Building a Multilingual Internet: The View from South Asia

By

Anushah Hossain

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Energy and Resources

in the

Graduate Division

of the

University of California, Berkeley

Summer 2022

Building a Multilingual Internet: The View from South Asia

Abstract

Building a Multilingual Internet: The View from South Asia

by

Anushah Hossain

Doctor of Philosophy in Energy and Resources

University of California, Berkeley

Professor Paul Duguid, Co-Chair
Professor Isha Ray, Co-Chair

In this dissertation, I argue that the digitization of the Bangla letter, khanda ta, tracks the transformation of the Western-oriented early consumer Internet into an open, global infrastructure that better met the needs of international, multilingual users longing for digital representation and equal affordances for communication. The story takes place primarily between 2000 and 2005, in South Asia and Silicon Valley, but ventures out in time and space to contextualize the events at hand. I follow the trajectories of standards-makers, software hobbyists, government officials, major software companies, and linguists in turn, showing what stakes and interests they held during this period, and how they shape the debate over khanda ta. The issue sits of khanda ta sits at the crux of several dichotomies that achieve a certain synthesis through its resolution: the gap between new open-channel internet governance institutions and traditional, closed-participation forms of governance; the opposition between the new entrants from the Global South and the mostly-Western actors who had designed the early Internet; and the tension between socially-meaningful language conventions and machine-readable technical standards.

Khanda ta's eventual inclusion, or "encoding", in the Unicode Standard represents the victory of the open-governance model of new industry consortia such as the Unicode Consortium, in which older authorities such as government ministries and international treaty organizations must fit themselves. It also represents a recognition of the values and expertise of the Global South, embodied by the South Asian experts in this debate, amongst those of the Western technical elite designing critical digital infrastructures. And it highlights the role of intermediaries who must accumulate both technical and linguistic expertise to build technical tools and standards for language. In the end, my goal is to show the multitudinous threads that came together to build a multilingual internet.

For Dada, Nanu, and Nanabhai.

Table of Contents

List of Figures

Acknowledgements

"When did you decide you were going to do your dissertation on Bangla?!" my mother had asked me in bewilderment after listening to me give a practice quals talk. This project was as much a surprise to me as it was to her. I grew up speaking Bangla but knew little about my birth country until a few years ago. But the exposure I had through my family was enough to seed the questions in my mind that became too fascinating to ignore and would eventually bloom into this dissertation. Equally influential was my time spent on the internet growing up -- I'm grateful that the hours I invested participating in virtual communities have finally found a productive use.

I first want to thank the institutions that generously funded this research: the Wood-Calvert Engineering Funds, the Chowdhury Center for Bangladesh Studies, the Berkeley Center for New Media's Lyman Fellowship, and the Ford and Sloan Foundations' program on Critical Digital Infrastructure. I am especially grateful to the Ford and Sloan's program managers, Michael Brennan and Josh Greenberg, who gave early encouragement to my research agenda and introduced me to the wonderful cohort of scholars who in turn helped shape this work.

Thank you to my community at ERG that continually validated my choice to study the internet while everyone was studying slightly more world-historical events like climate change. The genuine interest and support I felt from ERG helped me stay excited and passionate about this project. I want to thank my cohort, Water Group, the finer things listserv, Fort Row, and ERG Online, my micro-communities in this program that made graduate school so much fun and made my projects sharper. Thank you to my wonderful committee members, who took a chance on me and this project when they had little to go on. Thank you to Isha Ray for affirming I always had a place at ERG. Thank you to Cori Hayden for always excellent comments and for designing the seminar that influenced much of my thinking in this dissertation. Thank you, Cathy Koshland, for sharing your time, mentorship, and cross-displinary expertise with me. I am so grateful to have gotten the chance to work with you throughout this degree. And thank you to Paul Duguid, with whom I've gotten to spend many cherished afternoons discussing internet histories and treatises over tea in South Hall. Thank you for supporting this journey and being the perfect guide through it.

Beyond my department and committee, I had a wonderful network of mentors who encouraged me, advised me, and sharpened this work. Thank you to Ashwin Mathew, who told me I needed to apply for that first grant that launched this work. Thank you to Debbie Anderson, who helped me build relationships with the key actors that feature in this work. Thank you to Ken Whistler for being a generous and reflective storyteller, and for being such a diligent archivist — this dissertation could not exist without your careful curation. Thank you to Keith Murphy for being my Unicode partner-in-crime — one of the few people who knew exactly what I was talking about if I started going off about "dotted circles." Thank you to Isabelle Zaugg for generously sharing your work and helping build this nascent community of Unicode scholars. Thank you to Abhijeet Paul for teaching me to read Bangla and helping me understand what this project could mean to the Bengali community. Thank you to Matthew Berry, who took the time to delve into my project and helped me draw in scholarship on print cultures. Thank you to Manish Goregaokar for patiently answering my repeated questions about characters and glyphs and all of the other intricacies of Unicode.

My utmost gratitude to all of the keepers of the archives, mailing lists, and personal websites that constituted the corpus for this dissertation. A special thanks to the Internet Archive for preserving so much of the early Web — the stories featured here would be lost without the work of this important institution.

Thank you to each of my interviewees. I feel honoured to have the opportunity to tell your stories and share your work. I hope I have done it justice.

For the inspiring conversations and endless emotional support, I want to thank Arvin Javadi, Tejomay Gadgil, and Esther Shears. Thank you to Katherine Young and Annie Ferguson, who read over everything I send them, no matter the hour, and whose friendships mean the world to me. Thank you to Jonathan Breit, for listening to me talk about khanda ta incessantly over the last two years. Thank you for all of your love. Thank you to my parents, Naila Hussain and Asif Hossain, for being my biggest cheerleaders. Thank you for always asking to read drafts, for never letting me forget how much you love me, and for giving me all the things in life that brought me to this moment. Thank you to my best friend and sister, Sariah Hossain. All my love, always.

I can hardly believe this project and my time at Berkeley have come to an end. For someone who moved every two years growing up, it was an incredible and novel experience to stay in one city and one school for so long. Now two undergraduate degrees, a masters, and a PhD later, I am finally getting ready to leave. I owe as much thanks to the cafe workers that fueled this dissertation as I do my friends and family; thank you to everyone at Yali's on Oxford where I began every morning, Brewed Awakening where I spent my afternoons, and Philz on Shattuck where I'd stay until close in the evenings. These were the places where my friends knew to look for me, and they are the places I'll always picture in my mind when I think of this project.

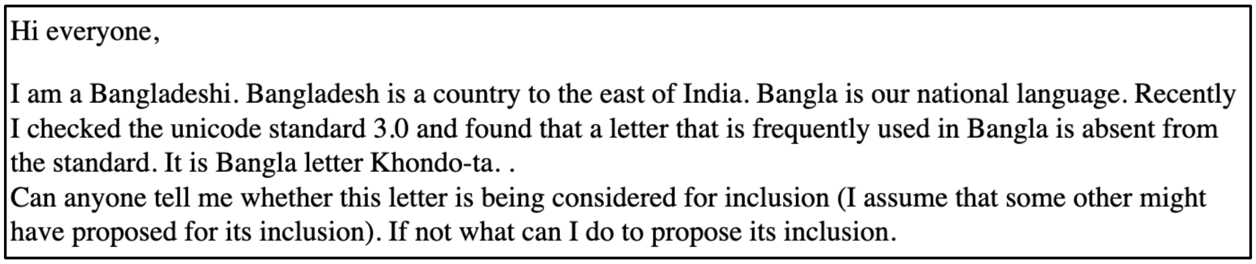
In many ways I found myself by accident in the perfect place to complete this research. Berkeley is the first university in the country to have a Bangladesh Studies center, which was started during my graduate program. It is the only university member of the Unicode Consortium, and the Script Encoding Initiative directed here by Dr. Debbie Anderson is responsible for getting hundreds of minority scripts encoded in Unicode. It is the home of the BSD open source operating system, and of course, the birthplace of the Free Speech Movement. The history of this university is not lost on me and I feel honoured to have been able to spend my undergraduate and graduate years here.

Introduction

"For a language community, a script isn't just a bunch of characters; it is a symbol of national and cultural identity, a political and emotional issue, a domain where 'angels fear to tread.'"[1]

Gautam Sengupta, a Bengali linguist, was typing with passion and frustration to the Unicode mailing list, where hundreds of messages were being passed back and forth over the digital representation of one Bangla letter. It was the letter *khanda ta*, a little used, but nonetheless essential, letter of the Bangla alphabet. *Khanda ta* was not appearing correctly on computers, and the Unicode Consortium, the nonprofit organization responsible for setting the rules of digital text, had not been doing enough to fix the issue – or so Sengupta felt.

Bengalis had been raising the issue of *khanda ta* for four years by this point, beginning with a humble message posted to the same listserv in the year 2000.[2]



Hi everyone,

I am a Bangladeshi. Bangladesh is a country to the east of India. Bangla is our national language. Recently I checked the unicode standard 3.0 and found that a letter that is frequently used in Bangla is absent from the standard. It is Bangla letter Khondo-ta. .
Can anyone tell me whether this letter is being considered for inclusion (I assume that some other might have proposed for its inclusion). If not what can I do to propose its inclusion.

There was a clear contrast in the tone of Sengupta's message and the initial post. The Bangladeshi newcomer had felt it necessary to explain where his country was located and what language was spoken there – striking disclaimers given that he was writing to the people who had already included it in their Standard. But most of the members of the Unicode Consortium were based in the West. The Consortium was founded by some of the most prominent North American software companies in 1991; its initial members included representatives from Xerox, Apple, IBM, and Microsoft. For those appealing from the Global South to this esteemed list of companies, it was easy to feel intimidated and out-of-place.

But by 2004, when Sengupta was writing, the world had already changed in significant ways. The potential of the Internet as a democratizing force — a way to cheaply access information and communicate with anyone around the world — was readily recognized. Though the mobile phone revolution had yet to transform access to the Internet in relatively-disconnected regions like South Asia, a vanguard of South Asian technologists had glimpsed a future marked by digital technologies and felt an urgency to make them available and accessible to their home populations. And in the context of the Unicode Consortium, there had been enough interactions

[1] Sengupta, Gautam. "[indic] Re: [Fwd: Re: All Bengali behaviours (not only khanda ta)]" Email, February 3, 2004.

[2] Rahman, Md Ziaur. "Bangla(Bengali) letter Missing." Email, July 27, 2000.

between the Bangla user community and the Western technologists that the issue of *khanda ta* had evolved from a polite request to a resolute demand for a solution.

In this dissertation, I argue that the deceptively small case of *khanda ta*'s encoding in the Unicode Standard tracks the transformation of the Western-oriented early consumer Internet into an open, global infrastructure that better met the needs of international, multilingual users longing for digital representation and equal affordances for communication. The issue sits of *khanda ta* sits at the crux of several dichotomies that achieve a certain synthesis through its resolution: the gap between new open-channel internet governance institutions and traditional, closed-participation forms of governance; the opposition between the new entrants from the Global South and the mostly-Western actors who had designed the early Internet; and the tension between socially-meaningful language conventions and machine-readable technical standards. As one Microsoft employee responded to Sengupta's declaration above, "I had not expected the extent to which the emotional issues would extend to how a script is encoded in bits and bytes."[3]

Khanda ta's eventual inclusion, or "encoding", in the Unicode Standard represents the victory of the open-governance model of new industry consortia such as the Unicode Consortium, in which older authorities such as government ministries and international treaty organizations must fit themselves. It also represents a recognition of the values and expertise of the Global South, embodied by the South Asian experts in this debate, amongst those of the Western technical elite designing critical digital infrastructures. And it highlights the role of intermediaries who must accumulate both technical and linguistic expertise to build technical tools and standards for language.

Over the course of this dissertation, I chronicle how *khanda ta* transforms from being an "absent" letter from the Unicode Standard, to a software bug, to an emblem of the ignorance of Western technocrats. The story takes place primarily between 2000 and 2005, in South Asia and Silicon Valley, but ventures out in time and space to contextualize the events at hand. I follow the trajectories of standards-makers, software hobbyists, government officials, major software companies, and linguists in turn, showing what stakes and interests they held during this period, and how they shape the debate over *khanda ta*. In the end, my goal is to show the multitudinous threads that came together to build a multilingual internet.

Internet histories

I argue that the period of the early 2000s was a critical moment of articulation, when the design of the digital environment that now supports much of our social lives was being determined. Personal computers were still the primary mode of accessing the internet. Internet Explorer was the most popular browser, followed by Netscape Navigator and Mozilla Firefox. Blogging was popular. Wikipedia was new and inconsistently reliable. There was still a feeling of potential, that

[3] Constable, Peter. "[indic] Re: [Fwd: Re: All Bengali behaviours (not only khanda ta)]" Email, February 3, 2004.

the Internet would democratize access to information, enable unprecedented connectivity, and cut out intermediaries.

Amidst all of the talk of these opportunities, there was simultaneous awareness that internet access was not spreading evenly around the world. There was a "digital divide," in which the poor, remote, and technologically-illiterate were being left out from reaping the benefits provided by the internet. The digital divide encompassed access to infrastructure and digital devices, but also the availability of local-language content. The New York Times published an article in 1996 with the headline: "Computer Speak; World, Wide, Web: 3 English Words." It said the Web was written mostly in English, creating a high barrier to entry for non-English speakers. How had this happened? It claimed:

> *The Internet started in the United States, and the computer hackers whose reality has always been virtual are almost all American. By the time the net spread, its linguistic patterns -- like its principal architecture and best software -- were all Made in the U.S.A.*[4]

Another popular piece published in the American Prospect in 2001 made a similar argument:

> *The Internet was basically an American development, and it naturally spread most rapidly among the other countries of the English-speaking world. Right now, for example, there are roughly as many Internet users in Australia as in either France or Italy, and the English-speaking world as a whole accounts for over 80 percent of top-level Internet hosts and generates close to 80 percent of Internet traffic. It isn't surprising, then, that the Web is dominated by English…*
>
> *But the tendency to use English doesn't disappear even when a lot of speakers of the local language have Internet access. Since the Web turns every document into a potentially "international" publication, there's often an incentive for publishing Web sites in English that wouldn't exist with print documents that don't ordinarily circulate outside national borders. And this in turn has made the use of English on the Web a status symbol in many nations, since it implies that you have something to say that might merit international attention.*[5]

Both of these were market-based explanations for why English dominated online. The user base spread from one English-speaking country to another; users seeking an audience wanted to appeal to the largest English-speaking one. But at the same time, there were issues in the supply. It was not yet easy to type in the writing systems of other languages. The digital infrastructure – a "stack" of technical tools including encoding standards, font formats, keyboards, user interfaces and more – that would eventually enable multilingual digital communication, were still being assembled. This stack was a work-in-progress until the mid-2000s, when, I argue, it suddenly became easier to open a Word file or webpage and type or read text that wasn't in English.

---

[4] Michael Specter, "Computer Speak;World, Wide, Web: 3 English Words," *The New York Times*, April 14, 1996, sec. Week in Review, https://www.nytimes.com/1996/04/14/weekinreview/computer-speak-world-wide-web-3-english-words.html.

[5] Geoffrey Nunberg, "Will the Internet Always Speak English?," The American Prospect, December 19, 2001, https://prospect.org/api/content/3e35e7bd-ce0d-57fe-bdc0-8327087966a9/.

Many of the gaps this dissertation fills were first articulated by computer historian, Michael Mahoney in "The History of Computing in the History of Technology".[6] As Mahoney wrote,

> *We speak of the computer industry as if it were a monolith rather than a network of interdependent industries with separate interests and concerns…What is truly revolutionary about the computer will become clear only when computing acquires a proper history, one that ties it to other technologies and thus uncovers the precedents that make its innovations significant.*

At his time of writing, the history of computing tended towards: corporate histories/"insider histories," first-hand and expert accounts, journalistic accounts, and "social impact" analyses. These works tended to be myopic, focused on the spectacular and unusual, or were polemical – better suited to eventually becoming primary sources.

The recent history of digital text covered in this dissertation, from about 1984 to 2004, is still in this nascent stage. As Haigh et al. have written, the dearth of scholarly attention following a historical approach may be related to the recency of the events at hand: "The ever-unfolding history of the Internet therefore risks falling into a kind of disciplinary no-man's-land—too old to be of interest to policy scholars or sociologists, but too recent and far too unstable for most historians to feel comfortable working there."[7] This period is just beginning to enter into historical view, as illustrated through new publications on the BBS networks of the 1990s, for example.[8]

The Unicode Standard, the base layer of the "stack" of technologies I explore in my dissertation, has already received attention from technology historians and Science and Technology Studies scholars. In his article, John emphasized that the design of the multilingual internet is based on historical contingencies and political-economic factors: in the case of Hebrew, John argues that it was Microsoft's adoption of one framework of representing Hebrew ("logical Hebrew") over another ("visual Hebrew") that led to its universal adoption. Similarly, he writes that Unicode should not be viewed as the teleological victor in the world of standards, but rather the dominant standard "because of the alliance of U.S. firms supporting it."[9] Unicode has been resisted in countries such as Korea in part for this very reason — seen as an emblem of Western culture. Initial antagonism towards the Unicode Standard led to the adoption of a 'dual standard' on Korean computers in the late 1990s, including both the existing Korean code and the new Unicode Standard. Dongoh Park interprets this reaction as "a product of a complex web of social

[6] Michael S. Mahoney, "The History of Computing in the History of Technology," *Annals of the History of Computing* 10, no. 2 (April 1988): 113–25, https://doi.org/10.1109/MAHC.1988.10011.

[7] Thomas Haigh, Andrew L. Russell, and William H. Dutton, "Histories of the Internet: Introducing a Special Issue of Information & Culture," *Information & Culture: A Journal of History* 50, no. 2 (2015): 143–59, https://doi.org/10.1353/lac.2015.0006.

[8] Kevin Driscoll, The Modem World: A Prehistory of Social Media (New Haven: Yale University Press, 2022).

[9] Nicholas A. John, "The Construction of the Multilingual Internet: Unicode, Hebrew, and Globalization," *Journal of Computer-Mediated Communication* 18, no. 3 (April 1, 2013): 321–38, https://doi.org/10.1111/jcc4.12015.

phenomena regarding globalization in the early 1990s in Korea...While some saw Western culture and knowledge as a modernizing force, there was also a growing anxiety that Western culture would eclipse Korean identity."[10] Other scholars such as Isabelle Zaugg have further interrogated the dissonance between language communities and Unicode standards-makers. Unicode's inclusion of Ethiopic, a major script of both Ethiopia and Eritrea, was delayed because of clashing interpretations of how the script was to be encoded -- in a "decomposed" form splitting glyphs into constitute consonant and vowel parts, or in forms more easily understood by the user community. Zaugg writes, "it was as if someone had tried to shatter what they perceived to be the indivisible atoms of their script… This offense..represented a common misunderstanding among digitally-disadvantaged language users about how Unicode works."[11]

These themes are highlighted in this work: the historical contingencies behind Unicode's widespread adoption; the gap between the perceptions of user communities and engineering experts; the clash of a "universal" standard against a national one. There are no works to date, however, that situate the Unicode Standard within the ecosystem of technologies and standards for computing and digital communication more broadly. In part, due to the choice of case study – the Bangla script – this dissertation indexes not only the Unicode Standard, but also the standards and software produced by specific technology companies, such as Microsoft and Apple, and the parallel technologies produced by hobbyists, all of which grapple with the Standard. As a result, we are able to see how the Unicode Standard is situated amongst these technologies, but also how it is forced to respond to them. Much like Mahoney wrote, what is truly innovative about the Unicode Standard only becomes clear when it is tied to other technologies and its precedents are uncovered. I therefore take care to note the intersections of the Unicode Standard and the stack of tools built atop it with histories that have hitherto received little scholarly attention, such as histories of text encoding and typography. Through my analysis of Microsoft alongside Unicode, I also contribute to a new field of histories of large international technology companies that situated them within local cultural and political contexts – a veer away from the "corporate" and "insider" histories that are more commonly found in the literature.[12]

The Bangla case

Khanda ta is the letter at the center of this study, and Bangla is the language and writing system that contains it. Bangla, often anglicized as "Bengali," is both a language and a script. The language is spoken primarily in the modern-day nation-states of Bangladesh and India. It is spoken in several Indian states, but it is the majority language of West Bengal – which is on the eastern side of India, but the western side of the Bengal delta. At the time of writing, there are

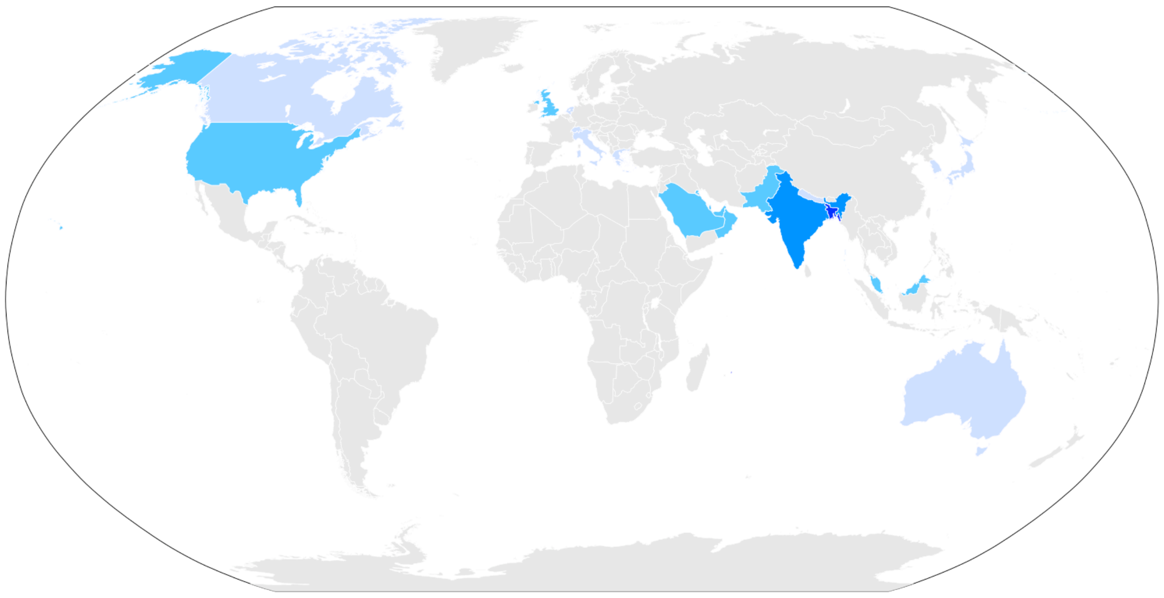
[10] Dongoh Park, "The Korean Character Code: A National Controversy, 1987–1995," *IEEE Annals of the History of Computing* 38, no. 2 (2016): 40–53, https://doi.org/10.1353/ahc.2016.0021.

[11] Isabelle Zaugg, "Digitizing Ethiopic: Coding for Linguistic Continuity in the Face of Digital Extinction" (PhD diss., American University, 2017.

[12] Others in this category include Colette Perold, "IBM's World Citizens: Valentim Bouças and the Politics of IT Expansion in Authoritarian Brazil," *IEEE Annals of the History of Computing* 42, no. 3 (July 2020): 38–52, https://doi.org/10.1109/MAHC.2020.3010892. and John 2013.

approximately 100 million first- or second-language Bangla speakers in Bangladesh, 85 million in India, and 25 million more across the diaspora – many of whom are settled in the United Kingdom, United States, and Middle East.[13] The story of Bangla digitization should not be considered a niche case study of the periphery; it is the seventh-most spoken language in the world, and its digitization affects millions. The Indic script family, for which this case study of Bangla is directly relevant, is used by nearly two billion people in the world, magnifying this story's significance.[14]

Figure 1. Bangla Speakers Worldwide



The Bangla language is typically written with the Bangla script. But the Bangla script is also used for other languages such as Assamese and Manipuri (though orthographic reform in recent decades has led to the replacement of the Bangla script with the traditional Meetei Mayek script for the Manipuri language).

There are differences in how the Bangla language is spoken across India and Bangladesh, and even within each country. Colloquialisms vary across the national borders, as do the spellings of some words. Within Bangladesh, there are also hundreds of different Bangla dialects, some of which are mutually unintelligible to each other. These differences are consequential when it comes to software translation, where different versions must exist for Indian Bangla and Bangladeshi Bangla, for example. An analogous system for English speakers may be the difference between American English and British English.

[13] "Bengali Language | Britannica," accessed June 28, 2022, https://www.britannica.com/topic/Bengali-language.

[14] Peter T. Daniels, "Indic Scripts: History, Typology, Study," in *Handbook of Literacy in Akshara Orthography*, ed. R. Malatesha Joshi and Catherine McBride, Literacy Studies (Cham: Springer International Publishing, 2019), 11–42, https://doi.org/10.1007/978-3-030-05977-4_2.

This dissertation focuses on script digitization for the Bangla language. In other words, I focus on the Bangla script, as used for the Bangla language. There are complex politics of how the Bangla script is used for languages such as Assamese, or whether it is even appropriate to call the script “Bangla” instead of “Assamese” or “Bangla-Assamese.” The Assamese alphabet contains several unique letters that are not used in Bangla, which have been the subject of their own controversies and discussions with Unicode.[15] These are not the subject of the present study, however, and I use the shorthand of the “Bangla script” as opposed to “Bangla-Assamese” for the limited purposes at hand.

The Bangla script

The Bangla script is derived from the Ancient Brahmi script, one of two historic scripts from the Indian subcontinent. The family of scripts descended from the Ancient Brahmi script are often known as “Indic scripts,” and are used throughout South and South East Asia.

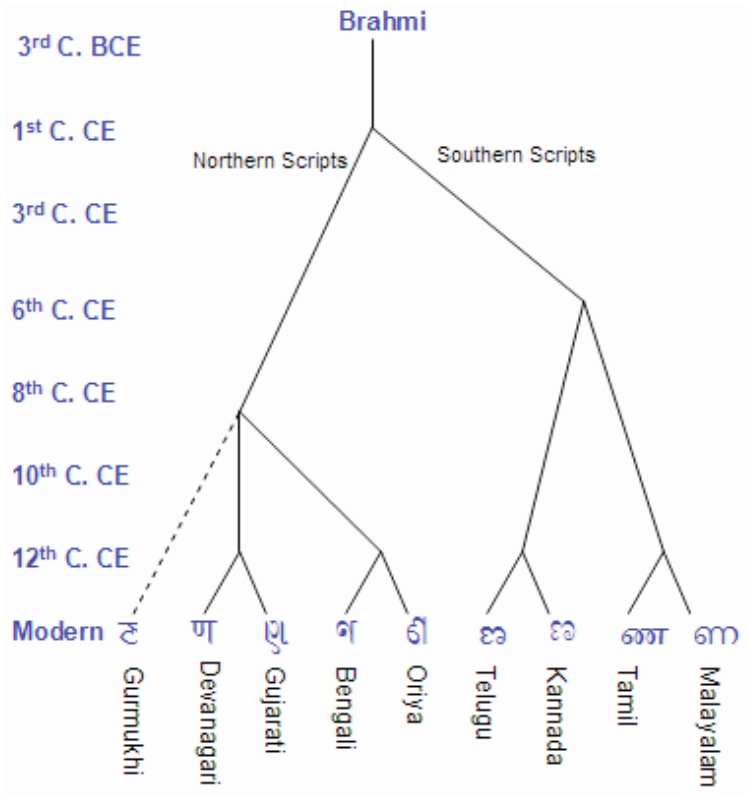
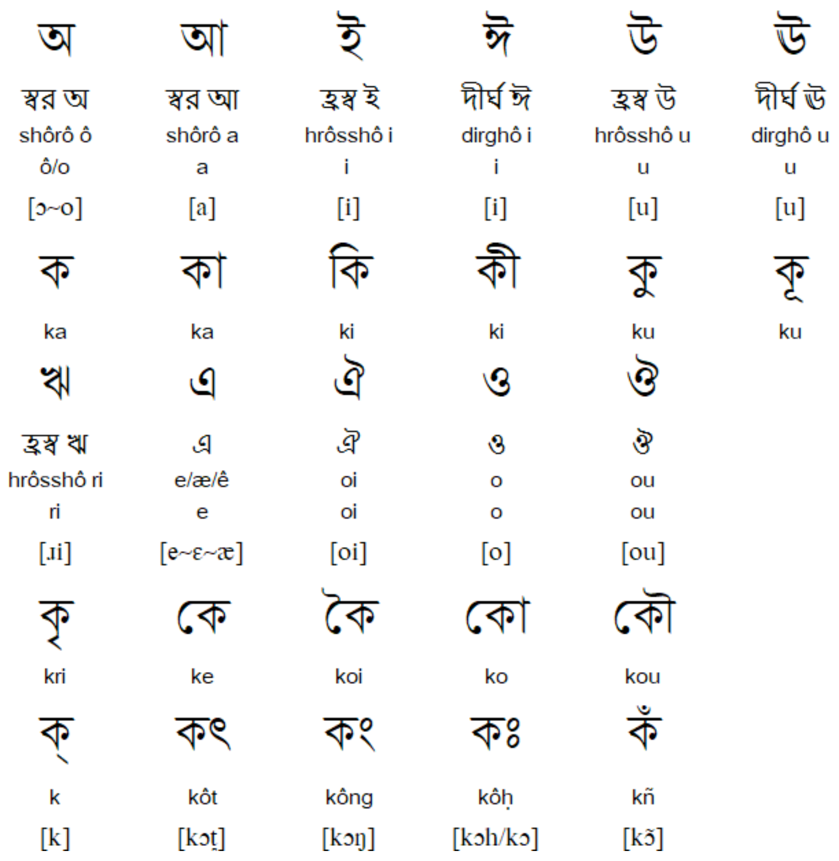


Figure 2. Evolution of Indic Scripts (Richard Ishida — Creative Commons)

[15] “Why Assamese Script Wants Its Own Slot, and What It Has Got Instead,” *The Indian Express* (blog), June 28, 2018, https://indianexpress.com/article/explained/why-assamese-script-wants-its-own-slot-and-what-it-has-got-instead-5236249/.

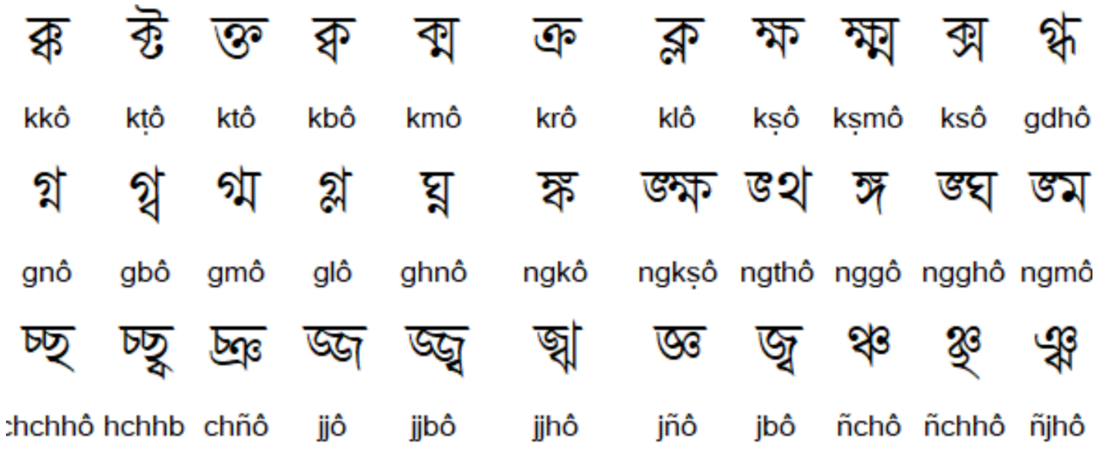
ক	খ	গ	ঘ	ঙ	চ
kô	khô	gô	ghô	ngô	chô/sô
[kɔ]	[k^{h}ɔ]	[ɡɔ]	[ɡ^{ɦ}ɔ]	[ŋɔ]	[tʃɔ~tsɔ~sɔ]
ছ	জ	ঝ	ঞ	ট	ঠ
chhô/ssô	ǰô	ǰhô	ñô	ṭô	ṭhô
[tʃ^{h}ɔ~ts^{h}ɔ~sɔ]	[dʒɔ~dzɔ~zɔ]	[dʒ^{ɦ}ɔ~dz^{ɦ}ɔ]	[nɔ~ẽɔ]	[ʈɔ]	[ʈ^{h}ɔ]
ড	ঢ	ণ	ত	থ	দ
ḍô	ḍhô	nô	tô	thô	dô
[ɖɔ]	[ɖ^{ɦ}ɔ]	[nɔ~ɳɔ]	[t̪ɔ]	[t̪^{h}ɔ]	[d̪ɔ]
ধ	ন	প	ফ	ব	ভ
dhô	nô	pô	phô	bô	bhô
[d̪^{ɦ}ɔ]	[nɔ]	[pɔ]	[p^{h}ɔ~ɸɔ~fɔ]	[bɔ]	[b^{ɦ}ɔ~βɔ]
ম	য	র	ল	শ	ষ
mô	jô	rô	lô	shô/sô	shô/sô
[mɔ]	[dʒɔ~dzɔ~zɔ]	[ɹɔ]	[lɔ]	[ʃɔ~ɕɔ~sɔ]	[ɕɔ~ʃɔ]
স	হ	ড়	ঢ়	য়	
shô/sô	hô	rô	rhô	yô	
[sɔ~ɕɔ~ʃɔ]	[ɦɔ~hɔ]	[ɽɔ]	[ɽ^{ɦ}ɔ]	[e̯~∅]	

Figure 3. Bangla Consonants (Source: Omniglot.Com)



অ	আ	ই	ঈ	উ	ঊ
স্বর অ	স্বর আ	হ্রস্ব ই	দীর্ঘ ঈ	হ্রস্ব উ	দীর্ঘ ঊ
shôrô ô	shôrô a	hrôsshô i	dirghô i	hrôsshô u	dirghô u
ô/o	a	i	i	u	u
[ɔ~o]	[a]	[i]	[i]	[u]	[u]
ক	কা	কি	কী	কু	কূ
ka	ka	ki	ki	ku	ku
ঋ	এ	ঐ	ও	ঔ	
হ্রস্ব ঋ	এ	ঐ	ও	ঔ	
hrôsshô ri	e/æ/ê	oi	o	ou	
ri	e	oi	o	ou	
[ɹi]	[e~ɛ~æ]	[oi]	[o]	[ou]	
কৃ	কে	কৈ	কো	কৌ	
kri	ke	koi	ko	kou	
ক্	কৎ	কং	কঃ	কঁ	
k	kôt	kông	kôḥ	kñ	
[k]	[kɔt]	[kɔŋ]	[kɔh/kɔ]	[kɔ̃]	

Figure 4. Bangla Vowels, Vowel Modifiers, and Diacritics (Source: Omniglot.Com)



ক্ক	ক্ট	ক্ত	ক্ব	ক্ম	ক্র	ক্ল	ক্ষ	ক্ষ্ম	ক্স	গ্ধ
kkô	kṭô	ktô	kbô	kmô	krô	klô	kṣô	kṣmô	ksô	gdhô
গ্ন	গ্ব	গ্ম	গ্ল	ঘ্ন	ঙ্ক	ঙ্ক্ষ	ঙ্থ	ঙ্গ	ঙ্ঘ	ঙ্ম
gnô	gbô	gmô	glô	ghnô	ngkô	ngkṣô	ngthô	nggô	ngghô	ngmô
চ্ছ	চ্ছ্ব	চ্ঞ	জ্জ	জ্জ্ব	জ্ঝ	জ্ঞ	জ্ব	ঞ্চ	ঞ্ছ	ঞ্ঝ
chchhô	hchhb	chñô	jjô	jjbô	jjhô	jñô	jbô	ñchô	ñchhô	ñjhô

Figure 5. Selection of Bangla Conjuncts (Source: Omniglot.Com)

Indic scripts have many common characteristics that make the incidents in this dissertation highly relevant to other Indic scripts beyond Bangla. They are alpha-syllabic, meaning each consonant has a default vowel associated with it, constituting a syllable. In Bangla, the inherent vowel is the phoneme /ɔ/, typically denoted with the letter 'a'. The Bangla letter 'ক' represents the sound /kɔ/ and is typically romanized as '*ka*'. The inherent vowel can be changed via 'vowel modifiers,' which attach onto the base consonant, or silenced with a symbol that typically appears under the consonant, called a *virama* or *halant* (e.g.ক্) . Vowel modifiers can appear on either side, below, or above the base consonant, depending on the Indic script. Vowels also exist in an independent form for cases when they are not modifying a consonant. When consonants appear next to each other, they can ligate into new graphical representations called "conjuncts". In sum, a common characteristic for Indic scripts is that the visual representations of letters can vary depending on what other letters are next to them. A single "glyph" or visual representation may also represent more than one letter of the alphabet. Unlike the Latin alphabet, there is not a one-to-one correspondence between letter and glyph. This quality, and divergence from the logic of Latin scripts, has proven difficult for modern computers to handle, as I discuss in Chapter 1.

Typographers working in both analog and digital media consider Bangla an especially difficult script to mechanize.[16] Letters and diacritics stack considerably atop one another. Vowel flourishes can extend on both sides of a base letter. Sequences of consonants form complicated ligatures that can depart quite drastically from their constituent parts. Type designers and engineers have had difficulty reckoning with these qualities since the birth of mechanical typesetting in the early 20th century, carrying through to the internet age. From a technical perspective, the Bangla script is an excellent case study for examining the limits of type and computing technologies, given its finicky structure.

Additional context that hangs in the background of this dissertation is the heavy politics of the Bangla language and script. A brief history: Bengal was first split into two regions during the late British colonial period, in 1905. It had been a hotbed of political activity, spurred by a vibrant regional literature; splitting East and West Bengal was an attempt by the British to "divide and conquer." Within years, the partition was overturned. But when India gained independence from the British in 1947, Bengal was again the site of partition. The 1947 Partition followed a religious logic, and East Bengal became the designed site for Muslims, and West Bengal for Hindus. Regional identity remained strong and the Bengal border remained porous for several years, however, until the formalization of national passports and passport checks in 1952.[17]

Religion, ethnicity, and language were all closely associated with identity in post-Independence South Asia. The early history of the Independence era is marked by battle and bloodshed over the right to gain recognized status for one's ethno-linguistic group. In India, this manifested as fights over the drawing of state lines according to language group. In Pakistan (then still a discontiguous region on either side of India), it manifest as a nine-month war in 1971 in which

[16] Riccardo Olocco, "Linotype Bengali and the Digital Bengali Typefaces," MA thesis, University of Reading, 2014.

[17] Pallavi Raghavan, "The Making of South Asia's Minorities: A Diplomatic History, 1947- 1952," *Economic and Political Weekly*, May 21, 2016.

the Eastern wing of Pakistan, what was once East Bengal, fought for independence and became the nation-state of Bangladesh. Bangladesh's Liberation War strengthened regional pride in the Bangla language and script; a precipitating event for the Liberation War had been a proposal by the Pakistani government to replace the Bangla script with Latin letters.[18] It also led to international recognition: in 1999, the United Nations Educational, Scientific and Cultural Organization (UNESCO) proclaimed February 21st of every year to be "International Mother Language Day," the day in 1952 when Bangla language activists in Dhaka were martyred and which inspired the decades-long Bangla language movement.

The Bangla language can thus be seen in some ways as representing a special case due to its history and politics. There is a strong emotional charge laden in the language, and the Liberation War had occurred recently enough that most of the individuals who feature in this study felt it impact their families or neighbors. As their personal histories will show, particularly in Chapter 2, understanding of the Liberation War is not confined to Bangladeshis alone; those across the border in India feel strong sympathies for their ethno-linguistic compatriots.

At the same time, I warn against overstating the importance of this political history, and denying some of the technical reasoning that emerges around the *khanda ta* debate from the Bangla user community. As several Unicode staffers were wont to do, it is easy to dismiss arguments made for Bangla as coming from a place of emotion or nationalism alone. The language politics are relevant and present, but not the only factors at play. Linguistic pride is a widespread phenomenon, and the desire to have one's language and script properly represented is not a unique sentiment. Nationalism-motivated resistance or interactions with the Unicode Standard have been documented by prior scholars, in contexts such as Israel, South Korea, China, Japan, and Ethiopia.[19] In each case, the particular history and politics of language provide a "script" (in the sociological sense) that governments and individuals have utilized to make their points; I share the history of the Bangla language to provide background on the scripts that appear in this story.

Finally, as a note on both case and method, I emphasize here the advantages of taking a language as the unit of study. Using the Bangla language as my entry point, I am able to study relations between the institutions and histories of two nation-states, India and Bangladesh. I am also able to explore the connections between diasporic Bangla users and those based in South Asia. Finally, I am also able to explore the relationship between native speakers and outsiders, or "foreigners" or "non-native speakers," as they are referred to by interlocutors. I am able to show how, due to India's relatively high economic and political status and promise of offering 'the next billion users', Bangla gains the attention of Western technology companies – even when Bangladesh is far off the Western radar. I also illustrate the warring allegiances of diasporic Bengalis, who strive for membership to an emerging class of Western-educated technical elite, while still feeling connection to their home countries. And I uncover the surprising leadership of non-native Bangla

[18] Dennis Kurzon, "Romanisation of Bengali and Other Indian Scripts," *Journal of the Royal Asiatic Society of Great Britain & Ireland* 20, no. 1 (January 2010): 61–74, https://doi.org/10.1017/S1356186309990319.

[19] See, for example, John 2013; Park 2016; Tsu 2022; Zaugg 2017.

speakers who, in many cases, can articulate the grammar and evolution of the Bangla language with greater learned expertise, in comparison to those for whom it is an instinctive mother tongue.

Histories of South Asia

This dissertation also contributes to the field of modern South Asian history, and particularly, South Asian histories of science and technology. I bring together histories of language politics and information technologies in this work.

The relationship between technology and South Asian society has most often been viewed in the extant scholarship through the lens of economic development.[20] Digital technology – in the form of internet access or mobile phones – has been conceived of as an intervention that can raise standards of living, increase access to information, reduce corruption, improve crop yields, provide access to education and healthcare, bolster disaster preparedness, and much more.[21]

There is much less literature historicizing technologies within the long sweep of South Asian history and situating it within social and political contexts. One illuminating example of this latter category is Lilly Irani's anthropological study, *Chasing Innovation*, which situates the figure of India's 21st century "entrepreneurial citizens" as a metamorphosis of the country's original nationalist development project. Irani examines how everyday actors such as designers and engineers came to constitute a distributed network of agents, performing planning much in the way the state did in the immediate postcolonial period.[22] This dissertation picks up on this theme, showing how members of the Bengali diaspora, in particular, become similar distributed agents in the sphere of language technology – picking up, in this case, where the state has faltered. Other recent publications taking a historical approach include Menon (2018), which details the contrasting vision for computers in 1950s India, in comparison to the United States, the United Kingdom, and Soviet Union during the same era. The digital computer was envisioned as a tool for state planning.[23] Subramanian (2003) highlights the close relationship between national politics and the computer industry in India, going through waves of openness and closure towards international computer companies.[24] Singh (2018) describes how American and

[20] Asif A. Siddiqi, "Technology in the South Asian Imaginary," *History and Technology* 31, no. 4 (October 2, 2015): 341–49, https://doi.org/10.1080/07341512.2016.1142632.

[21] William Mozzarella, "Beautiful Balloon: The Digital Divide and the Charisma of New Media in India," *American Ethnologist* 37, no. 4 (2010): 783–804.

[22] Lilly Irani, Chasing Innovation Making Entrepreneurial Citizens in Modern India, 2019, https://escholarship.org/uc/item/3239b1qv.

[23] Nikhil Menon, "'Fancy Calculating Machine': Computers and Planning in Independent India," *Modern Asian Studies* 52, no. 2 (March 2018): 421–57, https://doi.org/10.1017/S0026749X16000135. Menon chronicles a nascent precursor to Chile's computerized state-planning initiative, Project Cybersyn, whose rise and fall has been chronicled by Eden Medina (2011).

[24] Ramesh Subramanian, "India and Information Technology: A Historical & Critical Perspective," *Journal of Global Information Technology Management* 9, no. 4 (October 1, 2006): 28–46, https://doi.org/10.1080/1097198X.2006.10856431.

British entrants to metal typography during the late colonial period had to navigate British colonial stakeholders as well as a growing nationalist movement; much like the Unicode Consortium as presented in Chapter 4 of this dissertation, typography firms in the 1930s held inarguably political positions, as they were inevitably pulled into decisions about who had the authority to define a script, and what that typed script would look like.[25] Though the pre-Independence study on British India reflects on the modern-day country of Bangladesh, there are no other studies of the computing industry after 1947 that focus on Bangladesh. In this dissertation, I draw instead on first-hand accounts.

Scholarship on South Asian language movements has focused primarily on the precipitating factors and impacts on national policy[26] In several cases, as with the Bangladesh Liberation War, there is much work still to be done on unsettling the singular nationalist narrative of a uniform populace seeking independence from Pakistan.[27] Beyond this ongoing need for excavation of the language movements, there is a dearth of scholarship on the lasting repercussions of the language movement – how it remains in the memory of its participants, and how it is articulated and passed down to second and third generations. Chapter 2 of this dissertation picks up on these themes, showing how traces of the Bangla language movement have instilled a passion for language and language technology amongst those one generation removed from the war.

At the time of writing, there is little scholarly literature on the intersection of language and technology in South Asia. The few published sources in peer-reviewed journals are personal accounts and memoirs of the computing industry, which again serve as primary sources in this study.

Sociolinguistics for the digital age

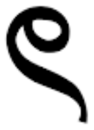


Figure 6. Bangla Letter *Khanda Ta*

"Is it truly an independent letter in the alphabet? Can it already be represented in the standard? Why didn't other Indic scripts have a symbol like this?" These were some of the questions at the heart of the *khanda ta* debate, and they spanned the technical *and* the linguistic. Alongside contributions to histories of the internet and histories of South Asia, this dissertation contributes

[25] Vaibhav Singh, "The Machine in the Colony: Technology, Politics, and the Typography of Devanagari in the Early Years of Mechanization," *Philological Encounters* 3, no. 4 (November 27, 2018): 469–95, https://doi.org/10.1163/24519197-12340051.

[26] Ramachandra Guha, India After Gandhi: The History of the World's Largest Democracy, Reprint edition (New York/N.Y: Ecco, 2008); Mithilesh Kumar Jha, Language Politics and Public Sphere in North India: Making of the Maithili Movement, n.d; Robert D. King, Nehru and the Language Politics of India, n.d.

[27] Anushah Hossain, "Remembering East Pakistan," *The Bengal Gazette* (blog), July 31, 2020, https://bengalgazette.org/2020/07/31/remembering-east-pakistan/.

to scholarship on the social place of linguistic expertise, particularly in the digital age. This is a nascent area of research within the field of sociolinguistics.

Sociolinguistics scholarship on the internet has most often focused on the concepts of *language change*, *language loss*, and *language regimes.* The largest body of scholarship at this intersection explores the impact of the internet *on* language, and specifically on language change and language regimes. This sub-field, termed "internet linguistics" by David Crystal, examines how features of language – the orthography, grammar, vocabulary, phonetics, etc – change due to the internet.[28] Crystal initially used a technological deterministic view to consider whether there existed an internet-specific language, "Netspeak," that followed new linguistic conventions. He found that indeed, the medium of the internet tended to change written communication, but that different regimes were emerging on different platforms. The linguistic conventions for email were different from chat boards, for example.[29] And contrary to the moral panic gripping the public in the early days of the Web, established regimes of spelling and grammar could continue to exist off the internet – formal writing would not experience widespread decay.[30]

Sociolinguists have since moved away from the deterministic framing of Crystal's work and have considered instead how Netspeak transforms along gender, racial, class, and geographic lines — essentially interrogating the role of culture in mediating technology.[31]

The concept of *language loss* has also featured prominently in the literature on language and the internet. As Kornai writes, "biological metaphor of viewing languages as long-lived organisms goes back at least to Herder [writing in 18th century]".[32] Paolillo and Pimienta, and Kornai have both presented frameworks for measuring linguistic diversity in the internet, transferring the analog concepts of speaker population to online population, and vigorous oral use to vigorous online use.[33] These frameworks respond to the popular claim that English will supersede other languages in digital space, and that minority languages will face "digital language death."[34] The metrics for digital vitality have slowly expanded as researchers have taken a more holistic view over time, ranging from availability of Wikipedia pages in one's language, to the presence of internationalized domain names, the availability of fonts and keyboards, and mature natural

[28] David Crystal, *Language and the Internet* (Cambridge: Cambridge University Press, 2001), https://doi.org/10.1017/CBO9781139164771.

[29] *Ibid.*

[30] *Ibid.*

[31] *Globalization of Language and Culture in Asia: The Impact of Globalization Processes on Language,* ed. Viniti Vaish (A&C Black, 2010); Susan E. Cook, "New Technologies and Language Change: Toward an Anthropology of Linguistic Frontiers," *Annual Review of Anthropology* 33, no. 1 (2004): 103–15, https://doi.org/10.1146/annurev.anthro.33.070203.143921.

[32] András Kornai, "Digital Language Death," *PLOS ONE* 8, no. 10 (October 22, 2013): e77056, https://doi.org/10.1371/journal.pone.0077056.

[33] *Ibid*; John C. Paolillo and Daniel Pimienta, "Measuring Linguistic Diversity on the Internet," 2005.

[34] Kornai 2013.

language processing tools.[35] Recognition of the disparities between languages along these lines has led to the defining of new terms such as "digitally-disadvantaged languages" – languages that may suffer from a dearth of digital communication tools and content, regardless of their status as majority or minority language outside of digital space.[36]

But returning to our interest in the social place of linguistics expertise, the most pertinent sociolinguistics concepts for this dissertation are *language planning and language standardization*. Language planning refers to allocation of state resources towards the elevation and refinement of languages and their scripts.[37] Standardization – of grammar, letterforms, spelling – is one aspect of language planning.[38] The seminal work bridging traditional language planning with the digital environment is David K. Jordan's 2002 article, "Languages left behind: Keeping Taiwanese off the World Wide Web."[39] Jordan argued that Unicode depended upon a source base of formalized scripts that it would then encode. "Unorthodox, unstable, or unofficial scripts", such as those used for Taiwanese dialects, were at risk of falling off the Internet, as Unicode would not be able to accommodate them. He posed,

> *When the new global standard is finally fully in place, when our operating systems are finally Unicode-based and our browsers and word processors routinely provided with full Unicode type fonts that offer Burmese and Russian and Arabic and Mongol on the same page (a moment that has already arrived for some of us), will it be too late to stabilize a new writing system for Hokkien? Will the age of innovative Chinese (and other) orthographies have drawn to a close? Will we have standardized at least some orthographies, perhaps some languages, into corners as curiosities that cannot be seriously used in a world in which literacy has become intimately bound to electronic information flow?*[40]

Not much has been written on the implications of Unicode on linguistic innovation since Jordan's piece. Of note, Jordan published in 2002, upon the recent release of Unicode 3.0 and the first wave of widespread adoption amongst "operating systems…and browsers and word processors." The events of this dissertation occur just after Jordan's time of writing and seek to answer many of his questions. I consider the extent to which instability is accommodated and scripts are permitted to innovate within Unicode in Chapter 4.

[35] Pratik Joshi et al., "Unsung Challenges of Building and Deploying Language Technologies for Low Resource Language Communities" (arXiv, December 7, 2019), http://arxiv.org/abs/1912.03457; Richard Littauer, "Open Source Code and Low Resource Languages" (Saarland University, 2018).

[36] Isabelle A. Zaugg, Anushah Hossain, and Brendan Molloy, "Digitally-Disadvantaged Languages," *Internet Policy Review* 11, no. 2 (April 11, 2022), https://policyreview.info/glossary/digitally-disadvantaged-languages.

[37] "Language Policy and Planning," obo, accessed June 29, 2022, https://www.oxfordbibliographies.com/view/document/obo-9780199772810/obo-9780199772810-0273.xml.

[38] "Language Standardization," obo, accessed June 28, 2022, https://www.oxfordbibliographies.com/view/document/obo-9780199772810/obo-9780199772810-0250.xml.

[39] D.K. Jordan, "Languages Left behind: Keeping Taiwanese off the World Wide Web," *Language Problems & Language Planning* 26, no. 2 (August 1, 2002): 111–27, https://doi.org/10.1075/lplp.26.2.02jor.

[40] *Ibid*, 121.

Most of the key concepts from sociolinguistics (beyond linguistic innovation) have just begun being translated, reinterpreted, or theorized from scratch for digital environments. That which does exist focuses primarily on the East Asian context, as in Jordan's work, and comments on the specific governance structures for East Asian scripts and on regional competition between those nation-states. Zhao's article, "Flows of Technology: Mandarin in Cyberspace" also used the framework of language planning to evaluate reactions amongst East Asian countries to Unicode.[41] China, which was seeking modernization and standardization, welcomed the work of Unicode's East Asian script committee. Japan, on the other hand, was displeased with the same committee's efforts as it had already completed script standardization and developed its own Unicode competitor. Tsu explains part of the discomfort with Unicode experienced by East Asian countries. Because additional script standardization needed to take place to prepare East Asian ideographs for Unicode encoding, a script committee was formed to perform this task, called the Ideographic Research Group (IRG). As Tsu writes, "Even though the IRG is made up of mainly computer engineers and scientists, they find themselves having to take on a Sinologist's or linguist's work."[42]

This dissertation analyzes how the regional dynamics, legacies of language planning, and linguistic structures of South Asia's scripts are brought forth into the digital age — revealing models of engagement between linguistic authorities and the Unicode Standard that greatly diverge from the East Asian model. Instead of inter-region or inter-script competition, we see South Asian officials grapple with unifying and enriching a conflict-ridden, poor nation inherited at independence. This case offers the field of sociolinguistics an example of what a postcolonial, digital language politics might look like.

Methods: Sources and Analysis

This dissertation takes an approach to the study of technology similar to those found in media studies, in which I examine an assemblage of standards, formats, and technologies that are ultimately packaged into one medium for users: digital text. Studies conducted in a similar vein include Sterne's *MP3: The Meaning of a Format* and Smith's *A Biography of a Pixel*, that focus on digital sound and digital images respectively.[43]

There is a vast infrastructure that supports digital text, and it can be difficult to scope and study. As Star and Ruhleder wrote, infrastructure typically exists in the background, it is invisible, and it

[41] Shouhui Zhao, "Flows of Technology: Mandarin in Cyberspace, " in *Globalization of Language and Culture in Asia: The Impact of Globalization Processes on Language,* ed. Viniti Vaish (A&C Black, 2010).

[42] Jing Tsu, Kingdom of Characters: The Language Revolution That Made China Modern (Riverhead Books, 2022).

[43] Jonathan Sterne, *MP3: The Meaning of a Format*, (Durham: Duke University Press Books, 2012); Alvy Ray Smith, *A Biography of the Pixel* (Cambridge, Massachusetts: The MIT Press, 2021).

is frequently taken for granted.[44] At the same time, one of the practical methods for observing its design and maintenance is through observing moments of breakdown.[45]

The subjects of my study are the stakeholders involved in the "breakdown" of *khanda ta*. They roughly fall into the categories of programmers, linguists, typographers, and policy makers, though in most cases each person wears more than one of those hats. More specifically, my subjects include members of the Unicode Technical Committee, or "UTC"; representatives from software companies like Microsoft, who are deploying the Standard and providing feedback at Unicode meetings; and government officials, academics, and software hobbyists who are also drawn into the discussions. I identified my sample through snowballing, beginning with Bangla software hobbyists and expanding from there to identify all the technical experts involved in Bangla language digitization. I collected oral histories from each subject, lasting between 1-2 hours, amounting to 36 interviews in total. I typically conducted interviews with Bangla speakers in Bangla, which I then translated into English; all other interviews were conducted in English. Oral histories were a valuable data source, as the events at hand were recent enough to remain in the memory of my participants, but were only partially documented in public archives. Oral histories were most essential for tracing out the development and motivations behind the OpenType format, for which there is a surprising dearth of information despite its importance to modern typography.

There were several large digital archives that also served as data sources for this project. I made use of well-kept Unicode meeting notes and technical documentation, which are publicly available on the Unicode Consortium's website. I also accessed documents from the Government of India's Ministry of Information Technology website, which stores newsletters and agendas from the year 2000 onward. Most importantly, I collected messages posted across several public mailing lists from the early 2000s, including the main Unicode list; the Unicode "Indic list," which discussed issues specific to Indic scripts; and three software hobbyist groups called "indic-computing", "Bengalinux/Ankur", and "freebanglafonts". Though these lists were all once public, some have since been hidden from public view by the list administrators; in these cases, I requested access and downloaded the materials for analysis. The number of messages posted to each of these lists numbered in the thousands. I reviewed each archive exhaustively, except for the main Unicode list, where I queried only for Bangla-related posts.

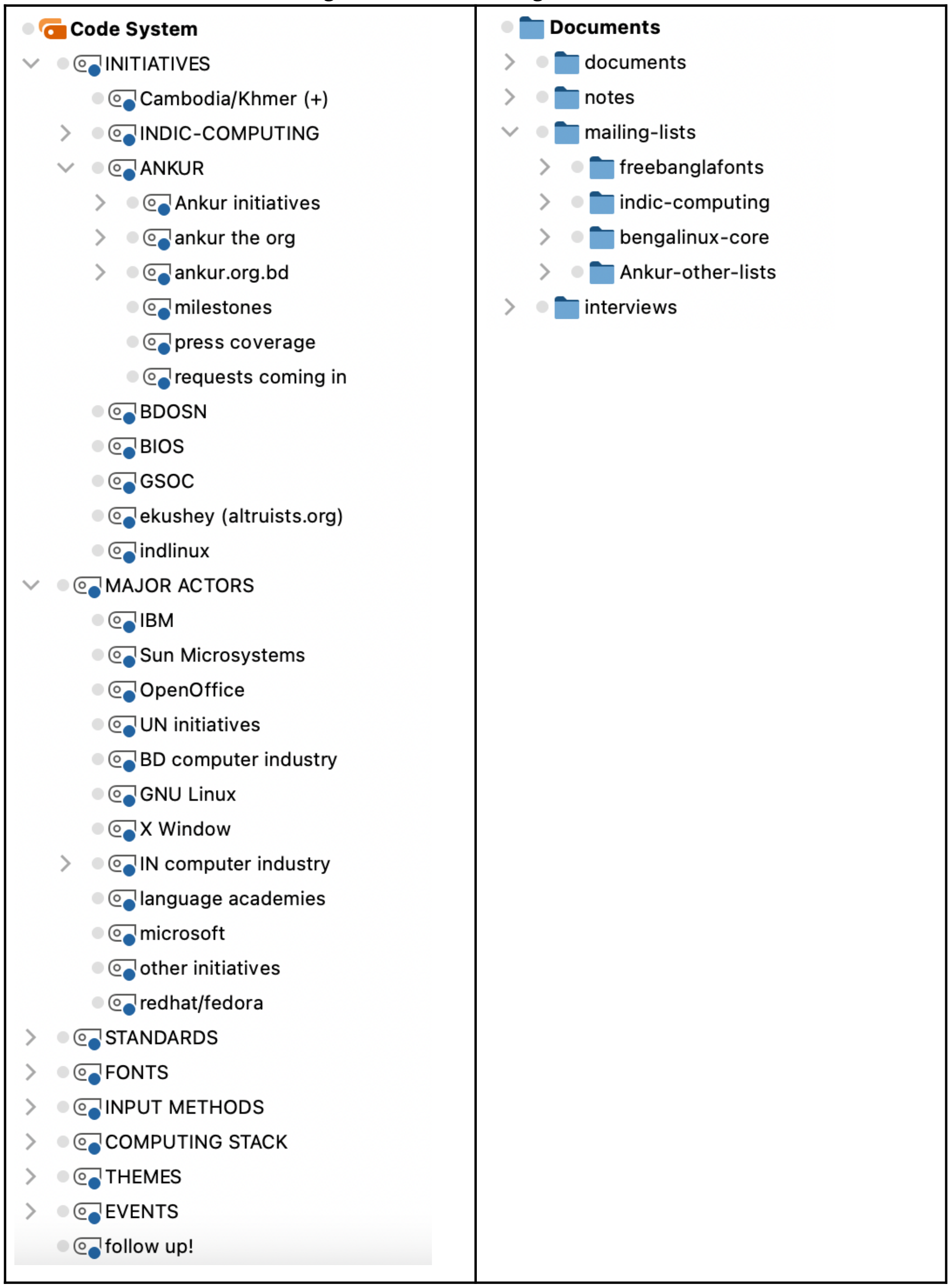
Following a networked/snowballing approach for my written sources as well, I looked for personal blogs and project websites that were mentioned in the oral histories or email archives. In most cases, the websites were no longer live, so I relied on snapshots stored within the Internet Archive's Wayback Machine. This public archive proved to be an invaluable resource for excavating this history of the early Web.

[44] Susan Leigh Star and Karen Ruhleder, "Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces," *Information Systems Research* 7, no. 1 (March 1996): 111–34, https://doi.org/10.1287/isre.7.1.111.

[45] Susan Leigh Star, "The Ethnography of Infrastructure," *American Behavioral Scientist* 43, no. 3 (November 1, 1999): 377–91, https://doi.org/10.1177/00027649921955326.

I analyzed all of the data sources above using MaxQDA, a qualitative data analysis (QDA) software application. I used an unstructured, interpretivist approach to first code all of the materials for recurring themes, projects, and organizations, producing over fifty codes. I re-coded to refine and organize the codes into easy-to-reference categories for the writing stage.

Figure 7. Data Coding Scheme



- **Code System**
  - INITIATIVES
    - Cambodia/Khmer (+)
    - INDIC-COMPUTING
    - ANKUR
      - Ankur initiatives
      - ankur the org
      - ankur.org.bd
      - milestones
      - press coverage
      - requests coming in
    - BDOSN
    - BIOS
    - GSOC
    - ekushey (altruists.org)
    - indlinux
  - MAJOR ACTORS
    - IBM
    - Sun Microsystems
    - OpenOffice
    - UN initiatives
    - BD computer industry
    - GNU Linux
    - X Window
    - IN computer industry
    - language academies
    - microsoft
    - other initiatives
    - redhat/fedora
  - STANDARDS
  - FONTS
  - INPUT METHODS
  - COMPUTING STACK
  - THEMES
  - EVENTS
  - follow up!

- **Documents**
  - documents
  - notes
  - mailing-lists
    - freebanglafonts
    - indic-computing
    - bengalinux-core
    - Ankur-other-lists
  - interviews

Though the sources used in this study have generally been public to view and download (or accessed with disclosure about the intent of this study), they reference individuals who are still alive, and in many cases, continue to work professionally in the same industry. We should take particular care in interpreting their communications, as they were produced at a time when the vast audience of the modern Web was not wholly conceivable. Despite their ‘public’ status, I caution that the spaces and discussions I document are better understood as ‘private’ (though not ‘secret’), with their own norms and rules that observers such as ourselves should make an effort to understand. I use names and quote directly from web archives throughout this work, with the understanding that the reader will take care to heed the context in which these communications were produced.

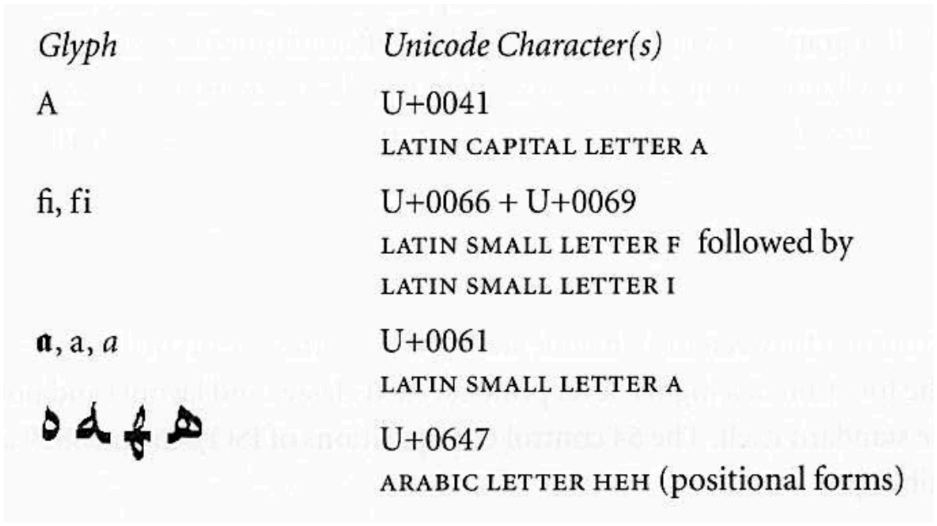
Key terms

There are a handful of terms used throughout this text that will be helpful for readers to grasp before we proceed. First, I want to distinguish the terms “**language**” and “**script.**” Broadly, language is a method of communication; we often use it to mean spoken or signed communication. To write down a language, one needs a script, or writing system. Scripts contain the letters, numbers, and other symbols needed to graphically represent a language. When conventions, such as spelling and grammar, are developed for a script, that system is called an “**orthography**.”

The Unicode Standard at the center of this work digitizes scripts, not languages. A script can be used for more than one language; a language can also use more than one script. Unicode aims to encode the minimum number of scripts to support all of the world’s languages. This means, for example, that Unicode encodes the Latin script, rather than the English or Swedish languages (both of which use the Latin script).

Second, I want to distinguish some similar but contrasting terms from linguistics, typography, and software internationalization that refer to the building blocks of writing systems: “**letter,” “glyph,” “grapheme,”** “**ligature,”** and **“character” and “codepoint.”** Perhaps the most familiar of these is the letter, the units that make up an alphabet. Grapheme is a similar term from linguistics that refers to the smallest units of the entire writing system. This includes letters, but also numbers and punctuation, as in the Latin capital letter A shown in row 1 of Figure 8.

Graphemes can be represented visually in more than one way – changing the visual design but not the underlying meaning, or “semantics.” For example, the vowel grapheme or letter a can be drawn in the following ways. These different visual representations are called **glyphs**, and are a term used predominantly in typography. Typefaces for a single script can have different glyphs, but represent the same graphemes, as in row 3 of Figure 8.



Glyph	Unicode Character(s)
A	U+0041 LATIN CAPITAL LETTER A
ﬁ, fi	U+0066 + U+0069 LATIN SMALL LETTER F followed by LATIN SMALL LETTER I
ɑ, a, *a*	U+0061 LATIN SMALL LETTER A
ﻩ ﻪ ﻬ ﻫ	U+0647 ARABIC LETTER HEH (positional forms)

Figure 8. Glyphs and Characters (Source: the Unicode Standard V.1)

When some graphemes show up next to each other in words, they take on new combined glyphs called **ligatures,** as in the fi ligatures shown in row 2 of Figure 8. Ligatures are also a typographic term and again represent a visual or stylistic difference, rather than a change in underlying semantics.

Next, there is a more abstract concept defined by the software internationalization sector, called "**characters.**" What are characters? They roughly map onto graphemes; here is the term as it appears in the Unicode glossary:[46]

> *Character*. (1) The smallest component of written language that has semantic value; refers to the abstract meaning and/or shape, rather than a specific shape (see also *glyph*), though in code tables some form of visual representation is essential for the reader's understanding. (2) Synonym for *abstract character*. (3) The basic unit of encoding for the Unicode character encoding. (4) The English name for the ideographic written elements of Chinese origin. [See *ideograph* (2).]

