), 473–93. doi: 10.1080/0950236X.2019.1580514
Myers, David Gershom, *The Elephants Teach: Creative Writing since 1880* [1996] (Chicago: University of Chicago Press, 2006).
Nakamura, Lisa, 'Indigenous Circuits: Navajo Women and the Racialization of Early Electronic Manufacture', *American Quarterly*, 66.4 (December 2014), 919–41.
NARMIC/American Friends Service Committee, *Automating Apartheid: U.S. Computer Exports to South Africa and the Arms Embargo* (NARMIC/American Friends Service Committee, 1982).
Needham, Roger, Letter to the Editor, *Times Literary Supplement* (8 June 1962), 429.
Nerlekar, Anjali, and Francesca Orsini, 'Introduction: Postcolonial Archives', *South Asia: Journal of South Asian Studies*, 45.2 (2022), 211–19. doi: 10.1080/00856401.2022.2040803
Nicita, Mike, and Rob Petrusha, 'Micro Reviews', *Popular Computing* (February 1985), 176.
Noble, Safiya Umoja, *Algorithms of Oppression: How Search Engines Reinforce Racism* (New York: New York University Press, 2018).
Nofre, David, Mark Priestley, and Gerard Alberts, 'When Tech Became Language: The Origins of the Linguistic Conception of Computer Programming, 1950–1960', *Technology & Culture*, 55.1 (January 2014), 40–75.
Noh, Jeff, 'Ralph Ellison's Computer Memory', *Contemporary Literature*, 62.4 (Winter 2021), 527–57.
Nolan, Carrie, 'Remediation and Diaspora: Kamau Brathwaite's Video-Style', in Carrie Nolan and Barrett Watten, eds., *Diasporic Avant-Gardes: Experimental Poetics and Cultural Displacement* (New York: Palgrave Macmillan, 2009), 77–97.
Nyhan, Julianne, 'Gender, Knowledge and Hierarchy: On Busa's Female Punch Card Operators', *Arche Logos: Notes on the Digital Humanities and Oral History* (3 May 2014), https://archelogos.hypotheses.org/135
Paquet, Sandra Pouchet, foreword in *The Zea Mexican Diary: 7 September 1920–7 September 1986* by Kamau Brathwaite (Madison: University of Wisconsin Press, 1993).

PEN America, '[Virtual] 90: A Celebration of Kamau Brathwaite', *PEN America* (11 May 2020), https://pen.org/event/a-celebration-of-kamau-brathwaite/

Perloff, Marjorie, 'Reading Modernism, after Hugh Kenner (1923–2003): Hugh Kenner and the Invention of Modernism', *Modernism/modernity*, 12.3 (September 2005), 465–9.

Peters, Benjamin, *How Not to Network a Nation: The Uneasy History of the Soviet Internet* (Cambridge, MA: MIT Press, 2016).

Peters, John Durham, *Speaking into the Air: A History of the Idea of Communication* (Chicago: University of Chicago Press, 1999).

Petrick, Elizabeth, 'Imagining the Personal Computer: Conceptualizations of the Homebrew Computer Club 1975–1977', *IEEE Annals of the History of Computing*, 39.4 (2017), 27–39.

'PM Gets Computers to Help with Memory Bank Project', *Daily Gleaner*, Kingston, Jamaica (25 September 1984), 15.

Pohl, Rebecca, 'Selling Difficulty: The Case of Christine Brooke-Rose', *Textual Practice*, 32.2 (2018), 283–99. doi: 10.1080/0950236X.2018.1413046

Pollock, Charles W., *New World Modernisms: T. S. Eliot, Derek Walcott, and Kamau Brathwaite* (Charlottesville: University of Virginia Press, 2004).

Posel, Deborah, 'The Apartheid Project, 1948–1970', in Robert Ross et al., *The Cambridge History of South Africa*, vol. 2 (Cambridge: Cambridge University Press, 2011), 319–68.

Pournelle, Jerry, 'Burnouts, Bargains and Two Sleek Portables', *Byte* (January 1983), 418+.

Pournelle, Jerry, 'Eagles, Text Editors, New Compilers, and Much More', *Byte* (September 1983), 307.

Pratt, Terrence W., *Programming Languages: Design and Implementation* (2nd edn, Englewood Cliffs, NJ: Prentice-Hall Inc, 1984).

