Keyboard layouts: Lessons from the Meꞌphaa and Sochiapam Chinantec designs

Hugh Paterson III

SIL International and the University of North Dakota

i@hp3.me

15 December 2020

Version: Post-Print

Preface

In contrast to the publication of the original chapter, this post-print includes the references which were cited in the text, directly after the main text. Several style sheet changes have also been made: The main font has been changed, several small spelling corrections have been fixed, some diacritics are shown with ◌ rather than appearing bare, a table of 'languages mentioned' in the chapter, a list of abbreviations used, and in some places titles and names have been italicized as is commonly done in some publishing styles. The in-text citations for items mentioned as "in same volume" as the original chapter have now been added as full refernces. The book was first released in 2014, but the printed date/copyright date, as is common in many publishing venues, indicates the year following—in this case 2015. Even though author affiliation changed since publication, affiliation has been left as it was at the time of original authorship.

Original publication

Paterson III, Hugh J. 2015. Keyboard layouts: Lessons from the Meꞌphaa and Sochiapam Chinantec designs. In Mari C. Jones (ed.), *Endangered Languages and New Technologies*, 49–66. Cambridge, UK: Cambridge University Press. https://doi.org/10.1017/CBO9781107279063.006.

1 Introduction

Codification represents a major challenge for writers of endangered languages. New technologies render the process of typing on a keyboard more accessible and less expensive than at any previous point in time. In the twenty-first century, widely used writing systems depend on electronic input methods for producing printed or electronic materials. This chapter explores keyboard layout design considerations as they were addressed in the creation of two keyboard layouts for the Latin script-based writing systems serving four languages in the Meꞌphaa language family[1] and Sochiapam Chinantec [cso]. In designing the typing experience for endangered language

[1] The four Meꞌphaa languages use a pan-lectal writing system that can be accommodated via a single keyboard layout.

writers, it was necessary to account for: (a) technical differences encountered across major computer operating systems (OS X and Windows); (b) computing culture issues such as the keyboard layout of the dominant language; (c) keystroke frequency of language specific segments; and (d) Unicode compatibility and input issues related to composite characters. The creation and use of a Unicode keyboard for data input facilitated the involvement of speakers of Meꞌphaa during the data-collection stage of a language documentation project by allowing for Unicode-encoded text documents to be generated by the speakers.

Early adaption of digital input methods may prove to better meet the needs of both the speech community and researchers. By giving the speech community a keyboard for its orthography, speakers were given the opportunity to enter into, and use, their language in new technological media and the language domains associated with communicating in those media.

2 Context

It is increasingly common for endangered language speech communities to take an active role in the documentation, preservation and development of their language (see, among other chapters in this volume: Hugo 2015, and Bel and Gasquet-Cyrus 2015). Members of these communities are now increasingly working within academia, which allows them to contribute their knowledge, experience and worldviews to new social circles. The global levelling of information access though the Internet also enables speakers of endangered languages and academics to engage more fully with each other – rather than, as before, operating in different social circles. Roles such as 'linguist', 'language documenter' or 'endangered language speaker', which might previously have been mutually exclusive, can therefore now be fulfilled by 'academics' and 'native speakers' alike. In designing technology to work with languages, especially keyboard layouts, it is therefore necessary to bear in mind the variety of backgrounds that can be represented.

2.1 The language documentation context

My initial involvement in the keyboard layout design for Meꞌphaa was in order to facilitate text creation and the typing as part of the NEH-funded project Documenting the Meꞌphaa Genus (Marlett 2010 NEH-DEL: FN-50079–10). My own responsibility was for Macintosh OS X,[2] and my colleague Kevin Cline worked with Windows-based operating systems. An existing keyboard layout was already in use by several Meꞌphaa writers, including some bilingual teachers in the Meꞌphaa-speaking region. Since some of these writers were also going to be involved in the text collection and creation process for the language documentation project, it was decided to use the existing keyboard layout as a starting point. In this way the documentation project would maximize continuity of experience.

The pre-existing keyboard layout and custom (non-Unicode) font were created by Mark L. Weathers and a team of Meꞌphaa speakers who have been involved in a longstanding language development project. It was decided that Unicode compliance was necessary for documents created

[2]The version of OS X used during this project and references throughout this chapter is OS X 10.6.8. At the time of writing, OS X is at version 10.9.

as part of the NEH-funded project. This meant designing keyboard layouts that would produce the expected input and also map the glyphs to their correct Unicode code points. This functionality was needed across several operating systems including: *Windows XP*, *Windows 7* and *Mac OS X*. To create these keyboards, the following software was used: *Ukelele* version 2.1.9 (Brownie 2012) and *Microsoft Keyboard Layout Creator* (MSKLC) version 1.4 (Microsoft 2012).