It is contrasted directly against glyph – Unicode famously encodes **only characters, not glyphs**, so as not to bloat the Standard (there is no finite number of glyphs).

Finally, I introduce another term from software internationalization: the **codepoint**. Codepoints are unique numbers assigned by Unicode to each character in the standard. Things get confusing when we see that not all graphemes are assigned single codepoints; they may instead be assigned a *codepoint sequence*, or multiple codepoints that computers must be programmed to recognize as a single grapheme. Here, the correspondence between graphemes and characters begins to falter. Users and engineers may have incompatible understandings of what it means for a grapheme to be "in the Unicode Standard." A user may not see it in the list of assigned codepoints, but the engineer may see that it has nonetheless been assigned a codepoint sequence that is hidden in

[46] "Glossary," accessed June 28, 2022, https://unicode.org/glossary/.

the documentation. These nuances and inconsistencies are important to keep in mind as they provide some of the fuel behind *khanda ta*'s fire.

Chapter Roadmap

Over the course of five chapters, this dissertation traces the evolution of the debate around how a single Bangla letter appears online. Each chapter advances the story incrementally, moving through the 1990s when the standards for multilingual text are developed, and settling in the early 2000s when those standards are being refined through debates like this one. Each chapter also focuses on one set of actors, from the technical designers of the Unicode Standard and OpenType format in Chapter 1, to software hobbyists who are beginning to implement these standards in Chapter 2, to government officials promoting plans for local-language technologies in Chapter 3, to industry actors again in Chapter 4, and finally academic linguists in Chapter 5. The conclusion provides a final verdict on *khanda ta*.

Language digitization is complex and interdisciplinary, bringing in many different fields of expertise and motivations. I tease out the various sets of motivations in each chapter – from business interests, to the open source ethic, to national pride, to academic rigor – that drive the debate forward. The advantage of this slow crawl towards a decision around *khanda ta* is demonstrates how standards are truly made, and how easily the narrative and foci of power can shift during the decision-making process. We see how the decision over a single letter can become inflamed and turn into a high-stakes tussle in which forms of technical and cultural expertise and the hierarchy of nation-states are challenged.

As Joe Becker, one of the founders of Unicode, once wrote, *"The many aspects of text character encoding are highly interrelated, and indeed each topic is best conceived in terms of a conception of all the others. Lacking hypertext or the ability to discuss all topics at once, the document is arranged as follows:"*[47]

Chapter 1: Assembling The Multilingual Internet provides a history of multilingual text through the second half of the 20th century. The reader becomes acquainted with what I call "the multilingual computing stack," the set of standards and technologies that layer atop each other to facilitate multilingual digital communication, from the Unicode Standard, to font technologies, to layout technologies that ensure text is displayed properly on screens. The main characters of this chapter are the engineers who set about developing the core standards that ensure multilingual text works on any machine, anywhere in the world: the designers of the Unicode Standard, and the designers of an equally important font format called the OpenType format. I chronicle the technological and economic context that drove their decision-making, from limits on the processing power of personal computers, to the needs of the two most important markets: Latin script users and East Asian script users.

[47] Becker, Joseph D., "Unicode 88," August 29, 1988, https://unicode.org/history/unicode88.pdf.

I show how the design of the multilingual computing stack that emerges disadvantages Indic script users. The logic of the Unicode Standard makes it relatively easy to design working fonts for Latin and East Asian scripts, but frustratingly difficult for Indic scripts where letters and glyphs do not have a 1:1 correspondence. As a result, several other technologies (such as the OpenType format) must be developed to ensure Indic scripts work properly, which only begin being developed at the turn of the millennium. I recount this history to establish the uphill battle that Bangla script users would face in the coming years, as they struggle against the systemic bias written into the Unicode Standard.

Chapter 2: Building Bangla Software provides a social history of Bangla-language software. I focus here on Bangla software hobbyists, who began building out the rest of the multilingual computing stack – fonts, keyboards, localized software applications – when they felt no one else was doing it. I demonstrate how these software hobbyists embody the “open source ethic” - to share work freely, not ask for permission, and build with whatever is at hand. Even as issues with *khanda ta* arose in the view of these groups, it was largely treated as a bug that they could easily build hacks around, a contrast to the stakeholders presented in later chapters for whom the issue carries additional weight. I begin the chapter in 2002, when the most prominent Bangla software group, *Bengalinux*, was formed. The Unicode Standard and OpenType format have launched but have yet to gain widespread recognition and adoption in South Asia. Local stakeholders working on language technologies – industry professionals, academic researchers, and government officials – remain ignorant or opposed, and continue to work in non-standard formats that inhibit widespread communication. Major companies such as Microsoft have not yet released their Bangla-language operating systems. I show how the Bangla software hobbyists work to advance the tools and standards they believe will define the future of digital communication, positioning themselves between the worlds of a elite, global software industry, and what they perceive to be inward- and backwards-looking South Asian governments.

Thus far the featured actors have been software engineers – whether those working for major western corporations or those working independently on open source software. In **Chapter 3: Digitizing Language Planning,** the perspective shifts to South Asian government officials, for whom language technology is explicitly tied to national pride and economic development. I present the philosophies and communications of leaders from India’s Technology Development for Indian language (TDIL) program, which began forging a strong relationship with the Unicode Consortium in 2003. I show how TDIL’s activities are best understood as an extension of the postcolonial practice of language planning, in which the state directs resources to promote languages and scripts in public space, such as in schools, on road signs, and in the media. For language planners, the issue of *khanda ta* is best understood as a “missing letter,” with implications for the status of the Bangla language, rather than a technical bug as the software hobbyists viewed it. In the emerging internet age, however, several shifts have occurred: the public space is *digital* space, mono-lingual regimes have given way to multilingualism, and language planners must have technological as well as linguistic expertise. I show how India’s TDIL program represents a new iteration of language planning, which must calibrate itself to the norms of emerging internet governance institutions such as the Unicode Consortium. Signed government letters and diplomatic visits to California do not hold the same weight as the informal, technical argumentation that will emerge from non-government actors in the following chapters.

Chapter 4: Accommodating Orthographic Reform carries forward the sociolinguistics lens of language planning, but hones in one aspect of it: orthographic reform, or state-sanctioned changes to writing systems. The events of this chapter take place in Unicode mailing list, where a range of orthographic reforms to Bangla is being discussed. We turn our eyes towards a set of intermediary actors, linguists-turned-technologists who are responsible for interpreting the Unicode Standard and implementing it in downstream software, such as rendering engines. I walk through four increasingly difficult demands that are made of the multilingual computing stack concerning the Bangla language. To what extent do these technologies and standards serve as enabling or limiting systems? Bringing together the frames of orthographic reform and techno-politics, I argue that the designers of the multilingual computing stack ultimately view themselves, and generally act in such a way, as to be agnostic accommodators of orthographic reform. Their goals are to understand the qualities and evolutions of a writing system, and translate them as best as they can to the digital medium. The question these technical "translators" ask is often *how,* rather than *if* a linguistic feature should be accommodated. This self-view becomes important to keep in mind as we progress to the next and final chapter, where the multiple perspectives and stakes presented throughout this dissertation finally converge.

Finally, I end with **Chapter 5: The Battle over *Khanda ta*.** Present in this chapter is the suite of actors introduced in previous chapters, as well as the new addition of academic linguists, who are historically the gatekeepers of orthographic reform. To them, it feels as though the very language itself is being compromised, not merely its digital encoding. We follow an increasingly contentious showdown between Bengali linguists and Western technocrats, in which the stakes of *khanda ta*'s encoding rise from disrespect towards linguistic expertise to neocolonialism perpetrated by Silicon Valley elite against non-English users. This chapter brings to the fore the emotional valences that simmered under the surface of previous chapters, and we see how the explicit expressions become productive in the fight to get *khanda ta* encoded.

The final chapter, ***Khanda ta*, encoded** concludes the story of *khanda ta* by relaying the verdict made by Unicode with respect to its encoding. I trace the implications of this wide-ranging and long-standing debate for each group of actors presented in previous chapters: the designers of the Unicode Standard, Bangla software hobbyists, government officials overseeing language, technology, and development, and linguistic experts. *Khanda ta* becomes a hallmark decision that begins to shift Unicode's treatment of all Indic scripts, it becomes a small victory for the software hobbyists and linguists, and convinces the Bangladeshi government to follow India's lead and begin direct engagement with the Unicode Consortium.

—-

Though the issue of *khanda ta* was resolved in 2005, it did not fall from popular memory. The story of *khanda ta* would be remembered and misremembered for many years. Bengali technologists recalled it as the critical episode that guaranteed the preservation of their language online. Unicode staffers would offhandedly mention it in meetings as late as the year 2020, symbolizing, for some, a time when "mob mentality" won out.

A viral blog post in 2015 was still criticizing the Unicode Consortium for its handling of *khanda ta*, and non-Latin scripts more broadly, whose users were "forced to make orthographic contortions just to write a simple email."[48] Though the blog post made mistakes in how the Unicode Standard worked (mistakes that were corrected generously in the many comment sections of the forums where the blog post was shared), it noted accurately that Unicode membership was "comprised largely of white men (and a few white women) whose first language was either English or another European language."[49] The author argued, "it's imperative that the writing system of the 21st century be driven by the needs of the people using it. In the end, a non-native speaker – even one who is fluent in the language – cannot truly speak on behalf [sic] the monolingual, native speaker."[50]

As this dissertation shows, none of these points are wholly true on their own. What appears at first glance to be a characteristic example of systemic bias gives way upon further investigation to complicated issues of digitizing a language whose alphabet is still being reformed, of distributing responsibilities between layers of a newly emerging 'multilingual computing stack' (of which the Unicode Standard is only one element), and of incorporating feedback from new multitudinous channels of feedback from user communities across the Internet.

For the non-technical reader, perhaps the historians, anthropologists, and sociologists of language or technology, my goal is to provide a thorough, empirical case study to help generate new hypotheses about the power structures of the modern Internet, the legacy of the nationalist South Asian development state, and the nature of language planning and standardization. I hope the findings that are surprising: Microsoft's productive role in expanding software internationalization; the push against Nehruvian ideas of state-led modernization; the absorption of language planning into technical rather than linguistic government agencies; and more. I work to contextualize the Internet not as a ground-breaking new paradigm, nor the teleological progression of the past, but as a dynamic space for encounter between the analog and digital worlds.

For the technical reader, this study is intended to concretely illustrate what is meant by the phrase, "code is political." The technical reader is likely to have found themselves reading through Unicode specifications at some point in time, trying to figure out how to get an edge case to work. Edge cases by definition are the situations that have not been addressed by the general model, divergences from the norm that hardly come up. It can be unclear how much the individual decision one makes in those cases matters. This story is about what happens when we look closely at an edge case, and suddenly discover there is a community of millions for whom

[48] Mukerjee, Aditya, "I Can Text You A Pile of Poo, But I Can't Write My Name," *Model View Culture* (blog), March 17, 2015, https://modelviewculture.com/pieces/i-can-text-you-a-pile-of-poo-but-i-cant-write-my-name.

[49] *Ibid*.

[50] Mukerjee 2015.

the outcome of one’s coding decision matters. There are long histories and deep values embedded in code, and this study excavates them for the Unicode Standard. This study shows the layers of bias built into the Standard, at the level of West versus the rest, but also at the regional level of Indic scripts and languages. We see how groups are privileged – exceptions for some, hard rules for others – and how outsiders must struggle to fit a system that was not designed around their needs. This case study also shows how user activism and user perceptions are consequential to changing the values embedded in such a system. In our current moment of rising tech activism, I hope this case is informative and instructive for those looking to make change.

Chapter 1: Assembling the Multilingual Internet

"It seems the developers of the computer and of word-processing software were coddled by the English language, which happens to have the simplest writing system of all: unadorned alphabetic letters laid out one after the other."

So wrote Joe Becker, a founder of the Unicode Standard, in an article on "Multilingual Word Processing" for *Scientific American* in 1984.[51] Indeed, the Latin script, used for writing the English language as well as others including German, French, and Vietnamese, was relatively simple in structure. There was a finite number of letters of the alphabet, which appeared in roughly consistent ways, and a handful of punctuation marks and numerals. A computer could store the entire writing system in an "8-bit encoding," or a table with 256 rows with information on each letter, number, and mark.

For much of the early digital computing era, from the 1960s to 1990, computers worked with 8-bit encodings, and the American 8-bit encoding ("ASCII") in particular. But the world's writing systems could not fit easily into this scheme. As computer suppliers began to eye international markets, and international markets expressed interest in purchasing computers, new ways to handle text needed to be invented.

This chapter traces the development of the set of technologies that eventually fulfilled that need. I tell the story of the Unicode Standard, the bottom layer of what I call the "multilingual computing stack." The term "stack" is commonly used in the computer networking literature to refer to the layered standards, protocols, and technologies that constitute our modern internet. The layers of a stack are modular but interdependent — each element has its own rules and procedures, but these rules and procedures must be communicated to and negotiated with the other elements.[52]

The history of the Unicode Standard has not yet been told from the perspective of Indic scripts. This vantage point is critical to understand because it best highlights the limitations of this standard, limitations that affect the digitization of scripts for nearly two billion people. In the following chapter, I describe how Unicode intended to handle Indic scripts: primarily through delegation to other layers of the multilingual computing stack. This act of delegation introduces a wider surface area for bugs (as the remainder of this dissertation shows) and releases the overseers of the Unicode Standard from responsibility for proper display of Indic scripts.

The second half of this chapter traces the development of the technical layer that augments the Unicode Standard and is most important for Indic script display: the OpenType font format. I argue that the OpenType format was equally critical as Unicode for displaying Indic scripts. The

[51] Joseph D. Becker, "Multilingual Word Processing," *Scientific American* 251, no. 1 (1984): 96–107.

[52] Benjamin H. Bratton, *The Stack: On Software and Sovereignty*, 1st edition (Cambridge, Massachusetts: The MIT Press, 2016), 52.

development of the OpenType format, and its attention to Indic scripts, has not yet been documented in the scholarship. I present it now, drawing from new oral histories with its developers. In sum, this new technical history of the digital text highlights the points at which Indic scripts received attention: as afterthoughts, in secondary layers. As the story of the Bangla letter *khanda ta* advances in the following chapters, we see how the consequences of Unicode's early design decisions play out and create an uphill battle for digitizers of Indic scripts.

Understanding the Multilingual Computing Stack

I begin first by describing the set of tools that constitute the modern "multilingual computing stack." This framework draws on several precedents: Becker (1984) first articulated the three functions that a computer needed to perform: *encoding,* or a way for text to be represented in the memory of a computer, *typing,* a way for text to be typed at the keyboard of a computer, and *rendering*, a way for the computer to present text to the typist.[53] These functions were expanded upon and connected to specific technologies and standards by Loomis et al (2017).[54] A similar framework has also been articulated in Translation Commons' digitization guide, "Zero to Digital" (2019).[55] I present a simplified version of Loomis et al (2017) model, selecting the components without which multilingual digital communication cannot proceed.[56]

At the lowest level is the **encoding standard**, which transforms text into bytes that a computer can understand.[57] Encoding standards are important for internal processing and for transmission. That is, an encoding standard tells computers how to process text ("internal processing") and lets a network of computers interpret the text sent between them, as they are all following the same set of rules ("transmission"). As such, encoding standards can be seen as a convergence of *computing* and *communications.* These two functions have not always been intertwined; for example, though ASCII reigned as a communications standard in the 1960s, it was not always used for internal processing in computers.[58] The Unicode Standard was designed to be amenable to both processing and exchange, goals which had implications for its design.[59]

[53] Becker, "Multilingual Word Processing."

[54] Steven Loomis, Anshuman Pandey, and Isabelle A Zaugg, "Full Stack Language Enablement," Steven R. Loomis, June 6, 2017, https://srl295.github.io/2017/06/06/full-stack-enablement/index.html.

[55]" Indigenous Languages Zero to Digital.Pdf," Translation Commons, accessed June 29, 2022, https://drive.google.com/file/d/1zpZK3jfF3bDt2e5YnEw8FXSkYXSRefKu/view?usp=embed_facebook.

[56] Loomis et al include "final frontiers" such as the availability of computer programming languages in one's mother tongue

[57] "Bits" are 0s or 1s that serve as data that a computer can process. A "byte" is eight bits in a row, for example 00000000 or 00101100. If a table of data is 8-bits, or 1-byte, then that means that it can fit 256 rows of information. This number (256) comes from the maximum combination of 0s and 1s that fit into eight spots ($2^8 = 256$).

[58] Computers are most efficient at processing information structured in bytes (or eight bits), for example. The first version of ASCII was a 7-bit standard, which was suboptimal for internal processing. See "Early History of ASCII?," accessed June 29, 2022, https://groups.google.com/g/alt.folklore.computers/c/gbg5YVFaT48/m/wlVFfJ2j4hYJ; Steven J. Searle, "Brief History of Character Codes in North America, Europe, and East Asia," TRON Web, 1999, http://tronweb.super-nova.co.jp/characcodehist.html#anchor953122.

[59] Becker, Joseph D., "Unicode 88," August 29, 1988, https://unicode.org/history/unicode88.pdf.

Encoding standards have primarily been oriented around sending *semantics* across network lines – the abstract ideas of letters. One example is early telegraph codes such as Morse code.[60] These would send codes that represented letters like an ‘a’ or punctuation such as a period, rather than information about the appearance of a letter — whether it should be boldface or italic or serif or sans serif.

Modern computers with screens need more information than the semantic encoding alone to represent text appropriately to the user. Towards this purpose, they need **digital fonts**. Digital fonts contain visual information for each letter. They turn text into images. The history of fonts over the late 20th century is a story of how the typography industry – traditionally concerned with the aesthetics and legibility of text over physical media – has become entangled with the computing industry. As a result, type has transformed into ‘information’ and come to be modeled and manipulated by software. Digital fonts can be seen, then, as a convergence of *computing* and *display.* I trace this particular evolution below in the section on “Digital Type.”

The third critical piece of technology needed for multilingual computing is the **input method**, typically a keyboard, that triangulates between the encoding standard, user, and font, to produce text. Virtual keyboards are relatively simple to build, but hard to get right; user preferences also vary widely.[61] As a result, there is a high proliferation of keyboards designed to various specifications. Though they are critical for the user, they feature in only limited ways in this dissertation, due to their relative distance from the standards at the center of this work.

For Latin scripts, these core pieces are enough to produce, display, and transmit text across the Internet. Granted, operating systems and applications need to also be able to support these pieces — the encoding standard, digital fonts, and keyboards — and much of the following story in the 1980s and 90s is about computer companies falling in line and adopting common standards. But with these basic tools in place, it becomes possible to perform higher-level commands such as text search, word processing, spell check, and even optical character recognition (OCR).

For Indic scripts, however, a critical other piece of software is needed: specialized software for – **rendering** or doing appropriate display. Unlike the Latin script, Indic scripts are made of letters that can appear in completely different visual forms depending on what other letters are next to them.

Consider the Bangla script as an example. The Bangla alphabet has fifty letters — 39 consonants and 11 vowels. But vowels can appear in independent and dependent forms. Consonants can combine with each other when next to other consonants, sometimes resulting in three or four letters combining into one graphical unit, called a “conjunct.” The total set of combinations is

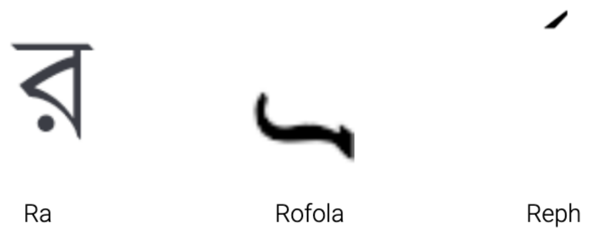
[60] Searle, “Brief History of Character Codes in North America, Europe, and East Asia.”

[61] Apple employee, interview with author, March 3, 2022.

neither standardized nor finite; estimates of the total number of visual units in Bangla range from 300 to 800.

A font can be designed to contain all of these possible "glyphs" (the various visual forms of the writing system), but if an encoding standard only contains semantics, it becomes unclear what visual to display. If a computer only has the information that the Bangla consonant *ra* was typed, how will it know which if the following three visual forms it should display?

Figure 9. Glyph Variants of *Ra*



A piece of software must therefore work with the encoding standard and font to interpret what graphic to display based on the context. This technology is called a "text rendering engine," and its role is essential not only for Bangla, but all Indic scripts. Not only does this contextual difference between letters and glyphs occur in Indic scripts, but also for Arabic scripts. Together, these two script families (Arabic and Indic) are termed "complex scripts" by the software internationalization industry, due to the additional work that must be done to display them correctly. The implicit "normal" assumed by this term ("complex script") is, of course, the Latin script, which does not become illegible without sophisticated text rendering.[62] The following sections illustrate how the notion of complex scripts came to be, beginning with the design of the Unicode Standard.

The Unicode Standard

The most widely implemented encoding standard in use today is the Unicode Standard, a scheme that aims to assign a unique code point for every unique semantic unit of all the world's scripts. As we will see, encoding semantics was not the only possible design principle, but it was the scheme promoted by Unicode and that which gained wide adoption leading into the new millennium.

At the time of Unicode's development in the late 1980s and first release in 1991, several individual character codes were in circulation globally. These were specific to different software vendors and to different languages. For example, IBM had its own standard called EBCDIC; the

[62] This is, of course, a matter of perspective. As one software engineer opined to me, "I learned Devanagari growing up, and it never seems complex to me. A lot of the complex comes from the lens you view it with, the way it's treated by technology."

United States had the popular ASCII standard; Japan had JIS.[63] There were also competing efforts to create a single universal standard that combined them all. These efforts towards unification were motivated by two important recognitions.

Firstly, a world of multiple conflicting standards would make displaying text slow and buggy at best, and illegible at worst.[64] This was due to the lack of uniqueness in the assigning of codepoints. If a document was produced on a machine following an American encoding scheme where 001 was assigned to ‘a’ and opened on another machine where the native encoding scheme assigned 001 to the Latin letter ‘p’ the Japanese character ‘は’ then the text would become garbled on that machine.

Workarounds for this frequently occurring confusion were developed by the International Organization for Standards (ISO) in 1987, to instruct computers how to switch between different encodings. ISO 2022 established “switch codes” which would flag when a new encoding scheme was being introduced.[65] Over fifteen country-specific schemes (most of which were variations on the 7-bit American standard, ASCII) could now be used across computers that were configured to understand ISO switch codes.[66]

The problem with switch codes, however, was that they proved troublesome for word processing. A cursor selecting a letter to *copy* could only produce the same letter upon *paste* if the computer internally stepped back in the text to first find the relevant switch code.[67] If one highlighted the letter ‘a’ that was mapped onto 001, how would the computer know if the 001 was from the American encoding scheme or the Swedish one? It would have to check its memory for the relevant switch code. This process made word processing time-consuming and error-prone. More challenges arose when handling “variable-width” schemes, as in documents using both Latin 8-bit and East Asian 16-bit encodings, as the machine could not count on characters having consistent lengths.[68] Simple functions, like knowing that deleting 8-bits would effectively erase one letter, could not be easily done.

In addition to these technical difficulties, the computing industry was increasingly cognizant of a coming era of increased globalization, where documents, emails, and software would need to travel frequently across national borders.[69] As computer companies became more international during the 1980s, there was a growing awareness that a universal, international standard was necessary.

[63] Anthony McEnery and Zhonghua Xiao, “Chapter 4 Character Encoding in Corpus Construction.” 3-4.

[64] *Ibid;* Ken Whistler, interview with author, January 29, 2020.

[65] Anthony McEnery and Zhonghua Xiao, “Chapter 4 Character Encoding in Corpus Construction.” 4.

[66] *Ibid.*

[67] Ken Whistler, interview.

[68] Anthony McEnery and Zhonghua Xiao, “Chapter 4 Character Encoding in Corpus Construction.” 9.

[69] Joseph D. Becker, “Multilingual Word Processing.”

With these needs in mind, at least three efforts were underway to develop a multilingual text standard by the late 1980s. The first was an effort by ISO to unify all existing standards *and* hold space for the rest of the world's scripts that had yet to be encoded. Following this aim, a working group was developing a 32-bit standard entitled ISO 10646.[70] 32 bits meant that 2^{32}, or nearly 4.3 billion, pieces of data could be stored int he standard, meaning that ISO 10646 would have enough codepoints for nearly 4.3 billion letters.

Another competing standard was the TRON standard, which was being developed out of the University of Tokyo beginning in 1984. TRON used switch codes like those introduced by ISO 2022, but switched between several 16-bit 'planes,' so that it too could encode billions of letters.[71]

The TRON standard's distinguishing features were its attention to East Asian scripts. The East Asian ideographs used for Chinese, Japanese, Korean, ("CJK"), Taiwanese, and sometimes Vietnamese, were far more numerous than the number of letters needed for the Latin alphabet. These scripts shared a common heritage and had some semantics in common, but each language had evolved different glyphs for the same semantics. For example, though CJK each had an ideograph for "return," the word was represented in the following five ways across Traditional Chinese, Simplified Chinese, Vietnamese, and Japanese.

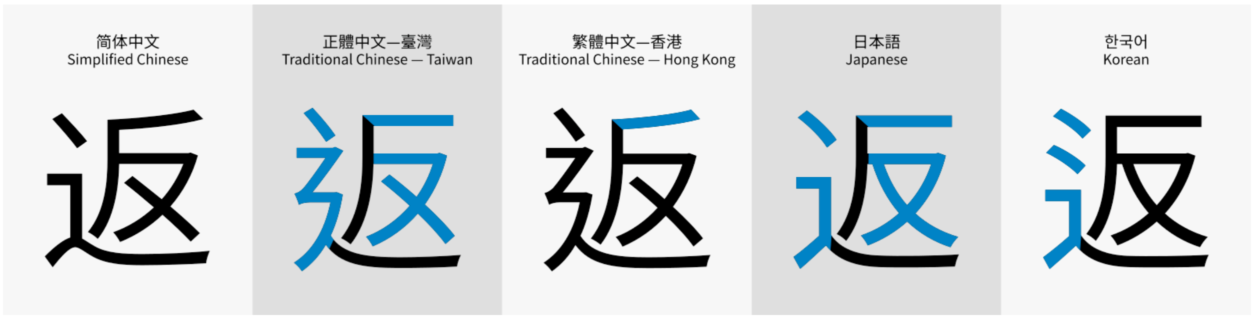


Figure 10. CJK Glyph Variants

In the TRON encoding system, each glyph was allocated its own codepoint. As a result, the project would require 200,000 codepoints for East Asian scripts alone.[72] Though it kept the processing difficulties of switch codes and required more computer memory to store the large encoding, TRON did honor the nuances of the CJK scripts by encoding all glyphs.

[70] Searle, "Brief History of Character Codes in North America, Europe, and East Asia."

[71] *Ibid.*

[72] *Ibid.*

In contrast to these initiatives, Unicode was designed to be an optimally-sized encoding system of 16-bits.[73] This length was determined to be still-manageable for contemporary computers, yet large enough to fit all major scripts, pursuant to a few notes.

The biggest challenge would be fitting in 200,000 CJK glyphs. A single 16-bit table would permit 65,536 rows of data, not nearly enough for even the East Asian scripts. But the early designers of the Unicode Standard, Joe Becker, Lee Collins, and Mark Davis, recognized the common semantics of the CJK ideographs and determined to *unify* them as a single set of codepoints. This effort, termed "Han unification", would reduce the 200,000 codepoints in TRON to approximately 20,000 points instead. This would still leave about 45,000 spaces available for the encoding of other scripts.[74]

The decision around Han unification led to Unicode's core design principle, of encoding only "characters not glyphs." The term "character" was invented by Unicode, and was rather tautologically defined. In an early scoping document, Becker posed, was 16-bits enough to fit the world's characters? Answering his own question, he wrote,

> *Since the definition of a "character" is itself part of the design of a text encoding scheme, the question is meaningless unless it is restated as: is it possible to engineer a reasonable definition of 'character' such that all the world's scripts contain fewer than 65,536 of them?*[75]

The answer was yes, pursuant to the principle of encoding only semantically distinguishable letters. It meant that for the example above, Unicode would only encode the concept of "return" as one number, or "codepoint," instead of allocating five codepoints for the five different glyphs shown above.

"Han unification" was easier said than done. There was no existing documentation of what the common set of characters was between the CJK scripts. A multinational committee would need to be formed to determine the set of characters for unification.[76] This meant determining whether two glyphs that looked different from one another were only graphical variants, or were representing the same underlying semantic value. If the difference was only graphical, then the responsibility of distinguishing and displaying the correct glyph fell to the font layer.

But the "character not glyphs" policy was not consistently applied within Unicode. To save space with the 16-bit standard, the founders of Unicode decreed that letters would also occasionally be

[73] Becker, Joseph D., "Unicode 88," August 29, 1988, https://unicode.org/history/unicode88.pdf, 4.

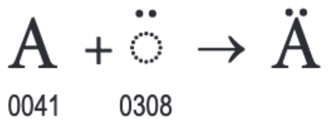
[74] "Unicode 88," 3.

[75] "Unicode 88," 4-5.

[76] See Shouhui Zhao, "Flows of Technology: Mandarin in Cyberspace, " in *Globalization of Language and Culture in Asia: The Impact of Globalization Processes on Language,* ed. Viniti Vaish (A&C Black, 2010) and Jing Tsu, Kingdom of Characters: The Language Revolution That Made China Modern (Riverhead Books, 2022).

defined by combinations of codepoints.[77] The canonical example was overhead accents on Latin vowels. Since the circumflex could appear on multiple vowels, Unicode's design principles decreed that the accented vowels be produced using a vowel+accent sequence like the example below. This move of 'composition' was another space-saving policy, though it added one more (small) processing step for computers to handle.

Figure 11. Codepoint Sequence Example



This policy was muddied, in turn, by yet another design principle – that of ensuring "round-trip, backwards compatibility' with major existing standards (other than in the case of CJK scripts).[78] This meant that existing encoding schemes would be included wholesale in the first version of Unicode, as a way of achieving greater completeness and adoption. This led to the 7-bit ASCII standard, then most widely implemented in software, being included exactly as is into the Unicode Standard as the virtually identical first 128 codepoints. It also led to the inclusion of several European standards, which had accented vowels such as ô ,î ,ê ,â, and û encoded directly in their national standards.

As a result, the Unicode Standard had several redundant encodings included, in contradiction to its other principles. Because of the topography of the computing industry – with the most mature markets being located in Europe and East Asia – most of the redundant, convenient (or unsequenced) encodings belonged to the Latin-lettered European scripts.[79] This scheme would be disparaged later by non-Latin users. When it came to East Asian scripts, Unicode reduced and flattened; when it came to circumflexed Latin letters, redundancies were readily accommodated.

But these inconsistencies could not later be overturned because of a final core design principle, the stability policy.[80] This held that once an encoding, or even a character name, was officially included in the Standard, it would be forever honored. This was a way to promise stability to Unicode adopters and help maintain the integrity of downstream software.

Unicode's launch was received with critiques of "cultural imperialism" by some, primarily because of the affiliations of those steering the project.[81] Unicode had been the brainchild of former Xerox and Apple employees who had been working on encoding schemes for their respective companies. ISO 10646, the unwieldy 32-bit standard, was overseen by an

[77]' *The Unicode Standard, version 1.0,* 10.

[78] *The Unicode Standard, version 1.0,* 3.

[79] *The Unicode Standard, version 1.0,* 19.

[80] *The Unicode Standard, version 2.0.*

[81] Searle, "Brief History of Character Codes in North America, Europe, and East Asia."

international treaty organization with country delegates voting on its development. In contrast, the Unicode Consortium was designed to be a non-profit organization with tiers of paid membership. Members could pay a fee of $10,000 USD to become a voting member. The initiative quickly attracted the membership of the largest software companies, most of which were based in the United States. To many observers, it appeared as if a cartel of American companies was unilaterally steering the direction of Internet communication.[82] As Dongoh Park has written, the Unicode Standard was seen as an unwelcome sign of globalization against the backdrop of nationalism and indigenous computing in South Korea.[83] Others felt its ultimate success was not owed to any technical superiority in its design, but rather the market power of its overseers.[84] Still others worried that the decisions over encoding would have practical impacts on the language itself, likening the technical decisions to orthographic reform.[85]

This was a key point that Unicode designers would reiterate to their users and detractors in coming years: the Standard was as technical blueprint for computers, not meant to be a faithful representation of a human-meaningful language. Even though a human might recognize î as a single letter, a computer could still reasonably store and process it as "i" + "^". This was the foundational principle of Unicode being a "logical encoding," as opposed to a graphical one. Its aims were only to enable reliable communication and computing.

The Indic Case

Not much has been written about the case of Indic digitization, but its handling by Unicode is illustrative of Unicode's design and limits.

Many major scripts were not encoded in Unicode until version 3, released in 1999. This was due to a lack of existing encodings to incorporate, or complexities in the script requiring more time and resources to properly digitize.[86] Indic scripts, however, were included in version 1 of Unicode, grandfathered in through a modification of the Indian government's ISCII standard. ISCII was an interesting multi-script scheme that aimed to encode nine of India's officially recognized in parallel form. It was conceptualized by researchers at IIT Kanpur, who had noticed commonalities in the phonetics and ordering of all the scripts that descended from the ancient Brahmi script.[87] The researchers decided to develop an encoding and keyboarding scheme that would lay out analogous letters of these Brahmi-based scripts alongside each other.

[82] Searle, "Brief History of Character Codes in North America, Europe, and East Asia."

[83] Dongoh Park, "The Korean Character Code: A National Controversy, 1987–1995," *IEEE Annals of the History of Computing* 38, no. 2 (2016): 40–53, https://doi.org/10.1353/ahc.2016.0021.

[84] Nicholas A. John, "The Construction of the Multilingual Internet: Unicode, Hebrew, and Globalization," *Journal of Computer-Mediated Communication* 18, no. 3 (April 1, 2013): 321–38, https://doi.org/10.1111/jcc4.12015.

[85] D.K. Jordan, "Languages Left behind: Keeping Taiwanese off the World Wide Web," *Language Problems & Language Planning* 26, no. 2 (August 1, 2002): 111–27, https://doi.org/10.1075/lplp.26.2.02jor.

[86] Whistler, interview.

[87] R. Mahesh K. Sinha, "A Journey from Indian Scripts Processing to Indian Language Processing," *IEEE Annals of the History of Computing* 31, no. 1 (January 2009): 8–31, https://doi.org/10.1109/MAHC.2009.1.

Each script would have its own encoding — akin to other 'switch code' schemes — but the envisioned advantage of laying out letters in the same order was that one could move between the scripts relatively easily, swapping out a keyboard cover for Devanagari with one for Bangla, and having the letter '*ka*' from both scripts be in the same place.[88] Multilingual text processing, which depending on the sorting order of the letters, might also become more easy to do. Functions like adding 3 to '*ka*' to get '*ga*' (akin to adding 3 to 'a' to get 'd') could be done with similar algorithms across the scripts.

While a clever linguistic analysis, ISCII had been perceived in India largely as an abstract experiment with limited practical value. Due to the idiosyncrasies between scripts that had evolved over many centuries, there were enough differences between them that translation between them was severely impaired. Furthermore, ISCII was controversial for seeming to privilege Devanagari, the script used to write the majority-language Hindi. Devanagari was placed in the first column of the encoding, against which all other scripts were made to correspond. As a result, many of the idiosyncrasies of the other scripts were left out of early versions of the ISCII standard.

Regardless, by 1988, ISCII had captured the attention of government officials in these Department of Electronics and was adopted by the Department of Official Language in the Ministry of Home Affairs.[89]

When Unicode was being prepared in the following years, ISCII came to be included merely "because it existed."[90] It made sense to pick it up — the "Government of India made it, were promulgating it, trying to implement it."[91] Since Unicode had a different aim of assigning unique codepoints to letters, it separated out the columns in ISCII and assigned each new codepoints. In accordance with the desire for backwards compatibility, only a trivial formula was needed to translate between the Unicode Standard and a machine that was already programmed with ISCII.

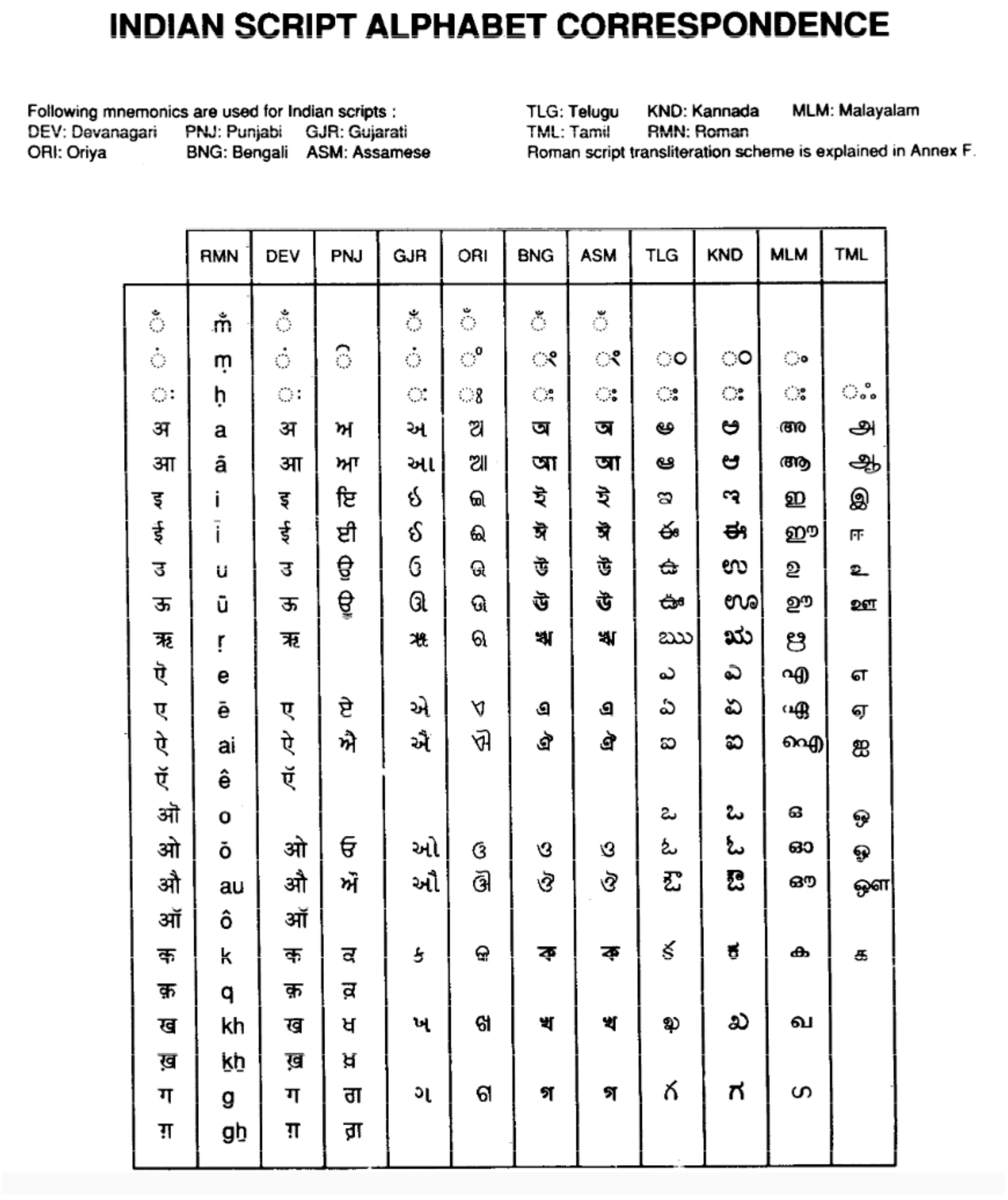
ISCII's design had accorded with the logical encoding philosophy that Unicode had sought to follow at the outset. In both Unicode and IIT Kanpur's estimation, for complex scripts such as these, computing was best served by storing only the semantics in the backend, and letting glyphs be handled on the surface level by other display technologies.

[88] "Development of ISCII and INSCRIPT Keyboarding - Dr. R. M. K. Sinha," accessed June 29, 2022, https://sites.google.com/site/profrmksinha/research-projects/development-of-iscii-and-inscript-keyboarding.

[89] "IS 13194 (1991): Indian Script Code for Information Interchange - ISCII," iv. https://law.resource.org/pub/in/bis/S04/is.13194.1991.pdf

[90] Ken Whistler, interview with author, April 23, 2020.

[91] *Ibid.*



INDIAN SCRIPT ALPHABET CORRESPONDENCE

Following mnemonics are used for Indian scripts :
DEV: Devanagari PNJ: Punjabi GJR: Gujarati
ORI: Oriya BNG: Bengali ASM: Assamese
TLG: Telugu KND: Kannada MLM: Malayalam
TML: Tamil RMN: Roman
Roman script transliteration scheme is explained in Annex F.

	RMN	DEV	PNJ	GJR	ORI	BNG	ASM	TLG	KND	MLM	TML
ँ	m̐	ँ		ઁ	ଁ	ঁ	ঁ				
ं	ṃ	ं	ੰ	ં	ଂ	ং	ং	ం	ಂ	ം	
ः	ḥ	ः		ઃ	ଃ	ঃ	ঃ	ః	ಃ	ഃ	ஃ
अ	a	अ	ਅ	અ	ଅ	অ	অ	అ	ಅ	അ	அ
आ	ā	आ	ਆ	આ	ଆ	আ	আ	ఆ	ಆ	ആ	ஆ
इ	i	इ	ਇ	ઇ	ଇ	ই	ই	ఇ	ಇ	ഇ	இ
ई	ī	ई	ਈ	ઈ	ଈ	ঈ	ঈ	ఈ	ಈ	ഈ	ஈ
उ	u	उ	ਉ	ઉ	ଉ	উ	উ	ఉ	ಉ	ഉ	உ
ऊ	ū	ऊ	ਊ	ઊ	ଊ	ঊ	ঊ	ఊ	ಊ	ഊ	ஊ
ऋ	ṛ	ऋ		ઋ	ଋ	ঋ	ঋ	ఋ	ಋ	ഋ	
ऎ	e							ఎ	ಎ	എ	எ
ए	ē	ए	ਏ	એ	ଏ	এ	এ	ఏ	ಏ	ഏ	ஏ
ऐ	ai	ऐ	ਐ	ઐ	ଐ	ঐ	ঐ	ఐ	ಐ	ഐ	ஐ
ऍ	ê	ऍ									
ऒ	o							ఒ	ಒ	ഒ	ஒ
ओ	ō	ओ	ਓ	ઓ	ଓ	ও	ও	ఓ	ಓ	ഓ	ஓ
औ	au	औ	ਔ	ઔ	ଔ	ঔ	ঔ	ఔ	ಔ	ഔ	ஔ
ऑ	ô	ऑ									
क	k	क	ਕ	ક	କ	ক	ক	క	ಕ	ക	க
क़	q	क़	ਕ਼								
ख	kh	ख	ਖ	ખ	ଖ	খ	খ	ఖ	ಖ	ഖ	
ख़	k̲h̲	ख़	ਖ਼								
ग	g	ग	ਗ	ગ	ଗ	গ	গ	గ	ಗ	ഗ	
ग़	g̲h̲	ग़	ਗ਼								

Figure 12. Alignment of Indic Scripts in ISCII (Bureau of Indian Standards, 1991)

Displaying Indic Scripts

The ISCII/Unicode encoding scheme was elegant in theory. In practice, however, rendering a ligature — a single glyph representing one or more characters — proved quite troublesome. There were two major sets of challenges with respect to displaying glyphs.

One issue had to do with the ordering of glyphs. Indic scripts were what linguists called an *abugida* writing system. This meant that consonants were the primary building block of the alphabet, and that they had a default vowel sound associated with them, called an “inherent vowel.” In Bangla, the inherent vowel was [ô], transliterated as ‘a.’ So the ক consonant did not represent the sound ‘k’, but rather ‘ka.’

Consonants could be modified to remove the vowel sound, using a silencer called a “virama” (Devanagari) or “halant/hasanta” (Bangla). The virama would appear under the letter (e.g. ক্ = ‘k’). The inherent vowel could also be adjusted to carry a different vowel sound. These “vowel modifiers” could appear on any side of a consonant. In some cases, a single vowel modifier could have two glyphs that appear on either side of the consonant (e.g. কো). Vowels could also appear in an “independent” form, usually at the beginning of a word.

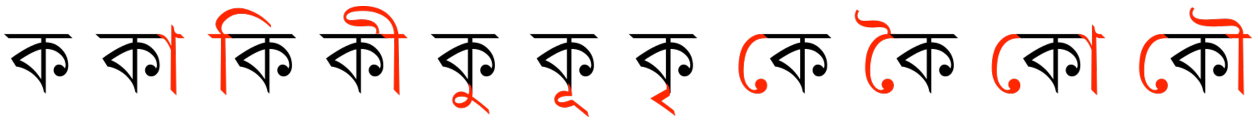


Figure 13. Vowel Modifiers On *Ka* (Bhargav Chowdhury - Creative Commons)

In ISCII, and thereby in Unicode, vowel modifiers, full vowels, and consonant full-forms were all independently encoded. The matter of re-ordering and re-positioning was relegated to higher-level software, however. Most keyboards were programmed to require the ‘base consonant’ to be inputted first, followed by the vowel modifier. Rendering software was responsible for finding the information about where the glyph should be displayed (placing a vowel modifier before or after, or sometimes below, the base consonant), and doing that reordering as a secondary process.[92] In this way, Unicode’s handling of Indic scripts defined a major expectation of anticipated digital type software.

There was a second more complicated expectation for rendering software with respect to consonant combinations, called “conjuncts.” In all Indic scripts, consonants could not only combine with vowels, but also with other consonants, as many as three forming a single conjunct. As conjuncts were perceived by Unicode to be variations in presentation, rather than semantically different from their constitutive parts, displaying the right conjunct glyph was also relegated to rendering software.

[92] *The Unicode Standard, version 1.0,* 13.

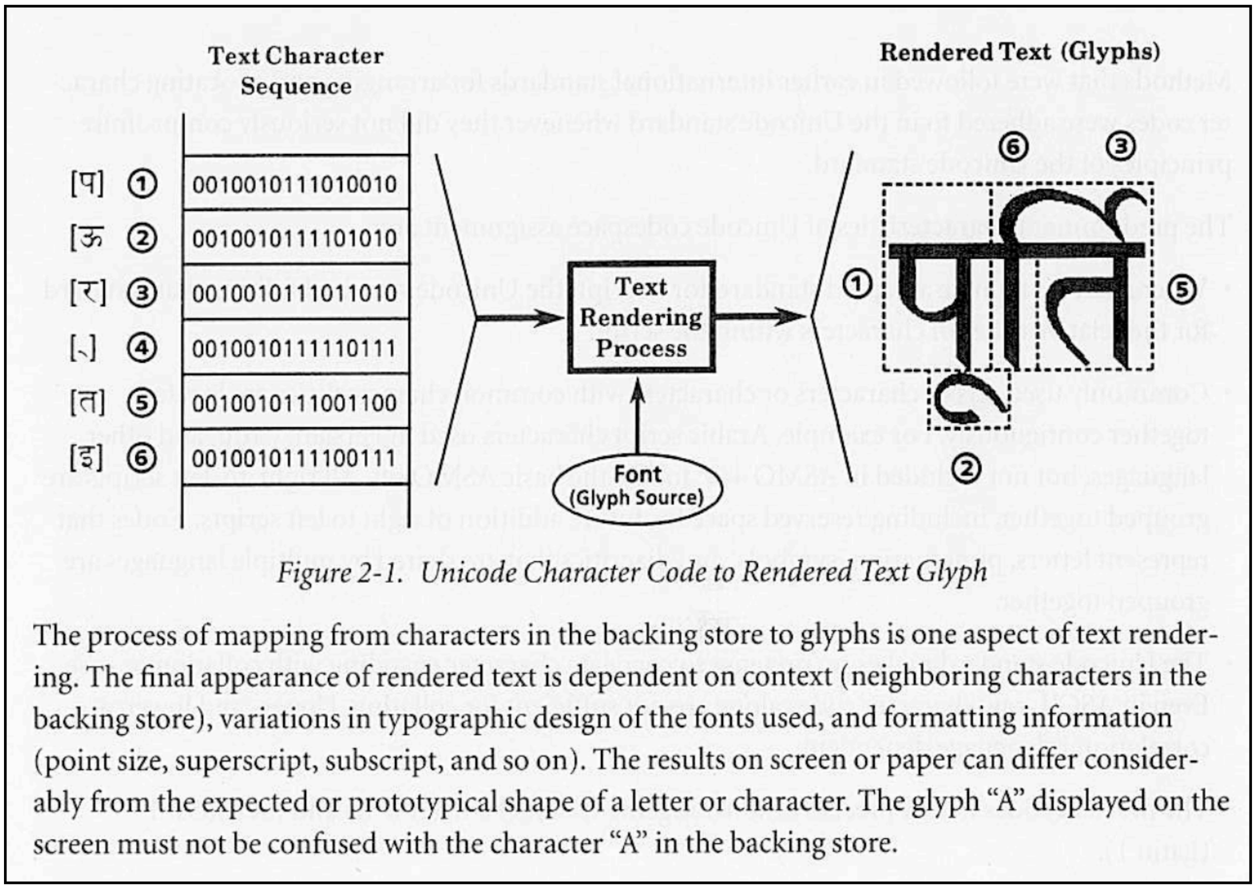


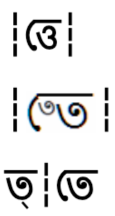
Figure 2-1. Unicode Character Code to Rendered Text Glyph

The process of mapping from characters in the backing store to glyphs is one aspect of text rendering. The final appearance of rendered text is dependent on context (neighboring characters in the backing store), variations in typographic design of the fonts used, and formatting information (point size, superscript, subscript, and so on). The results on screen or paper can differ considerably from the expected or prototypical shape of a letter or character. The glyph "A" displayed on the screen must not be confused with the character "A" in the backing store.

Figure 14. "Text Rendering Process" The Unicode Standard V.1

Deciding which conjunct glyph to display was not straightforward, however. Consonants could combine in a variety of ways. They could appear as a brand new ligature. There might be more than one form of the conjunct, with usage depending on the preferences of the user community. In some cases, the convention may be for the consonants to not combine at all, and instead appear side by side with a virama underneath. Though the below examples are semantically identical, their visual representation was anything but.

Figure 15. Examples of Ta+Ta+Akar in Ligated, Non-Ligated, and Galant-Forms (Unicode PRI-30)



How could the rendering software know which option to use in a given case? To help in this, Unicode encoded a handful of “non-printed characters,” or “control characters.” The Standard already had a *virama/halant* encoded from ISCII, which by default would trigger the *halant-form* of the conjunct from the font. If non-ligation was explicitly desired, then a “zero width nonjoiner”, or “zwnj” could be used to display case A above. Another control character, called a “zero width joiner” or “zwj” could follow a consonant to explicitly trigger its joining form.

If that scheme felt difficult to follow, your opinion would put you in the majority. In practice, ligating practices would be where Unicode would falter most for Indic scripts. The number of potential options for glyphs were so varied that control characters began needing to be combined in three or four codepoint sequences to cover them all. Furthermore, the logic for when to display which glyph varied across the Indic scripts, meaning the perceived simplicity and benefits of a common modeling system for all of the scripts quickly disappeared. What was initially perceived to be a straightforward, Devanagari-based system would receive crack after crack as the specific needs of other script communities came to light. And most importantly, the job of processing and displaying the right glyphs for Indic scripts was assigned in 1991 to a yet-to-be developed type software. Until its invention and widespread implementation, Unicode could not be used out of the box for any Indic script.

The OpenType Format

I move on now to a standard that became equally important as Unicode for Indic scripts: the OpenType font format. The OpenType Format (OTF) was a font format defined by Microsoft in 1997, but in partnership with major software design firm, Adobe. Though Microsoft receives most of the popular acknowledgement for OTF, the format built directly on Apple’s font technology, the TrueType format (TTF), released in 1991.[93] But in terms of global impact, OTF became far more significant for its wide reach and attractiveness to type designers.

The 1990s saw the development of several font formats that could support Indic scripts. Their development was the result of several intertwining trends, which I handle in turn in the following sections: advances in typesetting, typography becoming a computerized activity, and again, the desire for global expansion by technology companies.

Within type and design histories, where OTF has received the most attention, OTF is widely hailed for ending the “font wars” of the late 1980s and 1990s and for making advanced typography possible — subscripts, superscripts, ligatures, and custom design choices.[94] But its critical importance in the story of multilingual computing has rarely been acknowledged in the extant literature. Here, I present this untold history and use it to connect the histories of typography with that of computing and communications.

[93] An iteration on TTF, called TrueType GX, was actually the first to support the Unicode Standard and fill in some of the gaps for making complex scripts work.

[94] Robin Kinross, “The Digital Wave,” *Eye Magazine*, 1992, https://www.eyemagazine.com/feature/article/the-digital-wave.

A Brief History of Typography

Though Johannes Gutenberg often gets the credit for popularizing movable type, the basic technology had existed since its invention in 11th century China.[95] Gutenberg's main innovation was in developing "adjustable molds" — a method of printing faster and more cheaply. This was the driving concern of typography in the centuries that followed — *minimizing costs* for mass production.[96] To the extent that the technology allowed, typographers also sought to *maximize legibility* and *convey the spirit of the text* in the typeface design.[97]

These goals followed typography into the 20th century, when several developments helped transform the craft — evolving type from hot-metal to cold, and from analog to digital. The first major step in this direction was the development of phototypesetting, where instead of metal typefaces being forged and inked, text was produced by exposing light through cut patterns of letters onto chemically-treated paper.

Though phototypesetting removed some of the physicality of typesetting, the format was still considered "analog."[98] Type became "digital" with the invention of the Digiset machine in 1966, in which light was made to shine through grids of tiny points that represented the shape of a letter. As type historian Robert Kinross has written, "digital typesetting means that letters exist only where they can be generated by the rectilinear sweeps of a beam, either on or off."[99] Indeed the Digiset machine came with a terminal where the tiny points defining a letter, the "bits," could be edited and "saved" as data to be reused or transferred to another machine (via, for example, a floppy disk, which was invented in 1965.) This process improved efficiency and permitted customization; the patterning structure came to be known as the 'bitmap format' and the punched points would slowly transition to computer pixels.[100]

After this first example of digital typesetting, many others began working on making type mathematical and machine-readable. One problem with the bitmap format invented for the Digiset machine was that it was often discontinuous across font sizes. The bit patterns needed to be thoughtfully recomposed as the font became bigger or smaller, especially when trying to mind features like serif strokes or flourishes that were harder to define at smaller scales that had fewer bits available.[101] What was needed were "scalable fonts" or what is often called "vector fonts." Vector fonts came to be defined by mathematical formulae that combined straight and curved

[95] Robin Dodd, From Gutenberg to OpenType: An Illustrated History of Type from the Earliest Letterforms to the Latest Digital Fonts, Illustrated edition (Vancouver: Hartley and Marks Publishers, 2006).

[96] Dodd 2006.

[97] *Ibid.*

[98] *Ibid.*

[99] Kinross, "The digital wave."

[100] "Early Technologies of Digital Type," accessed June 29, 2022, http://www.designhistory.org/Digital_Revolution_pages/EarlyDigType.html.

[101] *Ibid;* Greg Hitchcock, interview with author, March 2, 2022.

lines to produce glyphs. These too were introduced by Dr. Rudolph Hell in 1974, the inventor of the Digiset machine.[102] Computer companies like Microsoft and Apple would later compete against each other on the size of their library of vector fonts.

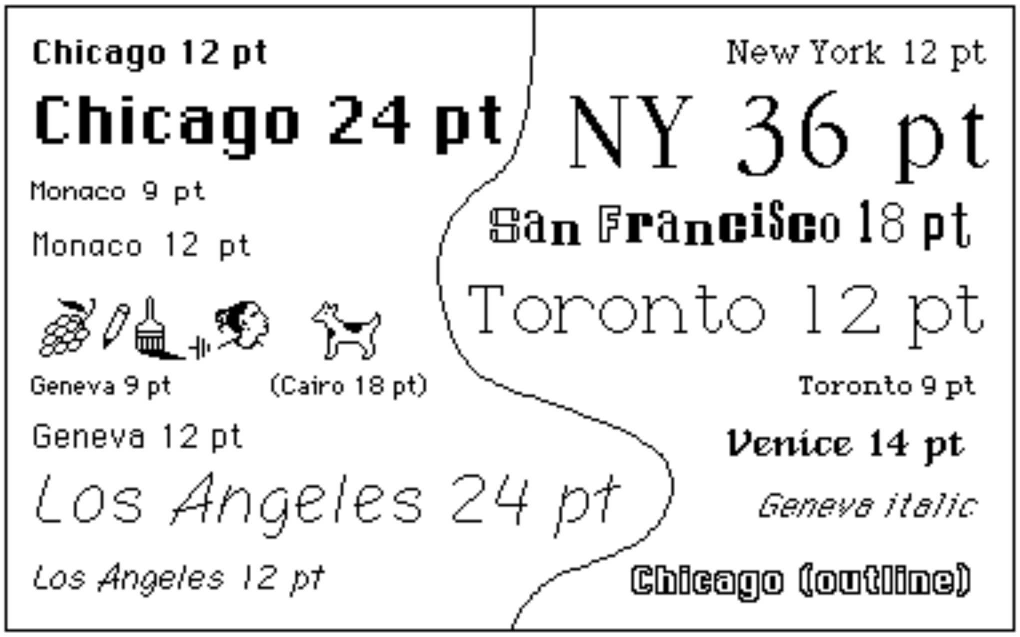


Figure 16. Original Mac Bitmap Fonts (David Remahl - Creative Commons)

These technological developments mostly affected high-volume typesetting, for clients such as newspapers. There were also several developments in "personal typesetting" that would set the path for personal desktop publishing in 1980s (which in turn would converge with the Unicode Standard in the 1990s). Typewriters became more customizable and supportive of advanced typography.

First was IBM's Selectric typewriter in 1961, which had golfball-shaped character molds that allowed faster, jam-free typing.[103] The modular, switchable "font balls" could also be changed to bold or italic type, or even to certain non-Latin symbols from Greek or Cyrillic.

---

[102] "Early Technologies of Digital Type."

[103] Dodd, 2006.

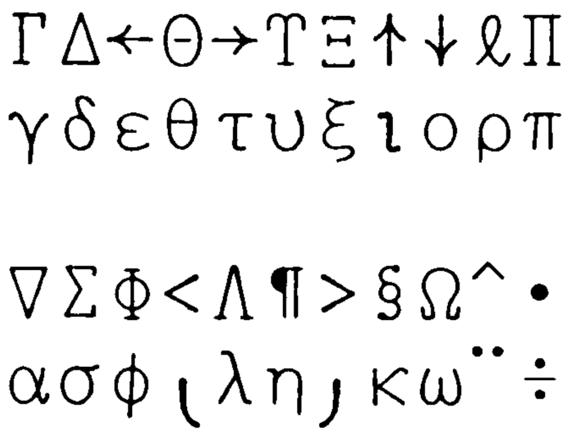


Figure 17. IBM Selectric "Symbol 10" Font

"Word processing" was also popularized as a concept, also by IBM through its release of the MT/ST typewriter, which had magnetic type to store text in memory.[104] A typist could now select text, copy-paste, backspace and rewrite. These developments are important to understand because they lay out how previously distinct activities were slowly converging in the late 20th century — as character codes can be seen as representing a convergence in computing and communications, digital fonts can be seen as convergence between computing and typography.

Digital Type on Personal Computers

The 1970s brought about the personal computer revolution. Early screens had low resolutions — these force the creation of jagged bitmap fonts in emerald green (a result of the constraints of the cathode ray tubes powering the screens).[105] Though there was text on computers, typography remained a largely separate industry from computing. Text on computers was mostly in Latin script, except for a handful of experiments with Japanese computerized word processing in the late 1970s.[106] Complex scripts were not initially supported by the 1970s personal computer revolution; they were still either typewritten in local language typewriters, or typeset for mass production on dedicated hot metal or photocomposition machines.

This state of affairs would change drastically with the release of a suite of technologies in 1984 and 1985. The following history is well known to the type design community. First was the Apple Macintosh computer, released in 1984 and the first personal computer to have a graphical user

[104] Brian Kunde, "A Brief History of Word Processing (Through 1986)," accessed June 29, 2022, https://web.stanford.edu/~bkunde/fb-press/articles/wdprhist.html

[105] Dodd 2006.

[106] K. Mori and T. Kawada, "From Kana to Kanji: Word Processing in Japan," *IEEE Spectrum* 27, no. 8 (August 1990): 46–48, https://doi.org/10.1109/6.58434.

interface.[107] Apple was positioning itself to be the premier tool for design. It did so by designing the Macintosh to be compatible with two other technologies: a ‘page description language’ called PostScript and a page design application called PageMaker, released by the Aldus Corporation.[108] PostScript was the invention of a newly established design firm, Adobe. Adobe was an outgrowth of the work done by former Xerox PARC employees on computerized word-processing and printers. PostScript provided the necessary instructions to computers to display on-screen fonts, and a set of instructions to output devices like printer to transfer them to a physical page. Together with Aldus’ design software for Macs, and Apple’s PostScript-compatible laser printer, the Laser Writer, it suddenly became possible for an individual to become a publisher. These technologies would constitute what we now refer to as Desktop Publishing (DTP). The design industry was agog.[109]

There were reasons to seek an improvement upon photocomposition. The holes on paper left edges soft where one might want them crisp; in contrast, the new DTP tools could support more beautiful and controlled typeface design.[110] Major type foundries such as Linotype began producing PostScript fonts, including for their non-Latin libraries, as early as 1987.[111]

Part of Adobe’s deal for PostScript fonts, however, was that designers had to license proprietary production tools from Adobe at steep costs.[112] Despite the cost, a major appeal of Adobe’s software was their font technology. Adobe had developed two categories of PostScript-compatible fonts: Type 1 and Type 3. Type 1 quickly supported both bitmap and vector fonts, and introduced the notion of “hinting.” Hints were where designers could specify exactly how the design should be scaled at different sizes and resolutions to maintain the integrity of the typeface features. This was especially important at small scales where bitmaps or poorly-defined vectors would lose stroke lines or serifs. Type 1 fonts were proprietary, whereas Type 3 were an open format and used bitmaps instead of vectors with hinting.

The high cost and evident superiority of Adobe’s font technology led competitors like Windows — still using choppy and memory-heavy bitmaps — to explore options of its own to keep up.[113] To the surprise of the competing and design industries, Windows ended up partnering with Apple to develop its own format in 1989, called TrueType (which would officially be released two years later). Apple had found Adobe’s licensing fees too high, on the order of the profit gained from selling the LaserWriter, and so was seeking an alternative. TrueType would be designed by Apple and licensed to Microsoft for free to ensure the format’s wide adoption. This event — the

[107] Phil Baines, “A Cast of Thousands,” *Eye Magazine*, 2002, https://www.eyemagazine.com/feature/article/a-cast-of-thousands.

[108] *Ibid.*

[109] *Ibid.*

[110] “Early Technologies of Digital Type.”

[111] Riccardo Olocco, “Linotype Bengali and the Digital Bengali Typefaces,” MA thesis, University of Reading, 2014.

[112] Hitchcock, interview.

[113] *Ibid.*

announcement of the partnership between the two computer companies — would launch the “font wars.”[114]

Digital Typography going Global

The font wars are folklore in the contemporary typography industry and are typically told as follows: Adobe sought to compete with the upcoming TrueType release by opening up their font formats and lowering the prices of their software. Apple and Microsoft nonetheless released their own technology, TrueType, in 1991 and released fonts for it on their operating systems. Microsoft split from Apple due to new, high licensing fees, leading the two companies to pursue separate ventures in 1992. Apple released the next version of TrueType, called TrueType GX, in 1994; Adobe released another format called Multiple Masters in 1992; and Microsoft released their own called TrueType Open.[115]

For users and designers, this stage of the competition was not ideal. Font formats were not compatible with one another, either needing to be hacked into operating systems, or making it simply impossible to share documents between devices. We should note here that in the background, several important technological milestones had taken place. The World Wide Web had been invented in 1991, giving a new platform and use-case for personal publishing. The steady opening of the internet to commercial service providers was taking it from an academic research network to a network for the masses. There were two implications of these trends for fonts: screen display was becoming more important than printed display, and interoperability, the ability to share content reliably across computer networks, was all the more important.

In 1997, the font wars were unexpectedly resolved, through the announcement of a partnership between Adobe and Microsoft on a new multi-platform format, OpenType. OpenType fonts would be usable on both Mac and Windows operating systems and would support both PostScript and TrueType font formats. They would also be Unicode-compliant. OpenType fonts could now work essentially anywhere and give designers a reliable format to design around.[116]

In the story of multilingual computing, these developments raise two key questions: what led to this truce, and how did the outcome result in support for multilingual type? The answers are, in fact, intertwined. Here I begin presenting the story of OpenType that has *not* been chronicled within the typography industry.

As noted in the previous section, Unicode was in development between 1988, through the launch in 1991. In the final years before the launch, the initiative had drawn in both Microsoft and Apple employees. As talk simultaneously grew of developing a new font format, it was clear to

---

[114] *Ibid*.

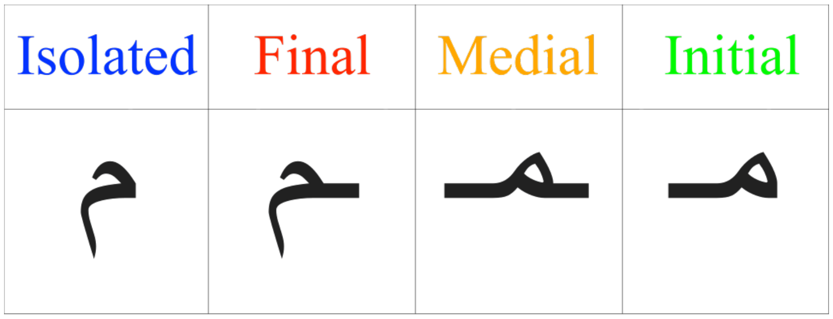
[115] Kinross, “The digital wave.”

[116] PeterCon, “OpenType Overview - Typography,” accessed June 29, 2022, https://docs.microsoft.com/en-us/typography/opentype/; Hitchcock, interview.

both companies that the format would need to support Unicode.[117] Though the initial release of TrueType did not yet support Unicode, it laid the groundwork for future formats to do so.

TrueType font files were structured to contain several data tables, each specifying different glyph parameters: widths, outline data, instructions for printers, and other such “metrics.” Amongst these was the character map (“CMAP”) which defined the relationship between characters and glyphs. Though Apple chose not to use Unicode mapping for the CMAP at this time, the structure was amenable to supporting it down the line.[118]

In the next version, Apple added an additional table to TrueType called the “MORPH” table. The MORPH table could essentially keep track of the characters and swap out glyphs as needed depending on the context, just as Unicode’s logical encoding system required the font layer to do.[119] To understand how this worked, here is an example from Arabic. In Arabic, glyphs would vary for letters depending on where in the word the letter appeared — at the start, in the middle, or at the end. Apple’s MORPH table would contain this information about the initial, medial, and ending graphical forms of a character and would work with backend software to display the correct glyph on a text editor. This format, TrueType GX, was then the first to make it technically possible to display complex digital text.[120] However, TrueType GX is largely forgotten to history as it gained limited adoption, not having the support of neither Microsoft nor Adobe.