Prescott, Andrew, 'Bias in Big Data, Machine Learning and AI: What Lessons for the Digital Humanities?', *DHQ*, 17.2 (2023), http://www.digitalhumanities.org/dhq/vol/17/2/000689/000689.html

Prescott, Andrew, and Alison Wiggins, eds., *Archives: Power, Truth, and Fiction* (Oxford: Oxford University Press, 2023).

Pressman, Jessica, *Digital Modernism: Making It New in New Media* (New York: Oxford University Press, 2014).

Pyle, Ian, 'Memories of Atlas Fortran', *Memories of the Ferranti Atlas Computer*, School of Computer Science, University of Manchester (December 2012), http://elearn.cs.man.ac.uk/~atlas/

Rajski, Brian, 'Writing "Systems": Richard Yates, Remington Rand, and the Univac', *Contemporary Literature*, 54.3 (Fall 2013), 550–77.

Raley, Rita, 'Machine Translation and Global English', *The Yale Journal of Criticism*, 16.2 (2003), 291–313.

Raley, Rita, *Tactical Media* (Minneapolis: University of Minnesota Press, 2009).

Rankin, Joy Lisi, *A People's Histories of Computing in the United States* (Cambridge, MA: Harvard University Press, 2018).

Reiss, Timothy J., 'Kamau Brathwaite, a Memoir: The Road to and from New York City', *Small Axe*, 26.1 (2022), 111–23. doi: 10.1215/07990537-9724121

Rippeon, Andrew, 'Bebop, Broadcast, Podcast, Audioglyph: Scanning Kamau Brathwaite's Mediated Sounds', *Contemporary Literature*, 55.2 (2014), 369–401.

Roach, Rebecca, 'Cross-Cultural Communication Theory: Basic English and Machine Translation at the Rockefeller Foundation', Rockefeller Archive Center Research Report (19 December 2019), https://rockarch.issuelab.org/resource/cross-cultural-communication-theory-basic-english-and-machine-translation-at-the-rockefeller-foundation.html

Roach, Rebecca, 'Expanding the History of Humanities Computing: Hugh Kenner's Conversational Criticism', *Digital Scholarship in the Humanities*, 38.2 (June 2023), 679–95. doi: 10.1093/llc/fqac066

Roach, Rebecca, 'Generative Transcription: The Interview in Post-World War 2 Anglophone African Literary Culture', *Journal of Postcolonial Writing*, 60.4 (2024), 491–505. doi: 10.1080/17449855.2024.2339395

Roach, Rebecca, 'Hero and Bad Motherland: J. M. Coetzee's Computational Critique', *Contemporary Literature*, 59.1 (Spring 2018), 80–111.

Roach, Rebecca, *Literature and the Rise of the Interview* (Oxford: Oxford University Press, 2018).

Roach, Rebecca, 'Talk Interfaces', in Max Saunders and Lisa Gee, eds., *Ego Media* (Stanford: Stanford University Press Digital, 2023), www.egomedia.org.

Rohlehr, Gordon, 'Dream Journeys', *World Literature Today*, 68.4 (Autumn 1994), 765–74.

Rohlehr, Gordon, 'A Literary Friendship: Selected Notes on the Correspondence with Kamau Brathwaite', *Small Axe*, 26.1 (2022), 124–44. doi: 10.1215/07990537-9724135

Sale, Roger, 'The Art of Hugh Kenner', Review of Hugh Kenner, *The Counterfeiters*, *The Hudson Review*, 22.3 (Autumn 1969), 521–7.

Sammet, Jean E., *Programming Languages: History and Fundamentals* (Upper Saddle River, NJ: Prentice Hall, 1969).

Sammet, Jean E., 'The Use of English as a Programming Language', *Communications of the ACM*, 9.3 (March 1966), 228–30.

Schryer, Stephen, 'Conservative Circuits: Hugh Kenner, Modernism, and *National Review*', *Modernism/modernity*, 26.3 (2019), 505–20. doi: 10.1353/mod.2019.0040

Shannon, Claude, 'Prediction and Entropy in Printed English', *The Bell System Technical Journal* 30 (January 1951), 50–64.

Shannon, Claude E., and Warren Weaver, *The Mathematical Theory of Communication* [1949] (Urbana: University of Illinois Press, 1963).

Sharma, Dinesh C., *The Outsourcer: The Story of India's IT Revolution* (Cambridge: MA: MIT Press, 2015).