The second keyboard layout discussed in this chapter was designed for Sochiapam Chinantec. A *Keyman* (Durdin and Durdin 2011) keyboard already existed and was in use by several people involved in a community literacy programme. However, as this keyboard would only function on Windows-based operating systems, I was asked by Wilfrido Flores, a native Chinantec speaker, writer, and one of the programme facilitators, to make it possible to type Chinantec on Macintosh computers too. I therefore created a functional copy of the Keyman layout but using OS X-based technologies.

2.2 The digital context

With the arrival of new technological media in the personal communications arena, it is important for linguists and language documenters to consider the effects of new modes of communication on the societies of endangered and minority language users (Eisenlohr 2004:21). Digital technologies sit increasingly between conversational interlocutors. Sometimes, new technologies enable speakers to bypass previous obstacles to distance communication. The mobile phone, for instance, enables speakers of endangered languages to bypass methods such as orthographical representation. With the advent of the smartphone and tablet, we see video conferencing, where reactions to both the oral channel and the visual channel are considered in communication. Audio-visual communication has increased in popularity since the first decade of the new millennium. This marks a change from writing or typing, which, in former decades, represented the primary means of encoding language. Notwithstanding these great digital advances or their social acceptance rates, oral and oral-visual communication is not always fitting, nor is it always sufficient.

Personal communication via digital technologies is no longer novel. The pace of cultural technologization is increasing (Holton 2011:373, 393–4). However, financial constraints often prevent the writers and media producers of endangered languages from acquiring a personal computer (although computers are commonly seen in remote mountain villages of Mexico). Smaller devices generally cost less, and are therefore acquired more easily by endangered language community members. Many speakers of endangered languages use local computer centres, (smart)phones, netbooks and tablets (see Scott Warren and Jennings 2015 and De Graaf et al. 2015, this volume). These devices all require some sort of keyboard layout, whether soft or hard. Language planners therefore need to consider the use of endangered languages in the digital medium.

Text-based communication continues to be a relevant need in the endangered language context for text messaging, e-mail, web-surfing, letter writing, certificate printing and a range of other communicative functions. This has prompted language planners (Diki-Kidiri 2011:231 [France];

Zhozhikov et al. 2011:251 [Russia]); educators (Galla 2009); Silva and Donaghy 2004 [Hawai'i]); governments (Bailey 2007 [South Africa]), and speakers (Bernard 1992 [Mexico]) alike to acknowledge the need for keyboarding solutions. Speech communities or institutions often call upon organizations such as SIL International (McLendon 2011:98–9) or companies such as Tavultesoft (2013) to help create technological solutions.

Encoding a language via text is not only needed by writers of endangered languages but is also of interest to language documenters. Best practice in language documentation calls for language-use events to be captured in oral and visual modes as primary data (Bird and Simons 2003:574). Best practice also calls for the documentation and analysis of primary data to include written elements such as transcriptions, annotations and translations (Himmelmann 1998:162–3; Seifart 2006:286). It is often linguists who are first aware of the need to keyboard endangered languages and who rise to the occasion (Harvey 2013). When keyboard layouts are designed and distributed by linguists, they are not always centrally or transparently available to communities, nor are they always designed with intent for use beyond the immediate project. While the issue of textual encoding is of great concern to both the language documenter and the speech community, the challenge to the endangered language writer/typist is often not how to read a given orthography but rather how to produce literature in that orthography.

Challenges to the keyboard designer and typist include sociolinguistic pressures, user experience issues, and technological limitations. Sociolinguistic challenges for the typist include the typical domains in which languages are used – the acceptance and usage of (digital) written literature within the community and balancing design for a particular endangered language with keyboard design needs for a multilingual environment. User experience challenges for the keyboard designer include visual stimulus and feedback for the user, key positioning based on frequency of character occurrence and the psychological ordering of keystrokes in order to produce the intended characters. Technological challenges to both typists and designers often centre around the underlying encoding processes and the ability of an Operating System (OS) to natively process input from a keyboard.