Isolated	Final	Medial	Initial
م	ـم	ـمـ	مـ

Figure 18. Arabic Forms for the Letter “Mim” (the SVG Effect - Creative Commons)

Microsoft, now on its own, iterated on TrueType to create TrueType Open. Instead of a MORPH table, its approach was to add ‘positioning’ and ‘substitution’ tables (“GPOS” and “GSUB”), which would provide information on how glyphs should be rearranged. Microsoft used this format to

117 Hitchcock, interview.

118 *Ibid;* Lee Collins, interview with author, March 22, 2022.

119 *Ibid.*

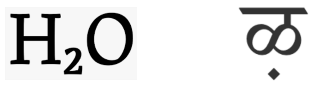
120 Apple employee, interview; Lee Collins, interview; Greg Hitchcock, interview.

release the Arabic version of Windows in 1995.[121] Seeing this, the Indian government released a paper soon after recommending its usage for Indic scripts.[122]

Despite the positive reception, TrueType Open was also short-lived. Industry pressure and the competition of the font wars led Adobe to seek a meeting with Microsoft in 1996.[123] Adobe's PostScript fonts did not have Unicode support, and though they printed with greater clarity than TrueType-based formats, screens were becoming the important medium for display because of the growth of the internet and the Web. TrueType had a sophisticated programming language built into the font, which was passed down to TrueType Open, and so it could readily produce high definition vector fonts for the screen, which Adobe wanted. For Microsoft, partnering with Adobe offered an opportunity to end the font wars — to finalize an interoperable font format and gain access, at the same time, to Adobe products as part of the deal.[124] And so, this meeting resulted in the announcement of the OpenType format only a year after the release of TrueType Open.

For Adobe, OpenType allowed many of the customizations they wanted to make advanced typography possible — subscripts and superscripts, ligatures. These features were previously relegated to "expert sets," font files that had to be purchased separately with the necessary glyphs.[125] This aspect of OpenType, "advanced Latin typography" is often hailed by typographers as its defining feature. But it was these same customization abilities that can be used to support multilingual communication — GPOS data could help position a Latin subscript but also a Devanagari nukta.[126] The turn of the millennium would bring about the search for the next billion users, and Microsoft now had the tools to reach out to them.

Figure 19. Subscript and Nukta



[121] Hitchcock, interview; Paul Nelson, interview with author, March 24, 2022.

[122] Shrinath Shanbhag, interview with author, March 28, 2022; S. P. Mudur et al., "An Architecture for the Shaping of Indic Texts," *Computers & Graphics* 23, no. 1 (February 1, 1999): 7–24, https://doi.org/10.1016/S0097-8493(98)00113-7.

[123] Hitchcock, interview.

[124] *Ibid;* PeterCon, "OpenType Overview - Typography."

[125] Hitchcock, interview.

[126] Constable, interview. Amelie Bonet, interview with author, July 21, 2021.

Rendering support

There was one final piece was necessary to make Unicode and OpenType workable; this was the "rendering engine." Because the Unicode Standard only provided codepoints and the OpenType format only provided tables of glyph data, another layer of software was required to assemble these pieces and tell a software application what to show. This task was performed by the rendering engine, also known as a "shaping engine."

A user would type letters on a keyboard that had keys mapped to Unicode code sequences. The rendering engine would then take the codepoints and find the relevant glyphs with the instructions on how to display them in the OpenType font file. The engine would then implement the necessary positioning and substitutions (following the GPOS and GSUB tables) and spit out the correct text in the application (e.g. a word processor, email client, or web browser).[127]

The engine was not only responsible for selecting the right glyphs and positions, but also for determining "fallbacks" — which glyph to show if the preferred one did not exist in the font. Especially for Indic scripts where many possible character combinations existed, a font designer would rarely draw all 300+ possibilities. If no ligature is available, for example, then the engine might search instead for the non-ligated form to show, instead of displaying a blank or square where the glyph was meant to go.

Finally, the rendering engine was also responsible for text-editing functions, such as determining "caret placement," or how the on-screen cursor travels across letters; in what order glyphs were backspaced and selected; and where to insert line breaks within words.[128] These functions depended on how the rendering engine defined "clusters" - roughly approximated to letters in alphabetic writing systems and syllables in alpha-syllabic/abugida writing systems. Cluster boundaries were also important for advanced natural language processing, such as text search and sorting.[129]

Rendering engines, at least for OpenType fonts, were thus required to be fairly complicated and comprehensive pieces of software with embedded knowledge of the encoding system, the font file, and the structure of the language itself. In Apple's font technology, much of the language-specific rendering information was programmed into the font itself, as in the MORPH table, rather than left to the engine. This had certain advantages — namely, fewer points of failure as various technologies attempted to coordinate together. But that same concentration, rather than distribution, of responsibilities meant that the font designer needed to acquire supplemental

[127] Each application would also need to be configured to work with the rendering engine. This configuration was called "layout support."

[128] For scripts where spaces between words may not be standard.

[129] F. Avery Bishop, David C. Brown, David M. Meltzer, "Supporting Multilanguage Text Layout and Complex Scripts with Windows NT 5.0," *Microsoft Systems Journal,* November 1999. https://web.archive.org/web/19990828165906/http://www.microsoft.com/MSJ/1198/MULTILANG/multilangtop.htm

linguistic and engineering knowledge, beyond only design, to prepare Apple fonts.[130] Apple's decision to build smart fonts meant it could therefore maintain a relatively simpler engine, which it called Core Text.

For Indic script users, OpenType's design was more relevant than TrueType's, as Windows had a deeper global market penetration than Apple. When OpenType was first being developed, Microsoft began by hardcoding the rendering logic for Arabic straight into the operating system.[131] This was their first experience of working with a complex script, in 1995. "Hard-coding" meant the code in the operating system was very Arabic-specific — how to parse the OpenType font tables, how cluster logic worked. By the release of Windows 2000, however, Microsoft had opted for a different approach of developing separate engines for "script families" that were perceived to have similar logic. These script specific engines, e.g. "Arabic" or "Indic," would plug into a more general text renderer called Uniscribe.[132]

Uniscribe support for Indic expanded rapidly. The Devanagari script was included in Windows 2000, followed by Gujarati, Gurmukhi, Kannada, Telugu, and Thaana in Windows XP, released the following year. Windows XP SP2, also released in 2001, nominally included support for Bangla and Malayalam, covering the majority of South Asia's language communities.[133] It appeared as though, by 2001, all of the pieces needed to create, send, and display Indic scripts between computers was in place.

However, as a study by Lancaster University reported in 2007, Unicode use in South Asian documents was extremely limited. Using data the researchers had scraped from the web between 2000 and 2003, they found that though there was a great deal of text in Indic scripts being produced online, they were in "a bewildering variety of fonts and formats."[134] Other essays in the early 2000s pointed to the same problem: Indic pages were not showing up on Google as the underlying encodings of Indic webpages were not in Unicode, which was what the Google engine utilized.[135] The search engine couldn't therefore match strings.

Why was this the case? To start, Unicode adoption took several years, filtering through operating systems, applications, and the Web. Unicode veterans tended to mark version 3, released in

[130] Hitchcock, interview; Apple employee, interview.

[131] Hitchcock, interview; Nelson, interview; Andrew Glass, interview with author, October 29, 2021.

[132] *Ibid.* John Hudson, "Making Fonts for the Universal Shaping Engine," http://tiro.com/John/Universal_Shaping_Engine_TYPOLabs.pdf.

[133] Michael S. Kaplan, "Script and Font Support in Windows," October 31, 2007, http://archives.miloush.net/michkap/archive/2007/10/31/5800258.html.

[134] Andrew Hardie, "From Legacy Encodings to Unicode: The Graphical and Logical Principles in the Scripts of South Asia," *Language Resources and Evaluation* 41, no. 1 (2007): 1–25.

[135] Mahesh, B. G. "From NRIs to Indians to Global Citizens: Evolution of the Indian Presence on the Internet" in *NetCh@kra: 15 years of Internet in India.* 2011.

1999, as the milestone where the Standard had covered all national and “commercially viable” scripts.[136] Unicode was also incorporated in all Microsoft software at that point.[137]

Another milestone was in 2007, when Google reported that Unicode encodings had finally overtaken ASCII in use on webpages.[138] Indeed, the oft-cited essay by Science and Technology Studies scholars Pargman and Perle, “ASCII Imperialism,” was published as late as 2007 and warned of the biases embedded in the American standard ASCII.[139] Pargman and Perle pointed towards Unicode as an improvement over ASCII, but were uncertain it would catch on.

But as the Lancaster study and Paolillo pointed out, the lackluster adoption of Unicode for Indic scripts had more to do with the lack of rendering support for complex scripts rather than Unicode’s generally slow adoption.[140] Though Windows had begun officially supporting Bangla in Windows XP SP2, it had yet to release a Bangla font and was still refining the rendering engine.

As previously discussed, the rendering engine was responsible for converting Unicode codepoints (or sequences of codepoints) into the appropriate glyphs. For Indic scripts, this required processing, but also *defining* various combinations of characters and control characters, using the *zwj*, *zwnj*, and *virama*. In some cases, Unicode provided guidance in the Standard. It would also put certain explanations on its online FAQ pages. But in the many edge cases that arose, the job of defining a sequence to portray a certain glyph fell to the maintainers of the rendering engine.

Though Microsoft’s Indic rendering engine was released with many scripts by 2001, it would take several years for all of the rendering logic to be worked through, at times requiring an upstream decision by Unicode. It is in that period that the events of the rest of this dissertation take place.

In some ways, this outcome of slow Indic script support can be seen as the result of Unicode’s early design decisions. By structuring the encoding layer as a set of puzzle pieces that other technologies needed to assemble for Indic scripts, responsibility for proper *display* moved up the technical stack. Gaps and bugs became more likely.

The Lancaster University study had noted that Indic font designers had taken an alternative path, of building “graphical encodings” that merged the encoding layer and the font layer.[141] These had a unique codepoint was assigned to each possible grapheme of a script, much like the TRON

---

[136] Whistler, interview.

[137] *Ibid*. Constable, interview.

[138] “Moving to Unicode 5.1,” *Official Google Blog* (blog), accessed June 29, 2022, https://googleblog.blogspot.com/2008/05/moving-to-unicode-51.html.

[139] Pargman D, Palme J. "ASCII imperialism” in *Standards and Their Stories: How Quantifying, Classifying, and Formalizing Practices Shape Everyday Life.* Ithaca: Cornell University Press; 2009. p. 177–99.

[140] Hardie, 2007; John C. Paolillo, “Language, the Internet and Access: Do We Recognize All the Issues?,” 2010, http://www.efnil.org/documents/conference-publications/thessaloniki-2010/language-languages-and-new-technologies/08-John-C-Paolillo.pdf.

[141] Hardie, 2007.

standard had started to do with East Asian scripts. But these graphical encodings for Indic scripts were also unstandardized, meaning every font developer followed their own set of mappings, reverting to a pre-Unicode era. As one designer said, this solved the "primary challenge of getting text to look right on screen," which Unicode had punted on.[142] But because neither the encoding nor the font were standardized, the typist data could not be transmitted or exchanged. For this reason, these fonts were known as "vendor-specific fonts" or "proprietary fonts." The fonts would typically be bundled with a keyboard and word processing software and sold as a full-suite of tools for Indic digital typesetting.[143] They fulfilled the needs of typesetting, but faltered when transferred to the Web. Only a user who also owned the same vendor-specific font package could view the webpage accurately. While Latin scripts often had the same problem of a font not displaying if not installed on one's own computer, in the complex script case, the text would appear completely broken, even with the same underlying encoding standard was used.

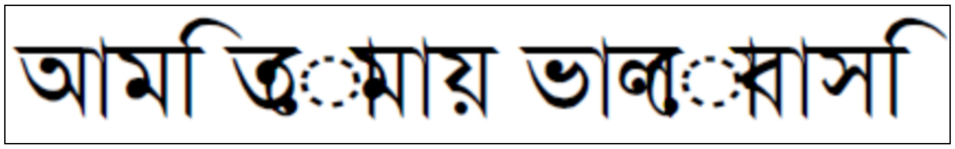


Figure 20. Example of Broken Bangla Font
(from https://github.com/dompdf/dompdf/issues/2627)

This was the true cost of Unicode's design. Its orientation around Latin and East Asian scripts — the most important and established markets at the time of Unicode's founding — led to policies such as "Characters not Glyphs." As a result, downstream tools such as clever font formats and complex rendering engines needed to be developed and refined to ultimately support Indic scripts. When those tools were inadequate or slow to arrive, swaths of Indic script internet users would adopt alternate schemes — schemes that displayed text correctly, but which hindered their ability to participate in the global internet. Proprietary fonts could not be easily transmitted across wires or indexed in search engines.

For some, the lack of interconnection was acceptable. But as the next chapter shows, a vanguard of South Asian internet users *were* longing for connection, and it would be these users that would seek to mold text standards to their needs and fulfill the software gaps that were lacking. Chapter 2 begins where this chapter leaves off, showing how independent Bangla software hobbyists would grapple with the fragile multilingual computing stack that had been developed in the West.

[142] Liang Hai, interview with author, December 9, 2020.

[143] Hardie, 2007; Olocco, 2014.

Chapter 2: Building Bangla Software

On October 30, 2002, Sayamindu Dasgupta sent out the following email:

এটাই বোধহয় বিশ্বের
সর্বপ্রথম বাংলায় লেখা মেল
হ্যাঁ, হয়তো এর আগে
বেঙ্গালিনাক্স-কোর মেলিনংগ
লিস্টে কিছু মেলে বাংলা ছিল,
কিন্তু পুরোটা বাংলায়
লেখা বৈদুতিন পত্র বোধহয়
এটাই
ই-মেল এড্রেসগুলো বাংলায়
লিখতে পারলে আরো ভালো হত,
কিন্তু, বাংলা ডোমেন
বোধহয় এখনো কেউ বুক করেনি
কিন্তু, সবচেয়ে বড়
ব্যাপার হল যে এই মেলটা
লিখতে কোন মাইক্রোসফ্ট
প্রোগ্রাম ব্যবহার করতে
হয়নি
আশা করা যেতে পারে যে
পৃথিবীর প্রথম বাংলা
অপারেটিংগ সিস্টেম-ও মুক্ত
সফটওয়্যার হবে

It said,

> *"This is probably the world's first mail written in Bangla. Yes, maybe before there was some Bangla in something someone posted on the mailing list, but this is probably the first that is completely in Bangla. If we could write the email addresses in Bangla that would be better, but probably no one has made Bangla domains. But, the biggest thing is that writing this email did not take any Microsoft programs. We can hope that the world's first Bangla operating system will also be free software."*[144]

Sayamindu was a seventeen year old high school student in Kolkata, India. In the past month, he had designed a Bangla font, found a working keyboard, and figured out how to configure an email client so it could send and receive Bangla messages. Now he was sending instructions to others on how to do the same, and celebrating his set up by sending what he believed was the first full email written in the Bangla script.

[144] Dasgupta, Sayamindu, "চিঠি ["letter"]" Email, October 30, 2002. Translation by the author.

Indeed it may have been. As we saw in the previous chapter, the tools that made a multilingual internet possible were just beginning to be implemented. The Unicode Standard had been released; the OpenType format was announced; rendering and layout software to interpret them were beginning to come out. But major firms like Microsoft had not yet actually released Bangla editions of their software. Though Hindi and Tamil Windows operating systems were available starting in 2000, the Bangla version was set for 2003. In the meantime, however, there were many keen internet users like Sayamindu who wanted to send messages in their scripts across the internet. Though the pieces to make this possible were available, it took technical savvy to assemble them correctly. It was hard to imagine a grandmother sitting at her computer and knowing how to turn the gibberish she received in her email into beautiful Bangla text. There was no out-of-the-box solution to local language computing yet.

This chapter tells the story of the independent hobbyists who began working on Bangla language software when no one else would. Individuals like Sayamindu would use the Web to research the latest standards and technologies that were emerging to support local language computing. They were not only interested in tools that would help them write Bangla language documents on their personal computers — the proprietary fonts mentioned at the end of Chapter 1 would allow them to do so —rather, they were interested in building fonts and keyboards that prescribed to new universal standards, ones they believed would be widely adopted and would allow them to participate in the global discussions occurring on the Internet. These individuals would form social groups — virtual communities — within which they would organize to build and advocate for Unicode and OpenType-compliant technologies.

At the same time, these individuals would be working mostly with free and open source software — software that was free to use, modify, and redistribute, and which found a natural ideological opposition to major private technologies companies like Microsoft. As Sayamindu wrote at the close of his email, a major aspect of his victory was not relying upon any Microsoft programs to send the email in Bangla script. There was an innate tension in despising Microsoft while promoting the font format it created, the only one that made it possible for Bangla script to be transmitted across the internet to any machine *and* show up correctly on the receiving screen.

Open source hobbyists groups such as these existed were not confined to South Asia — there are similar examples throughout Asia, Europe, Africa, and Latin America.[145] This chapter highlights their role in producing early versions of local language software, such as fonts, keyboards, and localized interfaces. While open source enthusiasts are often positioned in opposition to Microsoft, the case study in this dissertation eventually illustrates communication between the two parties, particularly over open standards that affected both. The relations between open source hobbyists and Microsoft however begin, in this chapter, from a place of suspicion and begrudging reliance.

I proceed with a brief history of free and open source software (FOSS), then present a series of case studies of three FOSS Bangla computing projects. I show how these hobbyists groups (*Free Bangla Fonts, Bengalinux,* and *Indic-computing)* played an important role in building a

[145] Hossain, Anushah. "Regional Open Source Software Communities: The View from Dhaka, Bangladesh" June 2021.

multilingual internet. They represented the vanguard in South Asian computing, recognizing the importance of emerging text standards before their national governments and academic institutions.

I then contextualize the activities and motivations of the members of these groups within the recent histories of computing and internet policy in India and Bangladesh. I show how heterogeneity emerges on either side of the border, in part because of each national government's role in promulgating information technologies — a rich national technology agenda leads to more resources but also a greater surface area for criticism, in contrast to a relative vacuum of policies in Bangladesh. I end the chapter by showing how the issues with Unicode's encoding of *khanda ta* are received by these open source communities. When issues like *khanda ta* appeared on their radars, they approached them from a technical mindset: a bug with the Unicode Standard that needed fixing.

Free and Open Source Software (FOSS)

Sayamindu was posting his email in 2002 to two mailing lists, the freebangfonts listserv and another called bengalinux. These were virtual communities set up that year to work towards the goal at the close of the email: building a Bangla language desktop.

The members of these groups were keen on building with *free software*. What was free software? It was "software that respects users' freedom and community…it means that the users have the freedom to run, copy, distribute, study, change and improve the software."[146] As the refrain went, it was software that was free in the manner of speech, not beer. A user could look at the code, tinker with it, and even re-distribute their adapted version. Unlike a Windows or Apple operating system, for example, free software empowered its users with a certain level of control and purported to exist outside of the market economy.

The free software movement began in 1984 when Richard Stallman published the GNU Manifesto.[147] Stallman had been an MIT engineer working at the artificial intelligence lab. His vision was of a full operating system and suite of applications that contained no proprietary components. The GNU system (GNU being a recursive acronym for "GNU's Not Unix!") would essentially be a free version of AT&T's Unix operating system. Though Stallman had developed nearly all the parts by 1991, it was the development and integration of the core component – the kernel – by a Finnish undergraduate student, Linus Torvalds, that completed the project that year. [148]

[146] "What Is Free Software? - GNU Project - Free Software Foundation," accessed June 29, 2022, https://www.gnu.org/philosophy/free-sw.en.html.

[147] Richard M. Stallman, *Free Software, Free Society: Selected Essays*, ed. Joshua Gay, 1st. ed (Boston, Mass: Free Software Foundation, 2002), 33.

[148] Stallman, 174.

Through the 1990s, the Linux project epitomized a new orientation towards producing goods: undirected co-production. There was virtually no difference between builder and user; each would identify and remedy the bugs that they could; all would benefit from differentiated skillsets of a large class of contributors. This model was characterized as "the Bazaar" by Eric Raymond, a self-proclaimed hacker, and contrasted against "the Cathedral" of corporations, where chosen experts doled out software improvements less efficiently and with lower quality.[149]

Stallman and Raymond would represent two approaches to the production of viewable, modifiable, distributable code. Where Stallman would emphasize the liberatory potential of free software for all of society (sometimes referencing it as *libre* software, as opposed to *gratuit*), Raymond wanted to emphasize its practical benefits of open source software for the software industry.[150]

Indeed, by the year 2000, Raymond's views had caught the attention of business people, policy makers, and scholars. The industry had been rocked just two years earlier when Netscape announced it would release the source code of its browser. The signal from a major technology company that it could be advantageous, rather than ludicrous, to make public the secret recipe for its product was revelatory.[151] Over the coming years, much ink would be spilled over whether other companies should 'open source' their products, how coordination worked in this leader-less model, and why people were freely contributing their high-skilled labour in the first place.[152] Over time, the distinction between *free* and *open source* software would be relegated to niche ingroup discussions, and be subsumed by contemporary terms such as FOSS (free and open source software).[153]

An idealized type emerged from these works. Eric Raymond, who fashioned himself hackerdom's ethnographer and tribal historian, wrote simply, "Hackers solve problems and build things, and they believe in freedom and voluntary mutual help."[154] He would also coin "Linus' law," which claimed that, with enough eyeballs, are bugs were shallow. He essentially espoused an ethic of self-interested cooperation. Though this "hacker ethic" has mostly been envisioned in the

---

[149] Eric Raymond, "The Cathedral and the Bazaar," accessed June 29, 2022, http://www.catb.org/~esr/writings/cathedral-bazaar/.

[150] Eric Raymond, "The Magic Cauldron," accessed June 29, 2022, http://www.catb.org/~esr/writings/cathedral-bazaar/magic-cauldron/.

[151] *Ibid.*

[152] See Steven Weber, *The Success of Open Source* (Cambridge, Mass.: Harvard University Press, 2005); Christopher M. Kelty, *Two Bits: The Cultural Significance of Free Software*, Illustrated edition (Durham: Duke University Press Books, 2008); Yochai Benkler, "Coase's Penguin, or, Linux and 'The Nature of the Firm,'" *The Yale Law Journal* 112, no. 3 (2002): 369–446, https://doi.org/10.2307/1562247.

[153] It is notable, however, that Sayamindu used the Bangla word *mukti* with respect to software, referencing freedom rather than no-cost.

[154] Eric Raymond, "How To Become A Hacker," accessed June 29, 2022, http://www.catb.org/esr/faqs/hacker-howto.html.

literature as a Western archetype, much of it rings true for the South Asian actors presented here.[155]

Sayamindu Dasgupta's trajectory tracked those of many self-proclaimed hackers in the early 2000s. He had gotten a computer at home in 2000 or 2001, he guessed: "I don't think I'm exaggerating when I say, getting a computer at home and starting to tinker with it was a lifechanger for me."[156] The broadband was spotty at home, but he still found going online a "fascinating experience." His journey into open source was also somewhat accidental. Every year he would go to the Kolkata book fair, and that year he happened to pick up a book about Linux there. It wasn't easy to figure out how to get the modem to work when he installed Linux at home, and that's when he came across the local Linux user group. Linux user groups had been forming around the world for this express need: trouble-shooting the somewhat finicky systems, much of which still lacked documentation. Though Sayamindu figured out how to solve his problem himself, he realized that others were facing the same challenges. So he began writing up HOW-TO documents – his first contributions to the world of open source.[157]

From there, he was joining several communities, meeting both in-person and online, on language and open source. He met in person with the Kolkata Linux User Group (ilug-cal) and found the bengalinux and banglapenguin groups online. These last two groups, in particular, were interested in building a Linux interface in Bangla. As they chatted it became clear, however, that before they could begin any translation, they needed to have a working Bangla font that was free to use and distribute. With that goal in mind, Sayamindu volunteered to create the Free Bangla Fonts (freebangfonts) project.[158] The first font he created for freebangfonts would be what he would use in the fully-Bangla email one month later.

Free Bangla Fonts

Sayamindu was no artist, but he was comfortable with the ways of open source projects – borrowing and repurposing from others as needed, and paying it forward. It was a strikingly different model than commercial software at the time, which was bent on filing patents, crushing or swallowing up the competition, and locking in customers to proprietary technology stacks.[159] Instead, free and open source software (FOSS) followed what Yochai Benkler calls a "commons-based peer production" model, in which individuals coordinated with each other on large-scale, complex tasks towards a common good.[160] Strikingly, there was little formal direction behind

[155] Kelty, *Two Bits;* E. Gabriella Coleman, Coding Freedom: The Ethics and Aesthetics of Hacking (Princeton: Princeton University Press, 2012).

[156] Dasgupta, Sayamindu, interview with author, April 17, 2020.

[157] Dasgupta, Sayamindu, interview with author, July 3, 2019.

[158] Dasgupta, interview, April 17, 2020.

[159] See Microsoft's leaked "Halloween documents." accessed June 29, 2022, http://www.catb.org/~esr/halloween/.

[160] Benkler, 2002.

these projects; self-interested individuals could proceed on their own, so long as previously-completed building blocks were well-documented, modular, and publicly available.

Thankfully, some earlier building blocks were there for Sayamindu to grab when he began the Free Bangla Fonts project. He found an open source font that seemed to be drawn by a physics professor in Kolkata, Professor Palash Baran Pal. Professor Pal was at the Saha Institute of Physics, and had developed a package, called Bangtex, for the popular document preparation system, Latex.[161] Latex built upon the Tex typesetting software, which had roots in both digital typography and open source software. Tex had been developed by Donald Knuth in 1978, after he got frustrated with the new phototypesetting technology that was being used to typesetting the latest edition of his textbook, *The Art of Computer Programming*.[162] Tex was the result of Knuth's tinkering with novel digital typesetting systems, though it remained relegated to mostly scientific settings rather than reaching PostScript or OpenType's widespread adoption.

Professor Pal had developed the Bangtex typesetting system that allowed the user to develop multilingual documents including Bangla text. The font glyphs were available as a separate file within the package and were licensed as a free software font. Sayamindu could open up the font, copy the glyphs, and paste into other programs to format them as Unicode fonts. The software license made the contents free to use, modify, and distribute, so Sayamindu never had to get in touch with Professor Pal at all. "I was just kind of intimidated," he recalled in an interview. The two met briefly many years later when Pal was a featured speaker at a science camp Sayamindu was attending, but even then their interactions involved other topics ("radio carbon or something").[163]

Pal's glyphs served as the artwork, but there was still a sizable programming task involved in developing a Unicode-compliant, OpenType font. Glyphs needed to be matched to Unicode codepoints. Information about positioning and substitutions needed to be filled into OpenType tables. It all needed to be done in a way that Microsoft's rendering software could process. There were two tools around to help with this process: an open source font design tool called PfaEdit, later renamed FontForge, and Microsoft's own software, called the Visual OpenType Layout Tool, or VOLT.

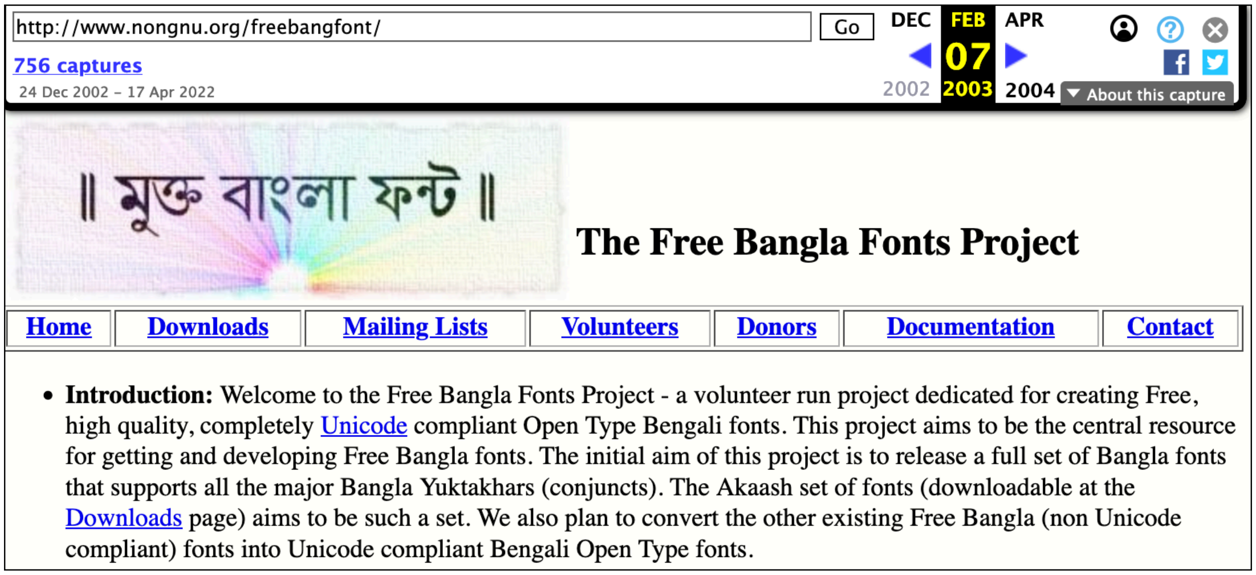
Sayamindu worked mostly in PfaEdit to build his first font, which he called *Akaash*, or "sky." On October 15, 2002, he posted the font on the freebangfonts homepage and shared it with his new colleagues in the Bengalinux project.[164]

[161] Palash B. Pal, "Bangtex," 2001, http://www.saha.ac.in/theory/palashbaran.pal/bangtex/bangtex.html.

[162] Donald E. Knuth, *Digital Typography*, Reissue edition (Stanford, Calif: Center for the Study of Language and Inf, 1998). 5.

[163] Dasgupta, interview, April 17, 2020.

[164] Dasgupta, Sayamindu, "[Bengalinux-core] Akaash-0.5" October 15, 2002.



The Free Bangla Fonts Project

- **Introduction:** Welcome to the Free Bangla Fonts Project - a volunteer run project dedicated for creating Free, high quality, completely Unicode compliant Open Type Bengali fonts. This project aims to be the central resource for getting and developing Free Bangla fonts. The initial aim of this project is to release a full set of Bangla fonts that supports all the major Bangla Yuktakhars (conjuncts). The Akaash set of fonts (downloadable at the Downloads page) aims to be such a set. We also plan to convert the other existing Free Bangla (non Unicode compliant) fonts into Unicode compliant Bengali Open Type fonts.

Figure 21. Snapshot of the Free Bangla Fonts Original Website (Wayback Machine)

Bengalinux

Just a few months prior, Taneem Ahmed had begun thinking about starting the Bengalinux mailing list. He too had been lucky to have been exposed to computers in Bangladesh at a young age.[165] His uncle had been working abroad and introduced Taneem to them in the mid-90s. He was also an early supporter of the first Bangladeshi magazine on computers, *Computer Jagat*, or "Computer World." Computers were still mostly conceived in the context of computation – used by banks and NGOs – or increasingly, desktop publishing. The idea of computing for communications, for *connection*, didn't resound to Taneem until he immigrated to North America to start his engineering degree in 1996. There, he learned about open source software, and about dial-up internet. He worked in the research lab of Professor Steve Mann, one of the early proponents of wearable technology. Taneem had been part of his lab during the creation of the traveling art installation, "SeatSale: Seating Made Simple."[166] This chair was free to use, so long as you had a paying subscription; once your subscription expired, jagged spikes would rise up from the seat. As Mann later described in an article, "The word "free" is used with jest, in the sense that although there is zero monetary cost… the true cost is the loss of privacy and the loss of freedom to sit without asking for permission from a global Seating Services™ provider."[167]

It wasn't until Taneem arrived in Silicon Valley after graduating that he began working earnestly on building open source software. He started by pushing a change to the GNU C library, a

[165] Ahmed, Taneem, interview with author, February 16, 2020.

[166] *Ibid.*

[167] Steve Mann, "Existential Technology: Wearable Computing Is Not the Real Issue!," *Leonardo* 36, no. 1 (February 2003): 19–25, https://doi.org/10.1162/002409403321152239.

codebase that was used by multiple Linux distributions, to make it possible to create a Bangla locale.[168] Though the dot-com bubble had recently burst, there was still a palpable energy and optimism in the region. "You suddenly start seeing the huge impact of the internet and the web. You realize the way people are sharing and storing information. It's not about libraries and books anymore," he recalled.[169] He felt anxiety about what communications scholars have termed the bias of communication: unless literature was translated into hardy mediums – digital bytes rather than fragile paper books – then their content might be lost to future generations.[170] There was so much Bengali literature to save; though he felt sure that "Rabindranath and Nazrul will survive," he felt that "other poets and songwriters – their work will be lost. If it's not online, you don't exist."[171] This was an aspect of multilingual computing that sometimes becomes lost; digital fonts are needed not only for contemporary communication, but to make it possible to digitize records of the past.

It was a similar thought that had motivated Deepayan Sarkar to start his Bengali archive project. In fall 2002, Deepayan was a PhD student in statistics at the University of Wisconsin-Madison. He intended for his literature archive to be similar to Project Gutenberg – records of public domain Bengali literature, stored and viewable on the internet.[172]

At first glance, the project seemed easy enough: "this was just a matter of typing up whatever I had the time and inclination for," he wrote in a memoir.[173] But of course, in practice there were many technical challenges, the foremost of which was having a reliable, usable font in which the text would appear. Like Sayamindu, Deepayan had searched for previous examples, coming across proprietary fonts that a site visitor might not have loaded on their computer and Unicode fonts that were not free to use. So he too began developing his own font, *Likhan (*"writing"). *Likhan* was drawn in PfaEdit and engineered to be an OpenType font in Microsoft's VOLT.[174] The reliance on Microsoft's tools at this point was significant; there was no way to fully work in an open source software stack and build a high-quality font (though Sayamindu had given it a try with *Akaash*).

Taneem, Deepayan, and Sayamindu had found each other simply by Googling for others working on Bangla computing. The group found another Indian Bengali expat studying in the US, Kaushik Ghose, who had built a text editor called *Lekho* ("Write"), though it was not yet Unicode-compatible. Their efforts were all proceeding independently, but Taneem saw the common goal, of creating a full operating system in Bangla. He reached out to each person and wrote, "Instead

---

[168] Ahmed, interview.

[169] *Ibid.*

[170] Harold A. Innis, *The Bias of Communication* (Toronto: University of Toronto Press, Scholarly Publishing Division, 1999).

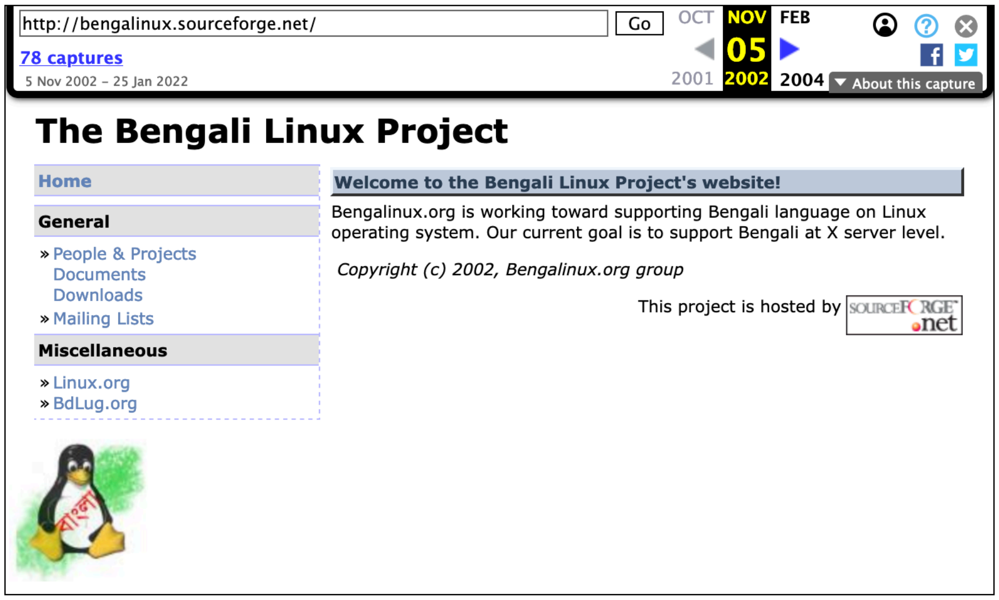
[171] Ahmed, interview.

[172] Sarkar, Deepayan, interview with author, June 16, 2021.

[173] Deepayan Sarkar, "Bangla Computing and I," 2020, https://deepayan.github.io/misc/bangla-computing.

[174] *Ibid.*

of me writing an editor, and you writing a font, maybe we do this together in a more organized way."[175] What would it take to support Bangla properly on Linux? Could they put their different pieces together? He officially started the bengalinux mailing list in September 2002 and put up a webpage hosted on SourceForge.net with their tools and intentions. The existing projects would each maintain their own mailing lists and names – Taneem was not trying to "grab other people's work under some umbrella" – but would try to work together.[176] This too epitomized the open source ethic: light touch coordination to build complex systems for the public.



The Bengali Linux Project

Welcome to the Bengali Linux Project's website!

Bengalinux.org is working toward supporting Bengali language on Linux operating system. Our current goal is to support Bengali at X server level.

Copyright (c) 2002, Bengalinux.org group

This project is hosted by SOURCEFORGE.net

Figure 22. Snapshot of Bengalinux Website (Wayback Machine)

Within the first few months, the group had grown in count to seven.[177] Some had common pathways into caring about language technology. Like most city-dwellers in South Asia, they had taken English-medium classes throughout grade school and knew how to type in Latin letters. But like all members of that generation, they had also grown up in the shadow of the Bangla language movement and the Bangladesh Liberation War, even across the border in India. In 2001, the United Nations had recognized the bloodshed over the Bangla language by commemorating February 21st of every year as International Mother Tongue Day. February 21st was a watershed date from the year 1952, when Pakistani police shot into a crowd of students on Dhaka University's campus. The students had been doing daily processions in protest of the national government's policy to recognize only Urdu as Pakistan's national language, despite

[175] Ahmed, interview.

[176] *Ibid.*

[177] "The Bengali Linux Project." http://web.archive.org/web/20021212083656/http://bengalinux.sourceforge.net/projects/

Bangla being the majority language of East Pakistan. After the death of four students, February 21st, or “Ekushey February” would be honored every year as Martyr’s day.

For Sayamindu, he had grown up hearing about the importance of one’s mother tongue, but had also been exposed to scholarly communities recognizing the importance and beauty of Bangla growing up. His father had been a language scholar, publishing primers for French speakers to learn Bangla. His mother was a Comparative Literature professor at Jadavpur University in Kolkata. Between the early exposure to computers and the immersion in academic communities, he naturally found himself circling the question of how to write Bangla on computers.[178]

For others like Taneem, there was a much more personal tie to Bangladesh’s political history. His father had been a senior officer in the 1971 Liberation War, a nine-month guerilla war that eventually resulted in the creation of Bangladesh as an independent country. The decades afterward did not pass smoothly, however, as alternating claims for political power in the new country led to assassination and violence against perceived supporters of the opposing side. Taneem’s father was executed in 1991 as a casualty of this power struggle. It wasn’t a backstory Taneem spoke about often, but it was an important experience for him. He had his father’s letters, and he felt he had an important connection to Bangladesh. “I’m not a very cultural person - I don’t sing or write. I don’t think I’ve ever fully written a Bengali letter or email,” he told me. “But I’m a technical guy and I felt I should be able to help here.” With Bengalinux, his motivation was just to find people who could work together, and “just go ahead and do it.”[179]

What was it like to “go ahead” and build Bangla language technology in the early 2000s? It was possible but not easy. As Deepayan wrote on his original project homepage,

> *Hypothetically speaking, a Bengali `document' can be stored in several forms. For example, they might be just images, or perhaps PDF files. However, one of the aims of this project is to use standards that are* ***open, cross-platform, and widely recognized, as well as appropriate for the task at hand****. A few years ago, it might have been difficult to meet these criteria, and even now, the only platform on which this will work exactly as intended is Microsoft Windows. However, the standards that are used are indeed open and widely recognized, it is only the lack of implementation that prevents users of other systems from seeing the results as seamlessly.*[180]

This message spoke to the primary challenge of building local language technology in the early 2000s. The critical advancements had been made: industry-accepted standards that would theoretically support the world’s scripts. But the rest of the stack was lagging. Microsoft’s rendering engine for complex scripts, Uniscribe, was incorporated into Windows 2000 and

[178] Dasgupta, interview, April 17, 2020.

[179] Ahmed, interview.

[180] Sarkar, Deepayan, ”Bengali Literature Archive.” http://web.archive.org/web/20021208180443/http://www.stat.wisc.edu/~deepayan/Bengali/WebPage/bengali.html

Windows XP. Layout support was coming but uneven across Microsoft applications, as different teams worked independently on each application, from the word processor Microsoft Word, to the web browser Internet Explorer, to the mail client Outlook.[181]

At this time, in 2002, the only popular application that supported Bangla text layout – meaning the software knew how to deal with Unicode encodings and OpenType instructions, and turn them into properly displaying Bangla text – was Internet Explorer.[182] But this at least gave the Bangla computing folks a place to start. If they could develop a Unicode, OpenType-compliant font, then they could use Internet Explorer to see Bangla. After that, they could work on getting similar support in the open source applications they used, on Linux systems.

Like Sayamindu and Taneem, Deepayan preferred to work in Linux. Unfortunately for the group, there wasn't much coordination or standardization beyond the OpenType font format. Those using Linux or Apple systems, or even non-Microsoft applications like Adobe Photoshop, had to wait for or build their own rendering engines and layout technologies to get Indic scripts to work. And as the previous chapter discussed, these technologies were heavily informed and dependent upon how Unicode and OpenType were defined for a given script. Microsoft was planning to release the Bangla specification of OpenType in 2002, but as the hobbyists would note, it was still just a "spec, not even a standard" —- they had to trust that Microsoft would maintain it or that patents would not be evoked against it.[183] This state of affairs again pointed to the concentration of power in the software industry in the early 2000s. Everyone had to follow Microsoft, even those trying to work outside of it.

This was evident in the process for building OpenType fonts. The open source font design tool PfaEdit provided an interface to draw the artwork, but it didn't yet support OpenType layouts. For this, the font designers had to download Microsoft's own tool, which only worked on their Windows operating systems. Deepayan used a student rebate to purchase Windows for this purpose alone, using dual boot to run both Linux and Windows operating systems on his machine.[184] He would design most of the font on Linux using the open source tool PfaEdit, do the font engineering in VOLT on Windows, then switch back to PfaEdit to make minor changes – software that wouldn't destroy, at least, OpenType-engineered fonts.

As the previous section highlighted, after the font layer, the multilingual computing stack had several intertwining pieces that needed to be developed in concert with one another: the rendering engine, which provided instructions for how to read the OpenType tables inside the font, and the layout technology, which was embedded in individual applications and drew from the rendering engine to ultimately display the text.

[181] Paul Nelson, interview with author, March 24, 2022.

[182] Sarkar, Deepayan, "Bengali Literature Archive."

[183] Dasgupta, interview, April 17, 2020.

[184] Sarkar, interview.

Microsoft had its own proprietary rendering engine and layout technologies to display fonts. The bengalinux members would try to build fonts that would work with Microsoft's Uniscribe rendering engine and Internet Explorer's layout technology.

But for Linux, they had several more moving targets. "Linux" itself referred more precisely to the Linux kernel, the open source piece of software that was originally developed by Linus Torvalds and released in 1991. Every computer has a software "kernel" which speaks to the computer hardware, performing essential tasks like memory management. Different initiatives have used the open source Linux kernel over time, and built their own software on top of it to create independent operating systems and applications. When the Linux kernel was combined with software packages like a desktop environment, internet browser, and other utilities, it was called a "distribution," or "distro" for short. One of the most famous examples of these was Richard M. Stallman's GNU/Linux distribution. Other major distros using the Linux kernel included Fedora/Red Hat, Ubuntu, and Debian.

Though these distros are often referred to offhandledly as "Linux," in truth they were different operating systems with their own options of "desktop environments" – what the user saw when using their computer. Each desktop environments used a different Graphical User Interface (GUI) toolkit. Some Bengalinux members used the GNOME desktop environment, which used the GTK GUI toolkit. Others preferred the KDE desktop environment, which used the Qt GUI toolkit. The GUI toolkits, in turn, used different rendering and layout technologies for text. GTK used a separate open source library called Pango to render and lay out text. Qt had its own layout engine. Finally there were popular open source applications like Open Office that had entirely independent rendering and layout technologies.[185]

It is easy to get lost in the acronyms of the Linux universe and the various layers of interdependent technologies. But the important takeaway for Bangla computing developers was that there were several different targets for their fonts. They could design a Unicode, Open-Type compliant font that worked with Microsoft's rendering and layout technologies, but was still buggy in many versions of Linux due to different OpenType implementations in Pango or Qt.

Though the flexibility and modularity of Linux systems was the boon that allowed localization teams around the world to begin developing local language software before giants like Microsoft began supporting them, those same qualities meant the source software could be unstable and error-prone. This state of affairs would lead to trouble later on, including on the issue of *khanda ta*.

Despite the challenges, the motivated group pushed ahead to design imperfect, but workable, Bangla OpenType fonts. By mid-November, 2002, three free Bangla fonts had been published. The group was admittedly pleased.

[185] Sarkar, interview.

From: Kaushik Ghose
To: Free Fonts
Date: 11/10/2002 12:55:48 AM

So, does that make it 3 bangla OTFs now ? Mukti, Likhan and Aakash ?
And we had zero (0) 3 months ago, is that right ?
And they said bengalis were lazy.
I guess we've become corrupted...
-kg[186]

Broader networks

Through 2002, both Taneem and Sayamindu were working to grow the number of members in the group. On the Bangladeshi side, Taneem reached out to the Bangladesh Linux Users Group (*bdlug*). Though there were plenty of subscribers, the topics of discussion were focused on using Linux for networking purposes: routers, firewalls, email servers.[187] Most of the active members were working at different internet service providers. Taneem tried to appeal to the potential for alternative use-cases for Linux, such as building a Bangla desktop. Beyond recruiting Jamil Ahmed, who, after receiving help from Taneem multiple times with setting up Linux for his office job, joined the Bengalinux project, bdlug brought in only occasional support.[188]

Sayamindu found more luck on the Indian side, particularly as his interaction with the Kolkata Linux User Group grew (ilug-cal). It was through ilug-cal that Sayamindu learned of the Indic-Computing conference that was being planned for September 2002. The Indic Computing Conference was the first major event planned by the indic-computing mailing list that had started the year prior. The Indic Computing Conference brought together representatives from several regional initiatives focusing on building local language computing.

It promulgated the same goals as the indic-computing mailing list itself. As the public spiel said, "The main purpose of this workshop is to build a community of people working in the space of developing local language development tools, applications, and content, to better coordinate their ideas and approaches towards the future of indic-computing."[189] The agenda included talks on the experiences of practitioners, those using local language technologies; walkthroughs of how encodings, text display, and input methods worked; and perspectives from those conducting linguistic analysis.[190]

[186] Ghose, Kaushik, "re: [Freebangfont-devel] Mukti Set of Fonts" Email, November 10, 2002.

[187] Ahmed, interview.

[188] Jamil Ahmed, interview with author, September 2, 2020.

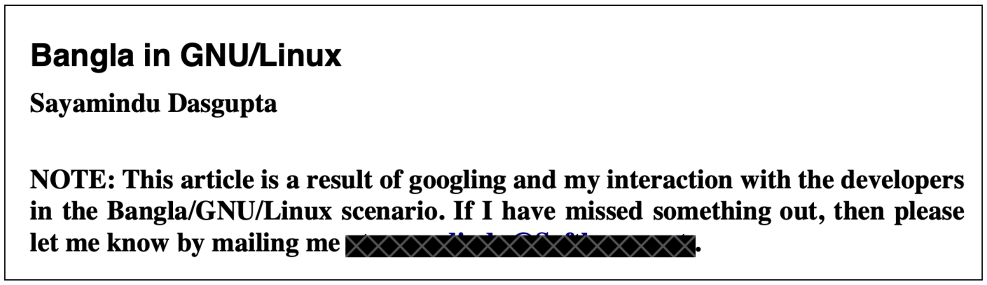
[189] Kotamkar, Ashish. "Indic Computing Workshop." Email, September 2, 2002.

[190] *Ibid.*

The indic-computing mailing list had been created in December 2001.[191] The people who created it were already involved together in another project and mailing list called IndLinux. IndLinux had much the same goals as Bengalinux - to create Indian language desktops in GNU/Linux.[192]

There were people in industry, however, who were also interested in guiding and supporting these efforts, such as Joseph Koshy whose day job was working in the private sector, at Hewlett-Packard Bangalore.[193] But in the evenings, Koshy was a regular contributor to the FreeBSD project, another open source Linux distro. Others like Guntupalli Karunakar or Tapan Parikh were current or recent graduates, keen on enabling local language computing, searching for an enticing project to work on.[194] Parikh, in particular, had travelled back to India for an internship in the middle of his doctoral program at the University of Washington, and had felt he had stumbled onto an issue important enough to change the direction of his research.[195] Koshy and Karunakar, along with a few colleagues, were interested in supporting collaboration between all of the people working in the language technology space. The indic-computing listserv was conceived to fulfill this purpose. Though their members would encourage the development free and open source software, the goal of this group would be to do information-sharing and agenda-setting; groups like IndLinux or Bengalinux could then go forth and build the actual software.

Sayamindu attended the September conference on behalf of the Bengalinux folks, and wrote a short whitepaper summarizing the state of Bangla computing to date.[196] He had been volunteered to attend by his Bangla computing compatriots. “I was incredibly scared. I was the youngest person in the group… they signed me up. I said I have my [high school] exams, I’ll try to make it.”[197]



Bangla in GNU/Linux

Sayamindu Dasgupta

NOTE: This article is a result of googling and my interaction with the developers in the Bangla/GNU/Linux scenario. If I have missed something out, then please let me know by mailing me [redacted].

[191] Karunakar, Guntupalli, personal communications to author, April 18, 2022.

[192] “IndLinuxSaga - IndLinux,” accessed June 29, 2022, https://www.indlinux.org/wiki/index.php/IndLinuxSaga.

[193] Karunakar, personal communications.

[194] Karunakar, personal communications.

[195] Parikh, Tapan. “[Indic-computing-devel] What are people working on?” Email, December 27, 2001.

[196] Dasgupta, Sayamindu. “Bangla in GNU/Linux.” *Indic-Computing Workshop. https://sourceforge.net/projects/indic-computing/files/workshop-proceedings/bangalore-september-2002/*

[197] Dasgupta, interview, July 3, 2019.

The conference seemed to accomplish its goals. The two-day workshop was where the group evolved from a virtual space to a real-life community that knew each other by face. It also successfully brought together folks from free software and private industry; in attendance were representatives from the Free Software Foundation and the Sarai Institute, as well as from Hewlett-Packard and Microsoft.[198] For Sayamindu, a strong believer in free software, the assortment of people at the conference puzzled him. "It was the late 90s. Microsoft was already the enemy. These organizers were sensible to invite people from Microsoft who seemed supportive, but it was weird to hang out with them."[199] The list of attendees also notably included a number of names who had indicated interest but could not attend, many of which included representatives from Unicode's core staff.[200] Bengalinux had mostly been working in isolation, with some communication with other open source hobbyist groups. But the Indic-computing folks were bringing in players with much more institutional authority — private sector leaders, government officials, academic researchers. The difference in participants represented a difference in goals, as well as a difference in the playing field between India and Bangladesh, which I discuss more in the section below.

Indic-Computing

The ethos of the indic-computing group was to encourage India's leaders to adopt the tools that would let India join the emerging global digital ecosystem, and thereby enrich itself. The communities they sought to bring together were stratified in two different ways. The first was the obvious splitting by different regional and language communities. Like the Bengalinux project that would form a year after for the Bangla language community, there were others beginning to fruit around Tegulu, Tamil, Marathi, Hindi, Kannada, and Malayalam.

But the indic-computing leadership was also aware of stratification by affiliation: there were those coming from academic or research backgrounds, who were working at universities like IIT Madras, Hyderabad, and Kanpur and at government agencies such as the National Center for Software Technology (NCST) or the Center for Development of Advanced Computing (CDAC). These academic groups were perceived as "mainly working in a closed way", and having had "years of experience in Indian computing area but small teams with no long term goals."[201]

These contrasted against the groups that were volunteer-based, such as Indigo or IndLinux.org. This second group was made up of initiatives that were part of the free software movement; they

[198] Noronha, Frederick, "[Indic-computing-users] IndicComputingNotes: Links, comments from overseas…" Email, September 20, 2002.

[199] Dasgupta, interview, July 3, 2019.

[200] Noronha, "[Indic-computing-users] IndicComputingNotes: Links, comments from overseas…" Email.

[201] Karunakar, Guntupalli, "[Indic-computing-standards] generic issues - first post." Email, December 7, 2001.

lacked experience, but had “clear long term goals and a big community of developers to depend on.”[202]

There was little existing coordination between the academic researchers and the free software folks. The lack of coordination was slowing the development of “future-proof” Indian language computing — software that abided by the open standards that the group’s leadership believed would structure all digital communications. Being able to typeset in an Indian language on a desktop was no longer enough. One needed to be able to *communicate* - be able to send that text in a document or email or webpage, and know with certainty that someone on the other side would be able to view it. This ability to communicate depended on common codes and common standards - this was the problem that Unicode was designed to solve. It was a contrast against the graphical or font-based encoding systems that had been profilerating throughout South Asia since the coming of desktop publishing. These proprietary systems had a small but important benefit – they were able to display the text appropriately, a problem that Unicode had only begun to solve. The OpenType format brought networked computers one step closer to being able to display Indic text properly, but in the early 2000s, its usefulness was still more idea than impact.

There were two pushes the indic-computing leadership was trying to make to change the status quo. For the FOSS groups working on building local language desktops, they advocated for more documentation and knowledge-sharing.[203] They were all contending with the same issues: interpreting the Unicode Standard and OpenType specifications to build a compliant font; hacking font development tools like PfaEdit to get them to work with Indic OpenType fonts; programming keyboards that could handle the multiple codepoints combinations that Unicode often assigned to represent a single Indic character; and lobbying applications and rendering libraries to bring in script-specific support.[204]

Greater documentation and communication could speed up the work of existing volunteers, and make it easier for new folks to dive in. As Karunakar wrote in an early agenda-setting email,

> *Most free software projects have started outsite India , so core teams composition is basically non indian, so although their project have support for western & eastern scripts, not much support is there for Indic scripts, since the team members are not knowledgebale enough abt our scripts. So we need to play our part there in helping them.*[205]

But for many of the “legacy” institutions working on Indic computing, the call was for them to build on open standards, and release software that was free to use and distribute. Many of these stakeholders were still resistant to Unicode or OpenType. Tapan Parikh wrote an impassioned plea on this topic in April 2002, in an email entitled, “New Blood - Forget the Establishment,”:

[202] *Ibid.*

[203] Karunakar, “[Indic-computing-standards] generic issues - first post.” Email.

[204] *Ibid; Proceedings of Indic-Computing Conference. https://sourceforge.net/projects/indic-computing/files/workshop-proceedings/bangalore-september-2002/*

[205] Karunakar, “[Indic-computing-standards] generic issues - first post.” Email.

> *All,*
>
> *I am usually very polite with other people's obserations and views. I feel I have to say this now because right now I am very frustrated with the kinds of things being discussed and proposed at these types of meetings, and am letting my emotions get the best of me. Sorry in advance.*
> *[We] were at a discussion at IIT-Powai where similar non-issues were re-hashed for hours. I wanted to yell. How long can we continue to talk and whine about technical non-issues? Are these people serious when they are basing all of their work on a font standard rather than a character standard? Can they not see the direction the rest of the world is going, with Open Type Fonts and Unicode? Do they feel we will somehow be using completely different and independent applications, operating systems, programming languages, libraries, APIs, etc? Do they not understand the power and benefits of interoperability?*
>
> *Will India always be this way?...*
>
> *As a technician it is hard to fight against these politics, vested interests, and ego wars. But we cannot allow it. We have to make sure technically-informed decisions are made. We must, it is our moral responsibility as technical leaders.*[206]

It was no accident that this call towards technical openness took on a particular moral and nationalist tenor – *"Will India always be this way?"* – one that was missing in the bengalinux group, despite similar appreciation of and adherence to open source principles. Over the past half century, India's technology sector had been a central focus of the national policy, leading to several decades of "indigenous computing" and bureaucratic mishaps. For the young generation of technicians, the inwardness and policies of indigenous technology had been tried and failed over the past half-century; they felt responsible for charting a new path. As Parikh wrote in his impassioned message, there was a "moral responsibility as technical leaders" to avoid repeating these mistakes.[207]

India's Indigenous Computing Policies

I pause here to provide a historical narrative of India's nationalist development policies, through the late 20th century. These policies are relevant in characterizing the milieu in which hobbyist groups such as Indic-computing and Bengalinux arose; these groups were not only emblems of the global open source software movement, but can be seen as reactions to their local contexts of development. This context can help us understand why the particular stance of open source hobbyists in this region is towards openness and interconnection with the West, in contrast to

[206] Parikh, Tapan, "[Indic-computing-standards] New Blood - Forget the Establishment" Email, April 11, 2002.

[207] *Ibid.*

protest and withdrawal[208] or a sense of subjugation to the West[209], as other case studies of FOSS communities in the Global South have shown.

When India achieved independence in 1947, Prime Minister Jawaharlal Nehru found a fractured, impoverished country on his hands. A wealthy, Western-educated lawyer with an interest in Soviet-style central planning, Nehru sought out to modernize India. He cultivated relationships with key entrepreneurs, industrialists, scientists, and academics, empowering them to build out space, nuclear, and electronics research institutes.[210] The first five Indian Institutes of Technology (IITs) were set up between 1951-61. (These would be supplemented by 18 more between 1994 and 2016), as well as the first two Indian Institutes for Management (IIMs) in 1961.[211] Though the other outcomes of a *self-reliance*—inspired industrial policy are considered to have had adverse outcomes on India's growth, the establishment of these foundational research and educational institutions have been hailed by scholars as a "bright spot" in Nehru's plans.[212] Indeed, in the history of multilingual computing, the IITs would become the grounds where ISCII and other Indian language technologies would begin being developed.

The next era in India's technology trajectory began in 1966, when the recently created Electronics Committee released a report laying out three goals for "indigenous industry":

1. India should participate in the ownership and control of foreign computer subsidiaries in the country
2. Wholly Indian producers should come to satisfy most of the country's computer needs, with foreign units only temporarily supplying exotic technologies and large systems
3. India should participate in the manufacturing of advanced systems available internationally[213]

In sum, India should own the foreign subsidiaries that lay inside its borders, have home-grown producers of computers, and participate in the international supply chain. At the time, there were two international computer companies with sales and manufacturing activities in India: the American company IBM and British company ICL (international Computers Limited). Over the next decade, the Indian government would pressure both companies to share ownership of its

[208] Anita Chan, "Coding Free Software, Coding Free States: Free Software Legislation and the Politics of Code in Peru," *Anthropological Quarterly* 77, no. 3 (2004): 531–45.

[209] Yuri Takhteyev, Coding Places: Software Practice in a South American City (Cambridge, Mass: The MIT Press, 2012).

[210] Ramesh Subramanian, "India and Information Technology: A Historical & Critical Perspective," *Journal of Global Information Technology Management* 9, no. 4 (October 1, 2006): 28–46, https://doi.org/10.1080/1097198X.2006.10856431.

[211] *Ibid.*

[212] Subramanian, 2006.

[213] *Ibid;* Joseph M. Grieco, "Between Dependency and Autonomy: India's Experience with the International Computer Industry," *International Organization* 36, no. 3 (1982): 609–32.

local activities with Indian nationals. ICL would make concessions after intense negotiations, but IBM would opt to instead withdraw from the country, which it would do in 1977.[214]

At the same time, the Indian government set up the Electronics Corporation of India Limited (ECIL) to advance indigenous manufacturing of computers, in pursuit of its second goal from the 1966 Electronics Committee report.[215] Part of the issue with IBM had been their policy of only *gradual upgrades* for developing countries. While the rest of the world was using advanced IBM 360s and 370s, the Indian government was only being offered IBM 1401s.[216] It was difficult for India to locally build these large mainframes. In the 1970s, the policy instead shifted towards buying a limited number of large systems from Britain's ICL, while simultaneously shifting towards minicomputer architectures that could be designed and assembled locally ("Indigenous assembly").[217] However, these efforts did not find much success. As Subramanian writes,

> *Despite IBM's faults, its presence had brought in ideas and processes for greater efficiency, and a talented, well trained and quality-minded sales and maintenance force. At the time of IBM's departure, India's home-grown efforts at developing computers was not producing much results…ECIL developed its own non-standard software which it could hardly sell. India was thus confined to the dark ages of computer development.*[218]

Where the establishment of research centers by Nehru in the 1950s spurred innovation and research, the indigenous computing policies of the 1960s and 70s seemed to stifle it in the private sector.

By the late 1970s many international competitors to giants IBM and ICL had emerged, particularly as mini- and micro-computer architectures became able to perform the same functions as mainframes at a cheaper cost.[219] These competitors were well-suited for Indian markets. However, these international imports were heavily gatekept by the Department of Electronics, created in 1970. If any organization wanted an advanced computer, they had to seek permission from the government agency, which would decide which computer the organization would be allowed to import. Users needed to show they had a different use case than what ECIL computers could provide. ECIL computers were still very costly and had long delivery delays, on the order of 1-2 years — by the time they arrived, they were already outdated and perhaps missed the initial purpose for which they were ordered.[220]

[214] *Ibid.*

[215] *Ibid.*

[216] *Ibid.*

[217] *Ibid¸*

[218] Subramanian, 38.

[219] Grieco, 1982.

[220] *Ibid.*

This set of affairs was termed “Permit Raj,” a play on the colonial-era administration, the British Raj.[221] The perception was that government policy was stifling innovation and productivity. There was widespread criticism of Professor MGK Menon, a physicist previously working at the Tata Institute of Fundamental Research, whose slow decision-making process was railed against in articles in the *Economic Times* and *Financial Express*.[222] This period would serve as a historical touchpoint for members of the indic-computing group — of bureaucratic, wrong-minded government agencies hampering the growth of the technology sector.

The tide turned again in the Indian computer industry beginning in the 1980s, with the rise of Rajiv Gandhi in national politics. Rajiv Gandhi was the son of then-Prime Minister Indira Gandhi, the daughter of Jawaharalal Nehru. Rajiv Gandhi was being groomed to enter politics. Rajiv Gandhi, an airline pilot, identified telecommunications and information technology as “core sectors” and happened to get a new Computer Policy approved by his mother just days before her assassination in 1984.[223]

The new computer policy was designed by Narasimiah Seshagiri, the directorate of the National Informatics Center that had been recently established under the Electronics Commission. Seshagiri was an advocate of trade liberalization.[224] He termed his computer policy, “flood in, flood out” - that is, allow free entry of software imports, and export an even greater amount. The policy signaled India’s openness towards international software markets, a stark contrast to the preceding Permit Raj era.

At the same time, India’s second foreign exchange crisis in 1989 (the first was in 1981), led to further trade liberalization.[225] India turned to the International Monetary Fund in 1991 and was forced to undergo “structural adjustment.” Structural adjustment mandated the opening of its borders to foreign investment, amongst other conditions. Despite the economic challenges, the new policies coming from the IMF and from internal players like Seshagiri began to show what India stood to gain from openness towards global markets.

During this period, the Indian computer industry pivoted from hardware towards software.[226] A generation of developers became trained at the IITs and IIMs. Important software-as-a-Service companies were founded in India: Infosys, Wipro, Satyam.[227] Two new research and training

---

221 *Ibid.*

222 *Ibid.*

223 Subramanian, 2006; Ramachandra Guha, “Rulers and Riches” in *India After Gandhi: The History of the World’s Largest Democracy*, Reprint edition (New York/N.Y: Ecco, 2008).

224 *Ibid.*

225 *Ibid.*

226 *Ibid.*

227 *Ibid.*

institutions, the National Center for Software Technology (NCST) and the National Institute for Information Technology (NIIT), were founded in the 1980s.[228] Cities like Bangalore that specialized in IT began to rise in national prominence.[229]

India's internet infrastructure also narrowly escaped another "Permit Raj" situation during this era. As in the United States, there were multiple computer networks emerging in India in the 1980s. Like the ARPANET, and later NSFNET networks, India had a network of linked government organizations and academic institutions beginning in 1986. "ERNET" connected the Department of Electronics, NCST, and several IITs and IISc (Indian Institute of Science).[230] ERNET initially followed both the competing TCP/IP and OSI-IP protocols, but eventually settled on the former.[231] Like the early internet in the United States, the infrastructure was far from the masses and limited in its uses.

At the same time, there was a flurry of recreational computer networks being formed over telephone lines. These bulletin board systems (BBS) were gaining popularity around the world, and India was no exception. These were important pre-cursors to the hobbyist groups described above; many of the older members recalled participating in them.

Networks such as Live Wire! BBS, BharatNet, and later FidoNet grew over telephone lines from the late 1980s into the following decade.[232] As an early Indian BBS founder wrote, "In a short period of time, we all became the watering holes of an eclectic mix of people ranging from computer nerds, to students, to entrepreneurs, or those with special interest and even those looking for an online date in the comfort of anonymity!"[233] The networks were not easy to establish, due to the long wait time of acquiring a phone line (up to seven years) and the high price of long distance phone rates, as BBSes became interconnected internationally and came to be charged higher fees.[234]

But here too, the 1991 economic liberalization reforms improved the state of affairs. The newly-created Department of Telecommunications (DoT) was tasked with opening up telecommunications to private sector investment. This formal opening-up came with a new Value Added Services (VAS) registration fee however – envisioned to encourage a minimum level of performance, but in practice, threatening to shut down BBSes because of the high cost.[235]

---

[228] *Ibid.*

[229]Guha, "Rulers and Riches."

[230]Rao, Madanmohan, "India's Internet Chronicles," in *NetCh@kra: 15 Years of Internet in India.* 2011.

[231] Rao, Madanmohan, "India's Internet Chronicles," in *NetCh@kra: 15 Years of Internet in India.* 2011.

[232] Nanda, Suchit, "From BB (Bulletin Boards) to BB (Broad Band) Internet," in *NetCh@kra: 15 Years of Internet in India.* 2011.

[233] Nanda, 86.

[234] *Ibid.*

[235] *Ibid.*

Advocacy against the VAS fee by the BBS community and the US-based nonprofit, the Electronic Frontier Foundation, led to the DoT backing down. As Nanda wrote, "This was probably the first instance of electronic advocacy in India and that too with quite a positive result. The license raj in Indian e-space was nipped in the bud!".[236] The tussle between government and hobbyists over internet infrastructure left the message that users might know better than the government, and could lobby for the right to access global networks.