Siraganian, Lisa, *Modernism's Other Work: The Art Object's Political Life* (Oxford: Oxford University Press, 2012).

Sokolov, Raymond, 'Computerese Spoken Here', Review of *Xorandor*, *Wall Street Journal* (19 August 1986).

Sterne, Jonathan, *The Audible Past: Cultural Origins of Sound Reproduction* (Durham, NC: Duke University Press, 2003).

Stewart, Victoria, *The Second World War in Contemporary British Fiction: Secret Histories* (Edinburgh: Edinburgh University Press, 2011).

Strachey, Christopher, 'The "Thinking" Machine', *Encounter* (October 1954), 25+.

Svensson, Patrik, 'Humanities Computing as Digital Humanities', *Digital Humanities Quarterly*, 3.3 (2009).

Tabbi, Joseph, *Postmodern Sublime: Technology and American Writing from Mailer to Cyberpunk* (Ithaca: Cornell University Press, 1995).

Tabbi, Joseph, 'The Technology of Quotation: William Gaddis's 'J R' and Contemporary Media', *Mosaic* 28.4 (1995), 143–64, 43.

Tabbi, Joseph and Rone Shavers, eds., *Paper Empire: William Gaddis and the World System* (Tuscaloosa: University of Alabama Press, 2007).

Taylor, Paul, 'Stochastic Parrots', *LRB Blog* (12 February 2021), https://www.lrb.co.uk/blog/2021/february/stochastic-parrots

Tenen, Dennis, Plain Text: The Poetics of Computation (Stanford: Stanford University Press, 2017).

Trettien, Whitney, *Cut/Copy/Paste: Fragments from the History of Bookwork* (Minneapolis: University of Minnesota Press, 2021).

Trettien, Whitney, 'Substrate, Platform, Interface, Format', *Textual Cultures*, 16.1 (2023), 286–312. doi: 10.14434/tc.v16i1.36107

Turing, A. M., 'Computing Machinery and Intelligence', *Mind*, 59.236 (1950), 433–60.

Turing, A. M., 'On Computable Numbers, with an Application to the Entscheidungsproblem', *Proceedings of the London Mathematical Society*, s2–42 (1937), 230–65. doi: 10.1112/plms/s2-42.1.230

Turner, Fred, *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism* (Chicago: University of Chicago Press, 2006).

Ullman, Ellen, *Close to the Machine: Technophilia and Its Discontents* (San Francisco: City Lights Books, 1997).

Vechinski, Matthew James, 'Christine Brooke-Rose and the Science of Fiction: Rethinking Conceptual Writing and Artificial Intelligence in *Xorandor*', *Critique: Studies in Contemporary Fiction*, 57.4 (2016), 448–63. doi: 10.1080/00111619.2015.1091288

Vincent, Chris, Untitled Review of *Xorandor* and *The Christine Brooke-Rose Omnibus*, *Women's Review*, 12 (October 1986).

von Hilgers, Philipp, 'The History of the Black Box: The Clash of a Thing and Its Concept', *Cultural Politics*, 7.1 (2011), 41–58.

von Neumann, John, 'First Draft of a Report on the EDVAC' (30 June 1945), Reprinted in *IEEE Annals of the History of Computing*, 15.4 (1993), 27–75.

Walmsley, Anne, *The Caribbean Artists Movement 1966–1972: A Literary & Cultural History* (London: New Beacon, 1992).

Weaver, Warren, 'Translation', [1949] Memo, Reprinted in *Machine Translation of Languages: Fourteen Essays*, eds. William N. Locke and A. Donald Booth (Cambridge, MA: MIT Press, 1955), 15–24.

Weizenbaum, Joseph, *Computer Power and Human Reason: From Judgement to Calculation* (W. H. Freeman, 1976).

Womack, Autumn, *The Matter of Black Living: The Aesthetic Experiment of Racial Data, 1880–1930* (Chicago, University of Chicago Press, 2022).

WReC (Warwick Research Collective), *Combined and Uneven Development: Towards a New Theory of World-Literature* (Liverpool: Liverpool University Press, 2015).

Yates, JoAnne, *Control through Communication: The Rise of System in American Management* (Baltimore: Johns Hopkins University Press, 1989).

Yates. JoAnne, 'The Emergence of the Memo as a Managerial Genre', *Management Communication Quarterly*, 2.4 (1989), 485–510.