Discussion of keyboard layout design is often missing from the literature on language documentation technology (Eisenlohr 2004), although it is not completely absent (Holton 2011:372). Much more technology-related ink is spilt discussing archiving platforms and formats, Internet usage and software tools for analysis. The challenges arising from orthographies which are difficult to type are more often acknowledged in language documentation literature which deals with orthography design (Csató and Nathan 2007; Guérin 2008:57; Jany 2010; Lüpke 2011:333–4; Seifart 2006:285–6). Treatments of the challenges faced by writers of endangered languages generally focus on the development of orthographies, including their social appeal and readability. The literature offers relatively little in terms of guiding principles for designers of keyboard layouts. This is not to say that nothing has been written on implementing minority and endangered language keyboarding solutions (Hosken 2001), only that the subject has not been well treated with regard to current technologies in discussions about language documentation and orthography design. The absence is not completely unexpected since human-computer interaction such as keyboarding is often treated and discussed as a sub-discipline of computer science or psychology (Krishna et al. 2005) rather than of linguistics.

Within the field of human-computer interaction, the last decade has witnessed several advances with respect to the keyboard. For many years, the keyboard was considered a device with a fixed arrangement. However, with the advent of touch-screen technology, it is no longer bound to a fixed position in terms of key location within a layout, characters displayed on the surface of the keys and characters visually returned to the user as output. Since there is no hard keyboard in these devices, there is no limitation to the design of layouts with graphemes that are specific to the majority language (Hinkle et al. 2010:191). However, even with such flexibility, issues of key location on the keyboard layout are yet to be solved for endangered language orthographies and their users. One challenge presented by devices with virtual keyboards is that the vendor must support keyboard layouts or app developers must create keyboards on a 'per app' basis. Again, this has hitherto been the concern of the software designer and product manufacturer rather than that of the linguist. However, it has always been a central concern for the speech community.

The need for an appropriate keyboard layout supporting a given orthography comes partially uninvited to the writer of an endangered language. Both the need and existing solutions come at the whims of the larger global society and are heavily impacted by manufacturers of keyboards and communicative devices. Therefore, the challenge of how users access characters on a keyboard is not so much a linguistic challenge (like semantic or syntactic analysis) as a language-use challenge. No one has told the endangered speech community that they must use digital technology: it is something that they draw from the majority culture around them. Linguists are called upon to solve the problem because they are often the ones representing technical expertise and are trusted by the endangered speech community. Language documenters attempt to solve the problem because they want the language to be documented and used. More broadly, the challenge lies in the hands of human interaction and interface designers. Delivering the solution lies in the hands of those manufacturing and marketing digital devices. Nevertheless, some practical guidance is beneficial for those new to designing keyboard layouts.

3 Good design

When language documenters and linguists build digital solutions such as keyboard layouts, they need to bear in mind that these products may have lasting effects on communities. As service providers, they have ethical and professional obligations to seek out not only solutions but also great solutions. In the manufacturing industry, manufacturers are often held accountable for the effects of their products on the users of their products. When linguistic and technical expertise is offered to communities of endangered language speakers and writers, we need to not only design solutions, we need also to offer well-designed solutions. Just because something is usable and useful does not mean it is desirable. When a speech community does not want to use a given input method (keyboard layout), the response should not be: ‘Well, they simply don’t want it enough.’ Keyboard layouts are not just products, they are experiences. Each keystroke in its place is a pattern created in an attempt to implement the orthography. It creates an experience that writers’ fingers will potentially encounter multiple times a day. This physical interaction is part of the user’s experience and should not be overlooked in the design process. Other parts of the user’s experience deal with the keyboard layout as software, so the keyboard layout should be considered and designed as software as well as an experience.

Consider the ambiguity of the term usability. In one sense it means no more than ‘Is a tool usable?’ However, just because a tool can be used as a hammer, this does not mean that every tool should be shaped like a hammer. Nor does it mean that every tool should be used as a hammer. Just because a keyboard layout can be used does not mean that it has a good layout. The term design in computing also suffers a similar fate. If a computer tool does something, it does so because it was designed to do so. Software is not generated by accident. The mere fact of its existence does not mean that a given computer tool is aesthetically pleasing nor that it creates a sexy or desirable impression upon its user (Anderson 2006, 2009, 2011a, 2011b). As designers, we strive to create software that both entices the user to come back to the software again and again and also meets their functional needs. It might even be said that we hope to create a symbiotic and addictive relationship between the user and the software.

The renowned industrial designer Dieter Rams proposed ten principles of good design (Vitsœ 2012). Due to considerations of space, the following sections only relate keyboard layout considerations to the first four of his principles. Good design, first, makes a product useful; second, makes a product understandable; third, is unobtrusive; and fourth, is thorough to the last detail.