Though India's large rural and remote population often makes it the subject of focus in discussions around the "digital divide," it is worth remembering that there was simultaneously a significant population that was highly, and excitedly, connected. By the late 1990s, BBS communities were rallying to establish internet for the public.[237] News of the World Wide Web had traveled through information networks, and there was high demand for its access. At first, the DoT authorized only one major phone company, Videsh Sanchar Nigam Limited (VSNL), to be an international internet carrier. VSNL service launched on August 15, 1995. But due to the expensive and limited service, two years after its introduction, there were just 40,000 subscribers, compared to 500,000 across the smaller area of Singapore, Hong Kong, Malaysia, and Taiwan.[238]

Public lobbying via the Email and Internet Service Providers Association of India (ESPAI) continued for an open policy that would allow private players to become internet service providers. Finally in 1997, the Telecom Regulatory Authority of India (TRAI) Act passed as a result of this lobbying, resulting in the establishment of an independent telecom regulator.[239] The following year, private internet service providers (ISPs) became legal and internet connectivity took off in metropolitan India. A similar 'opening of the internet' to commercial ISPs in 1991 in had greatly expanded the number of users and made the Internet a fixture of everyday life the United States.[240]

Though there were still many barriers to reaching the entire population – access to computers, internet access in rural areas, unreliable electricity – India had become connected to the "information superhighway" not long after the same services reached the masses in the United States. Going into the early 2000s, there was momentum towards India's rise with respect to information technologies. By the early 2000s, most major international software companies had set up offices in India: SAP, Peoplesoft, Oracle, IBM, Sun, CISCO, Nortel, Adobe, and most significantly, Microsoft.[241]

[236] Nanda, 90.

[237] Nanda, Suchit, "From BB (Bulletin Boards) to BB (Broad Band) Internet," in *NetCh@kra: 15 Years of Internet in India.* 2011.

[238] Chowdary, T. H. "Birth and Growth of Internet in India,"n *NetCh@kra: 15 Years of Internet in India.* 2011, 65.

[239] *Ibid.*

[240] "Internet History of 1990s | Internet History | Computer History Museum," accessed June 29, 2022, https://www.computerhistory.org/internethistory/1990s/.

[241] Subramanian, 2003.

Microsoft had a two-pronged vision of engagement with the Indian subcontinent. Like other software companies, it sought to employ India's large English-speaking and IT-proficient workforce. But it also was one of the only companies to identify India as a large consumer base.[242] Towards these ends, Microsoft began investing millions into local training and software localization. As it said in November 2002 article in the Indo-Asian News Service:

> *Contending that it was very important to localise software in India, Gates announced plans to market "Microsoft XP" and its next version "Office 11" – code named "cash cow" -- in Indian languages like Hindi, Bengali and Malayalam and extend it to nine more Indian languages in 2003. Work on this is underway at Microsoft's development centre in Hyderabad. The company plans to invest $100 million of the $400 million on the centre, which will have 500 employees by 2005, up from the current 200.*[243]

Additionally, the company would spend $20 million on a program to train over 80,000 school teachers and 3.5 million students in IT over five years.[244] This and other initiatives required partnerships with local government.[245] The same year, Microsoft announced plans to develop several products for e-governance initiatives: land record management, mail messaging systems, policy communication applications, registration services.[246]

Over the past fifty years, India had experienced isolation, limited negotiating power with the West (as with IBM's *gradual upgrades* policy and eventual withdrawal from India), and the swell of wealth that came from slowly opening its markets to international investment. The new millenia brought an eagerness to continue the upswing of the 1990s. At the same time, the history of the past fifty years brought suspicion of new players, reflected in articles dubbing Microsoft programs "cash cows". For software hobbyists, the experiences of the 1990s also showed that user activism had a role to play in directing national IT policy. The government had been slow to understand the high demand for and impact of the internet; it was BBS users who had glimpsed the future first. Now, in the early 2000s, the indic-computing group was left with a set of complicated politics: distrust of corporations, a belief in open standards and a desire for interconnection, and a sense of government lethargy. *Someone* needed to set a new agenda and direct these parties. This is where the Indic-computing group came in. Their work would entail bringing together open source enthusiasts and old-school academic researchers; it also meant inviting Microsoft representatives to those same gatherings. As Parikh wrote, "We have to make sure technically-informed decisions are made."[247]

---

[242] Constable, interview.

[243] "Gates to invest $400 million in India, unveils Tablet PC" *Indo-Asian News Service*. November 14, 2002.

[244] *Ibid.*

[245] Naazneen Karmali, "Microsoft's Passage to India," Forbes, accessed June 29, 2022, https://www.forbes.com/global/1998/0727/0108030a.html

[246] "Bengali Windows version by Oct 2003" *Times of India*, December 16, 2002.

[247] Parikh, "New Blood - Forget the Establishment," email.

Bangladesh's National Computing Policy: Contrasts and Counterpoints

In contrast, Bangladesh's information technology sector had only inched along through the 20th century, and the ethics of the bengalinux group reflected such. As many of its members would later reflect, they were doing their volunteer work because it felt like no one else would.[248] Where India had multiple research institutions, government agencies, a lively private sector, and even funding to support open source projects, Bangladesh's technology sector was scattered and sparse.

Its computing history contrasted starkly against India's. Bangladesh, then still East Pakistan, received its first computer in 1964 as a gift from IBM.[249] The Pakistani government could only find one suitable operator, based in East Pakistan, which led to its installation in Dhaka's Atomic Energy Centre. Though mainframes grew in presence in the country over the next decade, particularly in banks and the Bureau of Statistics, these machines had limited computing power (with Bangladesh too falling into IBM's gradual upgrades policy) and were quite costly. Where India jumped ahead to mini- and micro-computers in the 1970s and 80s, Bangladesh continued to use large mainframes.[250] Its major private universities, the Bangladesh University of Engineering and Technology (BUET) and Dhaka University only received these mainframe computers in 1979 and 1985, respectively.[251] Where India's major research universities would receive government resources and participate in international exchange programs (detailed further in the following chapter), Bangladesh's major universities lagged in even creating computer science training programs.[252] Hanging in the background were infamous sayings like that of Henry Kissinger in 1972, that the country of Bangladesh was destined to be a "basket case."

Bangladesh's computer policy straggled behind India's in many other ways. The counterpart to India's Department of Electronics (started in 1970) was Bangladesh's National Computer Committee, which was formed in 1983 to purchase computers for government use.[253] It was later re-organized and renamed the Bangladesh Computer Council in 1990, after which time it took on a more active role in computer education, organizational support, and infrastructure development.[254] India's Department of Electronics had followed a wayward policy of indigenous computing, but Bangladesh's analog had not done much at all. Indigenous assembly – the

[248] Taneem Ahmed, interview; Mahay Alam Khan, interview with author, July 8, 2019.

[249] Masud Hasan Chowdhury and Md Mahbub Murshed, "Computer - Banglapedia," accessed June 29, 2022, https://en.banglapedia.org/index.php/Computer.

[250] *Ibid.*

[251] *Ibid.*

[252] *Ibid.*

[253] Mahbubul Alam, "Bangladesh Computer Council - Banglapedia," accessed June 29, 2022, https://en.banglapedia.org/index.php?title=Bangladesh_Computer_Council.

[254] *Ibid.*

purchasing of parts for local assembly – finally became popular in the late 1990s as personal computers gained popularity and lowered in cost.[255]

Indeed, the popular belief amongst tech-savvy Bangladeshis was that their government was letting them down, particularly with respect to internet connectivity. In a July 1996 issue of *Computer Jagat*, an editorial stated,

> *Revolutions have been created round the world to use Internet for extension of knowledge, scientific activities and education. But, in Bangladesh we have no such initiative to provide Internet access to the educational institutions. Even most prestigious higher institutions like University of Dhaka and Bangladesh University of Engineering & Technology are beyond its reach.*[256]

The Bangladeshi government had rejected the offer to connect the country to the Internet via submarine cable twice, once in 1988 and again in 1994, citing "security concerns."[257] India had likewise rejected the first offer, but Bangladesh was a standout in 1994 when India, Pakistan, and Sri Lanka all accepted the interconnection offer.[258] Instead, internet in Bangladesh spread slowly beginning in 1996 with a satellite connection. The fees for the satellite connection were so high that internet use remained limited even in Dhaka until the coming of mobile internet in the mid-2000s.[259]

Those knowledgeable of and committed to the forthcoming "digital age" — perhaps from their experience with BBS networks from the early 1990s, or from hearsay beyond Bangladesh's borders — would travail to evangelize it at the turn of the millennium.[260] The Indian software hobbyists were seeking to steer a carriage that was well on its way, and Bangladeshi tech enthusiasts found themselves struggling to get onboard a coach at all.

The relative vacuum of activity in Bangladesh explains the context that many Bangladeshi bengalinux members bore in mind. Though the group spanned across India, Bangladesh, and the diaspora, its founder and about half of its members at any given time had ties to Bangladesh.[261] For them, the open source ethic or hacker mindset resonated even more than for perhaps the

---

[255] Chowdhury and Murshet, "Computer - Banglapedia."

[256] "Provide on-line Internet access to educational institutions immediately." *The Monthly Computer Jagat*, July 1996, 19. Quoted in "Overview of Internet Access in Bangladesh: Impact, Barriers, and Solutions," https://web.archive.org/web/20160103124803/https://www.isoc.org/inet97/proceedings/E3/E3_1.HTM

[257]Mohammad Ershadul Karim, "History of Submarine Cable," in *Cyber Law in Bangladesh* (Kluwer Law International B.V., 2020).

[258] *Ibid.*

[259] Bangladesh Network Operators Group, "History and Evolution of Bangladesh Internet," https://www.slideshare.net/bdnog/history-and-evolution-of-bangladesh-internet.

[260] For example: Shahidul Alam, "When a Modem Costs More Than a Cow," *Bytes for All*, April 30, 1999, https://shahidulnews.com/when-a-modem-costs-more-than-a-cow/.

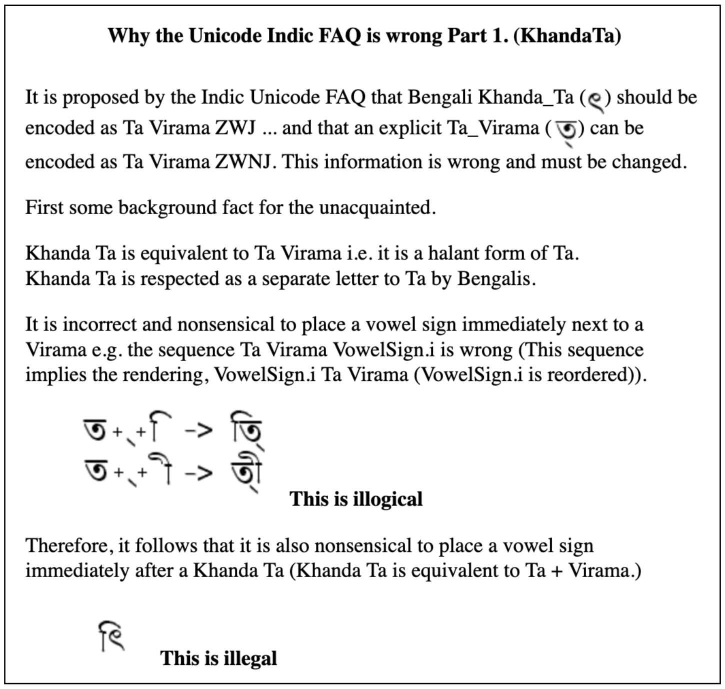
[261] Taneem Ahmed, interview with author, August 19, 2020.

average contributor — building independent tools gave them an opportunity to contribute and fill gaps that their governments had left unfilled. Again, the ability to do this rested on their technical expertise. Why shouldn't Bengalis be able to type in their own language online, or have their literature preserved for the future? If Microsoft or the government wasn't doing it, they could do it themselves.

"Khanda ta" & Bengalinux

It was in this context that we can begin to understand the steady interactions that would begin between the software hobbyist groups and the Unicode Consortium, as issues began to emerge with its Bangla encoding. As the introduction of this dissertation laid out, the issue with the Bangla letter *khanda ta* would have a long life and inspire vociferous debate. The issue would even take on a nationalist tenor and create a divide between Global North and South in the minds of some key actors. But this was not the universal reaction, and I show that here by tracing how issues with *khanda ta* were interpreted by technical-minded, software hobbyists — as a software bug that needed fixing, simply so that Bangla text would display properly.

On November 17, 2002, a new name would pop up on the FreeBanglaFonts listserv: Andy White. He posted, "Please see my Why the Unicode Indic FAQ is Wrong (Part 1) article," and a link to his personal blog.[262] It was highlighting errors in how *khanda ta* was showing up in text when placed next to certain vowels.



Why the Unicode Indic FAQ is wrong Part 1. (KhandaTa)

It is proposed by the Indic Unicode FAQ that Bengali Khanda_Ta (ৎ) should be encoded as Ta Virama ZWJ ... and that an explicit Ta_Virama (ত্) can be encoded as Ta Virama ZWNJ. This information is wrong and must be changed.

First some background fact for the unacquainted.

Khanda Ta is equivalent to Ta Virama i.e. it is a halant form of Ta.
Khanda Ta is respected as a separate letter to Ta by Bengalis.

It is incorrect and nonsensical to place a vowel sign immediately next to a Virama e.g. the sequence Ta Virama VowelSign.i is wrong (This sequence implies the rendering, VowelSign.i Ta Virama (VowelSign.i is reordered)).

ত + ্ + ি -> িত্
ত + ্ + ী -> ত্ী **This is illogical**

Therefore, it follows that it is also nonsensical to place a vowel sign immediately after a Khanda Ta (Khanda Ta is equivalent to Ta + Virama.)

িৎ **This is illegal**

Figure 23. Snapshot of Andy White's Blog Post

[262] White, Andy, "RE: [Freebangfont-devel] Bengali OT specification," Email, November 17, 2002.

I provide a layman's interpretation of this blog post below, but first, who was Andy White? White's messages conveyed expertise in text encoding and rendering, as well as deep interest in the Bangla language. He referenced Bangla manuscripts (from which he had identified that the half-ta was a short-lived innovation from the hot-metal type era) and seemed better acquainted with the orthography than many native speakers. As one Bengalinux member wrote in response to White's blog post, "I don't have such a deep knowledge of the orthography (I use bangla, I don't study it! :))"[263]

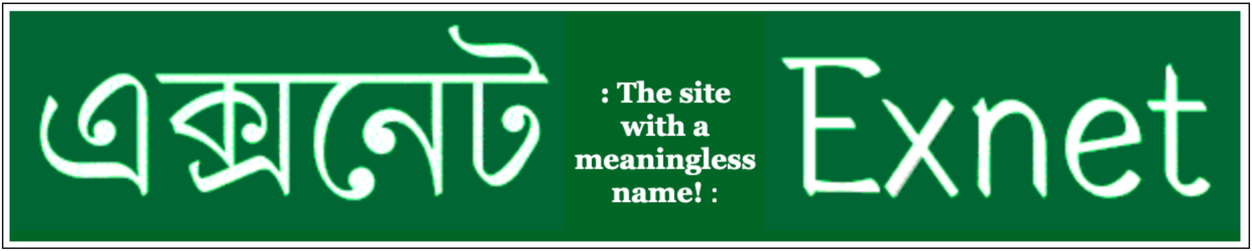


Figure 24. Banner From Andy White's Personal Website

Beyond that, however, few knew much about him. His website was called "Exnet: The site with a meaningless name!," with the header written in both Bangla and Latin script and a UK top-level domain.[264] The preamble did not give away much, beyond his interest in the "Bengali script (due to my own preferances)." The site hosted just a few pages with commentary on Unicode encoding; he did not appear to be a font designer or software developer, at least based on the content of the site.

In some ways, White epitomized the nature of identity on the early Web. As Emily van der Nagel has written, the internet has moved through several major waves of identity construction, from the usernames of the early Internet – ARPANET, UNIX, and email – to nicknames on bulletin boards, to the later stage of real-name social network profiles. The early and middle stages, in particular, were marked by malleability, or the absence of, "material attributes like gender, age, ethnicity and class." Bulletin boards were set up to emphasize interests over identities.[265] Users such as Andy White did not have to be tied to offline markers such as employer or education, as came to be encouraged on later social networks. As the adage went, "on the internet, nobody knows you're a dog." White would appear as an advocate for the Bangla language community at several points over the coming years; despite his lack of affiliation, he would gain the attention of Unicode overseers due to his sharp critiques, though the attention would be slow and uneven.

Andy had produced this blog post now, in November 2002, because *khanda ta* was not appearing correctly in digital text. It was a "dead consonant," meaning it carried no inherent vowel sound.

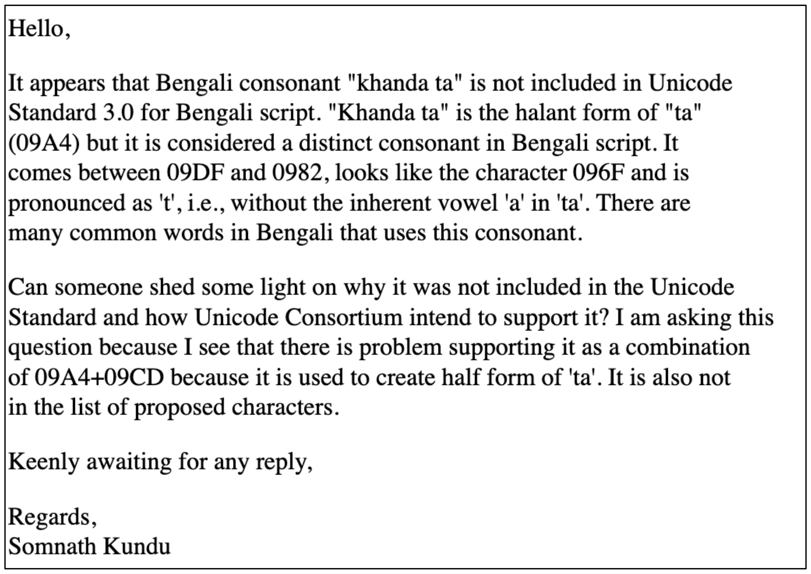
[263] Ghose, Kaushik. "RE: [Freebangfont-devel] Bengali OT specification," Email, November 17, 2002.

[264] "Exnet", https://web.archive.org/web/20121012000621/http://www.exnet.btinternet.co.uk/.

[265] Emily van der Nagel, "From Usernames to Profiles: The Development of Pseudonymity in Internet Communication," *Internet Histories* 1, no. 4 (September 2, 2017): 312–31, https://doi.org/10.1080/24701475.2017.1389548.

It was therefore “illegal,” as he wrote in his blog post, for vowel modifiers to appear around it. The layman’s description of his blog post was that, because of the instructions that Unicode had provided for representing *khanda ta* on its online “Indic FAQ” page, illogical behaviors were occurring.

This was a new rather than long-standing issue. The current Indic FAQ seemed to be the result of an earlier conversation that had taken place on the main Unicode list in May 2002. An individual named Somnath Kundu had started a new thread about *khanda ta* being missing from the Unicode Standard, much like Ziaur Rahman had done in 2000.[266]



Hello,

It appears that Bengali consonant "khanda ta" is not included in Unicode Standard 3.0 for Bengali script. "Khanda ta" is the halant form of "ta" (09A4) but it is considered a distinct consonant in Bengali script. It comes between 09DF and 0982, looks like the character 096F and is pronounced as 't', i.e., without the inherent vowel 'a' in 'ta'. There are many common words in Bengali that uses this consonant.

Can someone shed some light on why it was not included in the Unicode Standard and how Unicode Consortium intend to support it? I am asking this question because I see that there is problem supporting it as a combination of 09A4+09CD because it is used to create half form of 'ta'. It is also not in the list of proposed characters.

Keenly awaiting for any reply,

Regards,
Somnath Kundu

As with before, he highlighted the linguistic status of the letter (“considered a distinct consonant in Bengali script”) and technical ambiguity in its encoding (“there is a problem supporting it as a combination of 09A4+09CD”). Kundu wasn’t saying *khanda ta* was necessarily missing from the Standard, only that its representation was ambiguous.

Unlike Rahman’s post in 2000, Kundu’s received an authoritative response from a Microsoft employee, Apurva Joshi.

Apurva Joshi was Microsoft’s then-Program Manager for Font Technologies. Her father was the renowned Indian font designer, Professor R. K. Joshi, recognizable by name to anyone in the design industry in India. Professor Joshi had had a long career as a calligrapher, poet, teacher, and marketing guru in India. He had taught at the Sir J.J. Insitute of Applied Arts and later at IIT Bombay, training a whole generation of students in Indian typography.[267] He was also one of the

[266] Kundu, Somnath, “Bengali script - where is “khanda ta”?” Email, May 18, 2002.

[267] Olocco, 68; Shrinath Shanbhag, interview with author, March 28, 2022.

first non-Latin typographers to present regularly at the Association Typographical Internationale (ATyPi) annual conferences, which expanded his name recognition outside of India.[268]

In 1997, Professor Joshi had joined the Indian government's language technology initiative, the National Center for Software Technology (NCST) and was soon contacted in this post by Microsoft's Typography department to design their first set of Indian language fonts.[269] Between 1997 and 2004, Joshi and his team developed the Mangal, Latha, Gautami, Raavi, Shruti, Tunga, and Kartika fonts for Devanagari, Tamil, Telugu, Gurmukhi, Gujarati, Kannada, and Malayalam respectively. He also worked on the Vrinda font for Bangla, which was eventually released in 2004 with Windows XP Service Pack 2.[270]

While her father worked on Indic fonts, Apurva Joshi was leading Microsoft's rendering efforts. The program manager role at Microsoft encompassed equal parts relationship-building and technical direction. She was in touch with the folks leading the indic-computing effort, and headlined workshops co-organized by the hobbyist group and NCST to answer questions about Indic rendering.[271]

It was from this position of authority that she drafted a response to Somnath Kundu's question regarding *khanda ta*. Apurva Joshi responded that, though she was not an expert in the Bangla script, her understanding was that its structure could be analogized to Devanagari.[272] Where Devanagari had "half-forms", Bangla had "halant-forms" – cases where the inherent vowel is silenced – that would be represented in the same way.

Apurva Joshi's instructions were to treat Bangla halant-forms the same way as half-forms in Devanagari, since Bangla had no half-forms of its own. At the request of Rick McGowan, Unicode's Vice President, she wrote up an answer for the online FAQ about how to show the various forms, since so many questions had been asked about it.[273]

> *Q. I don't see the Khanda Ta encoded in the Bengali block? It has a distinct shape.*
>
> *Ans. Bengali does not have distinct half forms for consonants like Devanagari and Gujarati do. Hence for all practical purposes, the halant forms are also considered as half forms. Conjuncts that are used by the*

[268] Shanbhah, interview.

[269]Olocco, 68; Shanbhag, interview.

[270] Olocco, 70.

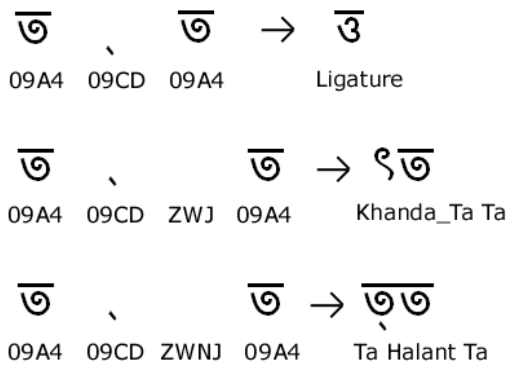
[271] Shroff, Keyur, "[Indic-computing-users] Open discussion and Q&A session on OpenType font," Email, October 14, 2002.

[272] Joshi, Apurva, "RE: Bengali script - where is "khanda ta"?" Email, May 19, 2002.

[273] Joshi, Apurva, "RE: RE: Bengali script - where is "khanda ta"?" Email, May 21, 2002.

Bengali language typically have ligatures to display them as. It is only in the case of words that are not native to Bengali and require the display of conjuncts that don't have ligatures; that these halant forms are used. The khanda Ta is not a consonant (or distinct character) by itself, but the halant form of a consonant. It is a special case graphically because it is not displayed as Ta Halant (as other consonants in Bengali), but has a distinct shape. Such alternate forms can be displayed using an OpenType font, that contains glyphs for such forms. Below are sample sequences to display the conjunct created using Ta Halant Ta in different ways. They assume that the font contains a glyph for: (i) the Khanda Ta, as well as (ii). a glyph for the ligature of this conjunct.

1. Ta Halant Ta -> Ligature for taTa
2. Ta Halant ZWJ Ta -> KhandaTa Ta
3. Ta Halant ZWNJ Ta -> Ta Halant Ta



As the previous chapter described, consonants could combine in multiple ways in all Indic scripts. Somewhere in the multilingual computing stack, a standard or piece of software needed to define which glyphs should be shown for a given combination of conjuncts. In Apurva Joshi's FAQ response, she offered three recipes, or Unicode codepoint sequences, that would trigger three different ways of producing *khanda ta* next to another consonant.

—

Jumping ahead to November 2002, Andy White was now raising the issue of *khanda ta* again because Apurva Joshi's clarification had led to more, rather than fewer, problems. The problem now was that in Microsoft's shaping engine, Joshi's instructions lead to misplaced vowel signs appearing around *khanda ta*. White was making a larger point; this misplaced vowel issue was symptomatic of mistakes in how Unicode was understanding the Bangla alphabet.[274]

He asserted that the three cases that Joshi covered in her FAQ response did not make sense – at least the third one of a ta with a visible halant (ত্*)*

> *To make it clear, I am not referring to any particular rendering mechanism (inc. MS's).If you look through a Bengali Dictionary I doubt that you will find a single occurrence of Ta with a visible Virama, khandata is always used.*[275]

The third option she had shown did not appear in text, and so that codepoint sequence should be the one used to display *khanda ta*. Making this change would fix the problems at the rendering level, in which vowel modifiers were appearing where they shouldn't.

Moreover, though half-*forms* did not exist in Bangla as they did in Devanagari, there were such things as half-*glyphs.* For half a century, "half-ta's" did exist in the Bangla alphabet, but more a result of a "printer's hack" than a change in the orthography.[276] Since metal linecasters did not have room for most Bangla ligatures, consonants instead had their own miniature forms that would be combined for conjuncts.

There were reasons to keep these half-glyphs now. Some people preferred them, and they could also be used to digitize old metal-typed manuscripts with fidelity. A half-ta might also be needed in a textbook to illustrate how certain conjuncts were composed.[277]

And so, White wrote that the second sequence Joshi had defined should be used to display half-*tas*.

> *Yes a* ***consonant+Virama+ZWJ*** *shows a half form but what makes you think that a half Ta should look like a KhandaTa? Why should the Bengali script not be allowed to have a Half Ta? In some fonts the Bengali half Ta is drawn as a smaller raised Ta whilst khandaTa is given as a separate glyph.*[278]

In sum, Andy's proposal was essentially saying: there are many visual representations of consonants we need to be able to account for in a font. Opting to show these should not trigger misplaced vowel modifiers. These mistakes were occurring in part because the rendering engine was not agnostic to linguistic structure – it interpreted control characters as representing a

[274] White, Andy, "RE: Extending the semantics of ZWJ and ZWNJ for Indic scripts," Email, November 18, 2002.

[275] White, Andy, "RE: Errors in the Indic FAQ" Email, November 17, 2002.

[276] Olocco, 2014; White, Andy, "RE: [Freebangfont-devel] Proposal to add Bengali Khanda Ta" Email, December 4, 2002.

[277] White, Andy, "RE: [Freebangfont-devel] Proposal to add Bengali Khanda Ta" Email, December 4, 2002.

[278] White, Andy, "RE: Errors in the Indic FAQ" Email, November 18, 2002.

common grammar across Indic scripts. A “virama + zwj” typically represented a half-form, so that sequence should align with half-glyphs in Bangla. A “virama + zwnj” represented halant-forms, and that was what should be used for *khanda ta*, which *was* the halant-form of *ta.*

It was Apurva Joshi’s conflation between halants and half forms between Bangla and Devanagari was causing problems.

When Andy posted to the freebangfonts page, others chimed in saying they had informally been using Andy’s advised sequence of codepoints anyways.

> *I have used Andy's proposal in all my fonts (Mukti Ani etc exept Mitra which is as yet grossly incomplete). The reason being it made more sense than the official Unicode proposition and possible to type using ITRANS keyboard of Yudit [using t.h].*[279]

This response is significant because it shows the relative independence with which the hobbyists were working. Their goal was to create working, Bangla language software; if it only worked in Linux, that was fine because at least it worked.

But of course, a more global solution, whether handled by Microsoft or by Unicode, would eventually be necessary for cross-platform reliability.

Andy’s proposal didn’t gather much steam at this point. The Bengalinux members agreed with him but otherwise let the thread die in the freebanglafonts list. Andy posted to the Unicode page and didn’t get any official response. In January of the following year, an employee at India’s NCST, Keyur Shroff, seemed to have noticed Andy’s argument and picked up the cause. When Andy called him out for apparently claiming credit, Shroff responded that “I tried to put it in other words because few people couldn’t understand what Andy meant :)”[280]

These points are important because they show the steady transformation of how the “missing letter” becomes an issue. When it first arose in 2000, it was a matter of confusion. Unicode was new, there was dissonance between what counted as a letter and what deserved its own codepoint. By 2002, there was work advancing in OpenType, with the Bangla specific specification being released just as Andy’s blog post went up. But there was still a strong need to coordinate between groups such as font designers, renderers, and the Unicode Standard. Protocols were not in place yet to facilitate these conversations. The discussions were informal, evident in the sources used throughout this chapter: mailing list threads, personal blogs that are no longer live, and technical specs that had not yet been finalized.

The technology was young, but so were the institutions. Unicode staffers were not engaging much on the mailing list, at least with respect to Indic scripts. Changes were made ad hoc to a

[279] Mitra, Anirban, “Re: [Freebangfont-devel] Khanda tha etc.” Email, November 18, 2002.

[280] Shroff, Keyur, “RE: [Indic-computing-standards] Re: [Indic-computing-users] Unicode” Email, January 23, 2003. The rest of this conversation, including Shroff’s post to Unicode, is lost because it was posted on Yahoogroups, which was deleted before being archived

web page hosted on the Unicode site that contained the “Indic FAQ.” Unless someone was scanning the Wayback Machine, these changes were not well documented nor were they promoted. As Andy White asked in a follow up email to the main Unicode listserv,

> *Is the Unicode FAQ officially part of the Unicode standard? If not why not?...this should be dealt with in detail in the next edition of the [The Unicode Standard]. IMHO, this [khanda ta’s encoding] is not a typographical detail that can be left to implementers to settle: it affects the interpretation of text.*[281]

To this, he received no response. Indeed the status of *khanda ta* was handled only by this informal FAQ page, which contains a smattering of questions related to how Unicode compares to ISCII and why Unicode names were given as they were.[282]

A related point is where the expertise was coming from in this conversation. *Khanda ta* did not emerge as an issue among the Bangla computing hobbyists, who either had not encountered the problem or had developed their own open source work around. *Khanda ta* was championed at this point by Andy White. Furthermore, the referenced “mistake” seemed to come from an Indian native, Apurva Joshi, whose decision to conflate Devanagari and Bangla introduced the bug. These facts add complications to a narrative that later assumes it is the fault of Westerners, disadvantaging the Bengalis.

These points are worth remembering as the narrative transforms in the coming chapters. The next chapter explores how such orthographic-like debates are handled between official representatives of the institutions themselves: Unicode Standard designers and Indian government representatives. Thus far, we’ve used the lens of technical design to talk about the shortcomings of the Standard and its downstream technologies for Indic scripts. Beginning in Chapter 3, we will consider how these technologies fit into a longer perceived narrative of language politics and planning: who holds the authority to determine how a script will be represented? What stakes does a script, and its digitization, hold for the identity of a nation?

[281] White, Andy, “ISO 464, Unicode & The FAQ” Email, November 21, 2002.

[282] “FAQ - Indic Scripts and Languages,” accessed June 29, 2022, http://www.unicode.org/faq/indic.html.

Chapter 3: Digitizing Language Planning

The Unicode Consortium first heard of Dr. Om Vikas and his government-sponsored program of Technology Development for Indian Languages (TDIL) in the year 2000, when Vikas sent them an unsolicited copy of his program's new newsletter. Intriguingly entitled "*Language Technology Flash: Moving up the knowledge chain…*," it set forth his nationalist agenda for Indian computing in bold, unapologetic terms:

> *India is the second largest population in the world with one billion populations. There are 18 constitutional languages with 10 scripts and over 1650 dialects. Development of the nation with such diversity depends on acquiring absorbing and communicating knowledge seamlessly. Information Technology (IT) has emerged as an enabling technology in reducing the knowledge gap across different linguistic groups encompassing over 95% of India's population that is not English-literate. It is, therefore, necessary that people should be able to use languages and derive benefits of enhanced productivity and better quality of life. National excellence in the millennium shall be determined by the extent to which the Information Technology can deliver its potential in Local Languages.*[283]

For Vikas and his team, in other words, multilingual computing was not just a nicety, but a matter of "national excellence," an essential prerequisite to the "development of the nation." As Vikas pointed out, India was both an enormous and an enormously diverse state — presenting a multilingual challenge on a scale almost unimaginable to the West. What Vikas didn't highlight, but nevertheless knew quite well, was that India was also poor — and he wanted to remedy that using information technology. The personal computer and the internet presented an incredible opportunity to improve India's living conditions, but only if the Indian masses were able to use them. And therein lay Vikas's problem: how to make information technologies available to all, not just the "English-literate" few? Over the next few years, Vikas and his team would not only become leading figures in India in the field of local-language computing, but would also succeed in building meaningful relationships with their counterparts abroad. Unlike the hobbyist listservs, Vikas had access to the international power players responsible for defining multilingual computing standards, including of course, the Unicode Consortium.

In this chapter, I lay out the emerging language politics of Vikas's new digital age, defined by an unprecedented access to international cooperation as well as an unapologetic pursuit of national greatness. I argue that Vikas' agenda is best understood through the sociolinguistic lens of "language planning," the state-sponsored activity of "manipulating language as a social resource in order to reach objectives."[284] Historians understand language planning to be a post-war, postcolonial practice, closely intertwined with theories of modernization and development. Those responsible for language planning included government agencies, educational institutions, and linguistic authorities.[285] As these forces worked to exploit "language as a social resource,"

[283] Vikas, Om. *Language Technology Flash*. Unicode Technical Committee Document Registry. L2/00-407.

[284] Carol M. Eastman, Language Planning, an Introduction (Chandler & Sharp, 1983), 29.

[285] *Ibid.*

they often conducted orthographic reform meant to refine and standardize writing systems, using state media and educational curricula.

At the same time, we see the contrasting approach of the Bangladeshi government. Where Vikas' engagement with the Unicode Consortium represents a recognition of new authority, Bangladesh's engagement with the International Organization for Standards (ISO) on the same issues represents a stagnant view and reliance on past authorities. Bangladesh's understanding of Unicode would evolve over the coming years through India's example and through its experience of *khanda ta* as the issue would begin to take off.

This chapter begins with a historical account of the activities of the TDIL program under Dr. Om Vikas, including the group's philosophies of language development. It then situates TDIL within the longer history of language politics in South Asia, arguing that TDIL represents the culmination of a crucial shift towards national modernization in the second half of the 20th century. Finally, it chronicles TDIL's interactions with the international Unicode Consortium to illustrate how it viewed script digitization policies as an extension of language planning. I bring in the Bangladeshi government at this point as a contrasting view.

In the end, I do not make a claim in this chapter about whether Unicode encodings are genuinely examples of language standardization or orthographic reform. As one of Unicode's most common refrains went, "We're creating a *technical standard*, not a *language standard.*" But I do argue that Unicode was understood as falling under the purview of language *planning*. Whereas a language *standard* dictates rules for spelling, vocabulary, and grammar, typically towards the goal of delineating its high status, language *planning* refers to the political activity of advancing a language in pursuit of social and political goals.[286] Of course, language planning can include language standardization efforts, but it can also involve modifying technical resources to better suit the needs of the language as it stands. For example, a 2000 whitepaper outlining "Fifty Years of Hindi Language Planning" described a wide variety of technical interventions meant to advance the development of the Hindi language, including "typewriters, stenographers, typists, machines for writing addresses, bilingual electronic machines, software with Hindi in Devanagari fonts and Unicode encoding, [all] made available in Hindi even before such resources are made available in any other language."[287]

For India's Hindi language planners, then, technological development was at least as important as any of their other major goals (e.g., defining the geographic borders of Hindi-speaking states, addressing public transport signs in Hindi, issuing Hindi license plates, and, yes, language-standardization efforts such as creating a simplified Hindi and publishing a standardized glossary for it). In other words, language technologies were absolutely central to language planning efforts – especially when it came to local language typewriters and (more importantly for our purposes) local-language fonts.

[286] "Language Standardization," obo, accessed June 28, 2022, https://www.oxfordbibliographies.com/view/document/obo-9780199772810/obo-9780199772810-0250.xml.

[287] Prof B. Mallikarjun, "FIFTY YEARS OF LANGUAGE PLANNING FOR MODERN HINDI The Official Language of India," January 1, 2000, 13.

This background helps us understand how the issue of *khanda ta* would come to take on such high stakes in the views of government officials and laypeople, as it would be viewed through the lens of language planning. The background provided in this chapter also helps us understand how we would come to see the puzzling assortment of dues-paying, voting members on the Unicode Technical Committee (an oversight board whose decisions hardly ever came to a vote). From 2000 onward, the point at which Dr. Om Vikas would set a new agenda for TDIL, Unicode membership would include major global software companies, one academic institution, and a handful of South Asian governments.[288]

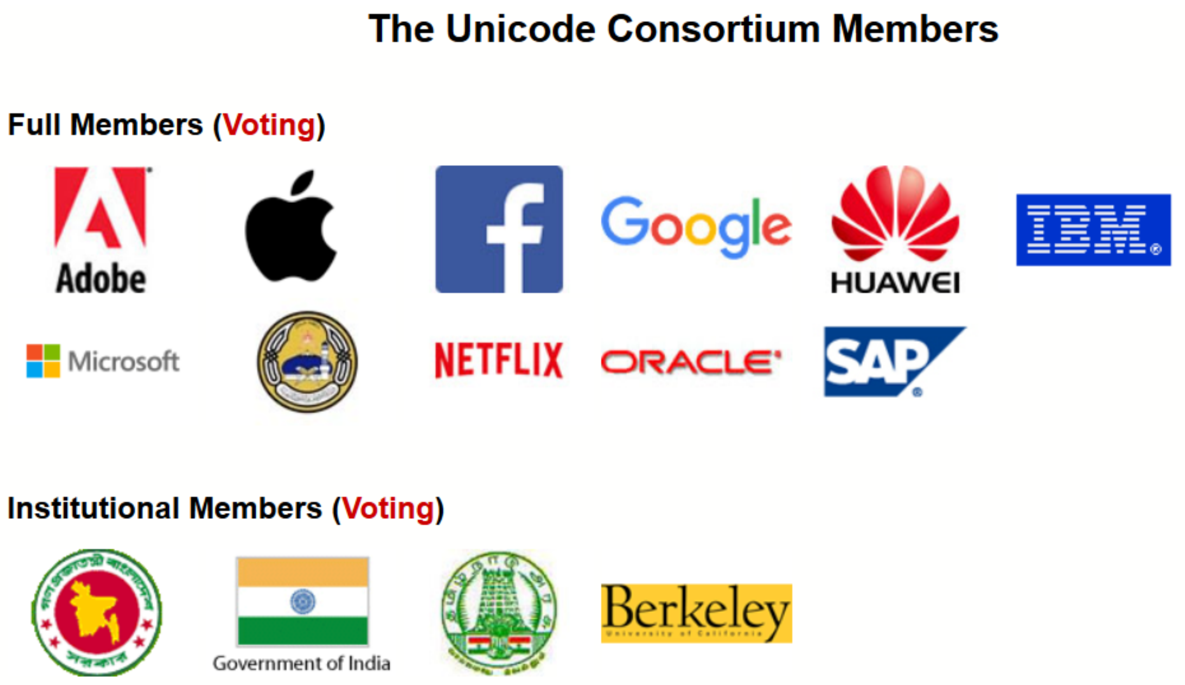


Figure 25. Snapshot of Unicode Membership (Pre-2019 Changes to Fee Structure)

What is crucial to understand about Vikas is that his TDIL program represents a significant shift in certain ways from the post-war language planning paradigm. Though the modernization of the nation remained an important goal, the means by which such modernization was to occur had changed. Specialized technical actors — including state-sponsored engineers and scientists-turned-policy-makers — had come to replace politicians and professors in the nation's language-planning projects. Their expertise in the intersection of language and technology helped ensure that their sphere of influence was no longer restricted within India's national borders, but also extended outward to the organizations overseeing international language standards. This evolution, I argue, showcases how national resource planning – including language resource

[288] This was the University of California, Berkeley, which would join the Consortium to aid in minority script digitization. I explain the role of UC Berkeley further in the Conclusion of this dissertation.

planning – has become increasingly corporatized in the new millennium, especially in the global South. In a world organized around the needs of multinational corporations, national governments may come to feel they can no longer dictate the course of language planning, but must instead negotiate with powerful corporate actors, such as the private companies directing the Unicode Standard.

The Roots of a New Language Technology Policy

Before we can appreciate how Dr. Om Vikas's program would transform the landscape of indic-language computing, we have to take a closer look at its origins. I begin here in the research institutions set up by India's first prime minister, Jawaharlal Nehru, where work in language technology, and language technology *standards*, began.

We start at the Indian Institute of Technology, Kanpur (IIT Kanpur, or IITK) where Vikas received his technical training.[289] There, he met two figures who would shape his research, and later policy, directions, and help produce key technical standards such as ISCII. IITK began work in this area in 1970, when an IITK professor, H. N. Mahabala, completed a visiting scholar position at the Massachusetts Institute of Technology (MIT).[290] There, Mahabala had been impressed by the development of an Optical Character Recognition (OCR) project for the blind. MIT's OCR project became an early prototype for modern-day screen readers, which are accessibility devices that interpret the text on a screen and read it aloud for those who cannot see. When Mahabala returned home in 1970,and relayed what he had seen, the concept intrigued one of the graduate students at IITK – R. M. K. Sinha – who soon decided that he wanted to develop a similar OCR system for the Devanagari script (used to write Sanskrit, Prākrit, Hindi, Marathi, and Nepali languages).[291]

As Sinha later wrote in his memoir,

> *Some of my colleagues expressed ridicule as well as surprise that I should choose to work on Indian languages at a time when it was almost inconceivable that Indian languages could be used on expensive computer systems, which remained within reach of only a few in India.*[292]

As this chapter later discusses, amongst well-educated Indians, native languages were relatively low status, in comparison to the English, the global tongue. Sinha's comment reflected the

[289] "Om Vikas - Biography" *Indira Gandhi Centre for the Arts*. http://ignca.gov.in/PDF_data/Dr_om_vikas_faculty.pdf

[290] R. Mahesh K. Sinha, "A Journey from Indian Scripts Processing to Indian Language Processing," *IEEE Annals of the History of Computing* 31, no. 1 (January 2009): 8–31, https://doi.org/10.1109/MAHC.2009.1, 15.

[291] *Ibid.*

[292] *Ibid.*

dissonance many felt between using a high-status technology — “expensive computer systems” — for a low-status purpose. Whom would that serve?

Sinha remained convinced, however, that “the benefits of computing technology could truly reach people only through their own language, and therefore we Indians had to make a beginning in this direction.”[293]

“Make a beginning” was exactly right. When Sinha was beginning his graduate work, IITK had only recently upgraded its computing infrastructure from the IBM 1620, which it had received through the Kanpur Indo-American program (KIAP) all the way back in 1963. KIAP was founded in 1962 as a cold war assistance program to allow the newly-established IITK to benefit from the resources of nine prominent American research universities.[294] At the time, India was an important Cold War ally for the United States (in contrast to the Soviet-leaning Pakistan), and in the interests of consolidating this allegiance, the US Agency for International Development (USAID) provided IITK with equipment, materials, and books not otherwise available in India. The program also provided funding for American faculty members to take up visiting positions at IITK, as well as opportunities for IITK faculty members to do the same at American member institutions such as MIT, which was how Mahabala had done his exchange in 1970. In total, KIAP ran for ten years (from 1962-72), at which point IITK was deemed self-sustaining and successful as an institute of higher education in the model of America’s premiere technological institutions. As a representative later characterized the program, “The Kanpur Indo-American Program is considered by many to be one of the most significant success stories in the rich history of bilateral higher education exchange programs between the United States and India.”[295] In many ways this was true, as it set into motion exchanges between key technical actors in the two countries that would persist into the new millennium.

Thus, in 1970, Sinha began his research on script “mechanization,” or translating the script to screens — the first step towards building a Devanagari OCR system. The first step was to analyze the underlying logic of Indic scripts. The eventual approach of tackling the full set of major scripts was not decided a priori. Only after discussing the possibility of a Devanagari OCR system with a Telugu-speaking colleague would he come to appreciate the similarities between India’s seemingly disparate Northern and Southern scripts, since he and his colleague “[came] as [they] did from those two different areas.”[296]

What were the linguistic similarities they noted? For starters, all Indic scripts — North and South — made the same distinction between full and pure consonants (namely, full consonants carry an inherent vowel sound (e.g., ka), whereas pure consonants have no such vowel sounds associated

[293] *Ibid.*

[294] “U.S.-India Partnership: Kanpur Indo-American Program and Beyond,” U.S. Department of State, accessed June 28, 2022, //2009-2017.state.gov/p/sca/rls/rmks/2010/144465.htm.

[295] “U.S.-India Partnership: Kanpur Indo-American Program and Beyond,” U.S. Department of State, accessed June 28, 2022, //2009-2017.state.gov/p/sca/rls/rmks/2010/144465.htm.

[296] Sinha, 15.

with them (e.g., k)). All Indic scripts also contained both vowels and vowel modifiers, and their alphabets followed a similar classification scheme and sorting order based on how their letters were articulated.

Though India's various scripts "differed in [terms of their] number of consonants and number of vowels, [with] some providing finer-grained articulation and some remaining at a coarser level," they could generally be modeled after one another. Thanks to this realization, Sinha and his colleagues were able to define a superset of all Indic script symbols, which they called the "enhanced Devanagari script."[297] This "enhanced" Devanagari would eventually form the basis of ISCII, the official character encoding standard of India, which would itself be folded into Unicode in 1991 (as discussed in Chapter 1).

Sinha's decision to base this supposedly universal script system on Devanagari, however, did not come without its costs. For many users, it would prove impossible to disentangle ISCII's reliance on Devanagari from the legacy of Hindi-language supremacy in India over the second half of the 20th century (as discussed later in this chapter). Of course, one can also paint a far more innocent picture of ISCII's development. It is possible that Devanagari was chosen as ISCII's "base" script simply because it was India's most widely used script. It is also possible that Devanagari was chosen simply because it was the primary language of the project's lead investigator, Dr. Sinha. For our purposes, however, it is telling that Sinha's move to elevate Devanagari in this way was near incidental. ISCII was the brainchild of a handful of scientists, who only stumbled into the role of developers because they were interested in similarities between scripts that were primarily clinical and scientific in nature.

Om Vikas, the policy maker whose TDIL program would help bring these kinds of language planning issues to the attention of the Unicode Consortium, did not directly participate in Dr. Sinha's research efforts to digitize the Devanagari script. However, when Vikas was completing his PhD at IITK, Sinha was also there as a professor, and was in fact in the middle of launching his language technology research lab, meaning that Vikas was well aware of such efforts.[298]

After Vikas graduated in 1977, he worked briefly as a Systems Engineer at Tata Consultancy Services before moving into a role in the National Informatics Center (NIC) within the Department of Electronics. This was where he met Professor MGK Menon, who had been with the Department of Electronics since the Permit Raj era of computing mentioned in the previous chapter. It was Menon who encouraged Vikas to pursue work in "Informatics for Indian Languages," a nascent policy area at that time as well.[299]

The idea took hold in Vikas. By 1978, Vikas was already organizing a national symposium on the "Linguistic Implications of Computer Based Information Systems." Even Dr. Sinha, one of the most prominent and senior researchers in the field, had to color himself impressed:"This

[297] Sinha, 17.

[298] Sinha, 16.

[299] Om Vikas, "Language Technology Development in India," 2001, 18.

symposium, a landmark in the history of Indian language computing, triggered numerous related research projects in India."[300]

One of the major projects "triggered" by the symposium was the creation of a new language standardization committee, sponsored by the Department of Electronics, which would be responsible for designing a character standard for Indic scripts along the lines of the American ASCII standard. As previously mentioned, this committee relied on the "enhanced" Devanagari script developed by Sinha's language lab to come up its first version of an official multi-script standard – which began in 1982 with a 7-bit code known as ISSCII-7 (Indian Scripts Standard Code for Information Interchange), and subsequently developed into an 8-bit code called ISSCII-8 the following year.[301]

ISSCII-8 represented an important leap forward because it complied with ISO's 8-bit code recommendations, which suggested all script standards — even those developed for non-Latin scripts — retain ASCII's Latin and control characters as their 128 initial codepoints. However, some members of the committee remained unsatisfied, and the 8-bit ISSCII-8 standard continued undergoing revisions. Finally, after further development, the standard was passed along (now renamed ISCII instead of ISSCII, dropping the second 'S' for Standard) to the Department of Electronics, which published the first official version in 1988.[302]

Another, equally important outgrowth of Vikas's linguistics symposium was a major grant to continue Sinha's work on local language computing at IITK, also funded by the Department of Electronics. During the same years that the standardization committee was developing what would become ISCII, between 1983 and 1988, IITK researchers were simultaneously working on a project that would become known as the Integrated Devanagari Terminal, which would finally allow Devanagari to be used on any UNIX computer.[303] The Integrated Devanagari Terminal was later expanded into the GIST (Graphics and Indian Script Terminal) card, which could support all major Indian scripts. GIST was important as being the only example of ISCII in practical use — serving as a touchstone for its proponents to say it was possible when it would fail to gain widespread adoption.

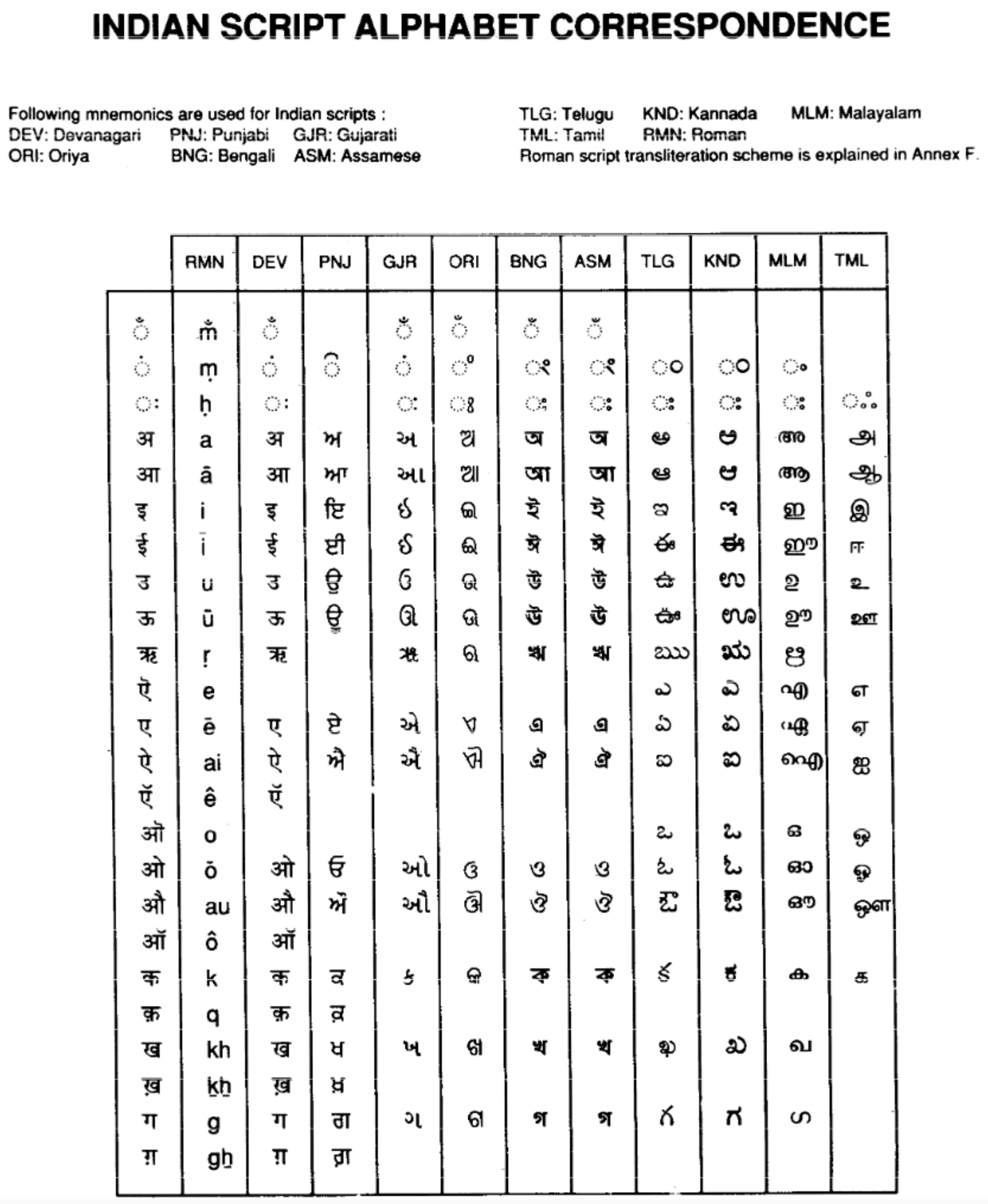
These twin developments — ISCII and the Integrated Devanagari Terminal — made it possible for the first time to view, enter, and process data in Indian languages. It was still just a preliminary model though. Efforts to refine and commercialize this technology were quickly taken up by the Center for Development of Advanced Computing at the Department for Information Technology — one of the government organizations that would be in the eye of the Indic-computing hobbyists as wrong-mindedly devoting resources to closed technologies.

[300] Sinha, 16.

[301] Sinha, 20.

[302] Sinha, 22.

[303] Sinha, 23-24.



INDIAN SCRIPT ALPHABET CORRESPONDENCE

Following mnemonics are used for Indian scripts :
DEV: Devanagari PNJ: Punjabi GJR: Gujarati
ORI: Oriya BNG: Bengali ASM: Assamese
TLG: Telugu KND: Kannada MLM: Malayalam
TML: Tamil RMN: Roman
Roman script transliteration scheme is explained in Annex F.

	RMN	DEV	PNJ	GJR	ORI	BNG	ASM	TLG	KND	MLM	TML
◌ँ	m̐	◌ँ		◌ઁ	◌ଁ	◌ঁ	◌ঁ				
◌ं	ṃ	◌ं	◌ਂ	◌ં	◌ଂ	◌ং	◌ং	◌ం	◌ಂ	◌ം	
◌ः	ḥ	◌ः		◌ઃ	◌ଃ	◌ঃ	◌ঃ	◌ః	◌ಃ	◌ഃ	◌ஃ
अ	a	अ	ਅ	અ	ଅ	অ	অ	అ	ಅ	അ	அ
आ	ā	आ	ਆ	આ	ଆ	আ	আ	ఆ	ಆ	ആ	ஆ
इ	i	इ	ਇ	ઇ	ଇ	ই	ই	ఇ	ಇ	ഇ	இ
ई	ī	ई	ਈ	ઈ	ଈ	ঈ	ঈ	ఈ	ಈ	ഈ	ஈ
उ	u	उ	ਉ	ઉ	ଉ	উ	উ	ఉ	ಉ	ഉ	உ
ऊ	ū	ऊ	ਊ	ઊ	ଊ	ঊ	ঊ	ఊ	ಊ	ഊ	ஊ
ऋ	ṛ	ऋ		ઋ	ଋ	ঋ	ঋ	ఋ	ಋ	ഋ	
ऎ	e							ఎ	ಎ	എ	எ
ए	ē	ए	ਏ	એ	ଏ	এ	এ	ఏ	ಏ	ഏ	ஏ
ऐ	ai	ऐ	ਐ	ઐ	ଐ	ঐ	ঐ	ఐ	ಐ	ഐ	ஐ
ऍ	ê	ऍ									
ऒ	o							ఒ	ಒ	ഒ	ஒ
ओ	ō	ओ	ਓ	ઓ	ଓ	ও	ও	ఓ	ಓ	ഓ	ஓ
औ	au	औ	ਔ	ઔ	ଔ	ঔ	ঔ	ఔ	ಔ	ഔ	ஔ
ऑ	ô	ऑ									
क	k	क	ਕ	ક	କ	ক	ক	క	ಕ	ക	க
क़	q	क़	ਕ਼								
ख	kh	ख	ਖ	ખ	ଖ	খ	খ	ఖ	ಖ	ഖ	
ख़	k̲h̲	ख़	ਖ਼								
ग	g	ग	ਗ	ગ	ଗ	গ	গ	గ	ಗ	ഗ	
ग़	gh̲	ग़	ਗ਼								

Figure 26. Alignment of Indic Scripts in ISCII (Bureau of Indian Standards, 1991)

Returning to Vikas, he would found the Technology Development for Indian Languages (TDIL) in 1991 under the aegis of India's Ministry of Information Technology where he took his next post. When it began, the program was focused exclusively on technology development, especially natural language processing. In this sense, it was a natural outgrowth of the research activities that had already been taking place within Indian universities over the course of the previous decades, particularly at the Indian Institute of Technology in Kanpur (IIT Kanpur, or IITK).

As the new millenium approached, however, Vikas's vision for TDIL began to evolve, and he orchestrated several moves to reflect this broadening vision. Whereas he had previously focused on supporting the government's national efforts at technology development, he now shifted his attention to trying to increase international engagement with companies that had shown themselves to be interested in targeting Indian consumers, such as Microsoft. Even more remarkably for our purposes, he began to situate the technology localization efforts of TDIL within a broader program of Indian development and modernization, quite apart from preexisting government initiatives.

Language Technology Development in India: Launching a New Era of International Cooperation

In a keynote speech he gave in October 2001 at the "Language Technology Business Meet" — a workshop designed to facilitate conversation between Indian policy makers, technologists, and researchers that included panel discussions such as "India as global player in Language technology" — Vikas laid out a remarkable four-part theory of techno-social transformation that drew important connections between local language technologies and human flourishing. It was later published as a whitepaper, "Language Technology Development in India."[304] What was remarkable about this piece was not only its resonance with modernization theory – i.e., the idea that there is a ladder of development that countries can "climb" step by step to become more prosperous, if only guided by the right economic policies – but also its efforts at "localizing" said modernization theory for the Indian subcontinent.

Vikas's keynote address began with a historical account of information technology, starting with the recognition that the past half-century had brought about a massive "paradigm shift from data to information to knowledge processing."[305] In other words, whereas computer science in the 1960s and 70s had largely been limited to processing numeric "data" alone, perhaps referring to statistical analyses of databases, in the 1980s and 90s, computers had become capable of processing "information" as well, such as building natural language processing models. Now, thanks to the computing advances of the 90s, Vikas claimed, "another paradigm shift from Information to Knowledge is taking place."[306] By this, he was referring to the emerging field of"knowledge engineering," which, as he explained, was "an important discipline especially in

[304] Om Vikas, "Language Technology Development in India," 2001.

[305] Vikas, "Language Technology Development in India," 6.

[306] *Ibid.*

the wake of convergence of computing, communication and content technologies."[307] Altogether, according to Vikas, humanity was in the midst of its fourth major information revolution, the first being the invention of writing systems, followed by the invention of the written book, followed by the invention of the printing press, and culminating now with the invention of the computer, which made it possible to communicate in new and unprecedented ways. Because the computing revolution had been concentrated in the West, however, "English [had become the] lingua franca of Science and Technology," which meant that entire swaths of the globe were getting left out of the conversation.[308] For Vikas, this deficiency also presented important opportunities for researchers to extend computing benefits to more of the population. After all, as he argued, over the course of the twentieth century, the time it took for new technologies to reach the masses had become shorter and shorter. After Edison's invention of the lightbulb, it took decades for electrical lighting to be installed in residential homes. Even compared to the radio or the television, the personal computer had reached the average consumer with remarkable speed.[309] Now all that remained was to ensure that all consumers around the globe had equal access. (For a researcher who spent most of his academic career in electrical engineering, Vikas appears to have devoted an astonishing amount of effort to the study of history to craft such a prehistory of his own technological efforts.)

TDIL MEET 2001'
on January 23-25 at
Ministry of Information Technology

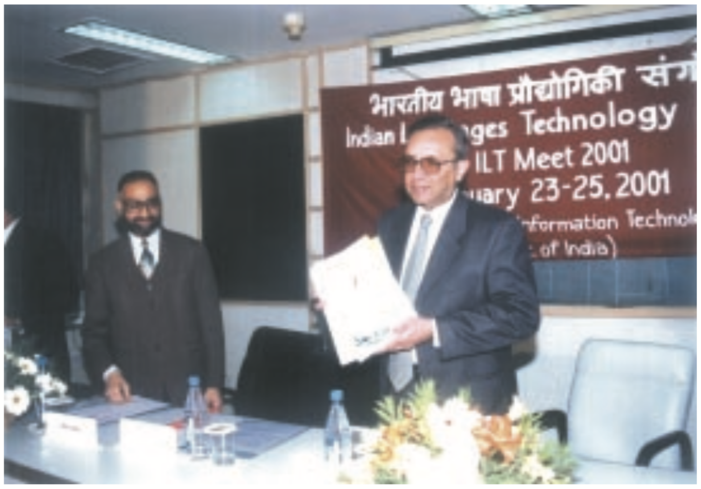


Figure 27. Om Vikas at TDIL Meet 2001 (*Language Technology Flash,* May 2001)

307 *Ibid.*

308 *Ibid.*

309 Vikas, 7.

In addition to researching the history of information technology, Vikas also had also clearly devoted himself to the study of economics. The second component of his keynote concerned the economics of information exchange, beginning with the premise that "Knowledge defies the economic principle of scarcity." In contrast to traditional goods and resources, Vikas argued, knowledge did not run out when more people tried to make use of it; on the contrary, in the case of knowledge, "the more you use it and pass it on the more it proliferates."[310]

On the basis of this presupposition, Vikas offered an equation for modeling the relationship between Technology (T), Economic Development (E) and Knowledge creation capacities (K), by which he meant its cultural resources — akin to a model one might learn in an introductory macro-economics class. In his conception, Technology (T) promised to boost Economic Development (E), but was just as likely to decrease overall Knowledge (K) as to increase it. In what cases might technology make cultural resources dwindle rather than flourish? Under circumstances in which technology had been dispersed unevenly across populations, leaving behind entire languages, literatures, and fields of expertise. In this new digital age of knowledge engineering, Vikas argued, knowledge would only be able defy the economic principle of scarcity if the *appropriate technology* was developed, capable of transforming "'digital divide' into 'digital unite.'"[311]

It's worth noting that Vikas's use of the term "appropriate technology" dates back to an influential 1973 essay by economist E. F. Schumacher, entitled "Small is Beautiful."[312] Schumacher's original concept, "intermediate technology," referred to tools that were simple enough for on-the-spot maintenance and repair – such as a hand-powered water pump, or a self-contained solar lamp. These tools needed to be cheap enough for anyone to afford, and accessible enough for anyone to use. Schumacher envisioned intermediate technologies as "helping people in the non-modern sector," by which he had something very specific in mind. According to Schumacher, the point of these intermediate technologies was not just their economic utility (e.g., filling infrastructure gaps in resource-poor areas where wealth was limited), but also their simplicity, which Schumacher envisioned as a protest against modernization writ large, which had made people "alienated from nature," deceived by the "illusion of unlimited power, nourished by astonishing scientific and technological achievements."[313] Intermediate technology was about a "more affective [e.g., emotionally astute] notion of production," which is to say, technology "with a human face." Inspired by the Buddhist tradition and the teachings of Mahatma Gandhi, which decried "machines that concentrate power in a few hands and turn the masses into mere machine minders,"[314] Schumacher had even

[310] *Ibid.*

[311] Vikas, 3.

[312] E. F. Schumacher, *Small Is Beautiful: Economics as If People Mattered*, Reprint edition (New York, N.Y: Harper Perennial, 2010).

[313] Schumacher, 5.

[314] Schumacher, 20.

published a work entitled *Buddhist Economics*, which combined questions of economic development with spiritual values like simplicity and egalitarianism.[315]

This context is striking in that the movement for appropriate technologies has deep roots in South Asian traditions and values, but was disregarded by Vikas, who reworked the concepts in service of his own vision of Indian development and ascension. These were outlined in the third component of his keynote: the "ABC Technology Development Phases."[316] According to Vikas, because "India [had always been] aware of the technological changes and the local constraints" of its circumstances, its original technology strategy had been to focus primarily on "**A**daptation," during the "A-Technology" phase from 1976-1990. Adaptation techniques included "abstraction of requisite technological designs and competence building in R&D institutions." This period, we can safely assume, included the development of the ISCII standard and GIST modules at major Indian research institutions. The next "Basic," or "B-Technology," phase lasted from 1991-2000, and was focused on ramping up language technology initiatives, including government-sponsored industry efforts (e.g. Indic-language word processors developed by Modular Infotech, the most prominent producer of proprietary language technologies in India).
According to Vikas, "**B**asic Technologies" included "generic information processing tools, interface technologies and cross-compatibility conversion utilities." Also, he added proudly, during this period, his own "TDIL program was initiated." Finally, the "Creative," or "C-Technology" phase, would mark the years from 2001-2010 defined by "developing **C**reative Technologies in the context of convergence of computing, communication and content technologies. Collaborative technology development is being encouraged to realise."[317]

More than ever before, this new era seemed to signify a greater desire to connect with international standards and multinational corporations, once considered to be agnostic or even hostile to Indian interests, but now seen as India's most promising path forward. It's worth noting that Vikas's "ABC Technology" periodization did not precisely map onto the historical account of broader information technology that he had begun his talk with. This is because Vikas's historical account concerned global stages of computing development, whereas his ABC phases referred to India's own language technology efforts – which were lagging slightly behind those of the rest of the world, but remained responsive to new computing developments.

By the early aughts, the TDIL program had become lauded around the world. By this point, it seemed, TDIL and the Ministry had begun sending copies of the newsletter to all potential stakeholders they could identify, Western and Indian. Its quarterly newsletter, renamed *Vishwa Bharat*, or "India Together," began its July 2002 issue with appreciative comments from international readers.[318] One professor from the Indian Institute of Science wrote in, "Thank you

[315] Schumacher, 35.

[316] Vikas, 8.

[317] Vikas, 8.

[318] "Language Technology Flash," *Vishwa Bharat@TDIL*, July 2002.

for sending me copies of the newsletter. These issues are excellently produced and carry a lot of useful information."[319] Similarly, a Cambridge University professor wrote, "Nowadays there is so much work under way on various aspects of the computerisation of Indian languages that no one person can hope to keep up to date with it all. Your Newsletter acts as a very valuable clearing house for this information."[320]

Others, such as a representative from the Instituto de Lengua y Cultura Aymara (an organization dedicated the Native American language known as Aymaran), even expressed the hope that TDIL might serve as a model for their own language-planning efforts: "We in ILCA are very interested in the advances in language technology that you are making, and wish to know more about them, to see how we can apply similar measures here."[321] Finally, topping the list of prominent officials was a representative from UNESCO, the cultural division of the United Nations, who wrote in that the newsletter was "very informative and may be useful for developing our programme Initiative B@bel" (dedicated to advancing "multilingual education" and "language on the internet").[322]

Yet in some sense, TDIL's efforts were hardly as novel as such responses made them seem. The TDIL program may have only officially begun in 1991, but its philosophies drew on a long history of language development on the Indian subcontinent. The language politics of the past three centuries provided the discursive context – the terminology – that allowed programs like TDIL to proceed. They also served to motivate certain changes in TDIL's all-India approach, such as advancing multilingualism. In the following sections, I trace the evolution of the language regimes that characterized the Indian subcontinent in the colonial and post-independence eras, including shifts in both the centers of linguistic authority and the stakes associated with each language. I review the language politics of South Asia at large in the following sections, but take an eye towards the subregion of Bengal, and later India and Bangladesh in the modern nation-state era.