Zimbler, Jarad, 'For Neither Love Nor Money: The Place of Political Art in Pierre Bourdieu's Literary Field', *Textual Practice*, 23.4 (2009), 599–620.

Zimbler, Jarad, *J. M. Coetzee and the Politics of Style* (Cambridge: Cambridge University Press, 2014).

Index

For the benefit of digital users, indexed terms that span two pages (e.g., 52–53) may, on occasion, appear on only one of those pages.

Accountability 184–5
Adler (typewriter) 116–17, 121–2
Aesthetic automatism 19, 24, 53
Aesthetic autonomy 16, 19, 162–3, 176–7, 187
Aldermaston 32–4
Alington, Jean 100–1, 101 n.20
Amateurism *see* Hobbyism
Anti-Apartheid campaigns 48–50
Apartheid 19, *see also* South Africa
Apple (company)
  MacBasic (programming language) 145
  Macintosh (computer line) 145–6
  MacSE (computer) 145
  MacWrite (program) 145
Archives 13–15, 69, 131–2, 144, 151–2, 183–4
Artificial Intelligence 4–5, 184–5
The Atlantic 156, 159
Atari ST (computer) 121–4
  ST Writer (program) 121–2
Atlas (supercomputer project) 30–4 *see also* Titan
  Autocode (programming language) 31–2
  Supervisor (operating system) 30–3, 40
Auto-Beatnik (program) 66–7
Automatization 23–4, 58–9, *see also* Linguistics
Atomic Weapons Research Establishment 32–4, *see also* Aldermaston

Backus, John 104
Barbados 127–8, 130
Bard College 177–8
Barth, John 173, 178
BASIC (programming language) 70–1, 73, 142–5
Beckett, Samuel 11, 25, 64, 91–2, 181
  Watt 41–3, 66–7
Binary logic 169–70
Binary thinking 19, 65–6, 185, *see also* Computationalism
Bletchley Park 20–1, 93–4, 96–102
Bloomsbury group 10–11, *see also* Modernism
Book History 7–9, *see also* Methodology
Brathwaite, Doris 141, 143–6
Brathwaite, Kamau 21–2, 126–55, 187
  Ancestors 151
  SVS 126–7, 130, 146–51
  Barabajan Poems 151, 155
  DreamStories 131–2, 144–6
  Nation language 129–30, 147, 153–5
  Conversations with Nathaniel Mackey 151, 153–5
  History of the Voice 129–30, 151–3
  X/Self 130, 141, 147–51
  Zea Mexican Diary 142–3
Brooke-Rose, Christine 4, 11–12, 16–17, 20–1, 93–125, 186–7
  Amalgamemnon 113–16, 120–2
  Between 111–12
  Out 108–9
  Poccum 43 (invented programming language) 119–20
  Remake 96–102, 105, 125
  Xorandor 113–14, 116–22
  Such 109–11
  Textermination 113–14, 124–5
  Thru 112–13
  Verbivore 113–14, 121–3
Brookes, Cleanth 64
Burroughs Corporation 17–18
Burroughs, William 17–18
Bush, Vannevar 159–60
Byte (magazine) 20, 81, 85–6, 89. *See also* Kenner, Hugh *Print Queue*

Caliban 149
Campaign for Nuclear Disarmament 33–4
Canada 16–17, 64, 135
Carcanet Press 113–14, 122, 124–5
Caribbean Arts Movement, 128
Caribbean Basin Initiative 134
Caribbean 16–17
  Computing industry 135–7
  Caribbean Digital 21, 132, 132 n.21
  Literature 126, 128–9, 129 n.12
Close reading 7, 16, 16 n.36, 65, 186
  see also Reading
COBOL (programming language) 14–15
Codework 117–19
Codex 9–10, 21–2
Coetzee, J. M. 4, 11–12, 16–17, 19, 23–63, 65–6, 183, 185, 187
  'Computer Poem' 23–4, 36–7
  Doubling the Point 58–9
  Dusklands 27–8, 46–7, 52–3
  'The English Fiction of Samel Beckett' 41–3
  Giving Offense 59–60
  'Hero and Bad Mother in Epic' 51
  'On Literary Thinking' 63
  'The Novel Today' 57–8
  'Surreal Metaphors and Random Processes' 51
  'The Works of Ford Madox Ford' 34–6
  Youth 23–4, 29, 31, 60–3
  In the Heart of the Country 27–8, 53–4, 56–7
Cold War 4–5, 26, 40, 163–4, 176–7
Colonialism 47 n.76, 126–8, 133–4
Colossus (computer) 93–4, 101–2
Command line interface 72 *see also* CP/M
Communication Theory 159–60
Computational thinking *see* computationalism
Computationalism 43–7, 52–3, 187
CP/M (operating system) 142–6
Creative writing 178–9
Creolization 126–8, 135, 138, 147
  Creolized computing 154–5
Cryptanalysis 93–4, 97–101, 103–4