[319] "Language Technology Flash, July 2002, 5.

[320] *Ibid.*

[321] "Language Technology Flash, July 2002, 5.

[322] *Ibid.*

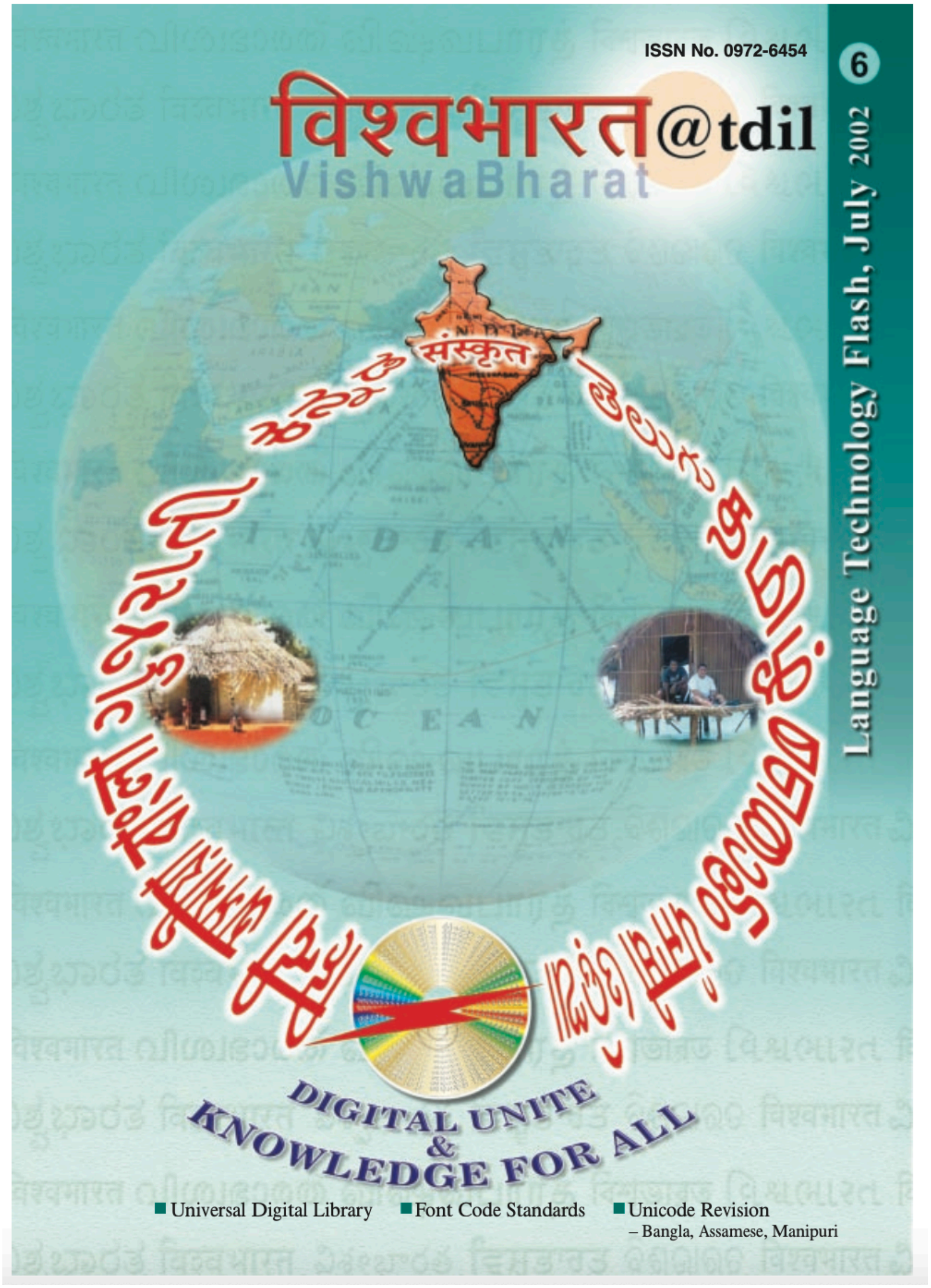


Figure 28. Cover of *Language Technology Flash, July 2002*

The British Colonial Era: Classification and Control

Prior to the imposition of British rule in India, the subcontinent was ruled for nearly two centuries by the Mughals, who had reached it by traveling East from the Ottoman Empire. Mughal control of India lasted from 1576 to 1757. During this period, Bengal represented the country's wealthiest and most industrially developed region, accounting for no less than 12% of the entire world's GDP.[323] Mughal rule introduced the Indian subcontinent to Persian, known as the language of "love, culture, literature, poetry, diplomacy, music, and charm."[324] Persian also served as the high-status language of governmental administration. It was necessary to learn for those performing administrative jobs, but otherwise coexisted with Sanskrit and many other local languages throughout the region.[325] By the end of Mughal rule, Bengal had become a quasi-independent state governed by the hereditary rulers known as the Nawabs of Bengal, whose use of Persian in high offices had a significant influence on Bengali literature, which began to incorporate Persian vocabulary into Bangla prose.[326] At the same time, there was no coordinated effort to constrain the use of Bangla in everyday life. To further complicate this picture of linguistic hybridity, Portuguese missionaries based in the colony of Goa also began printing local language books during this period, including Bengali-Portuguese dictionaries and grammars.[327]

After first setting foot in India in the early 17th century, the British East India Company overtook the Mughal Empire as India's primary administrators in 1757. That was the year of the Battle of Plassey, which brought the British East India Company a "decisive victory" over the Nawabs of Bengal, and ushered in a new era of British colonialism.[328]

In the ensuing years, British "Orientalists," or scholars of the supposedly "exotic" cultures and languages of the East, would establish linguistic hierarchies on the Indian subcontinent that still largely stand to this day. For these "Orientalists," part of the point of studying Indic languages was to advance certain overarching linguistic theories of their own.[329] By the late 18th century, European linguists had become concerned with mapping families of related languages, which sent them on a quest for the world's oldest and most supposedly "pure" languages. In this vein, linguist William Jones's "Third Discourse on the Hindus," which he presented to the British-colonial institution known as the Asiatic Society in 1786, proclaimed the "marvellous [sic.] structure" of Sanskrit grammar, and declared that its antiquity seemed to supersede that of both

[323] "Empire, Mughal - Document - Gale In Context: World History," accessed June 30, 2022,

[324] Robert D. King, *Nehru and the Language Politics of India*, 13.

[325] *Ibid.*

[326] Hanne-Ruth Thompson, *Bengali: A Comprehensive Grammar* (Taylor & Francis, 2010), 10.

[327] M. Siddiq Khan, "The Early History of Bengali Printing," *The Library Quarterly: Information, Community, Policy* 32, no. 1 (1962): 51–61.

[328] "Memoirs of the Revolution in Bengal, Anno Dom. 1757: | Library of Congress," accessed June 30, 2022, https://www.loc.gov/item/94840377.

[329] Khan, 53-4.

Latin and Greek.[330] After this revelation, India became a gold mine for researchers looking to develop their own pet linguistic theories. Jones's discovery suggested that Europeans could study Sanskrit to learn about themselves, and thereby elevated Sanskrit to the highest status in the hierarchy of Indic languages, with all other local languages relegated to "vernaculars" presumed to be degenerate products of Sanskrit's original "purity."[331]

Meanwhile, missionaries and typographers during this period began printing books from law and religion in various local languages. These printing practices were critical to establishing local vernacular literatures, and, in many cases, helped standardize local scripts.[332] But they also had their insidious angle. In some areas, British officers had been ordered to learn local languages to aid in their administration, and works printed in the vernacular came to serve them as lesson books or primers on the road to more efficient colonial exploitation. Indeed, according to historian M. Siddiq Khan, during Company rule, the extensive production of vernacular books was "aimed at destroying traditional patterns of authority through supplanting the Persian language which had been the official tongue since the days of the great Moguls."[333] Languages such as Bangla were purged of their Persian vocabulary, and forced to incorporate Sanskrit terms instead.

The "Orientalists" responsible for manipulating Bangla, Persian, and Sanskrit in this way were operating under the assumption that British governance would be easier, more efficient, and more effective if conducted in vernacular languages. Some even claimed to harbor an appreciation for vernacular languages on their own merits. Within the Company administration, however, these so-called "Orientalists" were opposed by those known as the "Anglicists" (or "Utilitarians"). The Anglicists insisted that colonial administration should be conducted entirely in the English language and in Roman script, mostly out of distaste for the "vernaculars."[334]

A fierce debate raged between Orientalists and Anglicists in the press throughout the early 19th century, culminating in an infamous 1835 memorandum written by Sir Thomas Macaulay entitled the "Minute Upon Indian Education." In it, Macaulay wrote disparagingly of Indian culture, claiming that a single shelf from a good European library was superior to the entire body of literature ever produced by India and "Arabia". English, he insisted, should be advanced as India's sole language of administration and education, which would in turn form a class of elite Indians — "English in taste, in opinions, in morals, and in intellect" — who could take up

[330] Annie Montaut, "Colonial Language Classification, Post-Colonial Language Movements, and the Grassroot Multilingualism Ethos in India.," in *Living Together Separately: Cultural India in History and Politics*, 2005.

[331] Khan, 53-4.

[332] Robert D. King, Nehru and the Language Politics of India.

[333] Khan, 53.

[334] Ramachandra Guha, India After Gandhi: The History of the World's Largest Democracy, Reprint edition (New York/N.Y: Ecco, 2008), 155.

positions within the Company and work to advance its interests.[335] Macaulay's open contempt for Indian culture notwithstanding, it's worth noting that many upper and middle-class Indians at the time also favored English education, seeing it as a means of raising their own social status, since they had access to English-language instruction and stood to benefit from that privilege.[336] The English Education Act of 1835 made Macaulay's recommendations official, transforming English from the language of India's foreign rulers, to one of the country's own official tongues, as indeed English would be declared post-independence.

By the turn of the 19th century, English had become so deeply entrenched in India that even the leaders of India's independence movement would generally converse with one another in English, as they were used to discussing affairs of state in that language (although they could and did still greet their friends and family in vernaculars).[337] What's more, the British had succeeded in stratifying India's languages into two categories: "pure and ancient," and "corrupted vernaculars," despite the fact that many so-called "vernaculars" had extensive independent histories of their own. In this context, "vernacular" itself became a loaded political term. Indeed, by the turn of the 20th century, many Indian nationals would rebel against the use of the term altogether, including the hobbyists discussed in the previous chapter. After their efforts were termed 'vernacular computing' in a press article, one member posted a rebuttal -- and a plea: "This was the word [i.e., vernacular] used by British while referring to the languages used by Indians, Africans, etc. meaning the language of slaves…I humbly request all patriotic Indians to refrain from using the word 'vernacular.'"[338]

The next important development in India's language policy came after another decisive battle that occurred precisely one hundred years after the beginning of British colonial rule: the 1857 Sepoy Mutiny. The Sepoy Mutiny refers to a large-scale uprising against the rule of the British East India Company by infantrymen in the Company's army. Though the mutiny failed, it was perceived by the Crown as a failure of the British East India Company to adequately govern its dominion. As a result, the British parliament passed the Government of India Act the following year, which established the British Raj. Under this new system, India would be administered directly by the British government, rather than by their proxy, the British East India Company, which carried important implications for the administration's approach to linguistic data collection.

The British Raj era of direct colonial rule transformed local understandings of language and identity through a variety of enumerative activities. One of the most critical of these activities was the Linguistic Survey of India, first proposed in 1886 by linguist and member of the Indian Civil Service, George Abraham Grierson. The Linguistic Survey, which was conducted annually

[335] "Minute on Education (1835) by Thomas Babington Macaulay," accessed June 30, 2022, http://www.columbia.edu/itc/mealac/pritchett/00generallinks/macaulay/txt_minute_education_1835.html.

[336] Acharya, 259.

[337] King, "Introduction."

[338] Pavanaja, U. B. "[Indic-computing-users] Vernacular" Email, April 25, 2003.

from 1894 to 1928, gathered important information about natives' "mother tongues" that could be used to more efficiently administer — and exploit — these populations.[339]

Despite the longstanding use of data from the Linguistic Survey of India to help govern the country into the 21st century, the uniformity and precision of the survey have since been called into question. This is in part because the assumptions of the survey did not adequately capture the nuances of language use on the ground. Before the British began collecting census data, Indians tended to practice what sociolinguists call "grassroots multilingualism," or switching languages as needed depending on their locale.[340] Montaut describes the classic example of "the Gujarati merchant who uses Kacchi (a dialect of Gujarati) in the local market, Marathi for wider transactions in the region, standard Gujarati for readings, Hindustani when he travels (railway station), Urdu in the mosque, with some Persian and Arabic, but also *sant bhasha* in devotional songs, his variety of Gujarati for family interaction, English when dealing with officials."[341]

Given this rich multilingual context, Indians understandably had a difficult time telling survey collectors what should be considered their proper "mother tongue." Local language use was simply "more intuitive," with more "fuzzy ways of locating" which language should be used in which context.[342] Over time, however, the Linguistic Surveys enumeration practices itself worked to help inculcate a "radically new representation of the relation of the speaker to his speech," indoctrinating everyday Indians in the belief that they ought to have "one language, one name, one identity."[343] For this reason, according to sociolinguists, the British Raj brought about a new and unprecedented "linguistic consciousness [that] seemed to have stemmed from the classificatory passion of the colonial agenda," which in turn would carry significant repercussions when it came to redefining Indian national identity in the independence era.[344]

The Independence Era

As we have already seen, over the course of the 20th century, language became increasingly intertwined with the sphere of politics, turning language itself into an essential component of national identity. Two core issues continued to plague India's political leaders up to and through the 1947 Partition splitting British India into two independent nation states: first, the question of national language, and second, the question of subnational linguistic states. These issues affected both India and Pakistan, eventually leading to policies affirming regional linguistic identities in India, and to the creation of the new nation state of Bangladesh from the former Pakistan.

[339] Montaut, 87.

[340] Montaut, 99.

[341] *Ibid.*

[342] Montaut, 87.

[343] Montaut, 85.

[344] Montaut, 87.

Before delving into these issues, however, it is important to understand the radical shift that national self-determination represented in comparison to the language politics of the preceding centuries. According to historian Robert D. King, it is only relatively recently that the boundaries of the nation-state were made to match up with the boundaries of a linguistically homogeneous population; prior to that, it did not matter if someone might be speaking a different language as they were ruling over you, as was the case in Mughal and British India. With the global ascension of nationalist ideology in the late 19th century, however, locals in India — much like locals in Europe — found it increasingly intolerable to have one language imposed upon speakers of another.[345]

In addition to the rise of nationalist ideology, national identity also came to play a critical role in transforming India's language policies. As linguists and historians have emphasized "the political link, more or less artificially created, between language and political or administrative needs."[346] As Montaut writes, "if we see language not merely as a tool for communication, nor even as a way of enacting one's social role(s), but as a means of asserting one's cultural or religious identity and an icon for a group identity, one can understand how it can become an intensely burning issue."[347] Indeed, as I discuss in the following chapter, the Unicode debates that roiled India in the 2000s had much less to do with technical specs than with deeply-felt beliefs about "cultural" and "group identity." So deep did the emotional valence of "cultural identity" run that language came to take on symbolic or even spiritual resonance as the core essence of what made Bengalis Bengali, and Indians Indian.

Perhaps understandably, at the time of India's inauguration as an independent nation-state, the dominant concern of its leaders was maintaining national unity despite the country's enormous diversity of religious, linguistic, and cultural differences. Religious differences had already resulted in widespread violence in the years leading up to the 1947 Partition, and had only become worse during the actual process of imposing the new borders.[348] In theory, the 1947 Partition divided the Indian subcontinent into two nation-states based on religion: Pakistan would serve as the Muslim homeland, and India would cover everyone else (mostly Hindus, but also a substantive Muslim minority, along with a number of Jains, Christians, Buddhists, and Sikhs). Dividing the two countries based on religion reflected a similar set of values as those that had already affected the region's linguistic identities: first, the enumerative practices of the British Raj had entrenched group identities among its colonized subjects, and second, the administrative practices of the British Raj had tied resource allocation and political power to these identities (as with "separate electorates" assigned at one point to Hindus and Muslims in the region).[349]

[345] King, 25.

[346] Montaut, 85.

[347] Montaut, 85.

[348] Guha, 222-245.

[349] *Ibid.*

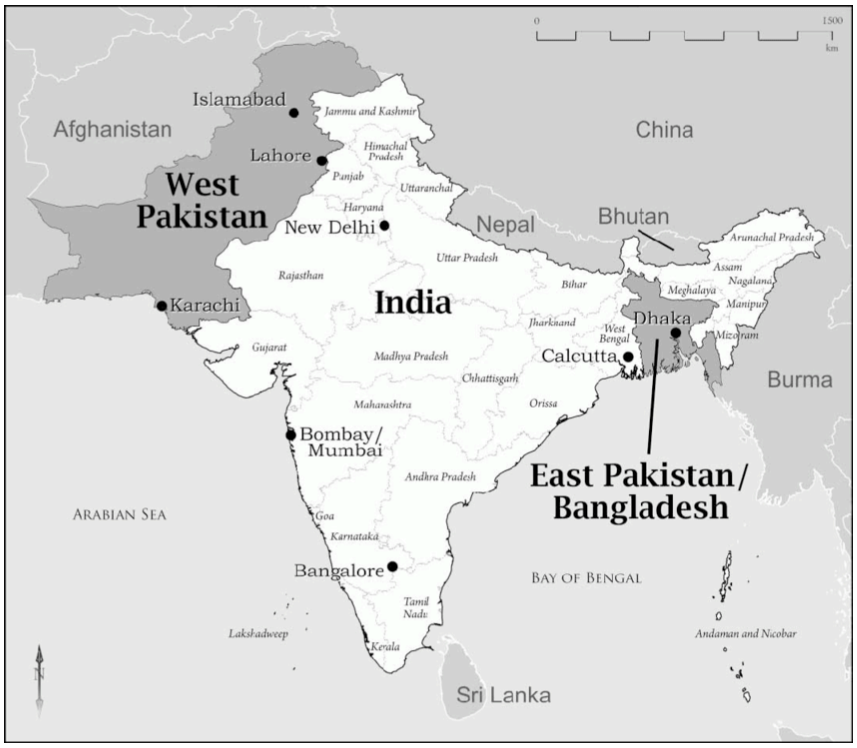


Figure 29. Map Of British India At 1947 Partition (Map By Julius Paolo)

In much the same way, during the first two decades of independent governance on the Indian subcontinent, language would prove to be a serious source of political disorder. First was the issue of linguistic states: should the country be subdivided into separate localities on the basis of language (which would mean, in essence, subdividing it on the basis of ethnicity)? This so-called "linguistic principle" had already been tried by the British when they had partitioned Bengal into Eastern and Western wings back in 1905, hoping to quash the region's anti-colonial agitation. There, the linguistic principle was used to justify transferring Oriya-speaking communities out of Bengal into Assam and Orissa. Although furious local protests had led to the reunification of Bengal's two wings within a decade, this first Bengali partition would have lasting linguistic effects, not least of which was making the Bengal presidency synonymous with the Bangla language.[350]

In a similar vein, in the 1930s, the idea of subdividing India into smaller linguistic states had attracted the attention of the Indian National Congress, the political party shepherding the new country into independence. In 1948, however, the committee that had been appointed to investigate the feasibility of linguistic states, known as the Dar Commission, strongly advised against the idea, arguing that such linguistic division would not be "in the larger interest of the

[350] Montaut, 85.

nation… [because it] would create new minorities."[351] At the time, India was already operating as a federation of smaller states, but each of these were linguistically heterogeneous, especially the area of modern-day Tamil Nadu (then known as Madras), which included speakers of Tamil, Telugu, Malayalam, Kannada, and others.

This uneasy status quo would be utterly upended on October 19, 1952. On this pivotal historic milestone, revolutionary activist Potti Sriramulu started a fast unto death in support of a Telugu-speaking state. As leader of the Andhra movement, Srimamulu represented an organization dating back to 1913: the Andhra Mahajana Sabha, or Mayajana Socialist Party. From the earliest days of its existence, this movement had demanded that Telugu be instituted as the [sole/primary] language of instruction in schools, and that it be granted separate administrative status in recognition of the majority presence of Telegu speakers in the region. The movement reached its tipping point in July 1952, when representatives of the Madras province officially filed a motion in the National Assembly for a Telugu-speaking state. While the motion gained the support of several Congress members, it was ultimately rejected by supporters of then-Prime Minister Jawaharlal Nehru, who argued that linguistic states were far too risky a prospect in the newly-independent and precariously-united India. In response, Sriramulu continued to fast for a total of fifty-eight days, all the way until his death on December 15, 1952. To the shock and dismay of Nehru his compatriots, Sriramulu's martyrdom threw the entirety of Andhra into widespread violence and devastating chaos. After two days of unchecked destruction, including several deaths, Nehru acquiesced to the demand for a Telegu-speaking Andhra state, which was formally inaugurated on October 1, 1953.[352] Nor was that the end of Sriramulu's influence. The Andhra movement for statehood proved historically significant across the continent, resulting in waves of emulative language movements. In 1953, a State Reorganization Commission recommended the creation of new linguistic states in light of this unrest, which resulted in the formation of fourteen new states by 1956.[353] Sociolinguist Annie Montaut has gone so far as to call this a "never-ending process of secession, if not balkanization, with a continuous creation of new minorities enduring increasingly worse conditions."[354] As earlier chapters in this dissertation have already explained, similar popular protests would occur in Pakistan as well, leading to the creation of Bangladesh as a new ethnolinguistic nation state, and similar language movements by other constituencies as well (though no others would result in secession).[355]

In India, there still remained the question of the country's overarching national language, which produced contending views amongst political leaders in the years leading up to Partition. Some wanted to promote Hindustani, a lingua franca of numerous Indian dialects that had been invented as a "golden mean" to bridge the Hindi commonly spoken in Northern India with the

[351] Guha, 225.

[352] Guha, 232.

[353] *Ibid.*

[354] Montaut, 87.

[355] Tariq Rahman, *Language and Politics in Pakistan* (Karachi: Oxford University Press, 1997).

Urdu commonly spoken in the South.[356] But this compromise position came with its own challenges: Hindus in the Indian National Congress supported the use of the Devanagari script for Hindustani, whereas the minority Muslim population wanted to see Hindustani written in Arabic script instead. In an attempt to bridge this seemingly insurmountable divide, some staunch supporters of Hindustani even proposed using the Latin script to transliterate the language, despite the uncomfortable colonial overtones of such a strategy. However, the idea didn't go far. As Mahatma Gandhi wrote at the time, India could only find true freedom by fully extricating itself from the destructive legacy of British rule, including the "crucial instrument of colonization, namely the English language."[357] As opposed to utilizing Latin script, then, Gandhi called for elevating "vernacular languages" instead - including granting schoolchildren the right to education in their mother tongue – and declaring Hindustani the national language of the country as a whole, written with the Devanagari script that the greatest number of Indian citizens would be able to understand.

Over the course of constitutional negotiations, legislators came to a compromise: rather than using Devanagari script for Hindustani, which actively excluded Muslim speakers, they would simply use Devanagari script for Hindi, the language it was normally associated with, and declare that the so-called "Official Language of the Union," with *no* language claiming the status of the country's "national language."[358] "Vernaculars" would be used in local schools and offices; Hindi would be used in federal courts and legislatures. To help appease Urdu-speaking South Indians who might feel shortchanged by this arrangement, legislators promised that federal institutions would also use English during an initial fifteen-year period. Over those fifteen years, officials would attempt to grandfather in Hindi across the nation through educational initiatives.

When that transitional period was up in 1965, however, and the time actually came to make the switch to Hindi-only institutions, major anti-Hindi protests broke out across the state of Tamil Nadu, marked by rampant violence between police and students. Self-immolation emerged as a common protest tactic. After a full month of uninterrupted chaos, agitation, national leaders declared English another official language of the Indian people, giving the language equal status with Hindi for the indefinite future.[359] Although this move perpetuated British colonial influence, it also served as an equalizing measure for India's non-Hindi speaking population.

Even as it was first debating the question of the country's "official language," Congress also identified fourteen major regional languages that would be designated "scheduled languages." These "scheduled languages" were each granted an official representative on the Official Languages Commission, which would steer national language planning policies in the years to come.[360] Speakers of "scheduled languages" were also granted the right to conduct local

[356] Guha, 225.

[357] Guha, 232.

[358] Guha, 389-92.

[359] *Ibid.*

[360] Montaut, 99.

education in that language. Although seemingly pluralistic, this institutional recognition of certain languages actually sparked significant competition between language groups – those who had *not* been included started to fear that they might be endangered.[361] Yet widespread recognition of local languages, along with the establishment of dedicated linguistic states, also led to compromise, cooperation, and a greater sense of national unity. Indeed, according to King, by the 1980s, "language [was] no longer the threat to the national unity of India that it was once considered to be."[362] In the modern era, India has come to embrace multilingualism at every level of governance: road signs in the region are often written in as many as four different scripts; educational curricula are designed to meet the specific needs of localized language communities; and every provincial state can "claim to have a literature, a history."[363]

By the 1980s, then, India, along with South Asia more broadly, had gone through several significant shifts with respect to language planning. A relatively ad-hoc "grassroots multilingualism" had given way to a "count and conquer" policy under British rule. This, in turn, exacerbated ethnic and linguistic cleavages to the point that post-Partition India faced literal language riots. The newly independent nation-state had to navigate rivaling tensions between monolingualism and state-sanctioned multilingualism, which mapped onto the tensions between national unity and pluralist political representation. Though I have chosen to focus primarily on post-independence language politics in India, many of the same conflicts regarding national language and the status of linguistic states were also unfolding in Pakistan, resulting in the totally independent linguistic state of Bangladesh separating from East Pakistan in 1971.

In recent years, scholars have come to situate the overarching practice of "language planning" during these volatile years within more specific historical contexts. Language planning, they argue, was not an abstract and idealized set of policy decisions, but instead a historical and culturally contingent phenomenon affected by the broader turmoil occurring in postcolonial states during its peak throughout the 1950s and 1960s. According to sociolinguist Jiří Nekvapil, "The issue of "language planning" arose only in connection with the decline of the colonial system and the processes of modernization in the developing countries," a context in which "it was considered a type of societal resource planning… firmly anchored in the theory of social and especially economic planning of the time."[364] Sociolinguist Jan Blommaert has similarly argued that the "decolonization of huge parts of the world after WWII (and especially in the 1960s) and the introduction of the paradigm of 'development' entailed invitations to experts worldwide to contribute to the development and modernization of third world societies."[365] Increasingly, researchers emphasize that language planning evolved to deal with the problem of multilingual,

[361] Montaut, 99.

[362] King, xii.

[363] Guha, 243.

[364] Jiří Nekvapil, "From Language Planning to Language Management," *Sociolinguistica Jahrbuch (2006)* 20, no. 2007 (February 20, 2007): 92–104, https://doi.org/10.1515/9783484604841.92, 92.

[365] Jan Blommaert, "Language Planning as a Discourse on Language and Society: The Linguistic Ideology of a Scholarly Tradition," *LANGUAGE PROBLEMS & LANGUAGE PLANNING* 20, no. 3 (1996): 199–222, 199.

multi-ethnic states – states that could only become unified and distinctively *modern* through the promotion of a single national language in precisely the vein of 'one nation, one people, one language.'

Despite the nationalistic insistence on linguistic unity as a guarantor of national unity, however, monolingualism has largely given way to a less dogmatic view of national language, thanks in large part to the persistent multilingualism of major postcolonial states like India and South Africa. Whereas former Prime Minister Nehru and his compatriots pushed for a single national language to unite all of India, and tried to forge state lines based on factors other than language usage to prevent linguistic fracturing, grassroots protests demanded otherwise.

Only in the full historical context of Indian language politics can we understand what was so significant about the language technology policy that emerged in India in the late 20th century. Even seemingly impersonal technical codes, such as ISCII, can hold powerful emotional resonances among those who were subject to Hindi-centric government policies as recently as one generation ago, and even an obvious coincidence or oversight, such as basing ISCII on the Devanagari script, can awaken a strong sense of grievance amongst non-Hindi language communities tired of being left out of the national conversation.

Yet by the time of Vikas' TDIL program, India had already largely embraced multilingualism at the highest administrative levels, including scheduling multiple regional languages and designating several of them "classical languages" in acknowledgement of their long histories and vibrant literatures. This historic commitment to multilingual forms of expression would in turn be codified into multi-script software schemes, such as ISCII and the GIST terminal discussed previously in this chapter. Throughout the remainder of this chapter, we will see TDIL leadership interrogating Unicode not only on the needs of India's Hindi speakers, but those of all of the country's major language communities. In other words, in the new era of technological development, multilingualism is no longer characterized as a symptom of backwardness, but instead as an essential tenet of modernization. Vikas's keynote situates multilingualism as an undeniably enriching force in Indian culture to date, and asserts that finding multilingual technology solutions will be part of the journey that raises both economic prosperity and human flourishing in India's future.

TDIL and Unicode: A Language Politics for the Internet Age

Having covered the high-stakes language-politics implications at play, we turn now to the confrontation between TDIL and external organizations such as the Unicode Consortium. By 2000, Vikas had decided that India needed to be a voting member of the world's major international committees on multilingual computing — namely, the Unicode Consortium and the W3C, or World Wide Web Consortium, both of which were volunteer-led organizations responsible for overseeing open standards. Vikas's insistence on Indian representation was a remarkable, unprecedented move, reflective of the new language politics of the digital age. Up to this point, both consortiums had been run by private software companies alone, and hence had been dominated by North American and European perspectives. Vikas, however, was able to exploit the unique structure of these volunteer organizations, which were themselves a novel trend in the technology sector. In theory, membership was open to any institute or individual, so long as they paid the appropriate membership dues — but these dues ran the rate amount of $10,000 USD per year at the time (although this rate has changed since). As critics have rightly noted, industry-led consortia were democratic and open typically only in name.[366] Would-be participants also faced high barriers to entry in the form of rarified technical expertise, which stood in the way of the consortia's supposed goals. In this context, India's unusual engagement with Unicode presents an alternative vision of the possibilities of multilingual international collaboration with the Global South.

Notably, Unicode's membership structure contrasted sharply with that of other international multi-stakeholder organizations at the time, particularly that of the Unicode Consortium's sister committee, the International Organization for Standardization (ISO) Working Group 2. ISO was an international treaty organization, born in 1947 from the rebuilding efforts following World War II, and originally had little to do with computing; members focused on setting uniform mechanical engineering standards instead ISO was based in Geneva, Switzerland, and thanks to its heritage as an Allied-forces organization, was largely a "European Club" at the time of its founding.[367] Over the course of the 1950s and 60s, ISO membership steadily grew thanks to the implementation of discounted membership fee programs, along with direct outreach to developing countries.[368] Then, throughout the 1960s and 70s, ISO's Information Technology subcommittee (known as Joint Technical Committee 1, or JTC1) developed various international character code standards, including ISO 8859 (used for switch codes) and ISO 646s (used for national character codes).

Ultimately, this character code subcommittee/working group (SC2/WG2) would become responsible for beginning to design an international character code (ISO 10646) in the late 1980s that would be similar in aim to the Unicode Standard. After Unicode version 1.0 was released in October 1991, however, the Unicode Consortium and ISO WG2 began to cooperate with one another in earnest so that they could establish a truly common standard. Specifically, the two consortia agreed to pursue "synchronization," wherein both standards would be aligned to have identical content, although each would still be maintained by its respective organizations. Unicode's next two releases, Unicode 1.1 and 2.0 (which came out in June 1993 and July 1995, respectively), aimed to align Unicode more closely with ISO, and hence revised or updated their component elements so as to match existing ISO standards.[369] After that point, synchronization took the form of reviewing proposals for new additions to Unicode at the annual Unicode Technical Committee (UTC) meeting, and then bringing those same proposals to the

[366] Laura DeNardis, *The Global War for Internet Governance* (Yale University Press, 2014), https://www.jstor.org/stable/j.ctt5vkz4n.

[367] International Organization for Standardization and Central Secretariat, *Friendship among Equals: Recollections from ISO's First Fifty Years.* (Geneva: ISO Central Secretariat, 1997).

[368] ISO, 46.

[369] Isabelle Zaugg, "Digitizing Ethiopic: Coding for Linguistic Continuity in the Face of Digital Extinction" (PhD diss., American University, 2017, 65.

annual ISO SC2/WG2 meeting for final review and acceptance into ISO 10646. In this way, the two standards remain utterly in sync.

These inter-consortia dynamics are important to understand because they represent two competing modes of engagement for those interested in shaping open-source standards. As an international treaty organization, ISO structures its membership by granting equal representation to various national delegates. Two of these representatives, who are chosen from their countries' respective national standards organizations, show up to vote each year on the next iteration of ISO 10646. ISO annual fees are somewhat opaque, determined by algorithm weighing a prospective member country's "economic importance."[370] In contrast, Unicode is agnostic to each of its member's affiliations; any dues-paying applicant, whether it be an individual, a company, a research institute, or a government body, can join the Unicode Consortium, as long as they pay the preset fee ($10000 during the events of this dissertation; $35 for students to $21000 for corporations at time of writing).[371] In contrast to ISO's older, Cold War-era model of mutual governance, Unicode's ad-hoc, free-market approach reflected the popularity in the 1990s of other industry-led voluntary consortia such as the Internet Engineering Task Force (IETF) and World Wide Web Consortium (W3C).

Now we can appreciate just how remarkable it was to see India's Ministry of Information Technologies join the Unicode Consortium as a full voting member in the year 2000, after having already become an ISO member in 1987. Not only that, but when India joined Unicode, it did so at the *voting member* registration tier, even though actual votes were rarely carried out at the annual UTC meetings. This was because Unicode's highly procedural, consensus-based approach kept truly contentious issues from ever coming up for debate; instead, controversial motions would be repeatedly tabled, which meant that some recurring discussions would drag on for multiple years without ever coming up for a decisive vote.[372] Even if becoming a voting member didn't grant India any significant influence over Unicode's decision-making process, it did get India in the door at the annual UTC meetings, along with granting India access to Unicode's internal mailing list and private document registry. India's choice of membership tier more importantly showed, as many UTC members acknowledged, a serious engagement with Unicode's goals.[373] Many private companies, in contrast, had become paying members at lower tiers, allowing them to sponsor and support Unicode's work more broadly, without necessarily having to engage with Unicode's specific character proposals. In all of these ways, India's membership represented an unprecedented level of participation in Unicode's aims.

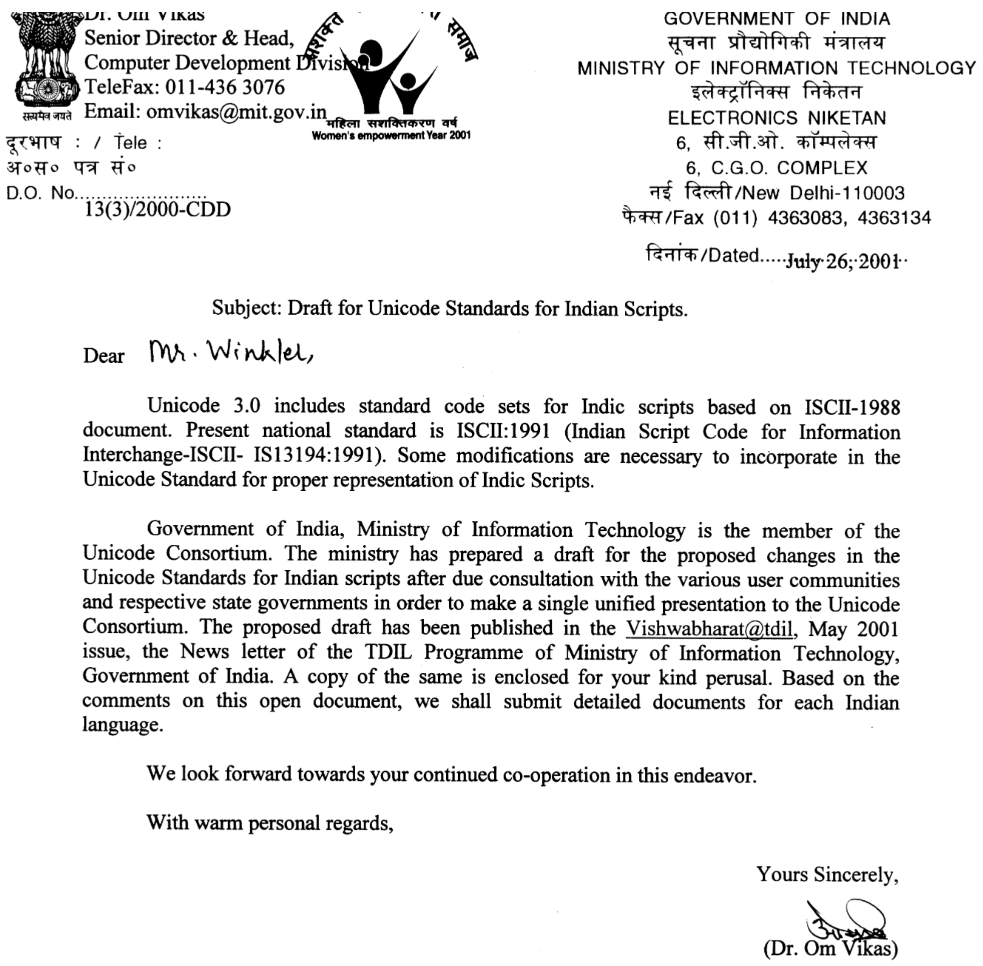
---

[370] "ISO Membership Manual." accessed June 28, 2022, https://www.iso.org/files/live/sites/isoorg/files/store/en/PUB100399.pdf.

[371] "Membership Levels." accessed June 28, 2022, https://home.unicode.org/membership/membership-levels/

[372] Ken Whistler, interview with author, February 12, 2020.

[373] *Ibid;* Lisa Moore, interview with author, April 26, 2022; Debbie Anderson, interview with author, January 17, 2020.



Dr. Om Vikas
Senior Director & Head,
Computer Development Division
TeleFax: 011-436 3076
Email: omvikas@mit.gov.in

सत्यमेव जयते

महिला सशक्तिकरण वर्ष
Women's empowerment Year 2001

दूरभाष : / Tele :
अ०स० पत्र सं०
D.O. No. 13(3)/2000-CDD

GOVERNMENT OF INDIA
सूचना प्रौद्योगिकी मंत्रालय
MINISTRY OF INFORMATION TECHNOLOGY
इलेक्ट्रॉनिक्स निकेतन
ELECTRONICS NIKETAN
6, सी.जी.ओ. कॉम्पलेक्स
6, C.G.O. COMPLEX
नई दिल्ली/New Delhi-110003
फैक्स/Fax (011) 4363083, 4363134

दिनांक/Dated July 26, 2001

Subject: Draft for Unicode Standards for Indian Scripts.

Dear Mr. Winkler,

Unicode 3.0 includes standard code sets for Indic scripts based on ISCII-1988 document. Present national standard is ISCII:1991 (Indian Script Code for Information Interchange-ISCII- IS13194:1991). Some modifications are necessary to incorporate in the Unicode Standard for proper representation of Indic Scripts.

Government of India, Ministry of Information Technology is the member of the Unicode Consortium. The ministry has prepared a draft for the proposed changes in the Unicode Standards for Indian scripts after due consultation with the various user communities and respective state governments in order to make a single unified presentation to the Unicode Consortium. The proposed draft has been published in the Vishwabharat@tdil, May 2001 issue, the News letter of the TDIL Programme of Ministry of Information Technology, Government of India. A copy of the same is enclosed for your kind perusal. Based on the comments on this open document, we shall submit detailed documents for each Indian language.

We look forward towards your continued co-operation in this endeavor.

With warm personal regards,

Yours Sincerely,

[illegible]

(Dr. Om Vikas)

Figure 30. Letter From Om Vikas to Unicode Technical Committee

India's initial application for membership was followed by a formal letter of introduction in 2001.[374] Vikas announced TDIL's intention to propose new alterations to the Unicode Standard, attaching draft recommendations in the form of the May 2001 edition of *Vishwa Bharat*, TDIL's monthly newsletter, which devoted a special issue to the recommendations.[375] The issue was twenty-four pages long, and included the usual updates on workshops and multilingual software development. Unlike previous issues, however, this edition included script-by-script comments on the Unicode Standard to date. One section, entitled "Feedback on Unicode 3.0," began,

> *Both the National [ISCII] standard and the Unicode standard will co-exist. It is expected that the coordinators of the Resource Centers for Indian Language Technology Solutions (RC-ILTS) will convene discussion meetings with representative(s) of the State Government and*

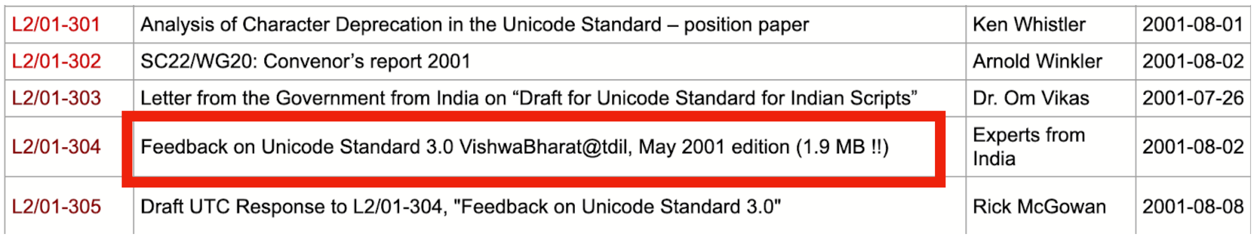
[374] Vikas, Om. "Letter from the Government from India on "Draft for Unicode Standard for Indian Scripts" Unicode Technical Committee Document Registry. L2/01-303.

[375] "Language Technology Flash," *Vishwa Bharat@TDIL*, May 2002.

> *the[ir] language expert[s], and finalize the updates for [the] UNICODE standard (version 3.0) on [a] priority basis.*[376]

The section continued by offering preliminary feedback on existing Unicode conventions from local Indian linguists and representatives of India's state governments. So thorough were their recommendations that when the newsletter issue was added to the official Unicode Document Registry, as per official procedure, the representative responsible for the upload couldn't resist adding a comment regarding its significant file size.[377]

Figure 31. Snapshot of Unicode Document Registry



L2/01-301	Analysis of Character Deprecation in the Unicode Standard – position paper	Ken Whistler	2001-08-01
L2/01-302	SC22/WG20: Convenor's report 2001	Arnold Winkler	2001-08-02
L2/01-303	Letter from the Government from India on "Draft for Unicode Standard for Indian Scripts"	Dr. Om Vikas	2001-07-26
L2/01-304	Feedback on Unicode Standard 3.0 VishwaBharat@tdil, May 2001 edition (1.9 MB !!)	Experts from India	2001-08-02
L2/01-305	Draft UTC Response to L2/01-304, "Feedback on Unicode Standard 3.0"	Rick McGowan	2001-08-08

Despite the double exclamation-point, Unicode's technical staff clearly took these recommendations seriously. After all, India's impressive membership application, along with their *Vishwa Bharat* special issue, proved that this was "the point at which India decided it could have a fruitful relationship with the Unicode Consortium."[378] Rick McGowan drafted a response on behalf of the Unicode Technical Committee within the week, which went up for discussion at the following UTC annual meeting in November. McGowan's initial draft began,

> *The document [i.e., the special issue of Vishwa Bharat] asks for some quite reasonable additional characters, provides some annotations and information for block introductions, and also requests a number of codepoint changes… UTC would like to thank the authors for writing this detailed analysis of Indic script encoding within the Unicode standard, and looks forward to discussion of the various points raised by the document.*[379]

Ultimately, the final version agreed upon by UTC voting members included much of the same preamble, but added the notable caveat that "this document [e.g., the UTC's official rejoinder] is an initial technical response, and the position of the UTC on specific points may change in view of additional information from the Government of India on particular characters. The committee

[376] "Language Technology Flash," May 2002, 15.

[377] "Unicode Document Registry." accessed June 28, 2022, https://www.unicode.org/L2/L2001/Register-2001.html.

[378] Ken Whistler, interview with author, April 23, 2020.

[379] McGowan, Rick. "Draft UTC Response to L2/01-304, "Feedback on Unicode Standard 3.0"" Unicode Technical Committee Document Registry. L2/01-305.

looks forward to discussion of the various points raised by the document, so that understanding and agreement can be reached about specific resolutions."[380]

Though the response was cordial and the general sentiment was one of welcoming exchange, the UTC's response held strict to its founding principles. Amongst several requests for other major language communities, the Indian Ministry had requested changes to Unicode's encoding of the Bangla language: they wanted to add Bangla-specific punctuation marks, as well as four new characters (including *khanda ta);* they also recommended changing the names of three existing codepoints.[381] UTC responded that the punctuation marks India was requesting had already been encoded in the Devanagari script section of Unicode. Of the four requested characters, UTC accepted only one; the others, they claimed, would need to be considered in greater detail, demanding that India present further evidence to justify the changes. Otherwise, they insisted, India's the requested character additions would lead to an unwanted "change to the model for Bengali."[382] Finally, the proposed name changes for existing codepoints were accepted as additions rather than revisions, with UTC citing its own stability policy, which prohibited making any retroactive changes to the Unicode Standard once an edition had already been published. In short, India got very little of what they wanted, demonstrating that a paid membership and signed official letters were not enough to make change in this new paradigm.

At the same time, it is worth noting that UTC's official communications with the Indian Ministry stood in stark contrast to those we saw in the previous chapter being exchanged between Unicode staffers in the West and local language computing hobbyists. Where the channels of communication were informal mailing lists in the previous case, there were now signed, sealed official letters being passed between the two parties: the Unicode Consortium and the Government of India. Where the sources were previously temporary blog posts that were both searchable online for everyone, but not delivered to anyone in particular, there were now regular newsletters that were systematically distributed. And where the bulk of expertise came from anonymous or self-taught sources, it was now being solicited from government and linguistics experts and brought to the fore. The very existence of these two forms of exchange highlighted a burgeoning space for public discourse on the internet in the early aughts.

After a brief lull in activity, the Unicode Consortium began hearing from Dr. Om Vikas and the Ministry of Communications and Information Technology once again in 2003. This time, Vikas was determined to make more of a dent in the encoding standard.

In March of 2003, the Unicode Consortium held its 94th technical committee meeting, which Dr. Vikas notably attended. There, Vikas presented original slides on the status of various Indic scripts, beginning with a description of what he called the general "linguistic scenario in India" – e.g., how many different languages were spoken, and where these languages ranked globally in

[380] McGowan, Rick. "UTC Response to L2/01-304, "Feedback on Unicode Standard 3.0", (repl. L2/01-305)" Unicode Technical Committee Document Registry. L2/01-430R.

[381] "Language Technology Flash," May 2002, 20.

[382] McGowan, Rick. "UTC Response to L2/01-304, "Feedback on Unicode Standard 3.0", (repl. L2/01-305)" 5-6.

terms of numbers of speakers.[383] He then moved on to TDIL's grand vision for India's techno-linguistic future, presenting the "A B C" Technology Development Phases that the agency had coined two years prior. India's current top priority, he said, was to determine "if the existing Unicode Standard for Indic scripts [is] fulfilling the requirements of all the Indian languages," rather than unfairly privileging only a few.[384] Toward this end, the Ministry of Information Technology had organized a number of meetings between linguistic and software industry experts within India's borders to identify existing deficiencies in the Unicode Standard as it stood. Reaching the agreement that he was about to present "took pretty good time."[385] But the results, he assured the Consortium, would be well worth it.

Before moving on to his specific encoding proposals, however, he emphasized how important it was to get the nomenclature of characters exactly right in the Unicode Standard: "once a character is encoded with the wrong name, it always creates confusion among the user group."[386] For instance, the Devanagari symbol for *Halant* (a computing term for a joining symbol) was called a *Virama* in the Unicode Standard, a term which the user community had always understood to refer to a kind of punctuation mark. This sloppy, inattentive terminology could only create ongoing confusion. (What Vikas didn't mention was that the wider community objected deeply to the use of the term "virama" altogether, as a "virama" only existed in any form in the Devanagari script, making the concept all but inscrutable to India's other language communities, including Bangla and Tamil, which used terms such as "hasanta" or "pulli" instead.) Vikas then presented several new character proposals, many of which echoed the requests from the special issue of *Vishwa Bharat* sent in 2000, and had been re-submitted as formal proposals by his TDIL colleague Manoj Jain a few days before the current presentation.[387]

Vikas ended his presentation with a bold, provocative proposal: he called for Unicode to host the next conference in India "to benefit the Multilingual Software Industry in India and the neighboring countries." Last, he closed with TDIL's utopian vision: "Government, Academia, Industry **together** to play **globally** and to serve **locally** in multilingual computing."[388]

The Unicode Technical Committee would make note of this presentation and take a few steps accordingly. It still wasn't willing to make the wholesale changes that the Indian Ministry of Information Technology was requesting. Many of these requests would require further investigation. But, the impact of India's efforts were apparent in the action items the UTC would post for itself in the days after Vikas' presentation. They included setting up a new "Indic"

[383] Vikas, Om. "Unicode Standard for Indic Scripts," Unicode Document Registry, L2/03-102, 3.

[384] Vikas, Om. "Unicode Standard for Indic Scripts," 6-11.

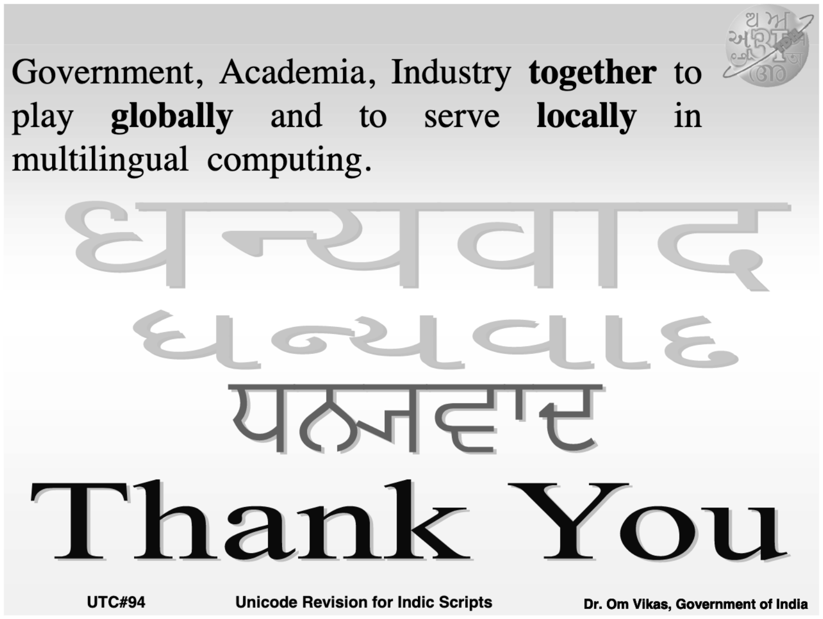
[385] Vikas, Om. "Unicode Standard for Indic Scripts," 12.

[386] Vikas, Om. "Unicode Standard for Indic Scripts," 13.

[387] Gov't of India. "Proposed Changes in Indic Scripts ," Unicode Document Registry, L2/03-101.

[388] Vikas, Om. "Unicode Standard for Indic Scripts," 39-40.

mailing list and arranging for a group of UTC representatives to visit India to engage in further dialog with local scholars.[389]


Government, Academia, Industry **together** to play **globally** and to serve **locally** in multilingual computing.

धन्यवाद

ધન્યવાદ

ਧਨਜਵਾਦ

Thank You

UTC#94 Unicode Revision for Indic Scripts Dr. Om Vikas, Government of India

Figure 32. Closing Slide of Om Vikas' UTC Presentation

The Indic mailing list started up only a few days later, beginning with posts sharing the documents that had been circulated at the recent UTC meeting, most prominently Manoj Jain's specifically character proposals.[390] Although seemingly humble in its origins, this Indic list will go on take particular significance in our final two chapters as the space where the *khanda ta* debate will finally be hashed out, bringing centuries of language controversies and international power imbalances to a head. For now, it's worth noting that the Indian government and its understanding of contemporary language politics was pushing the Unicode Consortium to recognize its diplomatic role, despite its technical founders' best efforts.

Bangladesh in the Digital Sphere?

On a final note for this chapter, whereas India had come to embrace the novel governance structures of technology consortia like UTC, Bangladesh — with its comparatively under-developed technological infrastructure — was still stuck reaching out over more traditional

[389] McGowan, Rick. "UTC #94 Action Items," Unicode Document Registry, L2/03-117, 94-A22-23.

[390] McGowan, Rick. "[indic] Document copies," Email. March 14, 2003.

channels. While India was joining the Unicode Consortium as a full member in 2000, Bangladesh was still relatively ignorant of its role in international digital standards-making.

Prior to the September 2000 ISO meeting in Athens, Bangladeshi delegates was requesting "harmonization" between the newly-released BDS 1520:2000 national standard (Bangladesh's analog to ISCII) and ISO 10646. Notably, the only difference between the two that they noted was the infamous character *khanda ta*.[391] Michael Everson, a UTC representative from Ireland, posted to the Unicode mailing list to pass along Bangladesh's request, along with an admission of his own lack of expertise in the matter: "I am at the WG2 meeting in Athens where the character is being discussed, but we don't know how to evaluate it."[392] Three country delegates had chimed in during the ISO discussion without coming to any conclusive answers. The delegate from Japan only expressed confusion about where this new standard would stand in relation to the previous one (BDS 1520:1997). The delegate from India, meanwhile, emphasized the intrinsic need to maintain alignment with other Indic scripts, or else the inter-workings of ISCII encodings would be compromised. Finally, the American delegate, Dr. Ken Whistler, who had also been an early developer and core member of Unicode, affirmed that interoperation with ISCII would indeed be an issue, and that many open questions remained on this issue.[393] In short, *khanda ta* was already gaining a reputation as a contentious, ambiguous issue. Hence the concurrent message out to the Unicode "experts list" – which was in fact the Unicode mailing list open to the public.

Everson's post about Bangladesh's request for *Khanda Ta* received a quick response. Abdul Malik, who was apparently a Bangladeshi native (based on his email domain), explained again that *khanda ta* was a form of *ta*, equivalent to *ta* with a silenced inherent vowel. The reason the BDS standard included *khanda ta,* he speculated, was because that standard did not have control characters (such as virama, zwnj, or zwj) to trigger a *khanda ta*. It was an "immature" standard that had yet to define rules for rendering, a process which, in theory, would lead to a process for displaying *khanda ta*. Malik did include an important caveat, however, that "A representative of the Bangladesh Standards and Testing Institution (i.e., the instigator of the proposal) should be better placed to answering these questions than me, anyway…"[394]

Even in the course of this short interaction, we can already see the hierarchical imbalances and structural differences between the ISO and Unicode communities. The Unicode "experts list" was understood to have a deeper knowledge of scripts and encodings, and a wider range of expertise over their implementation. UTC communications were as likely to happen virtually as in physical space, unlike ISO communications, which took place only during in-person meetings. And ISO

[391] Bangladesh SC2/WG2 N2261, "Synchronization of Bengali coded character set national standard and ISO 10646-1 on character U+09BA, KHANDATA," Unicode Document Registry, L2/00-303.

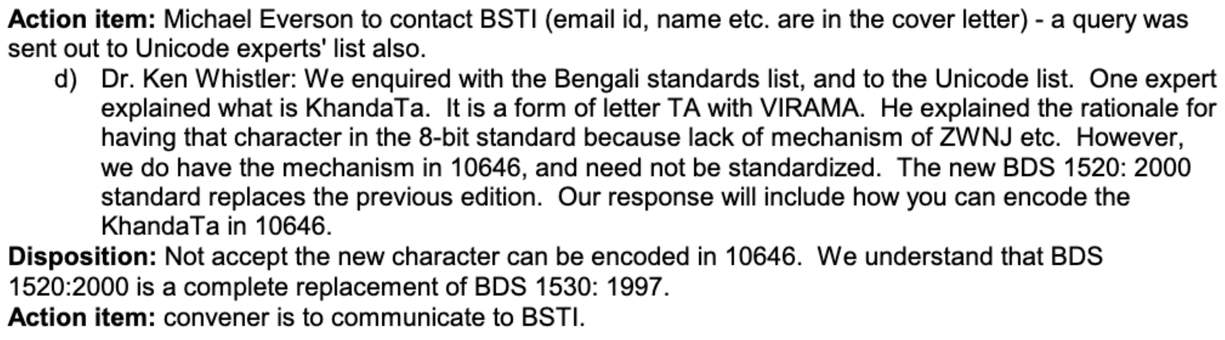
[392] Everson, Michael. "Request about Bangla/Bengali." Email, September 20, 2000.

[393] Umamaheswaran SC2/WG2 N2353, "Minutes of the SC2/WG2 meeting in Athens, September 2000 (repl. L2/00-318)" Unicode Document Registry, L2/01-050.

[394] Malik, Abdul, "Re: Request about Bengali/Bangla," Email. September 20, 2000.

was willing to defer expertise to the UTC over national standards bodies, especially due to the former's accessibility and promptness in responding.

At this point, Everson's action items included contacting the Bangladesh standards institute, BSTI, and the Unicode mailing list – although, as we might have expected, only the latter drew any kind of response timely enough to shape the conversation.[395] In the ISO meeting notes, the only verdict reached was:



Action item: Michael Everson to contact BSTI (email id, name etc. are in the cover letter) - a query was sent out to Unicode experts' list also.

d) Dr. Ken Whistler: We enquired with the Bengali standards list, and to the Unicode list. One expert explained what is KhandaTa. It is a form of letter TA with VIRAMA. He explained the rationale for having that character in the 8-bit standard because lack of mechanism of ZWNJ etc. However, we do have the mechanism in 10646, and need not be standardized. The new BDS 1520: 2000 standard replaces the previous edition. Our response will include how you can encode the KhandaTa in 10646.

Disposition: Not accept the new character can be encoded in 10646. We understand that BDS 1520:2000 is a complete replacement of BDS 1530: 1997.
Action item: convener is to communicate to BSTI.

Although terse and inconclusive, this ISO document would prove an important point of reference for Andy White when he first began to post critiques of the Indic FAQ in 2002. At that time, White noted that the response included the recommendation to render *khanda ta* according to the Unicode/ISO 10646 model, even though that diverged from what was currently posted in the Indic FAQ (by then, Apurva Joshi's edits had already been adopted).[396]

White's accusation of inconsistency instigated a long and hotly-debated thread that ultimately drew in UTC members to the conversation. Among these was Ken Whistler, who was a regular attendee of both ISO and Unicode meetings; angrily, he spit back that the ISO resolution could *not* be interpreted as definitively assigning the sequence of codepoints that would render *khanda ta* - the Indic FAQ still took precedence. The only point that the ISO resolution made, he insisted, was that the BSTI proposal had not been accepted, "on the basis of *this* feedback from a Bengali expert [Malik] on the Unicode list."[397]

Why do these intricate twists and turns on long-lost internet debate boards matter? As I've argued throughout this dissertation, these seemingly nit-picky decisions set the standards for language users worldwide, revealing much about the ideologies of various institutional stakeholders and shaping and constraining the power differentials inevitably embedded within the technical standard. As the discussion around *khanda ta* dragged on over the ensuing years,

[395] Umamaheswaran SC2/WG2 N2353, "Minutes of the SC2/WG2 meeting in Athens, September 2000 (repl. L2/00-318)" 28.

[396] White, Andy. "ISO 10464, Unicode & The FAQ." Email, November 21, 2002.

[397] Whistler, Kenneth, "Re: ISO 646, Unicode & The FAQ (Bengali Khanda Ta)." Email, November 21, 2002.

users reasonably started to ask why the Unicode staff seemed so stubbornly resistant to making what seemed like such a simple addition. As we can see here, one answer is that Unicode staff placed a strong priority on *principle* and *process,* as opposed to flexibility and an interest in outcome. It mattered precisely what was said in ISO meetings, and what had been approved there. It mattered precisely which documents could be said to supersede others. It mattered precisely what Unicode staffers considered it within the scope of the Standard to define. And while such dogmatic adherence to doctrine had the benefit of consistency for implementers of the Unicode Standard, it also risked propagating mistakes or inconveniences that language communities were beginning to raise.

Whistler's final word on the matter would be as follows:

*I have not digested all the argumentation in the last month about this topic, so cannot say what I feel the *right* answer [with respect to rendering rules], finally, is for this. But now, please, stop speculating about how things got to be the way they are, stop arguing about whose specification trumps whose (a statement in a WG2 resolution which is not reflected in the ISO 10646 standard or a statement in a Unicode website FAQ which is not reflected in the Unicode Standard), and focus on what is the technically best advice to give people about representing the Bengali Khanda Ta, given the context explained in the Unicode FAQ.*[398]

The following chapters showcase an equally contentious controversy over whose expertise should take priority in the *khanda ta* debate – that of the professional Bengali linguists who enter the chat or that of the Unicode technical experts.

Framing these seemingly petty squabbles is the enormous backdrop of centuries of techno-linguistic power struggles, including the historical and multi-layered interests introduced in our previous chapters: the interest of standards-makers in preserving an "efficient" technical code; the enthusiasm of a young, emerging Bengali diaspora committed to contributing to its countries' technical prowess; and the political agenda of government authorities taking on the complicated mantle of language planning in the rapidly-evolving era of an expanding consumer internet. As the case study of *khanda ta* that follows will demonstrate, the intertwining of these issues was what would propel small encoding issues into high-stakes debates.

[398] Whistler, Kenneth, "Re: ISO 646, Unicode & The FAQ (Bengali Khanda Ta)." Email, November 21, 2002.

Chapter 4: Accommodating Orthographic Reform

> *'Many can squabble over a single letter as though the well-being of Europe depended on it.' So said the eighteenth-century Dutch writer Z. H. Alewijn (1742–1788).*[399]

Orthographic wars are often fought letter by letter, diacritic by diacritic, taking no prisoners, according to sociolinguist Joshua A. Fishman.[400] In the previous chapter, I traced the historic practices of language planning across the Indian subcontinent – including how the state came to control the status and evolution of various languages, first under the British, then as an independent nation-state. These efforts at language planning included orthographic reform, which involves determining acceptable spellings within a given writing system. Orthographic reform might also include defining which letters are part of a given alphabet, determining how those letters should appear on the page, and deciding what foreign words (or "loan words") are permitted within a language.[401]

In this chapter, I evaluate how type technology molds (or refuses) to accommodate orthographic reforms. Before we begin to consider the "missing letter" in the following chapter, I walk through four increasingly difficult demands that are made of the multilingual computing stack concerning the Bangla language. To what extent do these technologies and standards serve as enabling or limiting systems? Taking the Science and Technology Studies premise that all technologies are political — all *artifacts have politics* — that "order" behavior, I consider the specific politics of the Unicode Standard and OpenType format.[402] As Winner wrote, the decisions to make or not make something, how those decisions are made, and who is involved in those decisions all reflect the politics of that system.[403]

Bringing together the frames of orthographic reform and techno-politics, I argue that the designers of the multilingual computing stack ultimately view themselves, and generally act in such a way, as to be agnostic accommodators of orthographic reform. Their goals are to understand the qualities and evolutions of a writing system, and translate them as best as they can to the digital medium. The question these technical "translators" ask is often *how,* rather than *if* a linguistic feature should be accommodated. At the same time, this mediating, middle role is challenged at times by difficult, esoteric asks, as we see in the final episode presented in this chapter of "garbage type." This self-view becomes important to keep in mind as we progress to the next and final chapter, where the multiple perspectives and stakes presented throughout

---

[399] Quoted in Mark Sebba, *Spelling and Society: The Culture and Politics of Orthography around the World* (Cambridge: Cambridge University Press, 2007), https://doi.org/10.1017/CBO9780511486739., 132.

[400] *Ibid.*

[401] Mark Sebba, *Spelling and Society: The Culture and Politics of Orthography around the World* (Cambridge: Cambridge University Press, 2007), https://doi.org/10.1017/CBO9780511486739., 82-99.

[402] Langdon Winner, "Do Artifacts Have Politics?," *Daedalus* 109, no. 1 (1980): 121–36.

[403] *Ibid.*

this dissertation finally converge. Where some Bengali observers see the technical mediators — the designers of the standards — as conducting orthographic reform, the mediators themselves view themselves as technicians, who are collecting information and documentation, and finding the appropriate point of intervention within a complex multilingual computing stack.

In the overall chronology of *khanda ta*, the episodes presented here also occur at a crucial time. It is mid-2003 and Unicode version 4 had just been released in April, including 52 new scripts and a total of 96,000 encoded characters, nearly double the previous numbers in version 3.[404] Several of the changes proposed by the Government of India had been officially accepted. Perhaps due to these changes, Unicode's brand-new Indic mailing list (created after TDIL's Vikas's persuasive presentation at the UTC annual meeting) remained relatively quiet during its first four months. The quiet breaks with the introduction of a handful of new characters who are beginning to work, now, on Bangla digitization.

"Making the Uniscribe engine work perfect for Bangla"

The silence on the Indic mailing list would break in June 2003 by a post by Rick McGowan, the then-Vice President of the Unicode Consortium, who would share that he had recently learned of the errata pages about *khanda ta* that Andy White had posted on his personal blog (as previously discussed in Chapter 2) and was promising to look into them.[405]

A few days later, Omi Azad followed up on McGowan's post by introducing himself on the Indic list as the person who had originally tipped off McGowan to White's errata pages.[406] Azad was involved in many active Bangla-language computing efforts. He contributed occasionally to Bengalinux; participated in another Bangladesh-based group called BIOS; and assisted most actively through his efforts with a group called *Altruists International*, an interesting and highly unusual group set up by Dr. Robin Upton, a UK-based "internet consultant."[407]

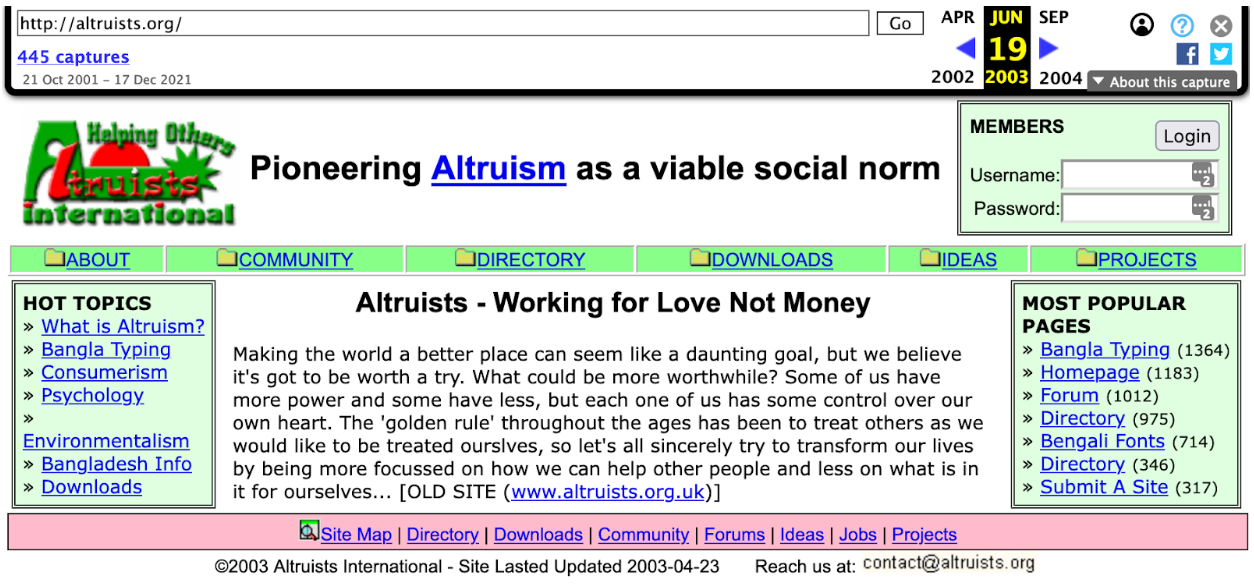
---

[404] *The Unicode Standard, version 4.0.*

[405] McGowan, Rick. "[indic] Indic FAQ "errata"" Email, June 26, 2003.

[406] Azad, Omi. "[indic] Langiage Processing…" Email, June 28, 2003.

[407] "Altruists International," accessed July 1, 2022, http://web.archive.org/web/20150206053738/http://www.altruists.org/about/.



http://altruists.org/ Go

445 captures
21 Oct 2001 – 17 Dec 2021

APR JUN SEP
19
2002 2003 2004
About this capture

Helping Others Altruists International

Pioneering Altruism as a viable social norm

MEMBERS Login
Username:
Password:

ABOUT | COMMUNITY | DIRECTORY | DOWNLOADS | IDEAS | PROJECTS

HOT TOPICS
» What is Altruism?
» Bangla Typing
» Consumerism
» Psychology
» Environmentalism
» Bangladesh Info
» Downloads

Altruists - Working for Love Not Money

Making the world a better place can seem like a daunting goal, but we believe it's got to be worth a try. What could be more worthwhile? Some of us have more power and some have less, but each one of us has some control over our own heart. The 'golden rule' throughout the ages has been to treat others as we would like to be treated ourslves, so let's all sincerely try to transform our lives by being more focussed on how we can help other people and less on what is in it for ourselves... [OLD SITE (www.altruists.org.uk)]

MOST POPULAR PAGES
» Bangla Typing (1364)
» Homepage (1183)
» Forum (1012)
» Directory (975)
» Bengali Fonts (714)
» Directory (346)
» Submit A Site (317)

Site Map | Directory | Downloads | Community | Forums | Ideas | Jobs | Projects

©2003 Altruists International - Site Lasted Updated 2003-04-23 Reach us at: contact@altruists.org

Figure 33. Snapshot of Altruists International Website (Wayback Machine)

Upton promoted an anti-capitalist, charity-oriented, and individually-motivated vision; according to him, it was within every person's power to perform altruistic deeds in everyday life, and thereby improve the world.[408] In this sense, Upton's vision was very similar to the "free software" ethic of the Western computing world (as discussed in Chapter 2). Under his guidance, Altruists International undertook a variety of projects, including compiling an "online database of electrical products that are no longer supported by the manufacturers" and creating a "Bangladesh Information" website to showcase a "different side to this beautiful country" rather than focusing only on "natural disasters such as floods, famine, or perhaps the man-made disasters of corruption."[409] Upton's interest to Bangladesh began after his first visit in 1998, when he felt "moved by the huge inequality between materially rich and poor nations and [felt] called to help in a direct way."[410] He was so committed to improving technological conditions there that he also spearheaded a Bangla computing and localization effort called *Ekushey* (named after "Ekushey February," Martyr's Day, an important commemoration within the Bangla Language Movement). As part of *Ekushey*'s efforts, *Altruists International* built a plug-in for Microsoft Word that allowed users to type in Bangla.[411] Recall that at this time in 2003, Microsoft had only released Hindi- and Tamil-language versions of its Windows systems, and had yet to release software in other Indic languages, though their encoding standards and rendering engine were already largely in place. Others like *Bengalinux* and *Indic-computing* were mostly working within the open source software space, so *Ekushey* was one of few initiatives tinkering directly with Microsoft products for local-language purposes.

[408] "Altruists International," accessed July 1, 2022, http://web.archive.org/web/20150206053738/http://www.altruists.org/about/.

[409] "Bangladesh Information Project," http://web.archive.org/web/20150302034508/http://www.altruists.org/projects/eo/bi/.

[410] "The Story of Ekushey Project - Ekushey," accessed July 1, 2022, http://ekushey.org/?page/Story_of_Ekushey_Project.

[411] *Ibid.*

For his part, Azad had always been passionate about the Bangla language. He lived in Bangladesh but would stay abreast of new technological developments worldwide through online publications. He enjoyed writing in Bangla, and would carry floppy disks with Bangla fonts loaded onto them back and forth between home and his school, so he could be sure to be able to type in Bangla no matter where he was.[412] Recall that at the time, typing in Bangla required proprietary fonts and programs, and the resulting document could not be reliably transferred between computers, nor could it be easily printed. As Azad looked for better solutions, a Google search brought him to the *Ekushey* project. He soon got involved in building fonts for *Altruists International*, and also began interacting with other hobbyist groups dedicated to Bangla-language computing (though many of these groups found him hard to keep track of; he had a hand in so many projects that he seemed to be everywhere and nowhere at once).

It was through Azad's involvement with *Ekushey* that Microsoft first became interested in him.[413] Specifically, he drew the attention of two program managers from Microsoft's typography department: Paul Nelson and Peter Constable. Nelson and Constable were both linguists-turned-technologists that had previously worked with SIL International, an evangelical Christian non-profit that was also a leader in language preservation. Historically, missionaries from SIL had helped many linguistic groups document their languages through dictionaries and grammar guides, along with developing tools – including computing tools – to facilitate the use of minority and indigenous languages. SIL had been responsible for developing several important software tools for text rendering, most significantly their open source rendering engine for complex scripts, Graphite.

While previous scholars have acknowledged SIL's critical role in the discipline of linguistics, its equally important role in the development of language technology has largely gone unrecognized.[414] Among SIL's most important contributions to local-language computing efforts was providing technology companies with trained employees, who rotated through the revolving door of SIL and industry. Paul Nelson, for instance, had been an SIL-employed Arabic linguist working on digital fonts for that script when Microsoft recruited him into their Typography team in the early 1990s.[415] A few years later, Nelson pulled in Peter Constable to Microsoft's Typography team, who had been previously been working mostly on Thai.[416] Both Nelson and Constable had extensive experience, in other words, with the difficulties of typing in non-Latin scripts – including both the technological challenges of non-standard fonts, and the cultural challenges of getting users to adopt a keyboard utilizing a non-standard script.

[412] Azad, Omi, interview with author, January 30, 2022.

[413] *Ibid*; Nelson, Paul, interview with author, March 24, 2022.

[414] Lise M. Dobrin, "SIL International and the Disciplinary Culture of Linguistics: Introduction," *Language* 85, no. 3 (2009): 618–19, https://doi.org/10.1353/lan.0.0132.

[415] Nelson, interview.

[416] *Ibid;* Constable, Peter, interview with author, February 4, 2022.

Because Microsoft was a member of the Unicode Consortium, Nelson and Constable were representatives to regular Unicode Technical Committee (UTC) meetings, where they helped mull over encoding proposals before they were brought to the joint UTC/ISO conferences. However, despite their industry affiliations, Nelson and Constable's history at SIL had proven them deeply sympathetic to the needs of minority and indigenous language users — a perspective that is important to keep in mind over the course of the ensuing Unicode debate, in which as hobbyists would begin associating Nelson and Constable with their corporate Microsoft identities rather than with their underlying social commitments. Nelson and Constable had to play a delicate intermediary role between user communities and the Unicode core staff. As Bangla users addressed them with comments like, "let's see what Microsoft decided for the sake of Bangla computing :)," the Microsoft duo was forced to shoulder a heavy burden of diplomacy.[417]

Back in 2003, however, when the duo first came into contact with Azad, they were in the middle of collecting user feedback about Microsoft's new Bangla font, Vrinda, and the rendering engine, Uniscribe, used to display it.[418] As described in Chapter 1, this rendering engine was tasked with several functions: first, it had to interpret the Unicode Standard and the data contained within the OpenType font file, and then it had to perform the necessary positionings and substitutions to display the multilingual text correctly. It was also responsible for display decisions, such as determining appropriate places for line breaks and caret placement (i.e., the on-screen cursor in a text file). Much of this work required them to administer user experience tests, which they conducted both internally, using Microsoft employees who spoke the relevant languages, and externally, by sourcing native speakers within each language community.[419] In order to locate a sufficient number of native speakers, Nelson and Constable spent meant much of their time building relationships with local governments, as well as scouring the web for native speakers who were already experts in the language technology space — which is what brought them to Azad. As Nelson later recalled in an interview, "There were not a lot of people in those areas [i.e., technology and linguistics]- it was a small community. There was a finite set of people and you figured out quickly who they were."[420] This "finite set of people" included Dr. R. K. Joshi, Apurva Joshi, and other NCST employees (discussed in Chapter 2), whom Microsoft brought on board to work on Indian fonts, especially the Devanagari OpenType and Uniscribe specification.[421] When it came to the Bangla specification, however, Microsoft found they needed to go beyond their contacts in the West Bengal government. Omi Azad was the perfect candidate. As Azad himself put it in his introduction to the Unicode Indic list, he had recently contracted "with Paul Nelson of Microsoft to make the Uniscribe engine work perfect for Bangla."[422] But as Azad would quickly

[417] Azad, Omi, "[indic] Re: RaJophola" Email, June 21, 2004.

[418] Azad, interview; Nelson, interview.

[419] Nelson, interview.

[420] *Ibid.*

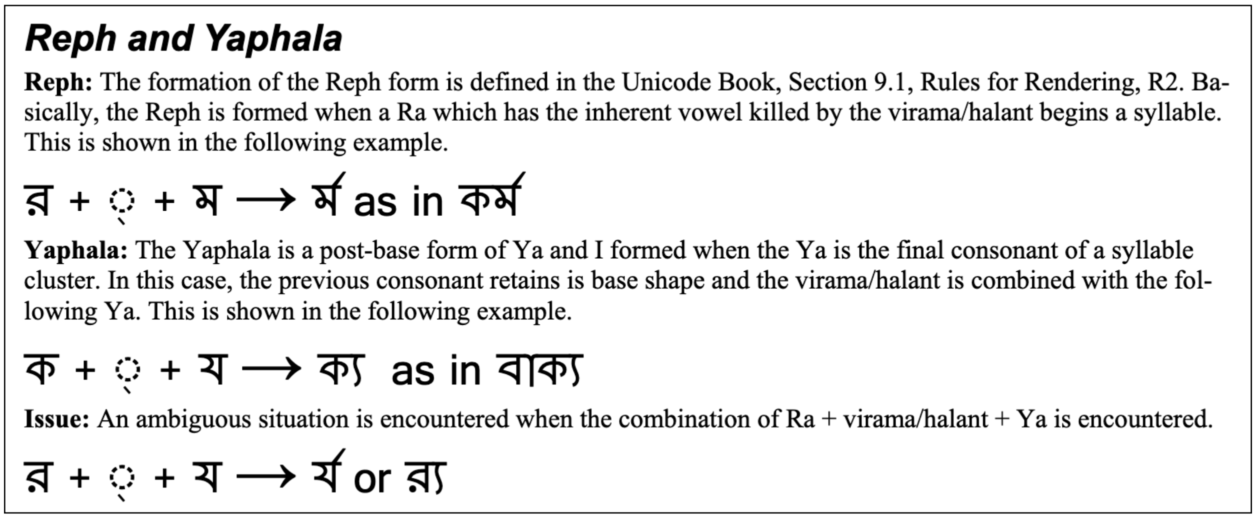
[421] *Ibid*; Shanbhag, Shrinath, interview with author, March 28, 2022.

[422] Azad, Omi. "[indic] Langiage Processing…" Email, June 28, 2003.

discover, making Bangla “work perfect” with Microsoft came with serious challenges, from displaying standard letters correctly, to incorporating special characters, to making space for character combinations yet to be invented.

Rendering Issues: Ra, Ja, and Khanda ta

One of the first issues Azad raised on the Indic list in June 2003 was an ambiguity in how two specific Bangla letters ought to appear when typed next to each other: *ja* and *ra*.[423] Both of these letters had radically different glyph shapes associated with them depending on their orthographic context.



Reph and Yaphala

Reph: The formation of the Reph form is defined in the Unicode Book, Section 9.1, Rules for Rendering, R2. Basically, the Reph is formed when a Ra which has the inherent vowel killed by the virama/halant begins a syllable. This is shown in the following example.

র + ◌্ + ম ⟶ র্ম as in কর্ম

Yaphala: The Yaphala is a post-base form of Ya and I formed when the Ya is the final consonant of a syllable cluster. In this case, the previous consonant retains is base shape and the virama/halant is combined with the following Ya. This is shown in the following example.

ক + ◌্ + য ⟶ ক্য as in বাক্য

Issue: An ambiguous situation is encountered when the combination of Ra + virama/halant + Ya is encountered.

র + ◌্ + য ⟶ র্য or র‍্য

Figure 34. Problem Statement From PRI-9 on *Ra (Reph) and Ja (Jofola/Yaphala)*

The question was, how could a user ensure the right set of glyphs would display in any given word? Unicode had not yet provided adequate guidance on this matter; in the absence of clear recommendations, implementers would resort to their own definitions that may be incompatible with one another. After Azad and others raised the issue, Paul Nelson drafted a relatively new type of Unicode-branded communication in response: the Public Review Issue, or “PRI.” PRIs would give community members an opportunity to provide targeted feedback on whether a proposed change should be incorporated into the Standard. In the case of *ja* and *ra*, the proposed change would have been written into the preamble of the Bangla section of the Standard itself.[424] This preamble offered guidance to the developers responsible for implementing the standard, such as rendering engine developers at Microsoft or independent hobbyists on the *Bengalinux* listserv.

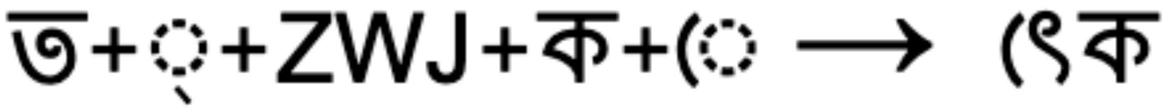
[423] Azad, Omi. “[indic] Langiage Processing…” Email, June 28, 2003.

[424] Paul Nelson, “Bengali Script: Formation of the Reph and Use of the ZERO WIDTH JOINER and ZERO WIDTH NON-JOINER,” June 24, 2003.

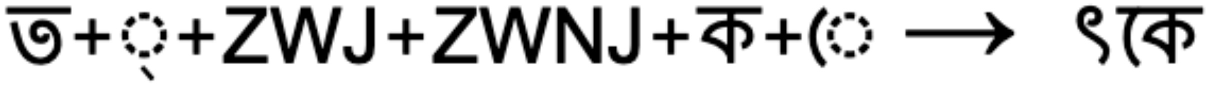
A few days later, Nelson updated the initial PRI regarding *ja* and *ra* to add a section on *khanda ta*. In it, he seemed to agree with Andy White's assessment from his blog post on *khanda ta* a few months earlier. In Nelson words, "The Unicode 4.0 preview makes a simplistic attempt at addressing the *Khanda Ta* [issue]. The documentation indicates that the *Khanda Ta* is the half form of the Ta (U+09A4)."[425] This "simplistic attempt" at a fix had proved inadequate. As White and others had already noted, treating *khanda ta* as a "half-form" (a kind of letter that was non-existent in Bangla, although it was used in Devanagari) would lead to incorrect renderings, such as vowel modifiers appearing around *khanda ta* instead of around the true "full consonants" they were intended to modify. To render *khanda ta* properly, its full input sequence – as in the combination of Unicode codepoints used to produce it– would need to be updated. In the case of *khanda ta*, this input sequence had already taken on a rather unwieldy form. Whereas most input sequences involved two or three codepoints, *khanda ta* required four, including the three control characters – the virama, the zwj, and the zwnj. Azad forwarded one comment to the Indic listserv from a Bangladeshi user who complained that they needed to "type four things" to get *khanda ta* to render correctly.[426] To be fair, this complaint stemmed from a misunderstanding – the *user* didn't need to type four different entries; only font and keyboard developers did, and then only when they were mapping the single keystroke used to type *khanda ta* to the four-code input sequence used to call up the correct encoding for it. But it was nonetheless true that this represented a potentially tricky combination for developers to work with.

Figure 35. Issues With Khanda Ta Rendering in PRI-9

Khanda ta with incorrect placement of vowel modifier — current encoding



Khanda ta with correct placement of vowel modifier —- suggested encoding



[425] Paul Nelson, "Bengali Script: Formation of the Reph and Yaphala, and Use of the ZERO WIDTH JOINER and ZERO WIDTH NON-JOINER," June 30, 2003.

[426] Azad, Omi, "[indic] Comments on Unicode" Email, July 10, 2003.

From Nelson's perspective, the problem with this four-code input sequence for producing *khanda ta* was that it broke several of the logical rules of Microsoft's rendering engine.[427] This was because the engine was designed to follow linguistic conventions (though in theory it could have been programmed any way developers liked). Within this system, there were meanings associated with each individual control character, and these meanings were aligned between the various Indic scripts. Using workarounds like the one used for *khanda ta* meant that control characters like *zwj* or *zwnj* no longer held a single consistent meaning for the engine to parse – which was manageable from a technical perspective, but inelegant, and a bit troubling. Nelson worried that if more such exceptions were added, the logic embedded in the engine would eventually fall apart, leading to broken characters and other text display errors. Nelson hinted at this possibility in his original PRI: "This necessity of using a ZWNJ to break the shaping should be an indicator that the Khanda Ta should be considered for its own code point if a different half form behavior for the Ta (U+09A4) can be defined."[428] Essentially, Nelson was saying: this workaround may work for now. But if other kinds of ta-glyphs become needed, then it would probably be easiest to encode *khanda ta* as its own codepoint than treat it as a variant of ta.

Ankur and Indic-Computing

Earlier in 2003, the *bengalinux* listserv (previously discussed in Chapter 2) had decided to rename itself '*Ankur*,' meaning "sapling," to give itself a stronger and more distinctive sense of identity.[429] In the months since the release of Unicode 4.0, *Ankur* had been busy. Sayamindu Dasgupta, an early member and the overseer of the *Free Bangla Fonts* subgroup, had built an entirely new font he called *Sagar* ("the sea"), using glyphs donated by none other than Omi Azad.[430] This font was also the first to use a templating system developed by Deepayan Sarkar, another of *Ankur*'s early members and the overseer of the Bangla Literature Archive project.[431]

One of the primary goals of the *Ankur* group had been to standardize as much of their Unicode-compliant, OpenType-format font design as possible, so anyone who came up with artwork for a new font (like Azad had in this case) could "plug and play" their glyph files to create a working Bangla font. Luckily, thanks to advances in font development tools, the group no longer needed to rely on Microsoft's VOLT font editor to design fonts; they could work entirely in the open source program PfaEdit.

[427] Nelson, "Bengali Script: Formation of the Reph and Yaphala, and Use of the ZERO WIDTH JOINER and ZERO WIDTH NON-JOINER," June 30, 2003, 6.

[428] *Ibid.*

[429] Ahmed, Taneem, "[Ankur-core] bengalinux.org is now Ankur" Email, March 3, 2003.

[430] Dasgupta, Sayamindu, "Subject: [Freebangfont-devel] [Announce] Sagar 0.5.0" Email, April 24, 2003.

[431] *Ibid;* "Bengali Template Font," accessed July 1, 2022, http://web.archive.org/web/20061230031531/http://www.stat.wisc.edu/~deepayan/Bengali/FreeBangTemplate/readme.html.

Ankur had also enlisted a number of volunteers to translate various Bangla text strings, with the aim of creating a fully-Bangla Linux desktop. To attract these volunteers, Ankur members were giving demos of their work at free software fairs across India and Bangladesh, which drew positive attention from the press.[432] By 2003, Ankur's goal had become to release a localized Live CD – as in a cd-rom containing a full operating system for one's computer – by the end of the year. This effort required a great deal of coordination and standardization across a distributed network of local volunteers. Participants needed to collaborate on questions of vocabulary, verb tenses, and registers ("should we *ask* the computer or *command* it?").[433] Within a few years' time, the group's translation efforts would have progressed to the point that they could offer developers standardized tools and common glossaries to help simplify the process of Bangla-language localization. But at this point in 2003, every individual act of translation required careful, conscientious, painstaking effort.

Perhaps most notably for our purposes, as they were working on these various efforts, Ankur members found themselves coming across the same encoding ambiguities that Azad had already documented with Paul Nelson. Namely, *rephs* and *jofolas*, certain glyphs for the letters *ra* and *ja*, were not appearing correctly in the open source rendering engine that Ankur was working with, known as Pango (which was an analog to Microsoft's Uniscribe).[434] In addition, the input sequence for *khanda ta* was producing what were known as 'illegal' behaviors (i.e. misplaced vowel modifiers) if it was inputted according to the erroneous Unicode Indic FAQ.[435]

As Ankur worked towards putting out a fully open-source Live CD, then, its members began filing these bug reports with Pango – much in the way Azad was simultaneously filing very bug reports with Microsoft. Ankur's members could see that there were widespread problems with the way Bangla letters were rendering on popular software, but as Ankur founder Taneem Ahmed wrote, "IMHO if uniscribe does not render [Bangla] properly then we need to let them know, not follow them :)"[436] By this, Taneem meant that coordination was desirable so that everyone followed the same input sequences, but they should coordinate to follow the dysfunctional recommendations. Perhaps they could be the originators of a solution that worked.

But even as Ankur members tried to come up with their own hacks for displaying *rephs*, *jofolas*, and *khanda ta* properly, they quickly ran into trouble: the Pango rendering engine was not nimble enough to handle the long series of control-character *zwjs* and *zwnjs* that were being proposed to render these symbols.[437] It was akin to the problem that Paul Nelson had identified

[432] E.g. Dasgupta, Sayamindu, "[Ankur-core] We are on The Telegraph, Kolkata," Email, December 8, 2003.

[433] Dasgupta, Sayamindu, "[Ankur-core] RFC - <bn>koro</bn> or <bn>kora hok</bn>" Email, July 17, 2003.

[434] Sarkar, Deepayan, "Subject: Re: [Freebangfont-devel] template font" Email, April 22, 2003.

[435] Sarkar, Deepayan. "Subject: Re: [Freebangfont-devel] khanda ta" Email, May 7, 2003.

[436] Ahmed, Taneem. "[Freebangfont-devel] [Bug 113551] Changed - Bugs in the Bengali rendering system of Pango." Email, June 1, 2003.

[437] "Bug 113551 – Bugs in the Bengali Rendering System of Pango.," accessed July 1, 2022, https://bugzilla.gnome.org/show_bug.cgi?id=113551.

with respect to Uniscribe. Because the *zwj* and *zwnj* control characters were already in use across all Indic-language complex scripts, and programmed to produce certain behaviors in those contexts, trying to write in exceptions to those rules to accommodate Bangla's idiosyncrasies produced all kinds of errors.

What could Ankur do in this situation? Their only real option was to continue appealing to Pango developers, and use their own hacks based on unauthorized/non-standard code, and acknowledge that these might produce bugs. For these hobbyists, the realm within which they were willing to make appeals was still the open source software space — they would either find their own fixes or accept the bugs.

One visitor to the freebangfonts list, Mehdi Hasan, even called Ankur out for using non-standard code sequences for *khanda ta*.[438] But Ankur was unapologetic: the truth was that there was no infallible way to render *khanda ta* at the moment. Ankur pointed towards Andy White's proposal as a possible solution in the works, but ultimately concluded: "you will have to accept the fact that *khanda ta* hasn't been standardized."[439] Hasan was not pleased with this response, as he was trying to incorporate Ankur's free Bangla fonts into his open-source virtual keyboard program, known as *Avro,* which had already garnered a large user base that was likely to be frustrated by the *khanda ta* display errors. If the letter's underlying encodings kept changing, or relied on non-standard inputs, his users would not be able to interact reliably with their *Avro*-produced Bangla text, including performing basic search-and-find functions.[440] But there was not much that Ankur — or anyone else — could do about the problem at the hobbyist level.

This sense of powerlessness began to change when Paul Nelson roped the Ankur membership into participating on Unicode's Indic mailing list. Nelson had been actively circulating PRI-9, the problem report on Bangla text rendering issues, among those who were not yet part of the Unicode mailing list "in hopes of being able to get some type of resolution/concensus on how Unicode should work with the Bengali script."[441] Among those he approached were Ankur, who decided to respond with a short comment posted to the Indic mailing list — their first contribution to the listserv. Sayamindu Dasgupta began,

> *Hello to all,*
> *I recently came across a mail by Mr Paul Nelson, where he solicited for*
> *the comments of the Linux community on the Bengali proposal submitted by*
> *him. We are currently working on bengali localization projects on Linux,*
> *and I believe that some members of our community has already communicated*

[438] Hasan, Mehdi. "Re: [Freebangfont-devel] Re: [Freebangfont-devel] FREE BanglaSoftwareAvro Keyboard - NewVersion Available" Email, September 21, 2003.

[439] Sarkar, Deepayan. "[Freebangfont-devel] Re: +AFs-Freebangfont-devel+AF0- RE: FREEBangla Software Avro Keyboard - NewVersion Available" Email, September 22, 2003.

[440] Hasan, Mehdi, "[Freebangfont-devel] Re: +AFs-Freebangfont-devel+AF0- RE: FREEBangla Software Avro Keyboard - NewVersion Available" Email, September 22, 2003.

[441] Nelson, Paul, " [indic] Re: Bengali Proposal" Email, July 3, 2003.

to Mr Nelson on this regard. However, I guess that it would be better to summarize our comments in some public forum, and hence, I post them in this list.[442]

As we can see from this introduction, Ankur expressed timidity but was ultimately encouraged by Nelson's efforts, and willing to provide their input on public forums. Sayamindu went on to report that the group approved of the sequence for *ja* and *ra* proposed in Nelson's PRI. He also commented on the ongoing issue with *khanda ta* and concluded that the proposed four-codepoint sequence for the letter would be acceptable, as it would at least improve upon the ambiguities of the current FAQ recommendation. White had proposed a different sequence for *khanda ta*, which was in some ways more reflective of both Bangla's grammar rules and Unicode's definitions for *zwj* and *zwnj* characters. But Nelson's proposal would do as well.

Thus far, despite Microsoft's near-monopoly position in the computer industry and opportunity to unilaterally dictate encoding and rendering rules, the approach taken by its program managers is one of transparency and consensus-building. In addition to reaching out to Ankur directly, Nelson also sent a similar email to Dr. Om Vikas's Technology Development for Indian Languages (TDIL) team, part of India's Ministry of Information Technology (previously discussed in Chapter 3). It was received by Manoj Jain, the TDIL team member who had last interfaced with Unicode, who immediately recognized the importance of Nelson's message and forwarded it, along with the accompanying PRI-9, to Vikas.

From: "Manoj Jain"
To:"TAMAL SEN"
Cc: "P.K Chaturvedi"; "Om Vikas"
Sent: Thursday, July 03, 2003 12:19 PM
Subject: [Indic] Bengali Proposal
> Dear Sir,
>
> Please see the forwarded mail from MR. Paul Nelson, regarding the Unicode
> for Bengali Script. Kindly send your comments to Unicode discussion forum
> with a copy to Department of IT, New Delhi.[443]

Though minor, this interaction was indicative of a much broader effort toward widespread collaboration. Throughout the course of 2003, the indic-computing community (previously discussed in Chapter 2) continued organizing workshops and building consensus regarding the adoption of uniform modern standards for Indic-language technology. In March 2003, the "First National Indic-Font Workshop" took place in Bangalore, bringing together 36 participants from

[442] Dasgupta, Sayamindu, "[indic] On the Bengali Proposal" Email, September 10, 2003.

[443] Jain, Manoj, "[Indic] Bengali Proposal" Email, July 3, 2003.

across India, Nepal, and Bangladesh, including at least one subscriber to the Ankur listserv.[444] Over three days, participants discussed why Indic scripts needed to make use of the OpenType font format and how Indic fonts could be built using open source tools, as well as developing strategies for information-sharing after the end of the event. The workshop led to the creation of new regional groups dedicated to continuing font development efforts within each local language community. Thus, although none of the core Ankur members attended this workshop, they became part of this regional community effort because conference attendees identified them as one of Bangla's most active local-language computing groups.[445]

The indic-computing group, for its part, was progressing more slowly than Ankur, but this was in part because it had a different aim: rather than developing working fonts, indic-computing aimed to raise awareness about the benefits of using Unicode and OpenType in India more generally, with the goal of building a pan-India movement for local language technology. In support of this goal, some indic-computing members attended another workshop in the fall, the "National Workshop on Unicode" in New Delhi, which was organized by the Ministry of Information Technology.[446] This workshop was important, not least because a number of prominent Unicode leaders, including President Mark Davis, would be in attendance for the first time, as per the action item at the most recent meeting of the Unicode Technical Committee. We will return to government thread in the storyline in Chapter 5, but first, we must cover a series of new techno-linguistic challenges cropping up on the Unicode Indic listserv in the middle of 2003.

The Ongoing "Reform and Rationalization" of the Bangla language

Before we can make sense of the next issue facing Bangla digitization, we must first review the history of the Bangla language. Previous chapters have discussed the technological history of Bangla typesetting (Chapters 1 and 2) and the political history of Bangla language policy (Chapter 3); here, I discuss the linguistic history of the language itself, which will help us understand the challenges facing digitizers in the 21st century.

Throughout the 20th century, the Bangla language and script has been undergoing continual "reform and rationalization," as termed by the West Bengal Language Academy.[447] Although Bangla has characteristics in common with other Indic languages, it first emerged as a language in its own right in approximately the 11th century, when distinctive verb inflections and pronouns such as *ami* and *tumi (me* and *you*) developed and the Bangla script took on characteristics separate from Devanagari.[448] Throughout the medieval era (12th-15th centuries),

[444] Aditya, Vijay Pratap Singh, "[Indic-computing-users] Action points First National Indic-Font Workshop, March, 2003" Email, April 7, 2003.

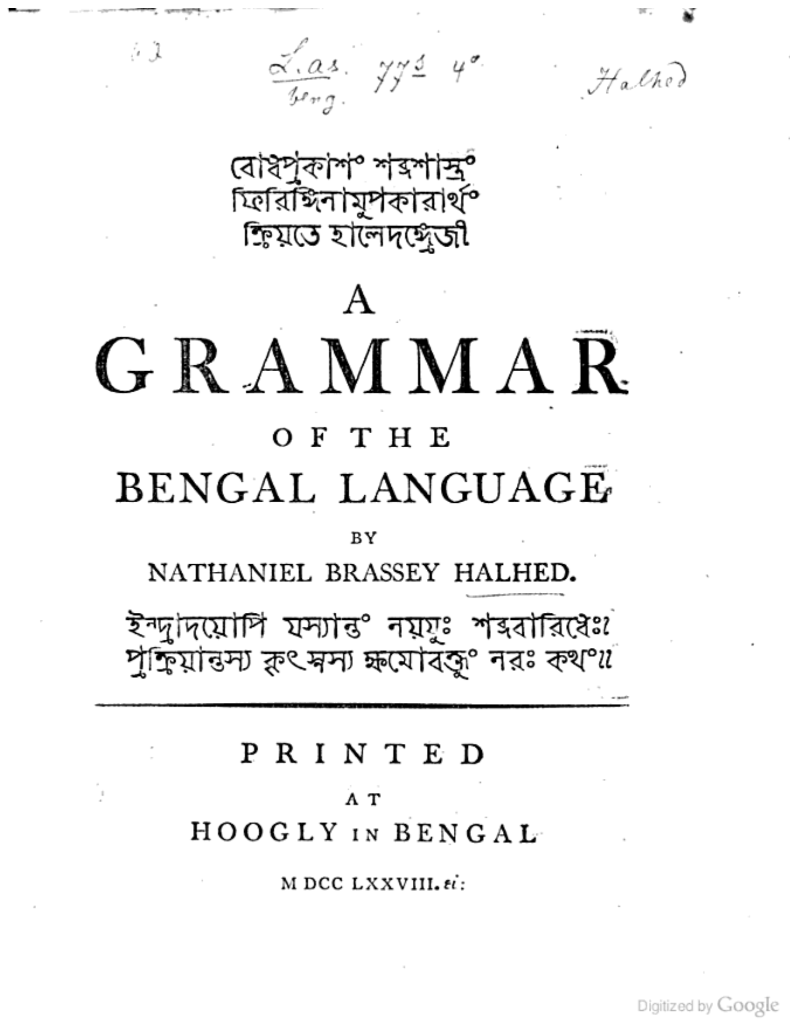
[445] *Ibid.*

[446] Dutta, Abhijeet, "[Indic-computing-users] Unicode Workshop - Sept24-26, 2003, New Delhi" Email, September 9, 2003.

[447] Riccardo Olocco, "Linotype Bengali and the Digital Bengali Typefaces," MA thesis, University of Reading, 2014.

[448] Hanne-Ruth Thompson, *Bengali: A Comprehensive Grammar* (Taylor & Francis, 2010), 10.

however, the Bangla writing system was mostly used for Sanskrit.[449] Additionally, as discussed in the previous chapter, Persian words continued to be added to the Bangla language throughout the Mughal period (16th-18th centuries), reflecting a fluidity of the language.
With the arrival of the British in the 18th century, the Bangla script went through its first major reform thanks to the efforts of European linguists and typographers (i.e. the "orientalists" discussed in Chapter 3). Among these was prominent British philologist, Nathaniel Brassey Halhed, who produced the first Bangla grammar in 1778, entitled *A Grammar of the Bengal Language*.[450]



L.as. / Beng. 77½ 4° Halhed

বোধপ্রকাশং শব্দশাস্ত্রং
ফিরিঙ্গিনামুপকারার্থং
ক্রিয়তে হালেদঙ্গ্রেজী

A
GRAMMAR
OF THE
BENGAL LANGUAGE
BY
NATHANIEL BRASSEY HALHED.

ইন্দ্রাদয়োপি যস্যান্তং নযযুঃ শব্দবারিধেঃ।
প্রক্রিয়ান্তস্য কৃৎস্নস্য ক্ষমোবক্তুং নরঃ কথং॥

PRINTED
AT
HOOGLY IN BENGAL
M DCC LXXVIII.

Figure 36. *Halhed's* A Grammar of the Bengal Language (1778)

[449] Hanne-Ruth Thompson, *Bengali: A Comprehensive Grammar* (Taylor & Francis, 2010), 10..

[450] Thompson, 11.

Of course, the typographers responsible for typesetting such works (in this case, Englishman Charles Wilkins and his Bengali apprentice Panchanan Karmakar), had an equally large role in standardizing the Bangla script. Wilkins and Karmakar succeeded in producing the first-ever wooden Bangla typeface, apparently gleaning their letterforms from medieval Bangla manuscripts.[451] Using wooden blocks, or "sorts," for these letterforms meant there was no need to restrict them in number, the Wilkins-Karmakar typeface included many complex conjunct forms. Interestingly, this typeface also included some reduced forms of consonants for use in combined forms to create conjuncts, which anticipated the typesetting strategy that would be used in the 20th century in the case of metal line-casters.[452]

The next era of reform and rationalization, known as the "Bangla Renaissance," speaks directly to the significance of *khanda ta*. The 19th century saw the rise of a rich Bangla literature, ushered in by poets and writers such as Michael Madhusudan Dutt, Bankim Chandra Chatterji, Ishwar Chandra Vidyasagar, and later Nobel laureate Rabindranath Tagore, all of whom composed histories, sonnets, and translations in the Bangla language.[453] Though these figures are often credited with establishing a genuine indigenous literature on the Indian subcontinent, their literary impact cannot be separated from the legacy of British colonialism. Because all of these writers belonged to the privileged *bhadralok* (or "gentleman") class in British India, they benefited from the English education and acculturation efforts set forth in "Macaulay's Minute," a treatise establishing the primacy of English amongst Indians (described further in Chapter 3).[454] Seeking to emulate the distinguished literary tradition of Great Britain, these educated Bengali *bhadralok* worked to develop a language that would provide a suitable medium for their literary and cultural ambitions. As Acharya writes, "The crisis [of establishing a high-status literary tradition] was resolved by developing a Bengali language and literature which had the sophistication of Sanskrit and the secularity of English to suit the growing new social class [of bhadralok]."[455] In other words, in order to make their prose as elevated as the British writing style they admired, the leaders of the Bangla Renaissances introduced heavy Sanskritization into their work. Vidyasagar and Bankim in particular worked to establish a formal literary standard for Bangla, known as *sadhu bhasha,* meaning "pure" or "chaste" language — an echo of the "purity" of Sanskrit exalted by European Orientalists.[456] By emphasizing Sanskrit in this way, Bangla authors helped elevate the status of Bangla amongst other Indic-language vernaculars, bolstering the argument that Bangla could serve as an appropriate vehicle for knowledge.[457]

---

[451] Fiona Ross, "The Evolution of the Printed Bengali Character from 1778 to 1978" (University of London, 1988), 33.

[452] Ross, 59.

[453] Thompson, 11.

[454] Poromesh Acharya, "Development of Modern Language Text-Books and the Social Context in 19th Century Bengal," *Economic and Political Weekly* 21, no. 17 (1986): 745–51, 749.

[455] Acharya, 749.

[456] Acharya, 750.

[457] *Ibid.*

The same author, Vidyasagar, played an equally important role in standardizing the modern Bangla alphabet. In 1855, he was tasked by the educational institutions of Bengal with writing a primer on the Bangla language for children. The resulting work, *Borno Porichoy* ("Introduction to the Alphabet"), refined the Bangla alphabet so that it consisted of 12 vowels and 40 consonants.[458] His preface presented his justifications for these orthographic changes, which included introducing three dotted characters, moving three diacritics to the list of consonants, moving another character to the list of conjuncts, and removing two letters that Vidyasagar argued had become obsolete.[459]

Most notably for our purposes, Vidyasagar's reformed alphabet specifically included *khanda ta*, establishing it as a distinct letter of the alphabet, rather than treating it as a mere visual flourish. Vidyasagar included standardized forms of several ligatures from the conjunct list as well, which would later become the target of reform in the 2000s, as the Bangladesh and West Bengal language academies sought to simplify the language. For Vidyasagar, however, conjuncts represented an especially important part of the Bangla language, as they were essential to writing Sanskrit words, and there was nothing Vidyasagar liked more than writing a Sanskrit-laden Bangla.

Among these Sanskrit words introduced into the Bangla language were *tatsama* terms, which had not been changed from their original or "pure" form, and *tadbhava* terms, which had been adapted to Bangla phonological patterns. The rest of the Bangla lexicon was comprised of *deshi* (indigenous) and *bideshi* (foreign) words, which entered Bangla from languages as diverse as Persian, Arabic, English, and Portuguese.[460] As we might imagine, the proportion of *tatsama* words used in Bangla literature has waxed and waned over time, depending on the aesthetic commitments of the author. By the start of the 20th century, writers such as Tagore had begun writing in a more colloquial Bangla instead, known as *cholito bhasha*, or "current language," which they believed better reflected the language used in everyday life.[461] Even as *sadhu bhasha* fell out of fashion, however, a number of Sanskrit loan words persisted in the language, meaning that the conventions for writing them in Bangla script persisted too — including the use of *Khanda ta*. One element that would find contemporary echoes from this history would be the tussle between simplifying and adding complexity to Bangla, which would affect the prevalence of loan words in the language and of complex conjuncts and diacritics in the alphabet.

[458] Ross, 258-9.

[459] *Ibid.*

[460] Thompson, 14.

[461] Thompson, 16.

Borno Spostikoron: “Clarifying” the Bangla Alphabet

Throughout the 20th century, the Vidyasagar alphabet continued to be what most children learned in school. By the mid-1970s, however, some Bangladeshi institutions had begun attempting to further reform the alphabet’s letterforms. The Bangladesh National Curriculum and Textbook Board announced the *Borno Spostikoron* (“Letter Clarification”) campaign in 1990, which called for the use of fewer ligatures and more “transparent” or “component-conjuncts.”[462] The goal, they claimed, was to aid reading comprehension.

As a Bangladeshi local, Omi Azad had grown up with the Vidyasagar alphabet — and with the reform efforts swirling around it. When Azad began working with Microsoft, he brought up the proposed *Spostikoron* reforms and advised the company to re-work their *Vrinda* font to support this reformed, modern version of the Bangla alphabet.[463]

Azad’s proposal caused quite a stir on Unicode’s Indic list, where the font developers and text renderers expressed initial frustration with the prospect of a reform to address. What exactly was this new orthographic standard? It wasn’t available anywhere online, nor did it seem to have any official documentation. The only proof of its existence was the word-of-mouth of Bangladeshi commenters — and this anecdotal evidence didn’t exactly suit the desire for documentation and proceduralism among the Unicode community. As one developer posted: “I will attempt to obtain a copy of the dictionary. Do you happen to know the full details (Author, Publisher, ISBN number, and edition) and/or a source for the publication in Bangladesh which would accept a cerdit card?”[464] When no further information was forthcoming, Paul Nelson quipped, “Looks like a trip to Bangladesh is required, or to trust those who are providing information from that region.”[465] This moment reflected a contrast that would recur between how knowledge was presented and accepted between the Bangla language community and the Western developers.

Finally, one of the Indic list’s font developers resolved the debate by reverse-engineering the changes he believed the Spostikoron reform had instituted, based on some textbook pages that had provided by Azad, though adding the caveat, “it is difficult to be clear what in the publications is a reflection of the rules, and what is simply a consequence of having to juggle the 200-odd available glyphs in an 8-bit font so as to construct the text.”[466] What he meant was that the non-Unicode, ASCII-based fonts that the Bangladesh government had used to print the textbook were themselves limited in the number of available characters and the lettering may reflect the technical constraint rather than the linguistic reform.

[462] Nelson, Paul, “[indic] Re: Bangla Academy and” Email, July 3, 2003.

[463] *Ibid.*

[464] Meir, Mike, “RE: Bangla Academy and “ Email, July 3, 2003.

[465] Nelson, Paul, “[indic] Re: Bangladeshi Text Book Standard as it may affect the Bangla shaping engine” July 29, 2003.

[466] Meir, Mike, "[indic] Bangladeshi Text Book Standard as it may affect the Bangla shaping engine” July 24, 2003.

Without any official guidelines regarding the Spostikoron reform, this is what the Indic list's font developer managed to glean on his own:

> *The reform concentrates on increasing the transparency of Bangla text.*
>
> *1. ligature forms which are irregular and not immediately obvious are suppressed*
>
> *1. Irregular Ukar (VowelSignU) and Uukar and Rikar forms are replaced with the standard base letter with standard subscripted vowel signs*
>
> *2. Antoshto Ba (indicated here by Va) is reintroduced at least as far as being the base form of Bophola, (considered since approx 1845 to be subscripted _Ba)*
>
> *3. New ligature forms are introduced for some common irregular ligatures. These are distinct forms which could not be constructed from the Half-Full formulation.*
>
> *4. Forms which were previously usually modified to become the top element in ligatures (S_, Ss_, M_, N_), with the modification of the vertical upright to a rounded form, are mainly reformulated to retain the vertical upright, and often placed to the left of the main consonant. (But not always in the case of Ssa, and not in an alternative font which is also used in the documents)*
>
> *5. Sa forming the first element in a conjunct drops its upright, and is written to the left, unless it is S_Va, but not always.*[467]

In short, the reform un-ligated several conjuncts, splitting them into component parts that were more clearly identifiable. Both of the examples in Figure 37 below have the same pronunciation; *Borno Spostikoron* advocated for the latter representation (un-ligated) over the former (ligated).

Figure 37. *Borno Spostikoron* Conjunct Reforms From

Ligated *kssa*: ক্ষ

Un-ligated *ka+sa:* ক্‌ষ

[467] *Meir, Mike, "[indic] Bangladeshi Text Book Standard as it may affect the Bangla shaping engine" July 24, 2003.* .

Now that the Indic list members had reached a tentative understanding of the Spostikoran reform, their next difficult decision would be to choose the appropriate component of the multilingual computing stack to handle the changes. Would all fonts need to be completely revised so that they only included the reformed shapes? Or was it possible for a single font to support both ligated and unligated forms? If so, what settings would the rendering engine need to support? And if higher-level tools like rendering engines were not capable of handling reformed Bangla script on their own, would the Unicode Standard itself need to be revised?

Further complicating the picture was the lack of universal consensus regarding the *Spostikoron* reform, even within the Indian subcontinent itself. For instance, India's Bangla language academy, which was based in West Bengal, had not officially supported the same changes.[468] Thus, developers concluded, fonts could not be transformed wholesale so that they only supported the *Spostikoron* style. The multilingual computing stack would somehow have to accommodate both versions of the language — reformed and unreformed. One Unicode-based solution would be to write new codepoint sequences for the modified Bangla letters, using the *zwj* and *zwnj* control characters, which would dictate explicitly when two consonants should ligate into the *Vidyasagar* style, and when they should be un-ligated in the *Spostikoron* style. The drawbacks of this approach were obvious: as one commenter wrote, "this begs the question of which [i.e., *Vidyasagar* or *Spostikoron*] should be considered the default forms, and increases the burden on typists unnecessarily [as they would need to insert extra control characters as they typed]."[469]

These types of questions reflected the internal politics of the multilingual computing stack. As described in Chapter 1, an inherent quality of a "technical stack" is its modularity and boundaries, but also inescapable interdependencies. As standards-makers encountered challenges like *Borno Spostikoron,* they were facing situations when the boundaries of their "technical layer" were being refined or strengthened.

To some of the native-born Bangladeshis participating in the thread, these objections from outsiders felt like an undeserved critique of their mother tongue.[470] But as the Western developers were quick to assert, "This concern has nothing to do with the desirability of the reform, it has to do with the desirability of modifying one standard (the Unicode shaping rules) to accommodate another standard (the textbook Board Standard) UNLESS THIS IS NECESSARY."[471]

A clearer sense of the Unicode Standard, in relation to the rest of the "stack", starts to emerge from these repeated episodes of conflict. For many of Unicode's adherents, the Standard had come to serve the role of a sacred, superior technical standard, which could not be challenged

[468] Nelson, Paul, " [indic] Re: Bangladeshi Text Book Standard as it may affect the Bangla shaping engine" Email, July 24, 2003.

[469] Meir, Mike, "[indic] Bangladeshi Text Book Standard as it may affect the Bangla shaping engine" Email, July 24, 2003.

[470] Shamsuddoha, Ranju, "[indic] Re: Bangladeshi Text Book Standard as it may affect the Bangla shaping engine" Email, July 29, 2003.

[471] Meir, Mike, "[indic] Re: Bangladeshi Text Book Standard as it may affect the Bangla shaping engine" Email, July 29, 2003.

without their sense of its authority being compromised. Ironically, in many ways, the Unicode Standard had come to seem more unchanging and conservative than any of the language standards it was created to encode – which is especially striking given how slow language standards themselves have historically been to change.

As Paul Nelson from Microsoft jumped in to insist, yet again:

> *Unicode is an encoding standard and not a linguistic standard. The use of Unicode should reflect the use of the language and not vice versa…the important part of this topic is that the Unicode stream *DOES NOT CHANGE* based on orthography. The Unicode character stream remains constant. It is only the typographic rendering of the stream that may have differences.*[472]

In essence, what the Indic list was really debating was the proper role of Unicode within the multilingual computing stack: which functions should the Unicode Standard fulfill, and which should be relegated to typographic or linguistic layers? As Nelson insisted, the design of the Unicode Standard was never intended to take the role of shaping how language works (as in imposing hard-and-fast rules), but nor was it intended to reflect every change in orthography at the encoding layer. Rather, intermediate layers, such as font formats and rendering engines, would have to accommodate orthographic changes.

Ultimately, the solution that mailing list participants (Paul Nelson, Omi Azad, and a handful of unaffiliated font designers) hit upon was to make use of a tagging option available in the OpenType font format. As discussed in Chapter 1, each OpenType font file included several tables of data specifying features of the script. Two categories within these tables were labeled "script" and "language." For Bangla-language OpenType fonts, the script tag would specify that the font made use of the Bangla script, whereas the language tag would specify which linguistic conventions that font would follow — which had previously been used to specify the language in question (e.g. Bangla, Assamese, or Manipuri), but could now be used to differentiate between *Vidyasagar* and *Spostikoron* as well. As Paul Nelson explained, the combined use of script and language tags "creates the definition of an orthographic system. The ability to specify the orthography to be used will be available in future version of Windows and Office applications."[473] Much like Malayalam fonts which already made use of both "traditional" and "reformed" conventions, Bangla fonts could be created using "Bangla, traditional" or "Bangla, reformed" language tags that the rendering engines would be able to parse appropriately.

In this way, the OpenType font format actually preserved the full range of orthographic preferences within a language community, particularly in places where there was no widespread consensus regarding how letterforms should be written — as was the case when comparing India and Bangladesh, or even when considering practices within Bangladesh itself.

[472] Nelson, Paul, "[indic] Re: Bangladeshi Text Book Standard as it may affect the Bangla shaping engine" Email, July 30, 2003.

[473] *Ibid.*

Given the Unicode community's conversations around this issue, and the eventual solution they came up with, we can see that Unicode's technical designers took a thoughtful, informed approach to state-sanctioned orthographic reforms, trying hard to stay agnostic as to the desirability of any changes. Even when one UK-based font designer allowed himself to delve into personal opinion, he quickly dialed it back:,

> *I wonder if the Bangla Academy has really considered why Vidyasagar allowed the forms they now wish to dispose of, or denigrate, while rejecting others, why they found their way into the printed script when it would have been easier not to have had them, and why printers and publishers using movable type preferred them when they did not have to use them? But this is not the place to go into these matters…*
>
> *With regard to standards such as those proposed by the Bangla Academy (which relate to the appearance of complex text), or indeed that propounded by Vidyasagar, the question to ask is: "Do the existing rules allow this to happen?" It is not "Do the existing rules make this the only or default solution?"*[474]

Only a few months later, the Ankur mailing list found themselves having the same conversation. There, Sayamindu noticed that Kolkata's major newspaper, *Anandabazaar Patrika,* had just announced a move towards transparent or component-conjuncts.[475] Ultimately, the Ankur team arrived independently at exactly the same solution that the Indic list had: using the language tags available in the OpenType file format to differentiate between reformed and unreformed script. Unlike the Western font developers, one of the Ankur members located in India could easily by a physical copy of the reforms from a bookstore for eleven rupees.[476] Nonetheless, the independently-reached consensus to use OpenType to handle it suggests this decision had more to do with the affordances each technical layer was understood to have than the Unicode Standard's conservatism.

By the end of August, Paul Nelson had finished up making the rounds regarding PRI-9 with all the Bangla language stakeholders he could find. To cap his efforts, he sent another email to Indian government officials:

> *I spoke with the Unicode Technical Committee yesterday. We came to an agreement that I should seek to build concensus on how to standardize Bengali before the next UTC meeting in late September. Being able to arrive at a concensus is critical as there is no way to standardize implementation without people agreeing how this should be implemented.*

[474] Meir, Mike, "RE: Bangla Academy and" Email, July 3, 2003.

[475] Dasgupta, Sayamindu, "[Freebangfont-devel] Changes in the Bangla Juktakhar system" Email, January 13, 2004.

[476] Mukhopadhyay, Sankarshan, "Re: [Freebangfont-devel] Changes in the Bangla Juktakhar system)" Email, February 2, 2004.

> *I am trying to finish up the Windows update of Bengali script support. I am aware that there is also an effort in the Linux community to have Bengali support. Because the only contention remaining is limited to a few issues, I hope that we can have some dialog to finally arrive at a unified approach…*[477]

He went on to summarize the suggestions that had been made in PRI-9 for handling *jofolas* and *rephs*, along with presenting the four-point sequence solution for *khanda ta*. He also described the OpenType language tag that would be used for handling the two different Bangla orthographies. Despite Nelson's claim that he "should seek to build concensus, [sic]" it seemed that consensus was already nearly achieved. But as Nelson himself noted, there was "no way to standardize implementation without people agreeing" — and soon a new note of contention would throw the Indic list again into controversy.

The Double Nukta

The newest problem was literary in nature, reflecting the cultural significance of Bangla as a mode of artistic expression. Dr. Ketaki Kushari Dyson, an acclaimed writer and translator (born in Kolkata, India, but then residing in Oxford, England), was facing a translation problem. Dyson frequently wrote in multilingual prose, and also frequently translated between multiple languages using multiple scripts. In this line of work, one orthographic convention she found particularly useful was the double nukta (a series of diacritic dots positioned below a base letter), which could be added to a Bangla *ja* to soften the enunciation, changing it from the hard "J" heard in an English word like *major*, to the soft "J" heard in the word *measure*.[478] Semantically speaking, the soft *ja* had no place in the Bangla language. But it was essential when transliterating French poetry, where the first-person pronoun (je) made frequent use of the soft "J." In addition, within Bangla itself, the double nukta convention had appeared prominently in the work of poet Buddhadeva Bose, who was so prolific and renowned that he was seen by many as the successor of Rabindranath Tagore.[479] In addition to his own original works, Bose had produced translations of the French Romantic poet Charles Baudelaire into Bangla, making use of the double nukta to capture the phonetics of the French language.

Dyson first brought this issue to the Indic list in July 2003, but it was not easily resolved. It would resurge as an issue over and over over the coming year. At some points, the mailing list's subscribers wondered: Was this important enough for Unicode to address? One commenter, a Kannada speaker, seemed to doubt the significance of a single author like Baudelaire. How often would the need for a double nukta really arise?[480] Dyson responded with resolute literary authority. It was not only the works of a single poet that required a soft *ja*. Common English

[477] Nelson, Paul, " [indic] Re: Bengali Proposal" Email, August 28, 2003.

[478] Meir, Mike, "[indic] Double Nukta on Ja" Email, July 31, 2003.

[479] *Ibid.*

[480] Pavanaja, U. B. "[indic] Re: [Bangla] not just khanda ta" Email, June 20, 2004.

words such as ‘genre’ and ‘collage’ also required the convention, as did the name of a beloved Shakespearean play, “Measure for Measure.” Therefore, Dyson insisted, students in Bangla-language schools needed to have access to the double nukta in their pronunciation guides, lest they go on to pronounce transliterated words with the traditional Bangla hard *ja*, “which is horrible.”[481]

In fact, Dyson’s argument here went beyond the academic context. As described in the section on linguistic evolution presented earlier in this chapter, the Bangla language had a long history of absorbing loan words from various languages, including Persian, Arabic, and English. One might write the transliterated word “স্কুল,” which says “school,” just as often as they might write the Bangla word “বিদ্যালয়”, which says “bidyalay” – two words with the same meaning but transliterated from English and Sanskrit respectively. In this sense, it was important for all Bangla speakers, not just students, to have the appropriate characters at hand to represent foreign pronunciations.

The presence of loan words in a language has traditionally been a target of orthographic reform. In postcolonial states in particular, reformers often make the case that continuing to use the loan words of the colonizer is the equivalent to continuing a form of linguistic subjugation, one that prioritizes modernization and internationalization over local culture and communication.[482] During state-mandated script changes, then, it’s common for loan words to become a topic of discussion. If a state were considering switching their writing system to the Latin script, for instance, legislators might express the concern that English words would slip into the language more easily.[483] Even in the case of reforms not involving a script change, such as the *Spostikoron* reform, issues of foreign pronunciation (like the soft *ja* presented by Dyson) frequently become cause for concern.[484] Should the word be pronounced using the conventions of the borrowing language, or the loaning language? Should Bangla speakers learn to say “meajur” with a hard *ja* or “measure” with the soft?

In some sense, Dyson and the Bangla Academy fell on opposite sides of this debate. Whereas the *Spostikoron* reform sought to remove conjuncts from the language to make it as easy as possible for native speakers to use, Dyson wanted to add additional markings to the language to expand its capacity for phonetic expression. As she herself put it, her proposal stood in opposition to the “simplification reforms” that had already been instituted in Bangladesh and were now looming in West Bengal.[485]

[481] Dyson, Ketaki Kushari, “[indic] Re: [Bangla] nuktas under ja” Email, June 22, 2004.

[482] Sebba, 99.

[483] *Ibid.*

[484] *Ibid.*

[485] Dyson, Ketaki Kushari, “[indic] Re: [Bangla] not just khanda ta” Email, June 22, 2004.

Yet in another sense, Dyson's double nutka had something in common with the Spostikoron academy's un-ligated forms. Both of these orthographic proposals had significant implications for the design of the multilingual computing stack. But again, it was a question of *how,* not *if.* As things stood, Unicode already had a nukta symbol encoded in the Devanagari block. The question was, what should be the protocol for displaying two nuktas below a single base letter? Should a new, standalone diacritic be added to the Standard? Or should users be tasked with typing two nuktas after one another, perhaps adding a *zwj* or *zwnj* in between to instruct the rendering system that the two nuktas should be paired together? If so, how should these "combined" nuktas be shown? And what about the concern that relying on too many control characters would lead to display errors — not to mention user frustration? As one frequent Unicode contributor chimed in, "Oh, for heaven's sake can we please stop using all of these ZW things as control-character fixes?"[486]

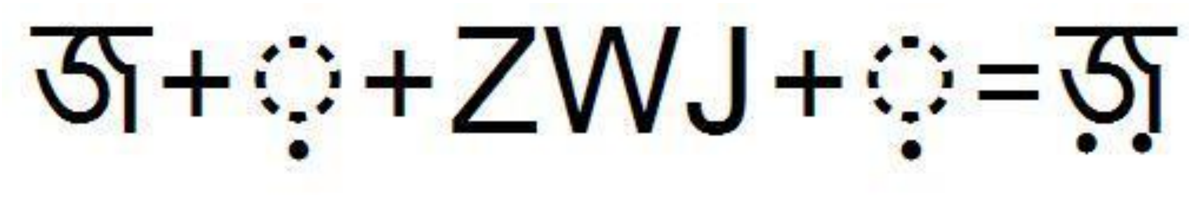


Figure 38. A Mock-Up of the Preferred Double Nukta Display

Ideally, the double nukta diacritics would appear side by side under the same letter, although it was also possible to stack them on top of one another vertically, as was the default case for diacritics in other scripts. In German, for instance, a user could stack five umlauts atop each other without penalty, perhaps to express a particularly emphatic "ouuu" sound.[487] Yet Indic scripts seemed to have been encoded following an entirely different logic, at least within the logic constraints of Microsoft's rendering engine. Nuktas needed to be associated with a single base letter, or else they would appear with a dotted circle over them, which indicated that an illegal sequence had been inputted.

This forced dotted circle "error" produced a great deal of controversy, even amongst designers of the same multilingual computing stack. At least one UTC member noted that "as a casual user I would have called [forcing dotted circles] dumb, and also fascist."[488] But from the perspective of Microsoft employees Paul Nelson and Peter Constable, Indic text could not be rendered properly without identifying valid "clusters" within the text. These "clusters," which mapped roughly onto individual syllables, were governed by rules written into Microsoft's rendering engine, which tried to ensure that text would appear appropriately (at least in the most common cases). The use of "clusters" also meant that higher level software, like text-to-speech readers and other

[486] Everson, Michael, "[indic] Re: Double Nukta on Ja" Email, July 31, 2003.

[487] McGowan, Rick, "[indic] Re: Double Nukta on Ja" Email, July 31, 2003.

[488] *Ibid.*

advanced natural language processing tools, could process Unicode sequences by interpreting a set of consistent rules.[489] In sum, changing the rules of the rendering engine for the relatively uncommon edge case of the double nukta would upset the delicately stacked set of rules that enabled regular Indic text to work.

Nor were the rules of the rendering engine the only justification. Nelson reported that the company "[had] gotten feedback from *a lot* of users who have found the enforcement of linguistic rules and dotted circle feedback helpful."[490] Thanks to this pushback on Microsoft's part, the double nukta issue created a seemingly unresolvable conflict between the Unicode encoding standard and the Microsoft rendering framework used to display it: which technology should be forced to make the necessary accommodation? In the words of the outspoken Ken Whistler, member of the Unicode Technical committee, "this sounds suspiciously to me like the implementation's tail wagging the standard's dog."[491] By this, he meant that the rendering layer was driving too many of the decisions in the tussle between layers of the multilingual computing stack. In addition to the desired adherence to Unicode's design principles as a motivating factor, Whistler added the warning that, if "representation and rendering" issues like the double nukta question got turned into encoding issues on the backend, that meant going through "the process of proposal bureaucracy and two years of tracking things for standardization before we have an issue to the text representation issue."[492] His reference to the synchronized Unicode/ISO 10646 decision-making process added a justification for the conservative nature of the encoding layer — this was a vast, official, multi-stakeholder process, in contrast to the flexibility and relative independence of Microsoft and Uniscribe.

Garbage Type

Long before the double nukta issue could come to a resolution, however, the multilingual computing community found itself facing a new "representation and rendering" concern: the demand for "garbage type." Omi Azad had been entirely absent from the double nukta debate; those who attempted to reach him received only an "out of office" vacation notification:

> *Hello,*
> *Your mail came to me but I have decided to take a brake from computer for a month. So I'll not check mails till a month. So, I'll reply your mail after September.*[493]

[489] Nelson, Paul, "[indic] Re: Double Nukta on Ja" Email, July 31, 2003.

[490] Nelson, Paul, "[indic] Re: Double Nukta on Ja" Email, July 31, 2003.

[491] Whistler, Ken, "[indic] Re: Double Nukta on Ja" Email, July 31, 2003.

[492] Whistler, Ken, "[indic] Re: Double Nukta on Ja" Email, July 31, 2003.

[493] Azad, Omi, "Hello, " Email, August 25, 2003.

When he did finally break his silence, however, it was impossible to miss his dramatic reappearance. Posting to the entire Indic list, Azad launched his complaint: "After coming back from the brake I learn that Paul is not implimenting garbage typing."[494]

Deepayan, still unclear on Azad's specific demands, shot back, "I have no idea what you are talking about."[495]

But gradually his request became not only clearer, but more and more obviously warranted. As Azad explained,

> *My issue was, If I can type "askludhgiweuynckvmzx" [in English] why I cannot type anything like that in Bangla….My question is why I cannot type something like that? Think once, if I want to make a book of wrong combinations,..then how I'll do that using Unicode based OTF font?*[496]

And indeed he was right! Azad had identified a telltale limitation within Unicode/OpenType fonts, one that denied Bangla users the full range of expression that English users took for granted — the ability to "mash" keyboard letters to express frustration, humor, or astonishment. This was because of the way Microsoft's Uniscribe engine (and others that mirrored its specifications, such as the open source analog, Pango) handled Bangla modifiers. If a user tried to bang out "garbage type," any time they happened to enter a solitary diacritic or a vowel modifier then those modifiers would appear around a dotted circle taking the place of the base letter was that expected, just as an unattached nukta would do. And there was no easy way to remove that blatant dotted-circle error symbol. If someone wanted to include a standalone *ikar* or *okar* for any reason — whether it be to clarify pronunciation within a textbook, or simply to express themselves as creatively and freely as possible — the dotted circle would be imposed upon them. One contributor raised the possibility of expressing "Baaaaaaaaah Humbug!!" in Bangla; there was no "authorized" convention for this, but might a typist not want to input a series of aakars, without dotted circles breaking up the expression?[497]

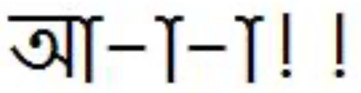


Figure 39. Example of Garbage Type: "AAAH!!"

Could this issue with garbage type have occurred in any other medium? Of course, any new printing technology introduces new constraints as well as new opportunities. One particularly apt demonstration of this phenomenon comes from the development of Bangla typography in the

[494] Azad, Omi, "[indic] Re: On the Bengali Proposal" Email, September 15, 2003.

[495] Sarkar, Deepayan, "[indic] Re: On the Bengali Proposal" Email, September 15, 2003.

[496] Azad, Omi, "[indic] Re: On the Bengali Proposal" Email, September 17, 2003.

[497] Meir, Mike, "[indic] Re: Bangla: [ZWJ], [VIRAMA] and CV sequences" October 9, 2003.

20th century. Bangla type, like other Indic type, had undergone a number of transformations in the days of early mass printing due to the physical constraints of hot-metal type machines. These hot-metal linecasters had become popular in the 1930s thanks to their high-speed production of prints, and demand for Indic-script printing quickly exploded, especially on the part of South Asian newspapers, such as Kolkata's *Ananda Bazaar Patrika*, and local university presses.[498] The linecaster had its limitations, however; the most popular model, developed by Linotype, had only a limited number of keys that could fit on the machine. For this reason, each Indic script could only make use of a maximum number of 90 glyphs, as opposed to the hundreds of glyphs that had previously been used in non-metal printing foundries.[499]

Thus, while the most frequently used conjuncts and letters made it onto the keyboard, everything else had to be implemented using "half-glyphs" (discussed previously in Chapter 2). These half-glyphs had to be squeezed together side-by-side, and often ended up looking very different from both handwritten Bangla and previous foundry type. Half-glyphs remained in use until well into the 1980s, when advances in type technology (such as the phototypesetter discussed in Chapter 1) would make it possible to expand the number of glyphs once more — at least, in theory. The great irony was that the very restrictions that had once been felt as a constraint had actually become the preference of many Bangla users in the interim. Even though Linotype was now capable of expanding the number of glyphs to their traditional, longstanding forms, then, many Bangla language institutions argued for a continuation of the hot-metal altered forms. What had started as a compromise accommodation had become the preferred linguistic solution, thanks to widespread exposure in *Ananda Bazaar Patrika* and other popular publications.[500] Over the decades, conjuncts typed with "half-glyphs" had become re-branded as "component-conjuncts" or "transparent conjuncts." These "transparent conjuncts" were exactly what were now being promoted by the official Bangla language academies and in campaigns such as *Borno Spostikoron* described above.[501]

When Omi Azad raised his desire for garbage type, then, Unicode veteran Ken Whistler brought up the above historical example as an important counterpoint. Much like the restricted Linotype glyphs led to evolutions in preferences, Whistler argued, the restricted Unicode encodings would lead to a new set of user behaviors. Whistler also provided an additional example that may be more familiar to American users: that of ASCII art, or digital images produced using the 128 ASCII characters.

As Whistler insisted,

> *By the way, neither was it ASCII's job to prescribe usage of writing systems, nor to prevent people from using their writing system as they saw fit. To the contrary, its very limitations*

[498] Ross, 267.

[499] *Ibid.*

[500] Riccardo Olocco, "Linotype Bengali and the Digital Bengali Typefaces," MA thesis, University of Reading, 2014.