Data processing 136–7, 140
Dialogue 97–100, 107–11, 118, 129–30, 146, 148, 172
  Questions 166–71
  Subconversation 108 n.51
Difficult Criticism 180
Difficulty
Difficulty 11–13, 68–9, 78–9, 88–9, 95, 97, 147, 171–4, 182–7
  as experimentalism 105–13
  and SVS 153–4
Digital divide 134
Digital humanities *see* Humanities computing
Disciplinarity 4–5, 65–6, 176–7, 186 *see also* Two culture
DIY computing 70–1, 76
Documentation 2–4, 22, 68 n.21, 71–2, 183–7
  see also Technical Writing

Eagle (computer) 144–6
EDVAC (computer) 164–5
Eliot, T. S. 64, 81, 91, 95
ELIZA (program) *see* Weizenbaum, Joseph
Ellison, Ralph 15 n.33, 83 n.60
Encounter (magazine) 10–11
Expertise 65–6, 72, 76, 88, 187
Explainability 185–6

Ford Foundation 158
Ford, Ford Madox 34–6
FORTRAN (programming language) 28–9, 31–2, 67–8, 73, 104, 138
Franzen, Jonathan 173, 180
Future tense 114–15, 146

Gaddis, William 11–12, 16–17, 156–83, 187
  Agapē Agape 156–7, 180–2
  Carpenter's Gothic 156, 180–1
  The Company You Keep 166–9
  A Frolic of His Own 156, 180–1
  J R 156, 171–4
  'Literature and Crisis' 175–6
  The Recognitions 156
  The Rush for Second Place 177–8
  Software 169–71
'Stop Player. Joke No. 4' 156–7
Gender 64 n.2, 140, 143, 149 *see also* Women
Global English 17, 17 n.38
Gold Coast (Ghana) 16–17, 127–8
Goldstine, Herman 102–3

Hartman, Charles O. 85–6
Harvard University 145
Heathkits 74–9
Higman, Barry 137–8

Historicity 186–7
History 5–7, 14–15, 17–18 *see also Archives*
  of computing 6 n.12, 26–8, 132
  as a discipline 128–9, 132
Hobbyism 19–20, 64–6, 69–75, 77–8, 88–9
Homebrew Computer Club 70
Hopper, Grace 104, 165–6
HP-65 *see* Programmable calculator
Human-Computer Interaction 3–4, 17–18
Humanities 162, 186
Humanities computing 19, 69, 89

Information theory 10–11, 25, 95
Interactivity 72
International Business Machines (IBM) 15, 17, 19, 26–30, 66–7, 75–6, 123–4, 133–4, 158, 165–6, 168–9
  1401 (computer) 27, 36
  1620 (computer) 137–8
  7094 (computer) 27
  and computer poetry 37–8
  and Jamaica 133–4
  and South Africa 48–50
  THINK 166, 175–6
International Computers and Tabulators (ICT) 30–4, 38–40, 49–50

Joyce, James 10–11, 25, 64, 66–7, 78, 80–1, 91, 95

Kaypro (computer) 142–3
Kenner, Hugh 19–20, 64–92, 187
  'Art in a Closed Field' 66–7
  'Beginners Corner' 71
  'Calcunuts' 69–71
  The Counterfeiters 67–9, 73–4
  Geodesic Math 68–9
  Heath/Zenith 74, 76–9
  The Mechanic Muse 90–2
  Mixpoem 83–6
  Print Queue 86–90
  The Pound Era 64, 70–1
  Sentences 85–6
  The Stoic Comedians 67, 73–4
  Travesty (program) 79–82, 84–6, 89
Knuth, Donald 1–4, 32, 73–4, 157, 184–5, 187