[501] *Ibid.*

> *were probably responsible for the development of emoticons -- a wholly unforeseen outburst of creativity that took writing systems off into new directions…*
>
> *As I see it, we don't want to have Unicode make any aspect of conventional orthographic representation in the various writing systems it covers impossible just by oversight or failure of design. But it also shouldn't be viewed as a universal writing engine to mimic the endless possibilities of handwritten form directly. On the contrary, Unicode's very limitations will inevitably result in creations of new forms in the future--in ways we won't anticipate until somebody dreams them up.*[502]

I share this series of Bangla digitization trials brought forth to the Unicode Indic listserv to highlight two points. Firstly, these episodes highlight the tussles occurring internally amongst overseers of the technical layers of the multilingual computing stack — namely, the Unicode Technical Committee authorities responsible for the Unicode Standard, and the Microsoft employees responsible for OpenType implementations in the text rendering layer. User interests — whether it be requests for the proper representation of *ja* and *ra* glyphs, to compliance with new letterform reforms, to adding flexibility for loan words, to permitting an unbounded range of typographic innovation — must fall somewhere between them. We see where the limits to user requests are drawn: though *Spostikoron* reforms and new diacritics can be accommodated, "garbage type" seems like it will fall outside of the entire stack's purview. For the most part, however, orthographic reforms are largely heard and responded to by the technical mediators on the Unicode mailing lists.

These episodes also lay important context for the upcoming *khanda ta* debate because they established a sense of confrontation occurring between the Bangla user community and the professional standards-makers on the Indic list. Genuine representation issues such as *ja, ra,* and *khanda ta* were getting lost among esoteric requests for double nuktas and garbage type. When the issue of *khanda ta* would erupt in the same forum in the coming months, the standards-makers would have to determine where it fell on the spectrum of importance: what authority lay behind the requests, how important was it to act, and, of course, which layer of the technical stack was best poised to handle it?

```
                               ___         ___       T__
                    ________   | |~~~~~~~~~| ||~~~~| |||
__|~~~~~~~~|   _/\_ |^^^^^^|  _| |---------| ||    | |##
|_|HHHHHHHH|  _|--| |------|_-##########################
```

Figure 40. ASCII Art of Fortress

[502] Whistler, Kenneth, "[indic] Re: more than one maatraa on a consonant" Email, October 13, 2003.

Chapter 5: The Battle over *Khanda ta*

Previous chapters have introduced us to the various stakeholders in the *khanda ta* debate. Chapter 1 laid out the origins of the Unicode Standard, a sociotechnical system designed with Western and East Asian scripts in mind, relegating the display of Indic scripts like Bangla to yet-to-be invented technologies. The overseers of this system, the Unicode Technical Committee (UTC), valued consistency, procedure, and efficiency. Chapter 2 introduced us to the social actors who inherited this system — local language hobbyist groups who had access to these open standards and their own faculties to build open source tools that could enable the Bangla language community to live online. They were solutions-oriented, independent-minded, and loyal to their nations. In Chapter 3, we saw the rise of government agencies, namely India's Ministry of Information Technology, that took on the mantle of language planners in the digital age. For them, language digitization was a matter of national pride and progress; their tactics were a blend of the analog and digital — paid memberships to join cutting edge industry consortia, but a preference for slow hand-posted communications over fast online forums. Finally, in Chapter 4 we began to see the internal workings of the multilingual computing stack, as they grappled with a burst of user requests. We saw the relative flexibility and swiftness of the text rendering layer, and its designers (Microsoft typography program staff), in comparison to the encoding layer and its overseers (the UTC).

Now, we are ready to appreciate how these multiple interest groups come to intersect around a single high-stakes battle: one decision, in one script, over one letter. This case represents something different to each party: an edge case, a bug, a letter dropped from the alphabet. As arguments come to a head, a new stakeholder group also enters the discussion: academic linguists, who typically stand at the forefront of any orthographic war, but had been missing from Unicode discussions until this point. We see how the introduction of the linguist's perspective transforms the *discussion* over *khanda ta* into an inflamed *debate,* providing an opportunity for the between each group mentioned above to come to the fore and attempt resolution. In the end we are able to see how language digitization is idiosyncratic and all-encompassing at once, both mundane and stunning.

—-

In the August of 2003, just before Omi Azad's "garbage type" outburst, the Unicode Technical Committee (UTC) met to address outstanding issues like PRI-9.[503] Although the committee did manage to finalize the encoding rules for *ra* and *ja* at these meetings, they did nothing to address PRI-9's section on *khanda ta*, for reasons that remain unclear.

In response to this curious silence, Andy White submitted a draft proposal to Unicode in September that contained the same explanation of *khanda ta* as his earlier blog post.[504] Unicode,

[503] Moore, Lisa. "UTC#96 Minutes," UTC Document Registry, L2/03-240.

[504] White, Andy, "[indic] Request to revise the current model for encoding Bengali Khanda-Ta" Email, September 21, 2003.

White reminded the committee, was currently treating *khanda ta* as if it behaved in the same manner as Devanagari's half-forms, but this mistaken approach was causing a number of display problems for Bangla users. This was because Bangla traditionally did not contain anything like half-forms; the only Bangla forms that bore a passing resemblance to half-forms were, in fact, the linecaster imposed corruptions, half-*glyphs*, that appeared briefly in the 20th century. For this reason, treating Bangla like Devanagari would cause rendering problems, including vowel modifiers appearing in the wrong places, i.e. around *khanda ta*.

White, acknowledged Paul Nelson's four-code proposal for inserting *khanda ta*, but insisted that Nelson's solution was "only a partial workaround."[505] This was because, as White laid out, Nelson's encoding rule would produce additional errors, as there were too many possibilities for how the base letter *ta* might appear in a font, making it impossible to determine which glyph might be preferred by the user in advance. Finally, White pointed out, although Bangla did not contain true half-forms as distinct semantic units, it did sometimes make use of half-*glyphs* within a certain font to form "component-conjuncts" in the style of the hot-metal Linotype (and subsequent *Spostikoron* reforms*)* discussed above. This was the exact case that Paul Nelson had warned would cause his suggested rule to falter. Despite these objections, however, White was not yet advocating for *khanda ta* to be encoded as a unique character in Unicode. He was only advising that Unicode opt for a different sequence of codepoints to produce *khanda ta* than Nelson's proposed solution – namely, to insert *khanda ta* by entering it as the *halant*-form of ta, as the latest versions of ISCII were now doing.[506]

With each new message, the hobbyists grew more and more confused as to why Unicode was not responding to their comments about *khanda ta*. After White submitted his proposal to the Indic list, Azad responded, "Dear Andy, We have talked on issues like this many times ago. But I don't think someone really care about these. So… Keep trying."[507] To this, White replied, "Try Try and Try again. I was having deja-vu when writing the document."[508] Indeed, the readers of this dissertation have heard this argument from some of its earliest pages; White's argument has been cooking since Chapter 2 (or November 2002 in the historical timeline).

Meanwhile, over at Ankur, Sayamindu and Deepayan were wondering why their joint comment on *khanda ta* had received so little response on the Indic mailing list, despite having been actively solicited by Paul Nelson to begin with.

> *Now that you mention it, there seems to have been no reaction to Sayamindu's prior post to the indic@unicode list. Maybe we should submit that more*

---

505 White, Andy. "A Request to revise the current model for encoding Bengali Khanda-Ta" http://web.archive.org/web/20040923213157/http://www.exnet.btinternet.co.uk/uniprop/KhandaTaEncode.pdf , 2.

506 *Ibid.*

507 Azad, Omi. "[indic] Re: Request to revise the current model for encoding Bengali Khanda-Ta" Email, September 21, 2003.

508 White, Andy, "[indic] Request to revise the current model for encoding Bengali Khanda-Ta" Email, September 21, 2003.

> *formally, and hope to get better results this time.*[509]

Even as they debated the best course of action to get an actual response, the Ankur members still claimed to be agnostic on the issue at hand: "I think there's nothing for us to do really, since we are happy with the way things currently are…"[510] Yet Sayamindu and Deepayan's joint statement *did* take a stance. It acknowledged that there was a serious problem with khanda ta, and stated a preference for White's proposal as well (since they, like White himself, had found Nelson's proposal to be too error-prone to be workable).

But perhaps Sayamindu and Deepayan were simply trying to draw a distinction between themselves and the government officials from India's Ministry of Information Technology, who were becoming increasingly forceful and aggressive in their recommendations. The Ministry had just hosted a "National Workshop on Unicode" in New Delhi in March 2003, which was attended by a handful of Unicode core staff, including President Mark Davis, who gave the keynote address. Davis's keynote showed that the Unicode consortium was trying to fulfill the promises it had made earlier in 2003, when it first began interfacing more actively with the Indian government's TDIL through in-person site visits, as well as collaboration on the Indic mailing list.

This "National Workshop on Unicode" included several presentations by representatives of different language communities, each of whom brought forth their own linguistic issues with respect to Unicode. One of the Ankur members residing in India, Indranil Dasgupta, attended the workshop and reported back:

> *It seems that TDIL (DIT/GoI) is once again poised to push forward another Bangla proposal to Unicode Consortium asking for allocation of code-points to khanda-taa, reph, ja-phala et al. (as evident from a presentation by Dr B B Choudhuri/ISI, Kolkata)*[511]

In fact, TDIL was planning to write proposals not only for Bangla, but for India's other major language communities as well — a strategy that was not at all clear would prove practical or effective. After all, although the Unicode Consortium seemed to be keen on building diplomatic relationships with India, it did not appear eager to grant the country's wishes. Indranil Dasgupta's email ended with the cautionary note that Unicode representatives encouraged anyone wishing to make changes to the Standard to write organized, well-reasoned proposals following the prescribed format. A careful, considered approach seemed like the best strategy for all involved; otherwise, requests risked getting denied before they got a proper hearing. As another Workshop attendee, V.S. Umamaheswaran ("Uma"), warned on the Ankur mailing list:

---

[509] Sarkar, Deepayan. "[Freebangfont-devel] Re: [Ankur-core] RE: Update on NationalUnicode Workshop & Bengali Proposal on Indic m/l" Email, October 11, 2003.

[510] Sarkar, Deepayan. " Re: [Ankur-core] RE: Update on NationalUnicode Workshop & Bengali Proposal on Indic m/l" Email, October 8, 2003.

[511] Dasgupta, Indranil. "[Ankur-core] Update on National Unicode Workshop & Bengali Proposal on Indic m/l" Email, October 4, 2003.

> *Before any proposal is made .. please do go through the steps I outlined in my presentation at the workshop .. on 'do your homework' .. especially the checking and ensuring that sequences described in Unicode V4.0 Ch 9 / FAQ / UTNs .. are indeed 'not adequate' for representing the characters you have mentioned.*
>
> *Even though presentations were made in earlier TDIL documents and at the workshop - such as by Prof Chowdhury, my impression is not that the workshop has opened up any path for new proposals on the characters you have mentioned. I am of the impression that sequences given for these are adequate for encoding them.*[512]

Uma's "do your homework" presentation on the proper way to submit proposals suggested that unreasonable, unwarranted requests had been slipping in from TDIL officials who were not sufficiently familiar with Unicode's design principles and encoding logic. In this context, Ankur wanted to position itself as a more informed, reasonable, and experienced group than TDIL. Ankur may not have had a strong stance on the question of a new encoding one way or another, but they wanted to make it abundantly clear that *they* understood the technical constraints, and that there *was* indeed an issue.

Introducing Gautam Sengupta

Indeed, it was so clear that there was still a problem that the issue soon escalated from the hobbyist into the academic realm. In December 2003, Peter Constable, the Typography program manager who had been working with Paul Nelson at Microsoft, attended a conference in Tokyo where he met Kolkata linguist Dr. Gautam Sengupta.[513] The conference was called the "International Symposium on Indic Scripts: Past and Future," and had been organized by the Research Institute for Languages and Cultures of Asia and Africa, which was located within Tokyo University. The conference itself was unremarkable – just another casual stop on an academic's annual conference circuit – but in retrospect, it marked the point when Sengupta took on the mantle of championing *khanda ta's* importance to the world.

Sengupta's conference paper emphasized the similarities between the Bangla letters *khanda ta* (ৎ) and *anusvara* (◌ং).[514] According to Sengupta, both of these letters were "silenced" consonants; *khanda ta* was the Bangla consonant *ta* (ত) with a silenced inherent vowel, making it analogous in sound to the Latin letter 't'. *Anusvara* was the Bangla consonant *nga* (ঙ) with a silenced inherent vowel, giving it the 'ng' sound, as in the middle of the word "Bangla." *Khanda*

[512] Umamaheswaran, Uma, "[Ankur-core] Update on National Unicode Workshop & Bengali Proposal on Indic m/l" Email, October 6, 2003.

[513] Sengupta, Gautam, interview with author, October 20, 2021.

[514]"International Symposium on Indic Scripts: Past and Future: Organized by Research Institute for Languages and Cultures of Asia and Africa Tokyo University of Foreign Studies Tokyo, December 17-19, 2003: Working Papers," 東京外国語大学附属図書館ＯＰＡＣ, accessed June 29, 2022, https://www-lib.tufs.ac.jp/opac/recordID/catalog.bib/BB13653529.

ta and *anusvara* had different linguistic genealogies, yet served a similar function in the present-day Bangla alphabet. This parallel made it all the more striking that *anusvara* had its own codepoint in Unicode — and *khanda ta* still did not.

With his carefully researched and professionally presented paper, Gautam Sengupta became the first person to officially begin advocating for a unique codepoint for *khanda ta* within Unicode. This marked a significant shift. Up until this point, there had only been requests for minor revisions to the way *khanda ta* was rendered, making *khanda ta* an ambiguous issue because it lay between Unicode's soft laws (i.e. the Indic FAQ page or the preamble of the Standard) and Microsoft's rendering engine rules. As long as *khanda ta* was presumed to be some graphical variant of *ta*, as previous listserv comments and even Andy White had treated it, developers assumed that whatever rendering issues it had could be handled at these layers. When Sengupta began proclaiming that *khanda ta* had significant linguistic status of its own, however, he asserted that *khanda ta* would need to be handled at the critical base layer instead: the Unicode Standard.

Just who was this professor to make such a bold claim? Sengupta would hold a unique position in the Unicode milieu. As Azad recalled in a later interview, "We used Bangla but we did not study it in school.... Gautam-da[515] explained really well to us, he wrote papers, he explained the grammar."[516] It could hardly come as a surprise that Sengupta explained so well, given that he was well-versed in both linguistics and technical systems. He had earned his PhD in Philosophy at the University of Massachusetts-Amherst, where he had studied formal linguistics. While there, he had worked as a research assistant for renowned Electrical and Computer Engineering Professor Bill Kilmer, and had developed an aptitude for formal logic and programming that he deeply enjoyed. By 2003, he had returned to India to become a professor of applied linguistics at the University of Hyderabad.[517]

Sengupta's foray into Unicode only came about as the result of a side project, which involved digitizing ancient Sanskrit texts.[518] As part of the digitization process, Sengupta began investigating encoding schemes and keyboard systems, including ISCII, Unicode, and Keyman (a make-your-own open source keyboard tool adopted by SIL). While Sengupta appreciated ISCII's cleverness, he recognized its code range (only 8 bits) was too small to be practical for multilingual documents. ISCII also produced numerous errors, in part because - as Sengupta knew quite well - its inventors were versed primarily in Devanagari. "Maybe they didn't even notice *khanda ta*," he would later muse.[519] Thus, even as Sengupta tried his best to make ISCII and its associated INSCRIPT keyboard do what they were designed to do – i.e. allow easy touch-

[515] "-da" or "-di" are Bengali honorifics to mark respect for elders

[516] Azad, Omi, interview with author, January 30, 2022.

[517] Sengupta, interview.

[518] *Ibid.*

[519] *Ibid.*

typing and translation between Indic scripts – his efforts proved so error-prone that any advantages gained were slim.

For these reasons, Sengupta soon moved on from ISCII to Unicode, and was immediately struck by how beautiful and expansive the Standard seemed in comparison. Much like Unicode's own founders, Sengupta intuitively understood the need for a single, unified, all-encompassing multilingual standard – one that overcame the limitations of the national and proprietary standards and switch codes of the 1980s (discussed previously in Chapter 1).[520] Yet as much as Sengupta admired Unicode, he noticed errors in it as well: extraneous or nonsensical characters that were encoded in the Indic block, as well as valid characters that were completely absent. When Sengupta had been working in ISCII, he had found himself wishing that more linguistic experts had been involved in its original design, as opposed to only computer engineers. When he began working with Unicode instead, he found himself wishing that South Asian governments had been able to offer more initial input on the Standard, as opposed to Western technology companies simply incorporating ISCII as-is without questioning its limitations.[521]

Because of these historical contingencies, for Sengupta, the *khanda ta* debacle - which he would spearhead for the next six months - represented more than a single, hard-to-type letter. It was paradigmatic of the failure of Western computing as a whole to take into account the specific techno-linguistic needs of language communities throughout the Global South. As such, it represented a crisis, or perhaps an opportunity, to reassert the quintessential importance of linguistic expertise in multilingual computing, even when it came to the design of a so-called purely "technical code." The cascading series of failures that engendered the *khanda ta* crisis demonstrated, more clearly than ever before, that South Asian perspectives must be heard and accommodated in the global computing world. For Sengupta, however credentialed the Microsoft program managers might be, they still represented the narrow-minded and self-interested American corporate agenda. In sum, when Sengupta entered the debate, he elevated the discussion from a single "missing" letter to a standoff between industry and academia, between the Global North and the Global South.

Re-Introducing Peter Constable

Soon after hearing Gautam Sengupta's presentation on the full linguistic implications of *khanda ta* in December of 2003, Peter Constable, Microsoft's head of typography, began posting in the Indic mailing list in late January of 2004 with questions about *khanda ta*'s linguistic behavior: "do vowel signs ever go around it?"[522] By this point, Constable had taken over from Paul Nelson as the Microsoft employee tasked with handling Bangla-related issues, and Constable was dutifully working on getting up to speed. In response to Constable's query, Deepayan Sarkar

[520] Sengupta, interview.

[521] *Ibid.*

[522] Constable, Peter. "[indic] Bengali khanda ta behaviours." Email, January 29, 2004.

answered promptly ("No") and added that the Unicode FAQ was still wrong – it listed *khanda ta* as a half-form of *ta*, when it should be described as the *halant*-form.[523]

This seemingly straightforward conversation quickly became inflamed when Omi Azad jumped in. A firebrand as always, Azad was unafraid to call out Microsoft for its long-delayed *khanda ta* response — and to make a few of his own demands as well:

> *Hello Peter,*
>
> *Thanks to God that someone from Microsoft again return with Bangla Issue. Last time I was working with Paul Nelson for Bangla issues and I also gave a hand to Vrinda. Paul modified the Uniscribe engine as per our request and after that he told me that his responsibility has changed and he cannot do anything for Bangla. I was also testing the BETA of the Uniscribe engine and Vrinda font by signing an agreement with Microsoft.*
>
> *Well, we had too many demands from Unicode and Microsoft. Unicode is not listening to India and Bangladesh for Bangla issues and it's true that we don't have much proof to make our demands stronger.*
>
> *I agree with Deepayan's reply to you regarding Bangla questions you made to him. But I want to point you to something else. We want garbage typing. I mean I can type whatever I want in Bangla. I don't know why other Indic languages did not demand for this solution, but we really need this.*
>
> *KhandaTo is a form of Ta+Halanth, but we also need the Ta + Halanth + consonant = Ta with postbase form of consonant. To give the solution, Paul made Ta+Halanth+ZWNJ, but the problem went to another fault. After that if I put another consonant and anther vowel sign, the vowel sign goes before the KhandaTa. To give a temporary solution, Paul suggested to put another ZWN after it, I mean Ta+Halanth+ZWNJ+ZWJ then the KhandaTa will not eat the vowel sign. But the whole concept went garbage to me.*
>
> *Bangla is Bangla and it has nothing to do with other Indic languages. Last time Paul said that the garbage typing issue is causing problem to other languages in the uniscribe engine. But I cannot make myself understand that why that is so! Cause, it's Bangla and your engine should work as Bangla users will demand.*[524]

Azad's striking message bears close examination. After reintroducing himself to Constable (explaining that his previous work had been with Paul Nelson), Azad expressed a palpable feeling of frustration, complaining that the repeated requests of Bangla users were still not being heard because "Unicode is not listening" to its South Asian constituents. Azad wasn't wrong-- at

[523] Sarkar, Deepayan, "[indic] Re: Bengali khanda ta behaviours" Email, January 29, 2004.

[524] Azad, Muhammad Shariqul Islam, "[indic] All Bengali behaviours (not only khanda ta)" Email, January 30, 2004.

the last two UTC meetings, the *khanda ta* issue had not appeared even once in the meeting minutes, despite over a year of ongoing appeals. Perhaps predictably, Azad also insisted, yet again, “We want garbage typing.” By this, he meant something more significant than the ability to produce random keyboard-mashed letters like “fnsawlnfl.” To Azad, the freedom to “type whatever I want” included an end to the “fascism” of dotted-circle error messages that limited users by telling them that their letter combination was illegal. It also included the ability to display *any* combination of Bangla glyphs – which would enable users to show the many different ways conjuncts had been drawn across the history of the Bangla script, not just the most common conjunct conventions of the contemporary era.

Azad’s demands went beyond the realm of font design; they demanded changes to the underlying rendering engine that also had to support his desired conjunct combinations. This is why we found Azad insisting that Microsoft’s “engine should work as Bangla users will demand”: the complexity of displaying Bangla script properly was Microsoft’s problem, and, depending on Microsoft’s limitations, it might be Unicode’s problem as well.

Azad’s final point would come to symbolize much of the overarching sociopolitical stakes associated with *khanda ta*. As Azad insisted, *khanda ta* was a feature of Bangla and Bangla alone. Yet Western developers, unfamiliar with the linguistic intricacies of the language, had been resting on the comfortable assumption that all Indic languages could be treated the same way. If Microsoft’s so-called “Indic engine,” or Unicode’s original ISCII-based Indic encodings, could not support the display requirements of *khanda ta*, that risked sending the message that Bangla users’ needs were peripheral.

In short, the problem could be summed up as follows: to render properly within the existing constraints of multilingual computing stack, Bangla’s unique features had to fall in line with other Indic scripts. But as Azad insisted, “Bangla is Bangla and it has nothing to do with other Indic languages.”

Not yet posting directly to the Indic list, Gautam Sengupta weighed in on Omi’s proclamations in a personal email to him. Despite having more academic training than Omi Azad, Sengupta heartily endorsed his conclusions:

> *I agree with you completely on this issue. Unicode is indeed giving the impression, perhaps inadvertently, that the opinion of Bengalis, even those who are well aware of the issues involved in script encoding, need not be taken seriously; and getting away with it simply because most Bangla speakers either in Bangladesh or in India - even those involved in language empowerment movements - are not even aware of the existence of the Consortium or its activities.*[525]

Sengupta’s tone may have been far more diplomatic than Azad’s, but he was still accusing Unicode of “giving the impression,” however “inadvertently,” that Bangla users “need not be taken seriously,” which had not yet been directly articulated by the software hobbyists nor

[525] Sengupta, Gautam, “Re: [Fwd: [indic] Re: All Bengali behaviours (not only khanda ta)]” Email, January 31, 2004.

government officials. Whether it was with Sengupta had given his blessing is unclear, but soon after receiving his message, Azad had quickly forwarded it on to the entire Indic list.

Here Peter Constable began to take on Sengupta's accusations against him. He claimed that the problem was purely a matter on *process*:

> *That's not the case. If Bengalis have not had their opinions taken into consideration, it is only because those opinions were not communicated adequately.*[526]

This defensive response revealed ignorance towards the technological frustrations that Bengalis had repeatedly expressed. As each chapter of this dissertation has shown, individuals from various stakeholder groups had already been identifying errors and recommending changes for over two years by this point. Yet to Constable, these users' objections were still "not communicated adequately" — an unnecessarily dismissive response that set a confrontational tone for the rest of the ensuing debate.

When Ankur's Deepayan responded to Constable, he clearly registered Constable's defensiveness — and responded in kind with a sarcastic apology: "Perhaps it's our fault for not being militant enough. Maybe we don't know the procedures well enough."[527] At this point, Rick McGowan, insisting that he was writing as a long-timer observer and was "NOT SPEAKING OFFICIALLY", tried to calm everybody down by chiming in to say that "militancy" was not necessary, but that the UTC would of course like to hear from everyone.[528] According to McGowan, Unicode's leadership had been under the (false) impression that all of the problems with Bangla had already been solved.

To Constable's point, in fairness, most of the requests that had brought to Unicode had taken the form of personal blog posts and messages on public mailing lists. Even the communications coming from official Government of India channels never had much impact because they had never mustered up technical justifications. The formal proposals and technical justifications had not yet aligned and brought to Unicode's attention.

To this end, Gautam Sengupta had recently submitted an official proposal to Unicode on February 1, 2004 requesting a change to the encoding for *khanda ta*.[529] Sengupta provided the same linguistic explanation as Andy White about how *khanda ta* should be considered the default form of a *ta* with a virama, and thus should be encoded simply as *ta*+virama. Control characters like zwj and zwnj could be used for less commonly used variations. Though his Tokyo paper had noted the inconsistency of giving *anusvar* its own "atomic" codepoint in Unicode, but not *khanda ta*, he refrained from making that argument here.

[526] Constable, Peter, "[indic] Re: [Fwd: Re: All Bengali behaviours (not only khanda ta)]]" Email, February 2, 2004.

[527] Sarkar, Deepayan, "[indic] Re: ENCODING BANGLA KHANDA-TA WITH TA+VIRAMA" Email, February 3, 2004.

[528] McGowan, Rick, "[indic] Re: ENCODING BANGLA KHANDA-TA WITH TA+VIRAMA" Email, February 3, 2004.

[529] Sengupta, Gautam, "Encoding Bangla Khanda-Ta With Ta+Virama" UTC Document Registry, L2/04-060.

In response, Paul Nelson expressed frustration, arguing that he had come up with a longer four-codepoint sequence precisely because it seemed like the only way to accommodate Azad's desire for garbage type.[530] He claimed that it was unclear that Sengupta's proposal would permit all the variations of *ta* to appear in text, while still allowing developers to program correct linguistic behavior for these variations (such as indicating where vowel signs should be displayed within the rules of their rendering engine). As Nelson insisted,

> *Frankly, it is really tough to build software to support the Bengali script because opinions keep changing, or little bits of information leak out hear and little bits leak out there. One person wants to define rules for specific cases and another want to type garbage. We end up with non-Bengali speaking "experts" putting together contributions to tell the world how the Bengali script works and the Bengali speaking group complaining that the implementation is broken.*[531]

Despite Nelson's clear annoyance that "opinions keep changing" and users keep "complaining," this moment actually marked a significant progression in his perspective. His opinion had actually shifted in repeated advocacy around this issue. The easiest and most sensible thing to do now, Nelson opined, would be to encode *khanda ta* as a standalone codepoint in the Unicode Standard.[532] The letter should be recognized as an independent character because that's how it functioned in the language.

As new commenters jumped in and the discussion began to spin out of hand, Constable began reconstructing the entire timeline for *khanda ta* to establish a clear record, beginning with the Bangladeshi Standards Institute's request to ISO in 2000 to do something about the letter, and continuing with the questions about *khanda ta* on the Indic list in spring 2002 that led to Apurva Joshi's error-ridden re-write of the Indic FAQ, followed by Andy White's blog posts from a few months later chronicling said errors. Constable also included Paul Nelson's recent PRI-9 error report, which contained important information about *khanda ta* in addition to *ra* and *ja*. Finally, Constable concluded with the revived conversation about *khanda ta* that was taking place at the present moment, including Gautam Sengupta's proposal for changing the letter's encoding within Unicode.[533]

Constable expanded upon his newly assembled digital archive with caveats, such as: "UTC did not at any time make any specific decisions regarding *khanda ta* (they did on reph/ya-phalaa [the buggy *ra* and *ja* glyphs], but not this)," which implied that users should not point their fingers at Unicode for failing to handle the issue before an actual decision was made. Constable also clarified that Joshi's faulty FAQ response had only become institutionalized in Version 4 of

---

[530] Nelson, Paul, "[indic] Re: ENCODING BANGLA KHANDA-TA WITH TA+VIRAMA" Email, February 3, 2004.

[531] *Ibid.*

[532] *Ibid.*

[533] Constable, Peter, "[indic] Re: ENCODING BANGLA KHANDA-TA WITH TA+VIRAMA" Email, February 3, 2004.

the Unicode Standard manual because another UTC representative had copied it in without knowing better. That same representative was now reportedly trying to retract this contribution, admitting that he "had no knowledge of [*khanda ta*'s] behaviors and that, had he known about them, he probably would have been trying to get a separate *khanda ta* character encoded."[534]

But the damage had already been done. A number of Bangla users had started complaining on the Indic list that they felt Unicode had been arrogant and dismissive of their concerns. Gautam Sengupta empathized, writing to them:

> *You will have to learn to live with ridicule that results from the deadly combination of ignorance and arrogance for the sake of your language and script, with the hope that in the end something positive will emerge out of all this. Remember that people in Bangladesh had to bear more than just insults and ridicule - they gave their lives and even went to war for their language. I understand exactly how you feel and have been through the same experience myself.*[535]

By calling the conflict "deadly" and comparing it to "war," Sengupta was evoking the hefty language politics of the Indian subcontinent from the past century. The "ignorance and arrogance" he identifies on the part of western developers spoke to the discomfort that many observers had expressed towards Unicode's assemblage of Western technology leaders at its founding, in contrast to ISO's treaty-bound array of governments. Nor did his heated rhetoric go unheard. Given the increasingly contentious discussion that had been ramping up, the UTC decided at their next meeting in early February that they had no choice but to address *khanda ta* directly.[536] The committee tasked Peter Constable with drafting a new PRI and inviting members of the Indic list to respond. In this way, they hoped to finally achieve a solution.

If the UTC hoped to tamp down the roiling controversy, they were sorely mistaken. In the days immediately following the UTC's February meeting, the Indic list quickly became overwhelmed with comments and questions about *khanda ta*. Constable responded directly to Gautam Sengupta's proposal, insisting that the *ta+virama* combination would still cause problems for the rendering engine. This was because, he argued, one of the tasks of a rendering engine was to identify "clusters" – units such as letters or syllables that could be composed of multiple Unicode codepoints, but needed to be treated as as a single unit for operations such as word-searching, line-breaking, and caret placement. The "cluster" approach meant that the processing issues involved in rendering *khanda ta* appropriately were significant, he said. But, "*khanda ta* has

---

[534] Constable, Peter, "[indic] Re: ENCODING BANGLA KHANDA-TA WITH TA+VIRAMA" Email, February 3, 2004.

[535] Sengupta, Gautam, "[indic] Re: Khanda Ta" Email, February 3, 2004.

[536] Moore, Lisa, "UTC #98 Minutes," UTC Document Registry, L2/04-003.

exceptional behavior and requires exceptional handling no matter what."[537] It was just a matter of figuring out what form of "exceptional handling" would be the least troublesome for developers.

Sengupta, meanwhile, was feeling frustrated with the role the UTC seemed to be taking on of orthographic reformers — the position we had seen the intermediaries so delicately steer clear of in the previous chapter. He wrote,

> *The fact that it is a distinct character in the Bangla script should be reason enough. It is NOT the business of the UC to analyze scripts in order to decide whether a certain character could be dispensed with. It has neither the competence nor the mandate to do so. That is the business of professional linguists researching the concerned language or script.*[538]

It felt like the level of pushback they were receiving, on what was an age-old mistake in ISCII that should have been resolved between South Asians and with linguistic expertise, was now being adjudicated by Western non-experts — and why?

Meanwhile, Dr. Ketaki Dyson, the Baudelaire translator, started posting about the need to have both *khanda ta* and *ta* with a visible *virama* available in the same document. Previously, these two forms of ta had been understood to be graphemic "allographs" — two glyphs that represented the same character, despite their differences in graphical appearance. In the Latin alphabet, for instance, the two different ways to write the lower-case letter "a" would be considered allographs.

a *a*

But Dyson argued that *khanda ta* and *ta+virama* were only allographs in the *orthographic*, rather than graphemic sense: the two letters represented the same sound, yet by convention, were used in entirely different words. To draw another parallel to the Latin alphabet, *khanda ta* and *ta+virama* functioned much like the character combinations 'c', 'ch', 'k', 'ck' within English, all of which could be used to represent the same sound, /k/ (as in cab, key, stomach, and lock), but, because of spelling conventions, could not be used interchangeably in different words (as in kab, chey, stomack, and loc). Similarly, in Bangla, Dr. Dyson explained, users would need to use *ta+virama* for the word 'tatsuma,' a variety of orange, or for a name like 'Marina Tsvetayeva'. In such contexts, treating *khanda ta* and *ta+virama* as interchangeable would lead to misspellings and confusion.[539]

---

[537] Constable, Peter "Encoding Bangla Khanda-Ta With Ta+Virama" UTC Document Registry, L2/04-062.

[538] Sengupta, Gautam, "[indic] Re: ENCODING BANGLA KHANDA-TA WITH TA+VIRAMA", Email, February 4, 2004

[539] Dyson, Ketaki Kushari, "[indic] Re: All Bengali behaviours (not only khanda ta)" Email, February 6, 2004.

When Sengupta took the opening Dr. Dyson had created to now insist, more forcefully than ever before, that “the BEST solution, would be to encode a distinct khanda-ta,” it seemed like Constable might be swayed.[540] But even as Constable mused that he could be convinced, he warned that this approach would not be without its costs: *khanda ta* and *ta+virama* would function as two different spellings, so search engines would not return results for one when searching for the other — which was a possible bug, or was it a feature?[541] And if this solution was accepted, then implementing it would not just be a matter of Unicode incorporating it into the Standard; those attending the annual ISO meeting would need to accept it as well.[542] That meant that the earliest that a solution would be available would still be well over a year away, in Unicode 4.1, and even that goalpost meant treating the issue as a rush order. Otherwise, without such a push, *khanda ta* would not be ‘live’ as a codepoint until the next full version of Unicode came out, which would not be for another two years or more.

Given these challenges, not all subscribers of the Indic list seemed to agree that this was the best way forward. One SIL colleague commented that he did not understand why *khanda ta* was being treated as so problematic in the first place – why couldn’t the documentation for producing the letter just be made clearer?[543] Alongside this, Unicode’s Rick McGowan asked about what specific advantages the separate encoding conferred: “All I want to know, and this is a question that I'm pretty sure UTC members will ask, is: What *cannot* be done unless Khanda-Ta is encoded?”[544] Meanwhile, some commenters were not limiting themselves to the boundaries of the Bangla language at all. One linguist from the University of Washington pointed out that *cillakṣarams,* also called *chillus,* in the Malayalam script (another Indic script), behaved similarly to *khanda ta*, undermining the case for a new codepoint for *khanda t*a: “Nobody has yet suggested separate codepoints for those [Malayalam letters], and I hope nobody will.”[545] Foreshadowing what was indeed to come, another linguist responded, “I also think [these] comments have highlighted the fact that it would be rash to take action on KHANDA-TA in isolation, without considering the possible parallels in other Indic scripts. A possible thin end of a wedge of unknown dimensions?”[546] By “thin end of a wedge,” this linguist was implying that the “possible parallels” in other languages might spin out of control, leading to a barrage of new encoding requests.

In an effort to help settle the matter, the conversation turned next to historical linguistics. Several of the linguists on the mailing list begin excavating *khanda ta*’s evolution as a letter within the linguistic record, investigating whether or not it had traditionally functioned as a unique

[540] Sengupta, Gautam, “[indic] Re: All Bengali behaviours (not only khanda ta)” Email, February 7, 2004.

[541] Constable, Peter, “[indic] Re: All Bengali behaviours (not only khanda ta)” Email, February 7, 2004.

[542] *Ibid.*

[543] Kew, Jonathan, “[indic] Re: All Bengali behaviours (not only khanda ta)” Email, February 9, 2004.

[544] McGowan, Rick, “[indic] Re: All Bengali behaviours (not only khanda ta)” Email, February 7, 2004.

[545] Baums, Stefan, “[indic] Khanda ta and Malayalam cillakṣarams” Email, February 9, 2004.

[546] Kew, Jonathan, “[indic] Re: [indic] RE: [indic] Re: Khanda ta and Malayalam cillakṣarams” Email, February 9, 2004.

semantic unit. In this vein, Dyson observed that *sadhu Bangla,* or "pure" Bangla, often used words that ended with an inherent vowel sound (*a)*, but that in *cholito Bangla*, or colloquial Bangla, that final vowel was often dropped.[547] *Cholito Bangla* then had the convention of using *ta+virama* to signal that something was a *sadhu Bangla* word with a dropped vowel, instead of employing a *khanda ta.* This case added to her argument about *khanda ta* and *ta+virama* not being graphemic allographs, or only visual variations. They had different functions in words.

Other linguists remained unconvinced by Dyson's argument, however. One claimed that Bangla's *khanda ta* (much like Malayalam's *chillus*) still behaved in fundamentally similar ways to its purported allograph, *ta+virama:*

> *Looking at the modern Indian scripts in general, there are FOUR (not three) distinct ways of writing a consonant without inherent a:*
>
> 1. *consonant sign plus overt virāma,*
>
> 2. *special glyphs that behave JUST LIKE no. 1 with respect to clustering, but look different,*
>
> 3. *half–forms,*
>
> 4. *component of a conjunct.*
>
> *Going back a thousand years or so, there were only categories no. 1 and 4. Half–forms (no. 3) are just a special horizontal way of writing ligatures that developed quite recently…*
>
> *No. 2, the category that includes both Bengali khanda ta and the Malayalam cillakṣarams, is a collection of glyphs that either have their origin in a cursive combination of base character and a virāma mark (as most of the cillakṣarams) or are historically fossilised reflections of the original virāma device (as khanda ta and some of the cillakṣarams).*[548]

Taking this long view of history meant recognizing the historical commonalities between categories 1 (e.g. *ta+virama*) and 2 (e.g. *khanda ta*), which in turn implied that there was no need for separate encodings for any letters that fell into category 2, "AS LONG AS there is no character in a modern Indian script for which all four forms of vowel cancellation are used."[549]

---

[547] Dyson, Ketaki Kushari, "[indic] Re: All Bengali behaviours (not only khanda ta)" Email, February 10, 2004.

[548] Baums, Stefan, "[indic] Khanda ta, Malayalam cillakṣarams & Co." Email, February 12, 2004.

[549] *Ibid.*

Coming from a more technical angle, Ankur's Deepayan Sarkar was forced to agree. A new *khanda ta* was not necessary, he concluded, "as long as there's a way to separately encode both TA-VIRAMA and KHANDA-TA. Encoding khanda-ta as a separate codepoint is [just] one of several suggested solutions to this problem."[550] In other words, it was still possible to address user concerns over *khanda ta* without opening up the can of worms involved in adding a separate encoding.

Given the fiercely divided linguistic debate, it seemed like Peter Constable might prefer to err on the side of not adding a unique *khanda ta* encoding after all. If the only reason for encoding *khanda ta* was to appease angry Bangla users, that might cause more harm than good. So far, he concluded, "There has been no evidence presented of any text elements in attested usage that cannot be represented and adequately distinguished without encoding a new character."[551]

Pulling together two weeks of hotly debated commentary, Constable soon went on to publish PRI-30, "Encoding of Bengali Khanda ta in Unicode," in February 2004. He accompanied the report with a strongly-worded warning:

> *UTC started utilizing PRIs as a way to make it easier for a wider variety of experts to provide input to the decision-making process. So, here's your chance. If you don't provide feedback, then either that means you don't have a strong opinion, or that you are choosing not to contribute your opinion in the decision-making process.*[552]

It was now or never: this would be the last "chance" for experts like Sengupta and Dyson to make their voices heard. Giving their input would require a high degree of commitment, however. Constable's PRI was long — thirteen pages — and included background on the Bangla script; information on the historical origins of *khanda ta* (as far as they were known); details on how *ta/khanda ta* might appear when placed next to consonants; and an assessment of four different possible models for encoding *khanda ta*.[553] These four models included:

1. Unicode's current approach (taken from Apurva Joshi's flawed FAQ response)
2. Paul Nelson's four-codepoint sequence suggestion
3. Andy White's three-codepoint suggestion, since echoed by Gautam Sengupta
4. The most recent proposal to add *khanda ta* as an entirely separate character.

[550] Sarkar, Deepayan, "[indic] Re: khanda ta using ZWJ" Email, February 12, 2004.

[551] Constable, Peter, "[indic] Re: khanda ta using ZWJ" Email, February 12, 2004.

[552] Constable, Peter, "[indic] khanda ta PRI" Email, February 24, 2004.

[553] Constable, Peter, ""Encoding of Bengali Khanda Ta in Unicode," UTC Document Registry, L2/04-262.

Diligent as always, Constable walked through the workings of each proposed model in the PRI. How would *khanda ta* be triggered in each case? How would this affect how the letter *ta* appeared in a ligature? What about in a non-ligated conjunct? How would vowel modifiers be displayed? How would cluster boundaries be drawn? Could “junk sequences” (a.k.a. Omi Azad’s long-vaunted “garbage typing”) be inputted without producing errors?

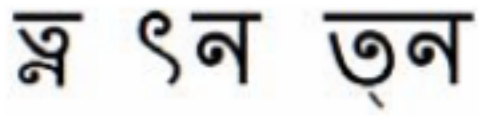


Figure 41. Three Ways Versions of Silenced *Ta* From PRI-30

Constable accompanied his discussions of the possible models with a considered analysis of the advantages and disadvantages of each scheme. Here, Constable considered factors such as: the divergence from the current Unicode encoding model, the accordance with current Indic text rendering frameworks; the difficulty level of identifying clusters; and the possible impact on end-users (such as how easy or difficult it might be to type a given sequence).

Figure 42. Text Sequences From Model C - *Khanda Ta* Rendered by *Ta+Virama+Zwj*

Pattern	Sample text sequence	Display	Note
< ta, virama, Cj, ekaar >	< ত, ◌্, ত, ে◌ >	¦ত্তে¦	
		¦েতত¦	fallback rendering
		ত্¦তে	fallback rendering
< ta, virama, Cn, ekaar >	< ত, ◌্, ক, ে◌ >	¦েতক¦	
		ত্¦কে	fallback rendering
< ta, virama, # >	< ত, ◌্, # >	ত্¦#	
		ত¦#	(reduced-size ta) other possible rendering
< ta, virama, ZWNJ, * >	< ত, ◌্, ZWNJ, * >	ত্¦*	
< ta, virama, ZWJ, * >	< ত, ◌্, ZWJ, * >	ৎ¦*	

In Constable’s final analysis, the evidence did not support a separate encoding for khanda ta:

While there appears to be particular preference for model D [a new codepoint for khanda ta] among some members of the Bengali community, it has not been shown that a new character is, in fact, required: there is no text that needs to be represented that cannot be represented without adding the new character…Unless additional technical advantages can be identified, there is not adequate justification to select model D as the preferred recommendation.[554]

Of the remaining three options, Model C (Andy White's proposal) would impose an additional burden on the end-user (who would have to manually press a control character key, zwj, on their keyboards every time they wanted to render *khanda ta*, rather than have the codepoint sequence function invisibly in the background). The other two models, A and B, seemed equally workable. Model A had the advantage of maintaining fidelity to the current Unicode Standard, making it easy to implement; all it needed was some clarifying language in Unicode's documentation. The other, Model B, would maintain the same programming approach used for other Indic scripts within the Uniscribe rendering engine, which would make it easier to explain to those responsible for implementing it, using the documentation tools in the OpenType specifications.[555]

Constable's own recommendation was to keep the current Unicode specification (Model A), but clarify within Unicode's documentation that this approach was not intended to imply that *khanda ta* should be treated as a half-form. Instead, it was intended to serve as an exceptional rule for the exceptional case of *khanda ta,* which would include adding additional clarifications as to how vowel modifiers should behave around this letter (which was the original issue raised by Andy White). The Unicode standard already included exceptions to the prototypical of *zwj/zwnj/viramas*, so there was plenty of precedent for this proposed solution.[556]

Given how well-researched and well-documented Constable's recommendation was, it might indeed have been accepted by the user community if it had been made only a few years prior. But by this point in the *khanda ta* debacle, too many parties had become aware of the Unicode Consortium's delay in addressing the problem for any of its decisions to seem neutral or reasonable to its Bangla user base. For many of these users, *khanda ta*'s encoding had come to seem like far more than a programming issue; it had turned into a test case to see whether the Unicode Consortium would ever start to listen to its users in South Asia. Omi Azad perfectly summed up this defiant stance when he wrote back to Constable after finally reading the PRI,

Peter, when we say we need it [i.e. a separate khanda ta encoding], then they [the UTC] should respect us, because we are the people who are going to use it in our way. I have doubt if UTC will use Bangla anyhow. So they should respect our decisions.[557]

---

[554] Constable, Peter, ""Encoding of Bengali Khanda Ta in Unicode," UTC Document Registry, L2/04-262.

[555] *Ibid.*

[556] *Ibid.*

[557] Azad, Omi, "[indic] Re: [Bangla] KhandaTa issues again" email, June 8, 2004.

By distinguishing between the "people who were going to use" Bangla and those who were not, Azad was making the case that user sentiments should supersede adherence to so-called principle. At the same time, however, even Azad had been forced to acknowledge, "I think no one can provide a strong approach on why KhandaTa should be encoded as a separate letter."[558] However strong the feelings of the language community may have been, the technical evidence just wasn't there.

That didn't stop the issue from picking up steam all around India and Bangladesh. Increasingly, Ankur's Sayamindu Dasgupta noted, whenever he met with anyone working in local governments or educational institutions, they would bring up the issue of *khanda ta*.[559] In Bangladesh in particular, all the local media outlets had begun covering the *khanda ta* debate, further inflaming this supposedly niche computing issue.[560] The media narrative presented the letter as "missing" from the Unicode Standard altogether, making it seem as if the Bangla language was being corrupted by Western developers. Taneem Ahmed, the leader of Ankur, recalls that "it seemed like we were becoming Indians based on that one character!"[561] Bangladeshi technologists had long expressed the sentiment that they had missed the boat when India's ISCII was incorporated into Unicode while Bangladesh then had no standard, or when India connected to high speed submarine cables for internet while Bangladesh lagged behind with a satellite connection.[562] Here was an opportunity for Bangladesh to take action and assert its identity, not fall along with whatever India had accomplished before it. As Omi Azad had once expressed, k*handa ta* was a feature of Bangla, and Bangla alone. And Bangladesh's language was Bangla, and Bangla alone — a contrast to the multilingual patchwork of India, in which Bangla was but one component.

Riled up by these incendiary media reports, many technologists in Bangladesh began to feel that their government was not doing enough to shift the needle. Why hadn't Bangladesh become a paying member of the Consortium, like India had? Why weren't Bangladeshi university professors jumping to provide their expert linguistic analysis, like Indian professors were? The complaints piled up across various personal blogging websites.[563]

Indeed, the bloggers had a point: the Government of India had quickly waded into the fray on behalf of their Bangla language users. Manoj Jain, the appointed interlocutor from the Ministry of Information Technology, forwarded a note to Unicode from a Professor Chaudhuri at the Indian Statistical Institute, which endorsed a standalone codepoint for khanda ta and added a

---

[558] Azad, Omi, "[indic] Re: PRI#30 - Bengali Khanda-ta" Email, May 20, 2004.

[559] Sayamindu Dasgupta, interview, April 17, 2020.

[560] Mahay Alam Khan, interview with author, July 8, 2019; Mamun Rashin, interview with author, March 29, 2022.

[561] Taneem Ahmed, interview with author, February 16, 2020.

[562] Progga, "[Bengalinux-core] (no subject)", email, September 21, 2002; Mahay Alam Khan, interview.

[563] Mamun Rashin, interview with author, March 29, 2022; most of these are lost to the Web but readily recalled by Bangladeshi technologists.

new line of argumentation — the usefulness of distinct codepoints for natural language processing:

> *This is about your enquiry on Bengali Khanda-ta coding in UNICODE format. Myself and my colleagues here think that Khanda-ta should be encoded as a separate character. This is because it will help in both scientific (eg NLP, Computational linguistics) and commercial applications. The typist will find it convenient to type it in a single keystroke. Moreover, by alphabet convention of Bengali script, it is treated as a separate Character.*[564]

Even TIL's Om Vikas himself wrote a letter supporting the standalone encoding, which he physically mailed to Unicode President Mark Davis. Though Vikas's letter did not present any specific technical arguments, Vikas claimed to have solicited the opinions of experts in writing it, and presented himself as the spokesperson for their consensus opinion.[565]

The "unresolved" issue of *khanda ta*, Vikas insinuated, was only still up for debate in the West; "all the experts in India" had already agreed on the "consensus opinion" of a separate encoding, implying that Unicode was behind in reflecting the status quo.

Comments from the Bengalis were not restricted to blog posts. They would make their way on to the Indic mailing list for all to see. Azad would later reflect, "We made a big fuss. We were young, easily mad. Our tone was harsh."[566] When Unicode Vice President Rick McGowan informed Azad that even if *khanda ta* was accepted as a separate codepoint, it could not possibly be included in the standard for several more years, Azad spat back:

> *Where you are living my brother? When my father was a kid then people could only think/dream about going to moon and now you know where are we. If giant vendors like Microsoft demand for that [separate khanda ta encoding], I believe we can get it done in 2 weeks. I don't know how, but it can be done.*[567]

Unicode's response was predictable: Azad was reminded, as always, that changing the Standard was not a matter of will, but instead a matter of adhering to the formalized, standardized procedures long followed by Unicode's multiple stakeholders.[568]

---

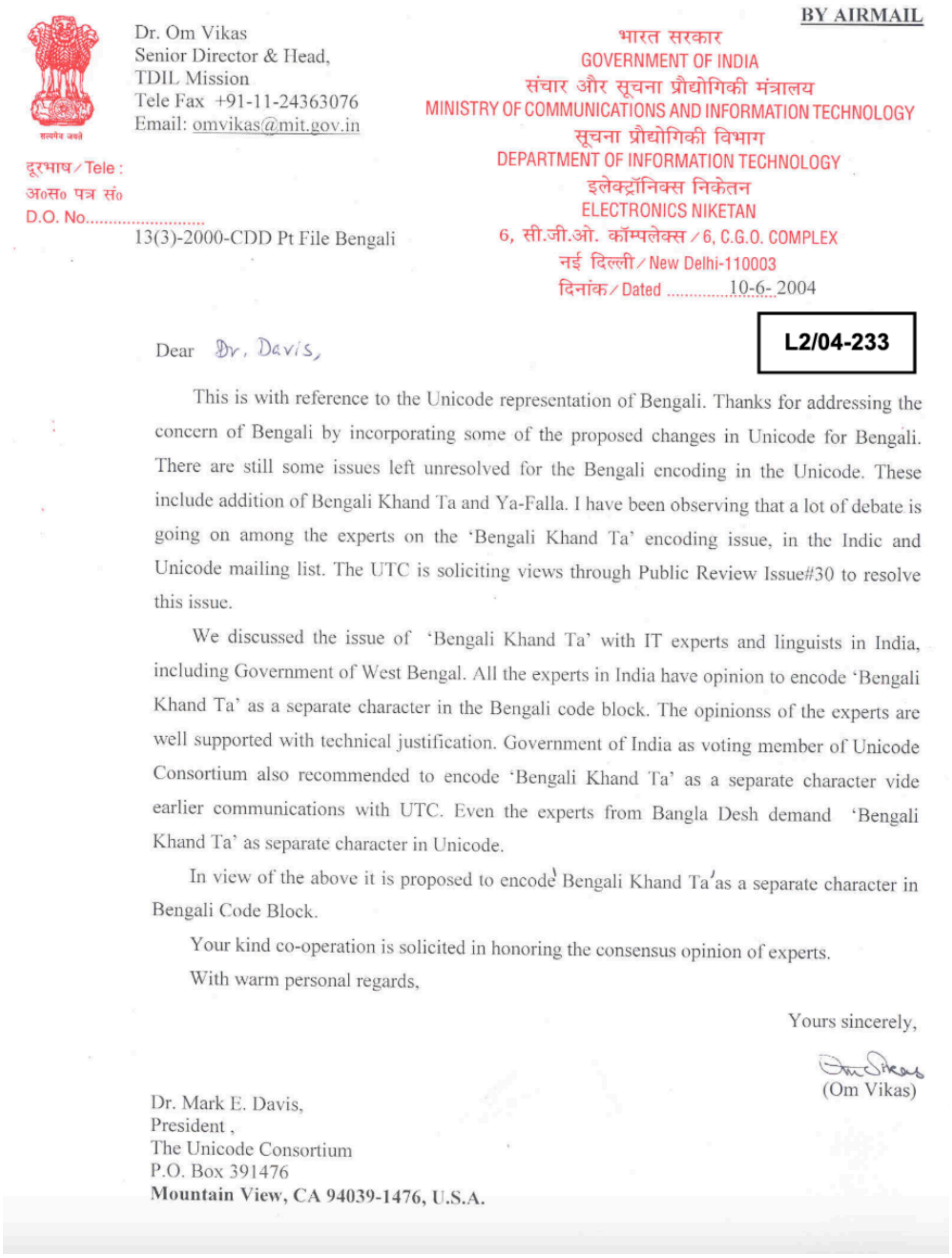
[564] Jain, Manoj, "[indic] PRI#30 - Bengali Khanda-ta" Email, May 19, 2004.

[565] Vikas, Om, "Letter to Mark Davis re Bengali Khanda Ta" UTC Document Registry, L2/04-233.

[566] Azad, interview.

[567] Azad, Omi, "[indic] Re: [Bangla] KhandaTa issues again" Email, June 11, 2004.

[568] Wissink, Cathy, "[indic] Re: [Bangla] KhandaTa issues again" Email, June 11, 2004.



BY AIRMAIL

सत्यमेव जयते

Dr. Om Vikas
Senior Director & Head,
TDIL Mission
Tele Fax +91-11-24363076
Email: omvikas@mit.gov.in

भारत सरकार
GOVERNMENT OF INDIA
संचार और सूचना प्रौद्योगिकी मंत्रालय
MINISTRY OF COMMUNICATIONS AND INFORMATION TECHNOLOGY
सूचना प्रौद्योगिकी विभाग
DEPARTMENT OF INFORMATION TECHNOLOGY
इलेक्ट्रॉनिक्स निकेतन
ELECTRONICS NIKETAN
6, सी.जी.ओ. कॉम्पलेक्स / 6, C.G.O. COMPLEX
नई दिल्ली / New Delhi-110003
दिनांक / Dated 10-6- 2004

दूरभाष / Tele :
अ०स० पत्र सं०
D.O. No. 13(3)-2000-CDD Pt File Bengali

L2/04-233

Dear Dr. Davis,

This is with reference to the Unicode representation of Bengali. Thanks for addressing the concern of Bengali by incorporating some of the proposed changes in Unicode for Bengali. There are still some issues left unresolved for the Bengali encoding in the Unicode. These include addition of Bengali Khand Ta and Ya-Falla. I have been observing that a lot of debate is going on among the experts on the 'Bengali Khand Ta' encoding issue, in the Indic and Unicode mailing list. The UTC is soliciting views through Public Review Issue#30 to resolve this issue.

We discussed the issue of 'Bengali Khand Ta' with IT experts and linguists in India, including Government of West Bengal. All the experts in India have opinion to encode 'Bengali Khand Ta' as a separate character in the Bengali code block. The opinionss of the experts are well supported with technical justification. Government of India as voting member of Unicode Consortium also recommended to encode 'Bengali Khand Ta' as a separate character vide earlier communications with UTC. Even the experts from Bangla Desh demand 'Bengali Khand Ta' as separate character in Unicode.

In view of the above it is proposed to encode 'Bengali Khand Ta' as a separate character in Bengali Code Block.

Your kind co-operation is solicited in honoring the consensus opinion of experts.

With warm personal regards,

Yours sincerely,

Om Vikas

(Om Vikas)

Dr. Mark E. Davis,
President ,
The Unicode Consortium
P.O. Box 391476
Mountain View, CA 94039-1476, U.S.A.

Figure 43. Letter of Support From Government of India

To users on the Indian subcontinent, however, nothing about Unicode's approach seemed standard or fair. Local commentators complained that Unicode's treatment of Indic scripts seemed uneven and unjust. As one Bangladeshi commented in a post, "I was watching the forum and what people are doing with Bangla. As far as my knowledge goes, I think it [i.e. the Bangla language] is [being treated as a] toy for Non-Native Bangali people to play with as they feel like. I am really disappointed with UTC & Microsoft."[569] The commentator supported his claim about Bangla being treated as a "toy" by referencing accented Latin characters that had been encoded straight into Unicode without any objection, even when those characters could have been inputted as multiple codepoint sequences instead. To be fair to Unicode, these Latin characters had only been included as standalone codepoints to ensure round-trip compatibility with previous Latin-based encoding standards (such as ASCII) that had already been in use when Unicode was released. And yet this compatibility policy itself reflected the needs and priorities of Unicode's mostly-Western user base, who may have been less willing to adopt a standard that did not incorporate the standards to which their machines already adhered.

To a non-Western observer, then, it felt as though Unicode was more than willing to make exceptions when they suited the needs of Latin-alphabet users, but when it came to South Asian scripts, suddenly nothing could be done without adhering to strict procedures. That same commentator lamented, "I have seen in Latin glyph that some of the character can be used [i.e. inputted] by using two letter but they encoded with one. My question will be why the have problem with Bangla where as they don't have problem with Latin or others to add new character."[570]

However exasperated such commentators might have felt, they doubted whether they would really be able to change the minds of the UTC. It was quite a different situation for someone like Gautam Sengupta, whose academic credentials and English-language skills made him harder for the UTC to ignore. As Omi Azad put it,, "he knew our sentiment and would explain it nicely. He wrote beautiful explanations for us."[571] For Azad, Sengupta's "beautiful explanations" mattered because neither the Bangladeshi nor the Indian government had ever made such an effort on behalf of Bangla users. Government officials might submit formal-sounding proposals, but they would never actually get involved in the forums, which is where the real debate was taking place. Similarly, government representatives would show up at international meetings, but when Unicode officials ignored their demands, they never seemed willing to push harder.

In this context, it fell to Gautam Sengupta to stand up for Bangla users, and he rose admirably to the occasion by developing a new line of argument for encoding *khanda ta* as a separate codepoint. In early June of 2004, he submitted his official response to PRI-30, while simultaneously withdrawing his previous codepoint-sequence proposal for *khanda ta* from

[569] Karim, Solaiman, "[indic] Re: [Bangla] KhandaTa issues again" Email, June 11, 2004.

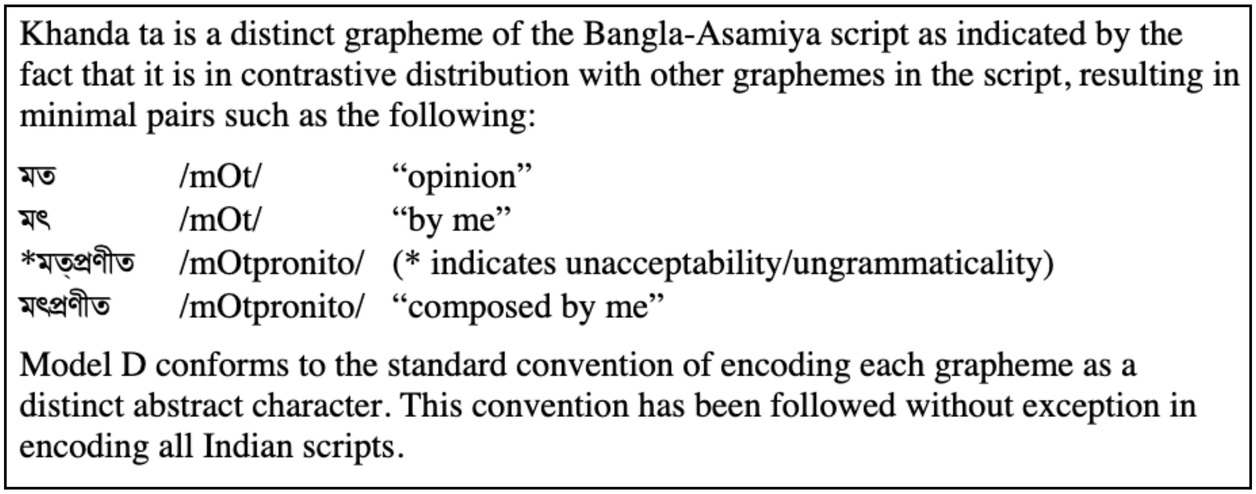
[570] Karim, Solaiman, "[indic] Re: [Bangla] KhandaTa issues again" Email, June 11, 2004.

[571] Azad, interview.

February.[572] Now, Sengupta was arguing that *khanda ta* deserved its own standalone codepoint because it needed to be independently searchable in text.[573] When a user used the Ctrl+F function to search for *khanda ta*, Sengupta insisted, the results should not return words spelled with *ta+virama* at all. This was because, as Dr. Dyson had already previously explained, the two letters did not function as equivalent spellings; some words used one, some used the other, and the distinction mattered. As Dyson had insisted, the difference between the two allographs was not graphical but orthographic, meaning it was not just a visual difference to handle within fonts.

To help illustrate this claim, Sengupta presented a list of examples of "minimal pairs," a linguistics concept referring to pairs of words with slightly different phonemes (sounds). An example of a minimal pair in English would be "let" and "lit," which illustrates the difference between two different vowel sounds. Sengupta's minimal pairs worked a little differently: he presented pairs of Bangla words that were pronounced exactly the same, but spelt differently (*ta+virama* vs. *khanda ta*), and consequently had different meanings (as illustrated below). These examples perfectly demonstrated why it was essential to be able to search for মত্ and মৎ separately within a document.[574]

Out of the four proposed encoding models for *khanda ta* that had been presented in Constable's PRI-30, however, only Models C and D would permit this kind of search behavior. And one of these, Model C Andy White's *ta+virama+zwj* sequence), came with severe disadvantages that Constable himself had already laid out (i.e. users would have to disrupt their typing flow to press a control key (zwj)). The only remaining option was Model D, the standalone encoding of *khanda ta*, which Sengupta pushed more strongly than ever before:



Khanda ta is a distinct grapheme of the Bangla-Asamiya script as indicated by the fact that it is in contrastive distribution with other graphemes in the script, resulting in minimal pairs such as the following:

মত	/mOt/	"opinion"
মৎ	/mOt/	"by me"
*মত্প্রণীত	/mOtpronito/	(* indicates unacceptability/ungrammaticality)
মৎপ্রণীত	/mOtpronito/	"composed by me"

Model D conforms to the standard convention of encoding each grapheme as a distinct abstract character. This convention has been followed without exception in encoding all Indian scripts.

Figure 44. Minimal Pairs Argument From Feedback on PRI-30

[572] Sengupta, Gautam, "FEEDBACK ON PR-30: Encoding of Bangla Khanda Ta in Unicode" UTC Document Registry, L2/04-192.

[573] *Ibid.*

[574] *Ibid.*

With this post, Sengupta cleverly turned the framing of the debate on its head by insisting that a standalone codepoint for *khanda ta* would not require a special "exception" at all, but would rather follow the same "convention" used for "all Indian scripts," in contrast to what Constable and others had claimed. By presenting Model D as a means of following Unicode's existing rules, rather than breaking them, Sengupta made the internal politicking of which technical layer should handle *khanda ta* moot; it was a "distinct grapheme" and should thus by independently encoded. In Sengupta's own words, one of the major advantages of this scheme was that "*Khanda ta* treated on par with *anusvar* and *visarga* with which it forms a natural class: all three represent dead consonants, none is able to bear a matra or other modifier and none can conjoin with a following consonant."[575] In short, Model D would make the Standard more consistent and legible to users, which fit perfectly with Unicode's proceduralist approach. Though the cultural understanding of all three letters as a natural class was elegant, Sengupta asserted that it was not the only reason to encode *khanda ta*; it existed alongside a purely technical, search-based argument:

> *Khanda ta should be encoded as a distinct character not because it is culturally perceived to be so but because it is in fact a distinct grapheme of the Bangla-Asamiya script, and because doing so would be the least expensive solution to the problems we are confronted with. The arguments presented here are technical and scientific. They have nothing to do with cultural perceptions (though the latter can NEVER be completely ignored in issues related to language and script).*[576]

Sengupta was trying to walk a fine line: even though he himself believed that "cultural perceptions" should "NEVER be completely ignored," he knew that the Unicode committee would only be persuaded by "technical and scientific" evidence. Would this prove to be the bulletproof argument they needed? Sengupta had indeed managed to articulate an important point that he, Dyson, and others had been dancing around throughout their previous messages on the Indic list. Knowing the value that the UTC placed on proceduralism, Sengupta began by providing rigorous technical documentation in favor of a single *khanda ta* codepoint. But he went on to make the case that there was value in constructing a standard that was human-meaningful, not just machine-readable. If the Unicode convention was for all "distinct graphemes" to be encoded as separate characters, he argued, then within the Bangla script, that should apply to *khanda ta* as much as it did to other "dead consonants," or consonants with the inherent vowel suppressed. In some sense, this argument wasn't new: he had begun this long journey making a very similar set of claims, as presented in his paper at the Tokyo linguistics conference in December of the previous year. What had changed in the intervening months was Sengupta's ability to appeal to the UTC's stated goals of consistency, efficiency, and proceduralism.

If Sengupta hoped that this thoughtful reply to PRI-30 would finally overcome Unicode's hesitations, he was sorely disappointed. To Sengupta's frustration, Peter Constable continued to

[575] Sengupta, Gautam, "FEEDBACK ON PR-30: Encoding of Bangla Khanda Ta in Unicode" UTC Document Registry, L2/04-192.

[576] Sengupta, Gautam, "FEEDBACK ON PR-30: Encoding of Bangla Khanda Ta in Unicode" UTC Document Registry, L2/04-192.

take issue with his claims. Firstly, Constable objected that it was not at all clear that the entire user community understood *khanda ta* to be a distinct grapheme — and if that were not the case, then that the user community would want word searches to pull up different results for *khanda ta* vs. *ta+virama*.[577] Indeed, for many years the Bangla language community had seemed to take for granted that *khanda ta* was nothing more than a visual variant of *ta+virama*. Several Bengalis had agree with that assumption. The question was, whose opinions should carry more weight: that of lay users, or that of a specialized Bangla linguist?

Secondly, though Constable found the minimal pairs demonstration interesting, he did not consider it incontrovertible proof that a unique encoding was needed. In theory, if one really wanted to search sometimes for মত্ and other times for মৎ, the search engine in question could be programmed to search not just for letters but for the associated code sequences (i.e. ta+virama+zwj vs. ta+virama). Though that admittedly convoluted search was *doable*, Constable acknowledged, it did not necessarily mean it was the best design. In short, the searching argument may be interesting and hold water, but Constable, for his part, was still evaluating.[578]

At this point, Sengupta, like Omi Azad before him, felt he was being treated with blatant disrespect by a Western outsider who did not even speak his language:

> *I am a native speaker of Bangla and a professor of linguistics and Director of a school specializing in language technology in one of the top five Indian Universities. I have spent most of my life working on Bangla linguistics and Indian language technology. If you have doubts about my qualifications and training, FYI, I studied linguistics at UMass and a list of the names of my teachers would sound like a who's who in contemporary formal linguistics. I have taught for years in three of the most distinguished universities in India. It is quite obvious that much of what you know about the Bangla-Asamiya writing system is from my ILCAA paper. You'd probably not be able to cite another paper with comparable depth and coverage on Bangla orthography. I have also spent considerable time and energy in devising software keyboards for almost all of the major Indian scripts on a single layout (http://geocities.com/indian_scripts). How many people do you know who would be better qualified to advise you, or would even bother to take the time to do so, on matters pertaining to encoding Bangla in Unicode?* [579]

As a “native speaker” and “professor… in one of the top five Indian Universities,” Sengupta insisted he deserved recognition as an authority on the Bangla language. When Sengupta lashed out at Constable for exploiting his expertise without crediting his advice — “much of what you know… is from my ILCAA paper” — he implies that Constable was taking strong positions without the expertise to back them. Stung, Sengupta presented a direct challenge: why “bother

[577] Constable, Peter, “[indic] Re: [Bangla] KhandaTa issues again” Email, June 8, 2004.

[578] *Ibid.*

[579] Sengupta, Gautam, “[indic] Re: [Bangla] KhandaTa issues again” Email, June 11, 2004.

to take the time" to provide his free academic labor if Constable was going to respond with ignorance all the same?

The thread became increasingly hostile, with Sengupta insinuating that Constable was nothing more than a corporate shill: "In the end," Sengupta insisted, "science is more important than personal and/or corporate interests."[580] Constable, realizing the time had come to dial it back, wrote, "For my part, this is not a matter of personal or corporate interests. I'm trying to discern what makes best sense, and to weigh the feedback of various contributors, which hasn't been easy due to opinions from users that don't all agree."[581]

It may have been true that users in the Bangla language community didn't "all agree" about how to use *khanda ta*. But from Sengupta's perspective, there was still something intensely puzzling and frustrating about Unicode's reluctance to add a new character to the Standard. He felt that "essentially the industry mandate was to say… Ok we have a lot of space, but don't fill it up."[582] Sengupta felt that his rigorous argument was so ironclad that it would have been embarrassing not to accept it in any kind of academic setting.[583] And even if the UTC required a different kind of proof than academia, he had gone out of his way to provide that, addressing not only the bugginess of rendering *khanda ta,* but the importance of making Unicode a more consistent and linguistically accurate standard. Why was it still so hard to get this letter in?

To Sengupta, Constable's responses may have seemed unnecessarily combative, but Constable wasn't actually trying to set himself up as Sengupta's enemy. The truth was, Constable had seen just how stubborn the UTC had been in the past about accepting so-called "cultural" arguments, and he had been approaching the conversation with the intention of helping Sengupta identify a strong enough rationale that the UTC would accept.[584] To Sengupta and other Bengalis, however, his challenges felt relentless and needlessly belligerent, making it hard for them to separate Constable's personal stance from that of Microsoft, or indeed from that of the UTC as a whole, which was perhaps the most hard-lined party of all.

After some time to cool down, Constable wrote back to Sengupta more contritely,

> *[My most recent] comment was only a reflection of the fact that it's hard to know, especially from a distance, to what extent needs expressed by one person are representative of the entire market. I personally know of no one more knowledgeable about Bengali language and script, and I know that you have done work on development of keyboard systems. But [even] a learned and technically-savvy person might not be fully informed about user requirements (I've no reason to think you are not -- this is just a general*

[580] Sengupta, Gautam, "[indic] Re: [Bangla] KhandaTa issues again" Email, June 9, 2004.

[581] Constable, Peter, "[indic] Re: [Bangla] KhandaTa issues again" Email, June 10, 2004.

[582] Sengupta, interview.

[583] Sengupta, interview.

[584] Peter Constable, interview with author, February 4, 2022.

> *statement), and so I could feel more confident saying to UTC "the sorting issue is an important concern for Bengalis" if there were more than one person saying so.*[585]

By calling Sengupta "learned and technically-savvy," Constable demonstrated that he recognized Sengupta's expertise, and by admitting that there was "no one more knowledgeable about Bengali," Constable acknowledged his own position as a newcomer. Most significantly of all, Constable insisted that he wanted to present Sengupta's argument to the UTC — but he needed the evidence that there was "more than one person" making these claims.

To this end, days before UTC's the scheduled June 15th meeting, Constable was looking for one more piece of evidence. If *khanda ta* was truly a unique letter of the Bangla alphabet, then the alphabetized sorting of Bangla dictionaries should reflect that. That is, a word spelt with *khanda ta* should not appear in the same interchangeable alphabetic order as a word spelt with *ta*. Unfortunately, Constable's own perusal of an official Bangla Academy dictionary turned up no such evidence. Though *khanda ta* appeared in a separate section than *ta* in the initial alphabetic listing, *khanda ta* and *ta* seemed to be showing up interchangeably in the actual dictionary word list. As Constable explained:

> *Seeing [khanda ta] listed within the ordering of consonants in the intro of the Bangla Academy's Bengali/English dictionary was certainly interesting, though in the same place they also list na-phalaa, ba-phalaa, ma-phalaa, la-phalaa and some other presentation forms. Even so, seeing [khanda ta] amongst the consonants raised the possibility of needing to be distinguished in sorting. I looked at this some months back to see if there was evidence for graphemic distinctiveness of khanda ta, and was actually disappointed when I discovered the dictionary did not actually sort it any differently from ta.*
>
> *I have not been unconvinced that [khanda ta] is considered a grapheme (though the only basis I have at the moment for believing it should be so considered is that you and Professor Dyson have told me it is.)*[586]

Constable had come a long way: by now, he was willing to admit that he was "not unconvinced" by Sengupta and Dyson's argument, even though he still thought it was lacking in hard evidence. Luckily, at just this moment, Ankur's Sayamindu Dasgupta — who for months had been following the thread without actively participating — chimed in with exactly the perfect citation. Attaching an image scan from the same dictionary that Constable had consulted, he drew Constable's attention to a crucial footnote he had missed.

> *In case it helps, here's what the Bangladesh Bangla Academy's "Bangla Banan Abidhan" says:*

[585] Constable, Peter, "[indic] Re: [Bangla] KhandaTa issues again" Email, June 11, 2004.

[586] Constable, Peter, "[indic] Re: [Bangla] KhandaTa issues again" Email, June 13, 2004.

> *<quote>* বর্তমান অভিধানে ৎ-কে স্বতন্ত্র বর্ণের মর্যাদা দিয়ে ত এবং ত-এর যুক্তবর্ণের পরে বিন্যাস করা হয়েছে। *</quote>*
>
> *(Bangla Academy Bangla Banan Abidhan - Jamil Chaudhury - Page 12, published by Bangla Academy, Dhaka - 1994)*
>
> *Rough translation of that would be:*
>
> *In this dictionary, khanda-ta has been treated as an unique/independent (swatantra) alphabet (barna), and it has been positioned/sorted after ta and the conjuncts of ta.*
>
> *-thanks-*
> *Sayamindu*[587]

A more perfect encapsulation of the linguistic power dynamics at play could hardly have been imagined. With his "rough translation" of this essential information that Constable hadn't been able to appreciate he was missing, Sayamindu Dasgupta, the "youngest person" of his Indic computing circles, demonstrated the natural understanding the Bengalis had of their language. In response, Constable informed him that this was very helpful, and that he would take this clarification regarding the dictionary's sorting order into account.[588] Though Constable stopped short of declaring himself convinced that *khanda ta* should be sorted separately from *ta* in search results, as Sengupta had argued, Constable did now have authoritative evidence to show the UTC in support of this claim.

Before the pivotal June UTC meeting, however, Sengupta managed to get in one last word. As much as he appreciated the dictionary citation provided by Sayamindu in the eleventh hour, he wrote, victory was still not certain. This was because, all evidence notwithstanding, the decision over *khanda ta* would come down to a test of the UTC's ability to accept the external expertise of non-Western users:

> *Recall that we have been told time and again that the UTC considers only formally submitted proposals and not mere postings on this list. Let us see if that is what happens at the upcoming UTC meeting or whether Mr Constable selectively extrapolates from the discussions here arguments that suit his end.*
>
> *Let us put our heads together and see if we can put up a more organized form of resistence to these neo-colonial, hegemonic aspiratons at national and international fora. After all, it is for us to decide what we want to buy. If we don't like the way a multinational operates, we can always boycott its products and look for alternatives even at the cost of some initial*

[587] Dasgupta, Sayamindu, "[indic] Re: [Bangla] KhandaTa issues again" Email, June 13, 2004.

[588] Constable, Peter, "[indic] Re: [Bangla] KhandaTa issues again" Email, June 13, 2004.

hardships. Let us look out for and encourage these alternatives. For monopoly always breeds the kind of arrogance we are witnessing now. The Govt of India, which is a corporate member of the Consortium, has formally presented a case for encoding Khnad-ta as a distinct character. Every native speaker on this list (without exception) has endorsed this position - including scholars and academicians technically qualified to adjudicate on such matters - and some have even presented meticulously worked out technical arguments and evidence. The extent to which UTC pays heed to this demand will be a reliable measure of its autonomy and fairness. Let us wait and see what happens at the upcoming UTC meeting.[589]

By casting doubt on the notion that even the seemingly sympathetic Constable would not "selectively extrapolate" from the thread to "suit his [own] end," Sengupta insinuated that Western technocrats could not be trusted to respect the needs of South Asians. Sengupta's bold rallying cry recalled the nationalist language wars of the post-independence era: calling for "organized resistance" to "neo-colonial" corporate exploitation in the form of a systematic "boycott". Given that "every native speaker" was united on this front, Sengupta argued, the UTC's decision would prove whether the committee could live up to its own stated values of "autonomy and fairness," or whether it would continue to be defined by "monopoly" and "arrogance" instead. Sengupta had backed the UTC into a corner: their reputations were on the line.

Over the previous, since the release of Unicode 4.0 in April 2003, a convergence of attention and interest from software hobbyists, government language planners, and academic linguists had forced the previously unchallenged Unicode Consortium to confront the limitations of their Indic encoding schemes. As the heated debate between Gautam Sengupta and Peter Constable had demonstrated, the battle over *khanda ta* had come to be a proxy war for postcolonial language politics writ large. In the digital era, would international computing standards evolve to take into consideration the perspectives of global actors? Or would Western tech experts committed to a set of principles developed in a time with smaller markets, fewer voices, and different technical constraints forever define the digital space? Could the historically embedded values of the Unicode Standard be made to shift?

These burning questions would have to wait until the in-person meeting of the UTC itself. One day after Sengupta's call for "resistance" to "multinational" domineering, the conversation was abruptly cut short by a pseudonymous moderator of the Unicode mailing list, "Sarasvati," who was named, appropriately enough, after the Hindu goddess of knowledge and wisdom. Citing the need for civility, Sarasvati imposed a unilateral block on all new messages. Not until the impending UTC meeting, where Peter Constable was slated to present the case for *khanda ta,* would conversation be allowed to resume.

This discussion appears to have become somewhat heated.
I would like to remind everyone that the comment period

[589] Sengupta, Gautam, "[indic] Re: [Bangla] KhandaTa issues again," Email, June 13, 2004.

has passed for public comment on the issue.

Let me take the opportunity to ask everyone to please review the new mail list policies, and try to keep the discussion calm.

http://www.unicode.org/policies/mail_policy.html

Sorry if this list was not informed when the policies took effect.

Regards from your,
-- Sarasvati [590]

[590] Sarasvati, "[indic] Khanda ta" Email, June 13, 2004.

Conclusion: *Khanda ta,* encoded

"The request from South Asians for a separate *khanda ta* character is not new," Peter Constable wrote. "It goes back at least three years, to the feedback that the Indian government gave on Unicode 3.0 (L2/01-304), and has been a recurring topic on email discussion lists."[591]

Constable was set to present to the Unicode Technical Committee (UTC) in a closed-door meeting in Toronto, Canada, that June 15th, 2004. The eleven-page document that he circulated in advance was intended to summarize the feedback he had received on the proposals he had analyzed for *khanda ta*, which he had written up as Public Review Issue 30 (PRI-30). He would present his case, and the UTC would now judge how to proceed on *khanda ta*'s encoding. In the initial PRI-30, Constable had recommended keeping the current Unicode encoding scheme, which used three codepoints in a sequence. They needed only to clarify the documentation and Microsoft would need to update its rendering engine.

Constable continued in his introduction, "Regrettably, most of the requests for a new *khanda ta* character have not been accompanied by a technical justification for why existing representations are inadequate and a new character is needed."

But:

> *On this occasion, though, feedback from one contributor, Gautam Sengupta (L2/04-192) did present some technical argumentation for encoding a new character. A key element of the case made for a new character is that the khanda ta is graphemically distinct from other forms of ta. This is new information that was not previously available, and has some bearing on how alternatives might be evaluated.*

Constable then listed out the evidence he had newly acquired to support the claim of "*Khanda ta* as a grapheme." Several university-level Bengali experts had affirmed that *khanda ta* was considered a distinct grapheme. He presented the dictionary entry and footnote provided by Sayamindu Dasgupta, which showed *khanda ta*'s distinct sorting from *ta*. He also presented Sengupta's example of minimal pairs. He closed by affirming that *khanda ta* and *ta-halant* were analogous to *anusvar* and *nga-halant*. *Anusvar* was the silenced version of the *nga* consonant, much like khanda ta was the silenced version of ta. "Clearly *anusvar* and *nga* have long been considered distinct graphemes." he wrote. "Given the similarities, it should come as no surprise that *khanda ta* is considered a distinct grapheme." This had been Sengupta's starting point in the paper he had presented in Tokyo in December 2003.

If *khanda ta was* a distinct grapheme, as incontestable evidence had now come to show, then was a distinct, atomic codepoint still needed to represent it? Here was the rub: "grapheme status has never been considered a sufficient condition for encoding a text element as a distinct, atomic

[591] Constable, Peter. *Review of Bengali Khanda Ta and PRI-30 Feedback*. Unicode Technical Committee Document Registry. L2/04-252.

character.” This went back to Unicode’s initial design, and the tautological definition of a “character” as an invented unit whose definition would allow all of the world’s characters to fit within a 16-bit codespace. Though, in effect, Unicode encoded all graphemes, it did not guarantee that it would assign all graphemes a codepoint. It had reserved the right to use sequences instead of atomic, or singular, codepoints.

Constable wrote, “The familiar answer to such a question in usual cases is to say that other mechanisms exist for that purpose. As has been shown, however, this is not a usual case. Alternative mechanisms have been considered yet found to have shortcomings.” *Khanda ta* would not be easily rendered, or backspaced, or searched in text, or analyzed in natural language processing, without “complex Boolean logic” in many of these cases.

“The only other obvious possibility,” he wrote, “is to encode a separate *khanda ta* character. The entire discussion thus far has constituted a case for this solution.” Though this verdict was made in plain, unemotional terms, each word carried astounding weight — affirmations of the months of fierce debate the Bangla user community had undertaken.

It seemed Sarasvati’s block on new messages to the Indic list had been lifted by June 16th. A new message by Gautam Sengupta came though:

> *I have now had the occasion to look at a document that Peter has prepared and hopefully submitted to the UTC by now. In my opinion the document is a fair representation of arguments for and against encoding khanda ta as a distinct abstract character. I therefore retract my earlier statements accusing him of harboring ill-will towards that much-maligned grapheme. Peter, please accept my unconditional apologies. Let's be friends again until the issue of Ya-phalaa comes up. :)*[592]

Sengupta was pleased with how Constable had incorporated his evidence and accepted the argument he had so long been developing. Until then, he had only Constable’s obstructions to go on, and was unaware that Constable had been preparing a document in the Bengalis’ favor.

Constable responded, “Thank you, Gautam. The apology is most willingly accepted. And the document has been submitted.”[593]

The UTC meeting took place over the next three days. The committee elected to encode *khanda ta* as a new Unicode codepoint, with the code U+09CE. Among their action items was to prepare a proposal for the upcoming ISO meeting, update the Indic FAQ page with more information, and prepare responses for the Indic list and for Om Vikas, with whom they had direct

[592] Sengupta, Gautam. “[indic] Re: [Bangla] not just khanda ta”. Email, June 16, 2004.

[593] Constable, Peter. “[indic] Re: [Bangla] not just khanda ta”. Email, June 16, 2004.

communication.[594] These steps would make official the final determination over the much-contested letter.

Constable felt that his goal had never been to arbitrarily resist the Bangla user community or deny its wishes. He simply knew that the UTC was unlikely to accept the arguments that had thus far been put forth.[595] It was in the days running up to the final June 15th meeting that he felt he had finally wrangled defensible arguments out of Sengupta and others. Ultimately, he wanted the language community to be satisfied; his intermediary position only put him in the role of a difficult coach rather than a cheerleader. This sentiment came through in the concluding paragraphs of his June 15th review document:

> *Up to now, the amount of careful analysis has not kept pace with the volume of words exchanged. At greatest risk has been that a decision would be made rejecting the Indian request without as best a case as possible having been made. This could only lead to a widening gulf of distrust between users in the Indian sub-continent and supporters of the Unicode program.*
>
> *At the very least, I hope to have shown that the case for a separate character is not completely without merit. Perhaps the analysis has revealed a case that is sufficiently convincing to grant the new character that has been requested, though I do not take that as assumed. At best, I hope to have provided a fair hearing for the Indian request such that, regardless of the outcome of a UTC decision, users in India will feel that their needs have been considered thoroughly, and that they can feel some confidence that implementations can be provided that will meet their reasonable needs.*[596]

Though he had an undeniable role in getting *khanda ta* encoded, Constable was never sure of his reception in the Bangla community. In an interview in 2021, Peter Constable wondered aloud to me, "I was always just curious, as Bengali people look back on it today… do I stand out in the storyline as this Westerner that was being insensitive?"[597]

Though Sengupta admittedly never fully overcame his frustration with Constable from the preceding months, Constable's contributions were recognized and lauded by others involved in the long debate. The lengthy *khanda ta* PRI documents were famous among Unicode staff – serving as one of the turning points when the UTC really started to take the issue seriously.[598] Omi Azad too felt in awe of Constable's work: "[Peter] wrote such a beautiful thirteen page-long

[594] "Approved Minutes of UTC 99 / L2 196," accessed June 28, 2022, https://www.unicode.org/L2/L2004/04156.htm.

[595] Peter Constable, interview by author, February 4, 2022.

[596] Constable, *Review of Bengali Khanda Ta and PRI-30 Feedback*.

[597] Constable, interview.

[598] Ken Whistler, interview by author, April 23, 2020.

document for us – beautiful explanations for why *khanda ta* was not working well. We didn't have that understanding, not like how Peter wrote it. He was the only one who did it, a Microsoft employee."[599]

Shifting Views of the Unicode Technical Committee

The UTC had accepted *khanda ta*'s encoding, but the verdict had not come easily. In an interview, Constable recalled the discussion that occurred inside the UTC meeting room. Even with the technical argumentation provided, the UTC was not overwhelmingly convinced. At that point, Peter said, "If we don't encode *khanda ta*, I don't think it will create significant problems in terms of the encoding model. But if we don't encode it, if we don't do it now, this won't be going away. How much more time do we want to spend continuing to deliberate?" It was at this point that he managed to tip the vote in favor of encoding *khanda ta*.[600]

The begrudging nature of the encoding decision came through in the message Unicode Vice President Rick McGowan eventually posted to the Indic list, announcing the new codepoint:

> *At last week's UTC meeting, the committee decided to encode Bengali Khanda Ta as a character in a future version of the Unicode Standard. The decision to encode this character was based primarily on the evidence and discussion presented in two recent documents:*
>
> *1. Peter Constable's paper "Review of Bengali Khanda Ta and PRI-30 Feedback" (L2/04-252), and*
>
> *2. Gautam Sengupta's paper "Feedback on PR-30: Encoding of Bangla Khanda Ta in Unicode" (L2/04-192).*
>
> *Both of those papers were posted to this forum earlier.*
>
> *In light of those documents, the committee was satisfied that Khanda Ta has gained enough of an independent existence in the modern writing system that it warrants encoding as a separate character. A consensus decision was then taken to encode it. A proposal summary form has been prepared by Peter Constable (L2/04-264, WG2 N2809) for submission to WG2, and it should be discussed this week.*
>
> *Sengupta wrote in his paper (cited above):*

[599] Omi Azad, interview by author, January 30, 2022.

[600] Constable, interview.

> > *"Model D conforms to the standard convention of encoding each grapheme as a distinct abstract character."*
>
> *Members of the committee wished me to specifically point out that it is *not* the case that each grapheme of a writing system is encoded, and that this is *not* a principle of the Unicode Standard. Also, the decision to encode Khanda Ta is *not* an endorsement of particular opinions or positions expressed in any documents presented to the committee.*[601]

In some ways, the UTC was hedging. They did not want to set a precedent that would send many more requests for new characters based on graphemes in their direction. They were still trying to separate the wheat from the chaff – many of these requests were, in fact, for graphical variants and not truly unique graphemes. Somehow it seemed refusing to endorse Sengupta's argument gave them more room to maneuver.

In truth, the UTC would only begin diving into the challenges with Indic script encoding after this first high-profile debacle. Up to this point, the UTC members had not been directly involved in tracing the etymology of *khanda ta*, or working through implementing the various models proposed for its encoding. An aspect of Unicode's governance that sometimes gets lost in the literature is the relatively removed stance of the UTC.

Over time, Unicode has begun to address this distance between the final decision makers and those with close expertise with the issues at hand. Beginning in 2010, Unicode created the "Script Ad Hoc Committee" – a subcommittee with the mandate to "provide recommendations to the Unicode Technical Committee (UTC) on encoding proposals and other documents, outside of CJK or emoji-related topics."[602] (Other designated subcommittees would handle CJK/East Asian scripts and emojis). The 10-15 people who were part of Script Ad Hoc committee would meet for an entire day once a month from 2010-onward and closely review active proposals for encoding. It would then prepare a summary document of recommendations for the UTC to review in its annual meetings. The Script Ad Hoc was also authorized to provide edits on existing proposals to help them better "meet the exacting technical requirements of the UTC."[603] These meetings would have voluntary and unpaid, but consistent, attendance from implementers and encoders, and would both streamline and add thoughtfulness to the encoding process for new characters.

An additional initiative to help incorporate new scripts was the creation of the Script Encoding Initiative (SEI), stewarded by Dr. Deborah Anderson of the University of California, Berkeley's Department of Linguistics. SEI was started in April 2002 on the premise that extensive research

---

[601] McGowan, Rick. "[indic] Encoding of Khanda Ta". Email, June 23, 2004.

[602] "Script Ad Hoc Group," accessed June 28, 2022, https://unicode.org/consortium/scriptadhoc.html.

[603] *Ibid.*

and resources regularly go into preparing historic and minority scripts for encoding.[604] SEI would provide funding to those willing to conduct this research – graduate students, professional linguists and academics, hobbyists, and others. The research and proposal writing would be aided by Dr. Anderson, herself a representative to the UTC, so as to meet the standards for Unicode. This had been the type of work Gautam Sengupta, and Peter Constable to some degree, had conducted for *khanda ta*; indeed, Sengupta would begin contributing occasionally to SEI in the years following.[605]

These initiatives helped institutionalize Unicode's overall receptiveness to new encoding proposals. However, a more subtle shift occurred after *khanda ta*'s encoding that would also contribute. As one commenter had noted in the previous chapter, there was a fear that the decision over *khanda ta* might trigger several more proposals for other Indic characters – "a possible thin end of a wedge of unknown dimensions?"[606] Indeed, conversation over *chillaksharams*, often called *chillus*, in the Malayalam script were picking up in the last months of the *khanda ta* discussion. Like *khanda ta*, chillus were similar idiosyncratic characters unique to the Malayalam script that denoted dead consonants, or consonants with the inherent vowel silenced. From late 2004 into 2007, ferocious debate would again take place over Unicode forums, engaging a similarly vast set of stakeholders, and eventually result in the atomic encoding of *chillus*.[607]

These and other conflicts would start to unsettle the ISCII model in the minds of UTC members. Unicode would have to appoint Tamil community liaisons through this period as well, as the movement for boycotting Unicode in favor of a novel Tamil-centric encoding gained momentum in the late 2000s.[608] Though the Tamil community would begrudgingly accept Unicode[609], the impact of these conflicts was clear: ISCII imperfectly addressed the needs of non-Devanagari communities, and it wasn't in Unicode's interest to continue upholding it. Slowly, the conservative nature of the UTC would give way to an openness towards new encodings. Thorough technical justifications were still necessary, but if multiple ways of representing a symbol were feasible, as was the case with *khanda ta*, the UTC was more amenable towards picking the one that made life easier for implementers. As one UTC member reflected in a recent interview, "if *khanda ta* came up now, there is no doubt it would be immediately encoded."[610]

---

[604] "Script Encoding Initiative," accessed June 28, 2022, https://linguistics.berkeley.edu/sei/.

[605] Gautam Sengupta, interview with author, October 20, 2021.

[606] Kew, Jonathan. "[indic] Re: [indic] RE: [indic] Re: Khanda ta and Malayalam cillakṣarams". Email, February 9, 2004.

[607] See the documents associated with Malayalam encodings U+0D7A to 0D7F listed here: https://en.wikipedia.org/wiki/Malayalam_(Unicode_block)

[608] Lisa Moore, interview with author, April 26, 2022; Muthu Nedumaran, interview with author, April 26, 2022; Whistler, interview.

[609] *Ibid*.

[610] Whistler, interview.

Many of the Unicode veterans would come to acknowledge the limitations of their previous perspectives. Ken Whistler, one of the strictest gatekeepers of the Standard, reflected,

> *[Khanda ta] was an interesting early case. One of the cases that showed the limitations of ISCII, because ISCII was a one-size fits all model. 'Pour everything into the Devanagari model, and it would be roughly correct for Brahmi-derived scripts.' But the problem was, they would each end up having edge cases and their own developments. By the time you got to South East Asia, the scripts were radically different. You can't pour them into the same model.*[611]

Their philosophy had changed in the intervening years. Now, "You take the writing system, and you analyze it on its own terms. Don't assume the Devanagari model is just going to work… The assessment is, if you're having to put in lots of joiners and non-joiners [zwjs and zwnjs], then it is just more trouble than it's worth."[612]

Constable, too, would explicitly acknowledge the failings of the ISCII-based, Unicode model. He would reflect, "I guess I went into it assuming, other people have researched this and figured out it's the right approach… We were a little naive. It's taken time to figure out. If we could do it all over, we might do these Indic scripts in significantly different ways."[613]

In his own work with Microsoft, Peter would continue working on the Uniscribe rendering engine for the nine major Indic scripts. By the time he got to the last remaining script, Oriya, the Devanagari-based shaping engine could just no longer work. The Indic engine, as it was called, had been written mostly to the specifications of Devanagari. Various hacks had been taped together whenever other exceptions for different scripts had come along, but Oriya diverged so greatly from Devanagari that it was not worth scraping together more ad hoc solutions.[614] And so, Peter would start from scratch and design a new Indic engine, called "Indic2," which would be released in 2007 in Windows Vista machines. In 2013, the Indic2 engine would be scrapped again, this time replaced by a new "Universal Shaping Engine" that handled all scripts in one system (rather than by script families, as had previously been the case), but each on their own terms.[615]

The final major factor driving changes in encoding and rendering philosophies was the addition of new members to each institution. As the viral 2015 blog post on *khanda ta* had quipped, "the composition of the Consortium's members, directors, and officers…[is] comprised largely of white men (and a few white women) whose first language was either English or another

[611] Whistler, interview.

[612] *Ibid*.

[613] Constable, interview.

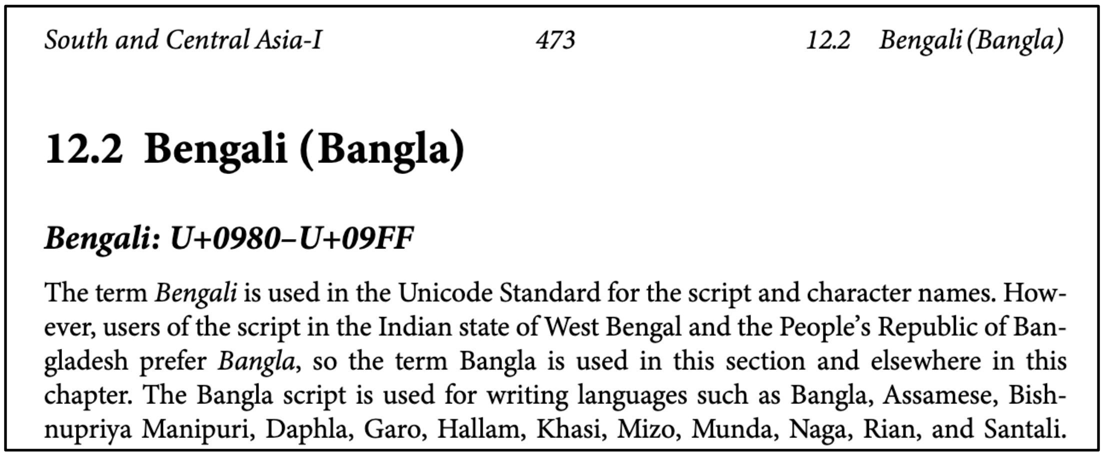
[614] Constable, interview; Andrew Glass, interview with author, October 29, 2021.

[615] The Universal Shaping Engine would be designed by Dr. Andrew Glass, an academic-turned-Microsoft program manager who had participated in Ankur/Bengalinux and Unicode discussions as a PhD student in the early 2000s.

European language."[616] Though many of these initial members did have PhD-level expertise in many languages, there were no core members specializing in complex scripts such as Arabic or Indic scripts. Several new members would slowly join the core staff and be able to say with authority how the ISCII-model was lacking, or why relying on zwjs and zwnjs introduced difficulties downstream.[617] This change in internal expertise would also help Unicode veterans shift their views on complex script encoding.

The New Language Politics

After the public drama over *khanda ta*, the Bangladeshi government would follow in India's footsteps and become a dues-paying full voting member of the Unicode Consortium.[618] The move was largely symbolic – they wanted to signal engagement much in the same way the Indian government had. Bangladesh would occasionally send representatives to UTC meetings and highlight a recurring set of issues: renaming "Bengali" as "Bangla" in the Standard, replacing existing sequenced characters with atomic ones, and adding characters to the Bangla block that were already included in the Devanagari block.[619] Though the first issue was well-received, Unicode's "absoluteness guarantee" meant that even the names associated with codepoints could not be changed without risking the collapse of downstream software. Though characters in the Unicode software library could not be renamed "***Bangla** letter khanda ta*", the supplementary documentation would be changed to say "Bengali (Bangla)."[620]


South and Central Asia-I *473* *12.2 Bengali (Bangla)*

12.2 Bengali (Bangla)

Bengali: U+0980–U+09FF

The term *Bengali* is used in the Unicode Standard for the script and character names. However, users of the script in the Indian state of West Bengal and the People's Republic of Bangladesh prefer *Bangla*, so the term Bangla is used in this section and elsewhere in this chapter. The Bangla script is used for writing languages such as Bangla, Assamese, Bishnupriya Manipuri, Daphla, Garo, Hallam, Khasi, Mizo, Munda, Naga, Rian, and Santali.

[616] "I Can Text You A Pile of Poo, But I Can't Write My Name," *Model View Culture* (blog), accessed June 28, 2022, https://modelviewculture.com/pieces/i-can-text-you-a-pile-of-poo-but-i-cant-write-my-name.

[617] Whistler, interview.

[618] Mahay Alam Khan, interview with author, July 8, 2019; Mamun Khan, interview with author, March 29, 2022.

[619] Mamun Khan, interview; Kabir, Mohammed Enamul and Md Ziauddin, "Bangladesh Proposal for UTC#159." Unicode Document Registry, L2-19-196.

[620] "The Unicode Standard, Version 14.0," 473.

Other requests from the Bangladeshi government for characters that could already easily be rendered (not required the gymnastics that *khanda ta* did) would be repeatedly denied. Though Unicode would become more generous over the years on the addition of new codepoints, they still held close to certain principles, including not introducing redundant characters if one already existed. And so, requests for Bangla punctuation that mirrored Devanagari, such as *danda*, or the Indic period mark, were denied.[621] Still, at the time of writing, Bangladesh maintains a voting membership and returns occasionally to Silicon Valley for UTC meetings to present these points.

India would also stay a voting member of the Consortium, though its appearances at UTC meetings also slowly dwindled down. Om Vikas would continue to steer the Technology Development for Indian Languages program until 2005. After that, government-sponsored work in language technology would continue, but in small occasional grants rather than a focused, mission-oriented program.[622] Engagement with Unicode from India has instead continued at the state-level, in many cases, as state governments have taken advantage of Unicode's agnoticism to the type of institution that becomes a member, and permitted states as well as national governments. In many years, then, the make up of the Unicode Consortium was a series of technology companies, the University of California, Berkeley (on behalf of the Script Encoding Initiative), and the Governments of India, Bangladesh, and Indian states such as West Bengal and Tamil Nadu.[623] These state representatives would advocate for the select issues relevant for their language communities, and then remove themselves from the Consortium.

Issues related to active Indian languages have died down in recent years. Unicode's focus with respect to Indic scripts has instead moved towards historic scripts whose digitization may help archivists and researchers.[624] Unicode has also come to recognize its place as a diplomatic player within regional language politics.[625] It handles with care the relationships between language community members and government officials, having learned from experience the high emotions that often emerge around language digitization. This careful work includes appointing liaisons from the UTC to regularly communicate with language community leaders and raising funds to permit in-person visits abroad to maintain positive relations.[626] In combination with the institutional changes to create the Script Ad Hoc Committee and Script Encoding Initiatives, Unicode's stance has changed slowly from a scarcity to abundance mindset – working carefully and extensively to encode the world's minority and historic scripts.

---

[621] Mamun Khan, interview.

[622] "Language Technology Journal of TDIL: Vishwabharat," accessed June 28, 2022, https://tdil.meity.gov.in/Publications/Vishwabharat.aspx.

[623] "Unicode Membership History," accessed June 28, 2022, https://www.unicode.org/history/contributors.html.

[624] Debbie Anderson, interview with author, January 17, 2020.

[625] Anderson, interview; Whistler, interview; Moore, interview.

[626] *Ibid.*

The Future of Bangla-Language Software

The most significant change for the Bangla computing hobbyists after the UTC decision was the need to now update their fonts and keyboards. Unicode 4.1 would be released in March 2005 and it would include the new U+09CE *khanda ta* codepoint.

> *From: "Jamil Ahmed" <jamil@bengalinux.org>*
> *To: <core@bengalinux.org>*
> *"Free Bangla Font Development" <freebangfont-devel@nongnu.org>*
> *Date: 4/6/2005 12:21:13 AM*
> *Subject: [Freebangfont-devel] Re: Unicode 4.1.0: Khanda Ta added*
>
> *We need to update our fonts to Khanda Ta compatible.*
> *`Jamil*
>
> *----- Original Message -----*
> *From: "Sharif Islam" <mislam@uiuc.edu>*
> *To: <core@bengalinux.org>*
> *Sent: Friday, April 01, 2005 12:53 AM*
> *Subject: [Ankur-core] Unicode 4.1.0: Khanda Ta added*
> *> http://www.unicode.org/versions/Unicode4.1.0/*
> *>*
> *> "U+09CE BENGALI LETTER KHANDA TA has been added. This will necessitate*
> *> adjustment of Bengali script implementations. In Unicode 4.1,*
> *> recommendations for the representation of Khanda-Ta in Bengali differ*
> *> from those documented in Version 4.0.1 and earlier."*[627]

For the tech-oriented hobbyists of *Ankur/Bengalinux* and the *Free Bangla Fonts* project, the decision over *khanda ta* held less symbolic weight than it did for other stakeholders such as linguists, media observers, and government officials. It was, still, a small victory. For many of them, it represented their influence on these burgeoning global projects – whether it be the Unicode Standard and Microsoft's OpenType specifications on the proprietary side, or Qt and Pango on the Linux side. For Deepayan Sarkar, one of Ankur's founding members, this was the greatest impact the group had over its years of existence: the contributions they made in helping others understand what was needed to digitize Bangla.[628]

Ankur would continue its work on Bangla computing over the next few years. Aside from tracking the Unicode debates, its members were working on building Live CDs through 2004.

[627] Ahmed, Jamil. "[Freebangfont-devel] Re: Unicode 4.1.0: Khanda Ta added." Email, April 6, 2005.

[628] Deepayan Sarkar, interview with author, June 16, 2021.

These were fully Bangla open source operating systems that users could easily slide into their computers. They would hand them out for free at internet fairs in India and Bangladesh, to much fanfare.[629] Dhaka had its first Internet Fair in April 2004, where Ankur's stall was prominent enough to draw in many new members to the virtual community.[630] From this point on, the priorities of the group would begin to shift. With much of the multilingual stack in place by this point – fonts, keyboards, standards – the group would begin to focus on translating interfaces.[631] This meant spending hours and hours translating strings before different software releases. They began working on Mozilla and Open Office localization. The interests of the new members were different from the early founders: many of them expressed deep appreciation for the Bangla language and for the craft of translation. Their dedication was potent and surprising to the earlier cohort of software engineers.[632]

For Taneem Ahmed, the group's fearless leader, his involvement naturally died out. His presence on the mailing lists would steadily lessen, sometimes leading to frantic goose chases as the other Ankur members searched for him to give them write-permissions to upload new translations before the next launch deadline.[633] As the new translators began to take greater ownership over the organization, Taneem appeared on the message boards to write one last message in 2006. He had been noticing that there were more frequent flame wars happening on the Ankur mailing lists, and that they were mostly occurring between Indian and Bangladeshi members – sometimes due to different preferences for translations, sometimes due to the different opportunities available to members on either side of the border, sometimes for other reasons.[634] Bangla was spoken and written slightly differently between the two countries, and while it hadn't mattered for lower-level tooling, the differences mattered now that the focus was on translations. India also had a richer technology scene, where open source translators were getting employed for their work by companies such as Red Hat. These same opportunities did not exist for Bangladeshis.

Though it was not his preferred path forward, Taneem proposed a split of the community into Ankur India and Ankur Bangladesh; the groups would cooperate when possible, but would otherwise coordinate their own activities, including producing separate translations for India and Bangladesh under the "BN_in" and "BN_bd" tags, respectively.[635] Though a few lurkers would post in resistance to Taneem's message, the damage was done, and the split would silently occur in the coming months. In an interview, Taneem would reflect, "Right from the beginning, we

[629] Taneem Ahmed, interview with author, August 19, 2020; Jamil Ahmed, interview with author, September 2, 2020; Mahay Alam Khan, interview; *Bengalinux-core* email archives.

[630] *Ibid*.

[631] *Ibid.*

[632] *Ibid.*

[633] Dasgupta, Sayamindu. "Re: [ooo-bn-trans] Re: [Ankur-core] Update on Taneem Bhai." Email, May 23, 2005.

[634] Ahmed, Taneem. "[Ankur-core] What Ankur means to me." Email, May 26, 2005.

[635] Ahmed, Taneem. "[Ankur-core] Proposal for the future of Ankur Foundation." Email, June 26, 2006.

knew something would happen. I'm sure you know, for whatever reason, between West Bengal and Bangladesh, things don't always work out."[636]

In the ensuing years, Ankur India and Ankur Bangladesh would follow different paths to acquire funding. Ankur India would be led by Runa Bhattacharjee and Sankarshan Mukhopadhyay, both of whom were employed by Red Hat to translate Linux distributions.[637] They would later become mentors under Google's Summer of Code program and bring in interns to do the same work. Sayamindu would get a fellowship to work on transforming proprietary Indic fonts into Unicode-compliant ones.[638] Soon after, he would leave the subcontinent to begin study in Human-Computer Interaction at the Massachusetts Institute of Technology.

Ankur Bangladesh would come to be led by some of the newer members to the original group, Jamil Ahmed and Mahay Alam Khan (who went by his initials, Mak). Jamil would meet a well-connected localizer, Javier Sola, at an open source conference in 2006, who would help Ankur Bangladesh acquire funding from his home government of Spain.[639] Sola had previously won a grant from the Spanish government to work on Khmer localization in Cambodia, and was willing to use his expertise to help Ankur.[640]

Ankur Bangladesh would need to first become a formal organization, however. It became incorporated as an NGO and set up a physical office in Dhaka.[641] The work sprawled in different directions over the three years of the grant: font development fell behind as it became hard to hire those with the right expertise; translations would rise and fall in activity, as new student recruits would join and then leave; Ankur Bangladesh would seek to build relationships with government staffers, but here too, leads would frequently dry up.[642] The problem soon became retaining talent. Jamil, one of the last active members, had gotten a visa to study in Canada. Mak, the only person working full-time for Ankur, lived across town from the office in Puran (Old) Dhaka. The Ankur office was rented in the neighborhood of Uttara, where much of the new development in Dhaka was happening. With Dhaka's infamous traffic, "it was basically a world away."[643] Mak would spend equal times commuting to the Ankur office as he did working there.

In both India and Bangladesh, by 2010, the groups would be dissolved.

---

[636] Taneem Ahmed, interview.

[637] Runa Bhattacharjee, interview with author, September 2, 2020.

[638] "FLOSS Fellowship Programme | s a r a i," accessed June 28, 2022, https://sarai.net/floss-fellowship/; Guide hosted at: https://unmad.in/conv_guide/

[639] Jamil Ahmed, interview; Javier Sola, interview with author, September 1, 2020.

[640] Sola, interview.

[641] Jamil Ahmed, interview; Sola, interview; Mahay Alam Khan, interview.

[642] *Ibid.*

[643] Sola, interview.

For the earliest members, the eventual fallout of the group was unfortunate but inevitable. Sayamindu would later express doubt over the decision for the two sides to incorporate themselves: "Ankur was never a formal organization. Just a thing over the internet, right?"[644]

Deepayan would write in a 2020 memoir,

> *Eventually, the online community of Ankur fizzled out. Partly this was because people moved on with their lives, and partly because we had accomplished what we had started out to do, which was to help enable Bengali on the web and on Free Software platforms; today, working with Bengali is as easy on GNU/Linux systems, if not easier, than it is on Windows or Mac OS.*[645]

At the same time, he wrote, "My only regret is that I never physically met anyone in the group other than Sayamindu, and never got to know them closely enough to know what motivated them to get involved in the first place."[646]

For Taneem Ahmed, Ankur had always been about the people involved. His goal was to help people connect around common goals. When he had arrived at the University of Toronto many years ago, he had noticed there was no Bangladeshi Students Association (BSA). So he had created one, and began an annual barbecue tradition that continues to this day. He would say, For International students, it helps them feel that they're not alone. For those who grew up here [in North America], you get to talk to people of a similar mindset, but with different experiences." Drawing the connection, he would say, "BSA was also just a meeting place. Just like Ankur. You come here, you find other people, together you try to make something."[647]

It is hard to quantify the direct impact of groups like Ankur, or others like Indic-computing (which faced a similar slowdown into the 2010s). The localized applications and desktops still see new download counts on SourceForge.[648] The Free Bangla Fonts were incorporated into other tools, such as the popular open source Avro keyboard for Bangla.[649] And the Unicode Standard has their fingerprints visible in footnotes and forum archives. But for the most part, they were creating tools at a time when there weren't many in languages other than English. As one Ankur member reflected, "Something changed in the mid-2000s. People started typing more in their

[644] Dasgupta, interview.

[645] "Bangla Computing and I," accessed June 28, 2022, https://deepayan.github.io/misc/bangla-computing.

[646] *Ibid.*

[647] Taneem Ahmed, interview with author, February 16, 2020.

[648] "Bengali on Linux," SourceForge, accessed June 28, 2022, https://sourceforge.net/projects/bengalinux/.

[649] Hasan, Mehdi. "[Freebangfont-devel] FREE Bangla Software Avro Keyboard - NewVersion." Email, September 18, 2003.

scripts. Why? Things started to come built-in. It seemed like one day the stack of tools just became available. You could just press some buttons and do it. You didn't have to search through search engines for hacks that only worked in half-formed ways."[650]

As Microsoft and others had first set about to do in 2000, slowly these major software companies had come to support the world's languages natively on their software. Today several Bangla keyboards come loaded onto my laptop. I can press a few buttons and the keyboard instantly changes. I can type Bangla text and it just works. I know that behind the scenes, however, my computer has a rendering engine that is looking through Bangla OpenType font files and matching to the Unicode Standard to make this work. But it is no longer a delicately-assembled puzzle for the user, but an internalized process for the machine. Whether this state of affairs is the product of capitalism, as companies chased more and more paying users, or benevolence, or the demands of users, one cannot definitively say. What I hope this dissertation shows, though, is that it was likely due to the combined efforts of many dedicated individuals and institutions, enacting old struggles in new digital systems to make it possible to write "চমৎকার!"

[650] Omi Azad, interview with author, January 30, 2022.

Bibliography

Archives

Bengalinux/Ankur mailing list (Bengalinux-core, Ankur-core)
Free Bangla Fonts Project mailing list (freebangfonts-devel)
Indic-Computing Project mailing list (indic-computing-users, indic-computing-standards)
Unicode Mailing list
Unicode Indic Mailing list (indic)
Unicode Technical Committee Document Registry (UTC Document Registry)

Primary Sources

"Altruists International." Accessed July 1, 2022. http://web.archive.org/web/20150206053738/http://www.altruists.org/about/.

"Approved Minutes of UTC 99 / L2 196." Accessed June 28, 2022. https://www.unicode.org/L2/L2004/04156.htm.

"Bangladesh Information Project," March 2, 2015. http://web.archive.org/web/20150302034508/http://www.altruists.org/projects/eo/bi/.

Becker, Joseph D. "Multilingual Word Processing." *Scientific American* 251, no. 1 (1984): 96–107.

Becker, Joseph D. "Unicode 88," August 29, 1988. https://unicode.org/history/unicode88.pdf.

SourceForge. "Bengali on Linux." Accessed June 28, 2022. https://sourceforge.net/projects/bengalinux/.

"Bengali Template Font." Accessed July 1, 2022. http://web.archive.org/web/20061230031531/http://www.stat.wisc.edu/~deepayan/Bengali/FreeBangTemplate/readme.html.

Bhaskararao, edited by Peri. "International Symposium on Indic Scripts: Past and Future: Organized by Research Institute for Languages and Cultures of Asia and Africa Tokyo University of Foreign Studies Tokyo, December 17-19, 2003: Working Papers." 東京外国語大学附属図書館ＯＰＡＣ. Accessed June 30, 2022. https://www-lib.tufs.ac.jp/opac/recordID/catalog.bib/BB13653529.

"Bug 113551 – Bugs in the Bengali Rendering System of Pango." Accessed July 1, 2022. https://bugzilla.gnome.org/show_bug.cgi?id=113551.

Constable, Peter. "Review of Bengali Khanda Ta and PRI-30 Feedback," n.d., 11.

"Development of ISCII and INSCRIPT Keyboarding - Dr. R. M. K. Sinha." Accessed June 29, 2022. https://sites.google.com/site/profrmksinha/research-projects/development-of-iscii-and-inscript-keyboarding.

IGNCA. "Dr. Om Vikas - Biography." Accessed June 28, 2022. http://ignca.gov.in/PDF_data/Dr_om_vikas_faculty.pdf.

“Early History of ASCII?” Accessed June 29, 2022. https://groups.google.com/g/alt.folklore.computers/c/gbg5YVFaT48/m/wlVFfJ2j4hYJ.

“FAQ - Indic Scripts and Languages.” Accessed June 29, 2022. http://www.unicode.org/faq/indic.html.

“FLOSS Fellowship Programme | s a r a i.” Accessed June 28, 2022. https://sarai.net/floss-fellowship/.

Hudson, John. “Making Fonts for the Universal Shaping Engine.” Presented at the TYPO Labs, May 10, 2016. http://tiro.com/John/Universal_Shaping_Engine_TYPOLabs.pdf.

“IndLinuxSaga - IndLinux.” Accessed June 29, 2022. https://www.indlinux.org/wiki/index.php/IndLinuxSaga.

“IS 13194 (1991): Indian Script Code for Information Interchange - ISCII,” n.d., 42.

Kaplan, Michael S. “Script and Font Support in Windows,” October 31, 2007. http://archives.miloush.net/michkap/archive/2007/10/31/5800258.html.

Karmali, Naazneen. “Microsoft’s Passage to India.” Forbes. Accessed June 29, 2022. https://www.forbes.com/global/1998/0727/0108030a.html.

Kinross, Robin. “The Digital Wave.” *Eye Magazine*, 1992. https://www.eyemagazine.com/feature/article/the-digital-wave.

Knuth, Donald E. *Digital Typography*. Reissue edition. Stanford, Calif: Center for the Study of Language and Inf, 1998.

“Language Technology Journal of TDIL: Vishwabharat.” Accessed June 28, 2022. https://tdil.meity.gov.in/Publications/Vishwabharat.aspx.

Mahesh K. Sinha, R. “A Journey from Indian Scripts Processing to Indian Language Processing.” *IEEE Annals of the History of Computing* 31, no. 1 (January 2009): 8–31. https://doi.org/10.1109/MAHC.2009.1.

Mann, Steve. “Existential Technology: Wearable Computing Is Not the Real Issue!” *Leonardo* 36, no. 1 (February 2003): 19–25. https://doi.org/10.1162/002409403321152239.

“Minute on Education (1835) by Thomas Babington Macaulay.” Accessed June 30, 2022. http://www.columbia.edu/itc/mealac/pritchett/00generallinks/macaulay/txt_minute_education_1835.html.

Official Google Blog. “Moving to Unicode 5.1.” Accessed June 29, 2022. https://googleblog.blogspot.com/2008/05/moving-to-unicode-51.html.

Mudur, S. P., Niranjan Nayak, Shrinath Shanbhag, and R. K. Joshi. “An Architecture for the Shaping of Indic Texts.” *Computers & Graphics* 23, no. 1 (February 1, 1999): 7–24. https://doi.org/10.1016/S0097-8493(98)00113-7.

Mukerjee, Aditya. “I Can Text You A Pile of Poo, But I Can’t Write My Name.” *Model View Culture* (blog), March 17, 2015. https://modelviewculture.com/pieces/i-can-text-you-a-pile-of-poo-but-i-cant-write-my-name.

Nelson, Paul. “Bengali Script: Formation of the Reph and Use of the ZERO WIDTH JOINER and ZERO WIDTH NON-JOINER,” n.d., 1.

———. “Bengali Script: Formation of the Reph and Yaphala, and Use of the ZERO WIDTH JOINER and ZERO WIDTH NON-JOINER,” n.d., 6.

Nunberg, Geoffrey. “Will the Internet Always Speak English?” The American Prospect, December 19, 2001. https://prospect.org/api/content/3e35e7bd-ce0d-57fe-bdc0-8327087966a9/.

Pal, Palash B. “Bangtex,” 2001. http://www.saha.ac.in/theory/palashbaran.pal/bangtex/bangtex.html.

Raymond, Eric. “Halloween Document 8.” Accessed June 29, 2022. http://www.catb.org/~esr/halloween/.

———. “How To Become A Hacker.” Accessed June 29, 2022. http://www.catb.org/esr/faqs/hacker-howto.html.

———. “The Cathedral and the Bazaar.” Accessed June 29, 2022. http://www.catb.org/~esr/writings/cathedral-bazaar/.

———. “The Magic Cauldron.” Accessed June 29, 2022. http://www.catb.org/~esr/writings/cathedral-bazaar/magic-cauldron/.

Sarkar, Deepayan. “Bangla Computing and I,” 2020. https://deepayan.github.io/misc/bangla-computing.

“Script Ad Hoc Group.” Accessed June 28, 2022. https://unicode.org/consortium/scriptadhoc.html.

“Script Encoding Initiative.” Accessed June 28, 2022. https://linguistics.berkeley.edu/sei/.

Searle, Steven J. “Brief History of Character Codes in North America, Europe, and East Asia.” TRON Web, 1999. http://tronweb.super-nova.co.jp/characcodehist.html#anchor953122.

Specter, Michael. “Computer Speak;World, Wide, Web: 3 English Words.” *The New York Times*, April 14, 1996, sec. Week in Review. https://www.nytimes.com/1996/04/14/weekinreview/computer-speak-world-wide-web-3-english-words.html.

Stallman, Richard, and Richard M. Stallman. *Free Software, Free Society: Selected Essays*. Edited by Joshua Gay. 1st. ed. Boston, Mass: Free Software Foundation, 2002.

“The Story of Ekushey Project - Ekushey.” Accessed July 1, 2022. http://ekushey.org/?page/Story_of_Ekushey_Project.

“The Unicode Standard, Version 14.0,” n.d., 1048.

“Unicode Glossary.” Accessed June 28, 2022. https://unicode.org/glossary/.

“Unicode Membership History.” Accessed June 28, 2022. https://www.unicode.org/history/contributors.html.

U.S. Department of State. “U.S.-India Partnership: Kanpur Indo-American Program and Beyond.” Accessed June 28, 2022. //2009-2017.state.gov/p/sca/rls/rmks/2010/144465.htm.

Vikas, Dr Om. “Language Technology Development in India,” 2001, 23.

“What Is Free Software? - GNU Project - Free Software Foundation.” Accessed June 29, 2022. https://www.gnu.org/philosophy/free-sw.en.html.

Secondary Sources

Acharya, Poromesh. “Development of Modern Language Text-Books and the Social Context in 19th Century Bengal.” *Economic and Political Weekly* 21, no. 17 (1986): 745–51.

Alam, Mahbubul. “Bangladesh Computer Council - Banglapedia.” Accessed June 29, 2022. https://en.banglapedia.org/index.php?title=Bangladesh_Computer_Council.

“Bengali Language | Britannica.” Accessed June 28, 2022. https://www.britannica.com/topic/Bengali-language.

Benkler, Yochai. “Coase’s Penguin, or, Linux and ‘The Nature of the Firm.’” *The Yale Law Journal* 112, no. 3 (2002): 369–446. https://doi.org/10.2307/1562247.

Blommaert, Jan. “Language Planning as a Discourse on Language and Society: The Linguistic Ideology of a Scholarly Tradition.” *LANGUAGE PROBLEMS & LANGUAGE PLANNING* 20, no. 3 (1996): 199–222.

Bratton, Benjamin H. *The Stack: On Software and Sovereignty*. 1st edition. Cambridge, Massachusetts: The MIT Press, 2016.

Chan, Anita. “Coding Free Software, Coding Free States: Free Software Legislation and the Politics of Code in Peru.” *Anthropological Quarterly* 77, no. 3 (2004): 531–45.

Chowdhury, Masud Hasan, and Md Mahbub Murshed. “Computer - Banglapedia.” Accessed June 29, 2022. https://en.banglapedia.org/index.php/Computer.

Coleman, E. Gabriella. *Coding Freedom: The Ethics and Aesthetics of Hacking*. Princeton: Princeton University Press, 2012.

Cook, Susan E. “New Technologies and Language Change: Toward an Anthropology of Linguistic Frontiers.” *Annual Review of Anthropology* 33, no. 1 (2004): 103–15. https://doi.org/10.1146/annurev.anthro.33.070203.143921.

Crystal, David. *Language and the Internet*. Cambridge: Cambridge University Press, 2001. https://doi.org/10.1017/CBO9781139164771.

Daniels, Peter T. “Indic Scripts: History, Typology, Study.” In *Handbook of Literacy in Akshara Orthography*, edited by R. Malatesha Joshi and Catherine McBride, 11–42. Literacy Studies. Cham: Springer International Publishing, 2019. https://doi.org/10.1007/978-3-030-05977-4_2.

DeNardis, Laura. *The Global War for Internet Governance*. Yale University Press, 2014. https://www.jstor.org/stable/j.ctt5vkz4n.

Dodd, Robin. *From Gutenberg to OpenType: An Illustrated History of Type from the Earliest Letterforms to the Latest Digital Fonts*. Illustrated edition. Vancouver: Hartley and Marks Publishers, 2006.

Driscoll, Kevin. *The Modem World: A Prehistory of Social Media*. New Haven: Yale University Press, 2022.

“Early Technologies of Digital Type.” Accessed June 29, 2022. http://www.designhistory.org/Digital_Revolution_pages/EarlyDigType.html.

Eastman, Carol M. *Language Planning, an Introduction*. Chandler & Sharp, 1983.

"Empire, Mughal - Document - Gale In Context: World History." Accessed June 30, 2022. https://go.gale.com/ps/i.do?p=WHIC&u=seat24826&id=GALE|CX3447600139&v=2.1&it=r&asid=6b597320.

Grieco, Joseph M. "Between Dependency and Autonomy: India's Experience with the International Computer Industry." *International Organization* 36, no. 3 (1982): 609–32.

Guha, Ramachandra. *India After Gandhi: The History of the World's Largest Democracy*. Reprint edition. New York/N.Y: Ecco, 2008.

Haigh, Thomas, Andrew L. Russell, and William H. Dutton. "Histories of the Internet: Introducing a Special Issue of Information & Culture." *Information & Culture: A Journal of History* 50, no. 2 (2015): 143–59. https://doi.org/10.1353/lac.2015.0006.

Hardie, Andrew. "From Legacy Encodings to Unicode: The Graphical and Logical Principles in the Scripts of South Asia." *Language Resources and Evaluation* 41, no. 1 (2007): 1–25.

Hossain, Anushah. "Remembering East Pakistan." *The Bengal Gazette* (blog), July 31, 2020. https://bengalgazette.org/2020/07/31/remembering-east-pakistan/.

Huq, Mohammad Daniul. "Sadhu Bhasa - Banglapedia." Accessed June 30, 2022. https://en.banglapedia.org/index.php/Sadhu_Bhasa.

Translation Commons. "Indigenous Languages Zero to Digital.Pdf." Accessed June 29, 2022. https://drive.google.com/file/d/1zpZK3jfF3bDt2e5YnEw8FXSkYXSRefKu/view?usp=embed_facebook.

Innis, Harold A. *The Bias of Communication*. Toronto: University of Toronto Press, Scholarly Publishing Division, 1999.

"Internet History of 1990s | Internet History | Computer History Museum." Accessed June 29, 2022. https://www.computerhistory.org/internethistory/1990s/.

Irani, Lilly. *Chasing Innovation Making Entrepreneurial Citizens in Modern India*, 2019. https://escholarship.org/uc/item/3239b1qv.

Jha, Mithilesh Kumar. *Language Politics and Public Sphere in North India: Making of the Maithili Movement*, n.d.

John, Nicholas A. "The Construction of the Multilingual Internet: Unicode, Hebrew, and Globalization." *Journal of Computer-Mediated Communication* 18, no. 3 (April 1, 2013): 321–38. https://doi.org/10.1111/jcc4.12015.

Jordan, D.K. "Languages Left behind: Keeping Taiwanese off the World Wide Web." *Language Problems & Language Planning* 26, no. 2 (August 1, 2002): 111–27. https://doi.org/10.1075/lplp.26.2.02jor.

Joshi, Pratik, Christain Barnes, Sebastin Santy, Simran Khanuja, Sanket Shah, Anirudh Srinivasan, Satwik Bhattamishra, Sunayana Sitaram, Monojit Choudhury, and Kalika Bali. "Unsung Challenges of Building and Deploying Language Technologies for Low Resource Language Communities." arXiv, December 7, 2019. http://arxiv.org/abs/1912.03457.

Karim, Mohammad Ershadul. *Cyber Law in Bangladesh*. Kluwer Law International B.V., 2020.

Kelty, Christopher M. *Two Bits: The Cultural Significance of Free Software*. Illustrated edition. Durham: Duke University Press Books, 2008.

Khan, M. Siddiq. "The Early History of Bengali Printing." *The Library Quarterly: Information, Community, Policy* 32, no. 1 (1962): 51–61.

King, Robert D. *Nehru and the Language Politics of India*, n.d.

Kornai, András. "Digital Language Death." *PLOS ONE* 8, no. 10 (October 22, 2013): e77056. https://doi.org/10.1371/journal.pone.0077056.

Kunde, Brian. "A Brief History of Word Processing (Through 1986)." Accessed June 29, 2022. https://web.stanford.edu/~bkunde/fb-press/articles/wdprhist.html?fbclid=IwAR2FIc7YfyzQjGVvIxajMUxyWYhpCN3voQOq2AIofkrbPjM4ww4vVNdlZkY#16.

Kurzon, Dennis. "Romanisation of Bengali and Other Indian Scripts." *Journal of the Royal Asiatic Society of Great Britain & Ireland* 20, no. 1 (January 2010): 61–74. https://doi.org/10.1017/S1356186309990319.

obo. "Language Policy and Planning." Accessed June 30, 2022. https://www.oxfordbibliographies.com/view/document/obo-9780199772810/obo-9780199772810-0273.xml.

obo. "Language Standardization." Accessed June 28, 2022. https://www.oxfordbibliographies.com/view/document/obo-9780199772810/obo-9780199772810-0250.xml.

Lise M. Dobrin. "SIL International and the Disciplinary Culture of Linguistics: Introduction." *Language* 85, no. 3 (2009): 618–19. https://doi.org/10.1353/lan.0.0132.

Littauer, Richard. "Open Source Code and Low Resource Languages." Saarland University, 2018. https://raw.githubusercontent.com/RichardLitt/thesis/master/thesis.pdf.

Loomis, Steven, Anshuman Pandey, and Isabelle A Zaugg. "Full Stack Language Enablement." Steven R. Loomis, June 6, 2017. https://srl295.github.io/2017/06/06/full-stack-enablement/index.html.

Mahoney, Michael S. "The History of Computing in the History of Technology." *Annals of the History of Computing* 10, no. 2 (April 1988): 113–25. https://doi.org/10.1109/MAHC.1988.10011.

Mazzarella, William. "Beautiful Balloon: The Digital Divide and the Charisma of New Media in India." *American Ethnologist* 37, no. 4 (2010): 783–804.

McEnery, Anthony, and Zhonghua Xiao. "Chapter 4 Character Encoding in Corpus Construction," n.d., 11.

"Memoirs of the Revolution in Bengal, Anno Dom. 1757: | Library of Congress." Accessed June 30, 2022. https://www.loc.gov/item/94840377.

Menon, Nikhil. "'Fancy Calculating Machine': Computers and Planning in Independent India." *Modern Asian Studies* 52, no. 2 (March 2018): 421–57. https://doi.org/10.1017/S0026749X16000135.

Montaut, Annie. “Colonial Language Classification, Post-Colonial Language Movements, and the Grassroot Multilingualism Ethos in India.” In *Living Together Separately: Cultural India in History and Politics*, 2005.

Mori, K., and T. Kawada. “From Kana to Kanji: Word Processing in Japan.” *IEEE Spectrum* 27, no. 8 (August 1990): 46–48. https://doi.org/10.1109/6.58434.

Mudur, S. P., Niranjan Nayak, Shrinath Shanbhag, and R. K. Joshi. “An Architecture for the Shaping of Indic Texts.” *Computers & Graphics* 23, no. 1 (February 1, 1999): 7–24. https://doi.org/10.1016/S0097-8493(98)00113-7.

Nagel, Emily van der. “From Usernames to Profiles: The Development of Pseudonymity in Internet Communication.” *Internet Histories* 1, no. 4 (September 2, 2017): 312–31. https://doi.org/10.1080/24701475.2017.1389548.

Nekvapil, Jiří. “From Language Planning to Language Management.” *Sociolinguistica Jahrbuch (2006)* 20, no. 2007 (February 20, 2007): 92–104. https://doi.org/10.1515/9783484604841.92.

Olocco, Riccardo. “Linotype Bengali and the Digital Bengali Typefaces,” n.d., 118.

Paolillo, John C., and Daniel Pimienta. “Measuring Linguistic Diversity on the Internet.” *Undefined*, 2005. https://www.semanticscholar.org/paper/Measuring-linguistic-diversity-on-the-internet-Paolillo-Pimienta/38ee472199bfe795066a3f5fb5472f9d16104de0.

Paolillo, John C. “Language, the Internet and Access: Do We Recognize All the Issues?,” 2010. http://www.efnil.org/documents/conference-publications/thessaloniki-2010/language-languages-and-new-technologies/08-John-C-Paolillo.pdf.

Park, Dongoh. “The Korean Character Code: A National Controversy, 1987–1995.” *IEEE Annals of the History of Computing* 38, no. 2 (2016): 40–53. https://doi.org/10.1353/ahc.2016.0021.

Perold, Colette. “IBM’s World Citizens: Valentim Bouças and the Politics of IT Expansion in Authoritarian Brazil.” *IEEE Annals of the History of Computing* 42, no. 3 (July 2020): 38–52. https://doi.org/10.1109/MAHC.2020.3010892.

Raghavan, Pallavi. “The Making of South Asia’s Minorities: A Diplomatic History, 1947- 1952.” *Economic and Political Weekly*, May 21, 2016.

Rahman, Tariq. *Language and Politics in Pakistan*. Karachi: Oxford University Press, 1997.

Ross, Fiona. “The Evolution of the Printed Bengali Character from 1778 to 1978.” University of London, 1988.

Schumacher, E. F. *Small Is Beautiful: Economics as If People Mattered*. Reprint edition. New York, N.Y: Harper Perennial, 2010.

Sebba, Mark. *Spelling and Society: The Culture and Politics of Orthography around the World*. Cambridge: Cambridge University Press, 2007. https://doi.org/10.1017/CBO9780511486739.

Siddiqi, Asif A. “Technology in the South Asian Imaginary.” *History and Technology* 31, no. 4 (October 2, 2015): 341–49. https://doi.org/10.1080/07341512.2016.1142632.

Singh, Vaibhav. “The Machine in the Colony: Technology, Politics, and the Typography of Devanagari in the Early Years of Mechanization.” *Philological Encounters* 3, no. 4 (November 27, 2018): 469–95. https://doi.org/10.1163/24519197-12340051.

Smith, Alvy Ray. *A Biography of the Pixel*. Cambridge, Massachusetts: The MIT Press, 2021.

Star, Susan Leigh. “The Ethnography of Infrastructure.” *American Behavioral Scientist* 43, no. 3 (November 1, 1999): 377–91. https://doi.org/10.1177/00027649921955326.

Star, Susan Leigh, and Karen Ruhleder. “Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces.” *Information Systems Research* 7, no. 1 (March 1996): 111–34. https://doi.org/10.1287/isre.7.1.111.

Sterne, Jonathan. *MP3: The Meaning of a Format*. 0 edition. Durham: Duke University Press Books, 2012.

Subramanian, Ramesh. “India and Information Technology: A Historical & Critical Perspective.” *Journal of Global Information Technology Management* 9, no. 4 (October 1, 2006): 28–46. https://doi.org/10.1080/1097198X.2006.10856431.

Takhteyev, Yuri. *Coding Places: Software Practice in a South American City*. Cambridge, Mass: The MIT Press, 2012.

Thompson, Hanne-Ruth. *Bengali: A Comprehensive Grammar*. Taylor & Francis, 2010. https://www.google.com/books/edition/Bengali_A_Comprehensive_Grammar/4GoHEAAAQBAJ?hl=en&gbpv=0.

Tsu, Jing. *Kingdom of Characters: The Language Revolution That Made China Modern*. Riverhead Books, 2022.

Vaish, Viniti. *Globalization of Language and Culture in Asia: The Impact of Globalization Processes on Language*. A&C Black, 2010.

Weber, Steven. *The Success of Open Source*. Cambridge, Mass.: Harvard University Press, 2005.

The Indian Express. “Why Assamese Script Wants Its Own Slot, and What It Has Got Instead,” June 28, 2018. https://indianexpress.com/article/explained/why-assamese-script-wants-its-own-slot-and-what-it-has-got-instead-5236249/.

Winner, Langdon. “Do Artifacts Have Politics?” *Daedalus* 109, no. 1 (1980): 121–36.

ETHW. “Word Processing for the Japanese Language,” January 9, 2015. https://ethw.org/Word_Processing_for_the_Japanese_Language.

Zaugg, Isabelle A., Anushah Hossain, and Brendan Molloy. “Digitally-Disadvantaged Languages.” *Internet Policy Review* 11, no. 2 (April 11, 2022). https://policyreview.info/glossary/digitally-disadvantaged-languages.