Large Language Models 184–5
Laughlin, James 83–6
Le Nouveau Roman 104, 106–7
Leavis, F. R. 127–8
Linguistics 40–4 *see also* narratology
Literary criticism 20, 64–5, 73–4, 76–8, 82, 90–2, 183–6
Literary studies (discipline) 2, 16, 21, 64–5, 162–3
Literate programming *see* Knuth, Donald

Machine translation 57 *see also* Weaver, Warren
Mainframes 165–6
Marketing 166–7
Massachusetts Institute of Technology 32
Materiality 7–8, 95–7, 187
Mathematics 23–4, 26, 34
McLuhan, Marshall 25, 35–6, 64 *see also* Media Studies
Media Studies 35–6, 64, 67 *see also* McLuhan, Marshall
Memory Bank Project 142–3, 151–2
Methodology 5, 7–9, 15
  and Archives 13–15
Minitel (computer network) 20–1, 114
MIS (computer system) 167–9
Modernism (literary) 10–12, 16, 25–6, 55–6, 65–7, 78, 81–2, 92, 95, 126–7, 176–7

Narratology 113–16, 119–22
National Endowment for the Arts 178–9
The National Review 65 n.5, 75–6, 89–90
New Criticism 25, 64, 162–3, 176–7
New York University 126, 130
The New Yorker 158

O'Rourke, Joseph 79–82
Offshoring 138–40
Opacity 184–5
Orality 127–30, 146, 151–3 *see also* Memory Bank Project

The Paris Review, 83–4
Pascal (programming language) 83–4, 91–2
PC Revolution 19, 64–5, 69–70
Pedagogy 65–6
Personal computer (PC) 21, 69–70, 78–9, 106, 121–4, 130–1, 144–5
Player piano 156–8 *see also* Gaddis, William
  Agapē Agape
Popular Computing 77–8
Pound, Ezra 10–11, 64, 66–7, 74, 84–5, 91
PPC Journal 70–1
Printed English 95 *see also* Shannon, Claude

Processual computing 21–2, 138, 147
Programmable calculator 5, 69–72
Programming language history 104–5
Pynchon, Thomas 15, 178–9, 183

Racter (program) 86, 89
Reading 54–6, 183
  and the humanities 172
  and difficulty 55–6
  human 1–2, 22, 65, 68–9, 72–3, 88–9, 97, 120, 153–4, 184–5, 187
  machine 1, 62–3, 73, 187
Richards, I. A. 25 n.9, 95

Savacou 128–9, 141–3
Sextant 70–1, 77–8
Shannon, Claude 10–11, 25, 81–2,
South Africa 19, 46, 48–51
  computer industry 48–51
Stein, Gertrude 170
Strachey, Christopher 10–11
Structuralism 106, 112–13
Style 77, 82, 87
Swift, Jonathan 67–9
SycoraX aesthetics 21, 126, 130 *see also* Brathwaite, Kamau
Sycorax (character) 149

Technical Writing 74–9, 156–9 *see also* documentation
  Definition 160–1
  History of 161–3
  Style 161–3
Telematics 114, 122–3
Time-sharing 32
Titan (computer) 30–4, 38–40 *see also* Atlas
Tower of Babel 20, 31–2, 95, 103, 105, 111–13, 123, 125
Translation 102–4
Transparency 12–13, 12 n.24, 121–2, 168, 185–7
Turing Test 67–9, 153–4, 173–4
Turing, Alan 4–6, 20–1, 26, 93
Two cultures 77, 93–4, 162, 175–6
Typewriter 114, 168

UNESCO 111, 158, 164–5
Universal (Turing) machine 5–6, 67–9, 102–4, 159–60, 183–4, 187
Universal language 95
Universalism 149–50, 186
University 16, 94, 96–7, 126
University of Cambridge 30, 127–8
University of Cape Town 26, 36, 43–4, 50–1
University of Texas at Austin 34–5, 40–3, 45
University of the West Indies, Mona 128–9, 137, 147
  History department 137–8
  Social History Project 138
Updike, John 173

Von Neumann, John 102–3
Vonnegut, Kurt 173, 179–80, 183

Weaver, Warren 10–11, 25, 103–4, 159–60
Weizenbaum, Joseph 46–8
Women 94–5, 105–6 *see also* Gender
WordPerfect (program) 123–4
Word Plus (program) 123–4
Word processing 12–13, 21, 74, 123–4, 126–7, 130–1, 146–8, 187
World Wide Web 17–18

Yates, Richard 179