3.1 Useful design

Rams suggests that a product is not useful if it does not also meet certain aesthetic, functional and psychological criteria. Aesthetics of keyboard layouts are physically dictated by the keyboard or by the combination of device and software on touch-screen tablets. These functional and psychological criteria are in the purview of keyboard layout designers (linguists and language documenters).

3.1.1 Special characters

Some of the functional criteria are obvious. The keyboard layout must be able to implement the orthography of the target language and as far as possible the orthography of the majority language. For indigenous languages such as Meꞌphaa and Chinantec, in the Mexican context, this would mean being able to also type Spanish. It is important to notice the directionality of composition: typing a document in Spanish and adding a few words or sentences in Meꞌphaa is drastically different from typing a Meꞌphaa document and adding a few words or sentences in Spanish. Even if the writing systems are ‘similar’ in that they both use Latin scripts and try and show social affinity by ‘looking similar’, when composing the document there is still a difference in terms of the user experience. For instance, two semantically equivalent texts were analysed for the purpose of evaluating the efficiency of the Meꞌphaa keyboard layout.[3] The Meꞌphaa glyph ⟨ á ⟩ is used 880 times, whereas the same content written in Spanish uses the same glyph only 59 times. A keyboard that accounts for the input of a complex glyph should also account for the frequency that that glyph is accessed. To input 59 ⟨ á ⟩ glyphs in Spanish on the Spanish ISO keyboard requires 118 keystrokes. Alternatively, on a standard OS X ANSI US QWERTY keyboard, 177 keystrokes are needed to form the same 59 ⟨ á ⟩ glyphs. Using the Meꞌphaa layout we created it still only takes 118 keystrokes to produce the 59 ⟨ á ⟩ glyphs in Spanish. However, writing the same content in the Meꞌphaa language requires 880 ⟨ á ⟩ glyphs (1,760 keystrokes). At this point, the layout designer needs to ask whether the Spanish ISO layout is an efficient option for typing Meꞌphaa.

In terms of psychological criteria, designers need to consider how much work is required to produce each character and how this impacts upon an individual's desire to type in a given language. Psychological factors also include user experience and the process of typing complex characters such as characters with diacritics representing tone and stress. The placement of a frequently typed character must be considered for the keyboard layout. For example, in the Meꞌphaa text, the LATIN SMALL LETTER SALTILLO U+A78C ⟨ ꞌ ⟩, which indicates a glottal stop, is used 1,189 times. This accounts for almost 8 per cent of all characters in the text and is the second most common non-complex character. This character on the Meꞌphaa layout is at one of the furthest places on the keyboard for the little finger to reach.[4] This distance can have an effect on a typist's speed and rhythm. The Saltillo is also a character that does not form part of the Mexican-Spanish orthography.[5] All these factors make it more compelling to type in Spanish than in Meꞌphaa.

3.1.2 Diacritics and dead keys

Access to diacritics has a significant bearing on both Meꞌphaa and Chinantec layouts. Both layouts use dead keys[6] to assign diacritic marks to base characters. Designers need to consider the order in which their keyboard layouts will require users to access diacritics. Should keyboard layout users hit the tone mark (or diacritic) key first or should they hit the base character first and then the tone mark key? In keyboard layout design terminology, the method of typing a diacritic first and then the base is known as the 'dead key method', whereas the method of typing the base character first and then the diacritic is known as the 'operator method'. Both may be valid ways to consider input, but user experience needs to be considered.

Hosken (2001:§5.2) points out that using the dead key method does not provide the user with any visual feedback, whereas the operator key method provides a visual change for every

[3] The texts were both translations of the New Testament Epistle of James. The Spanish word count was 2,165; the Meꞌphaa word count was 2,856 for the same set of verses.

[4] On a physical ANSI keyboard, this character lies a little further away than it does on a physical ISO keyboard.

[5] The Saltillo also does not appear graphically on the physical keyboard. This is not a major challenge to implement but it does mean that the user has to learn to strike a key that does not return an input corresponding to the image on the key top.

[6] Dead keys are often used for diacritics that occur with base characters. The dead key is a key that does not return an input to the text document, but waits for a second key to be struck and then returns some modified combination of the two keys.

keystroke. OS X addresses this by rendering the diacritic without a base prior to the base being struck. Additionally, if the diacritic is one that can become part of a pre-composed character, it is backgrounded in yellow (see Figure 1). However, since the advent of Windows 7, the diacritic is not displayed before the base is struck.

Figure 1 First stroke of a two-keystroke character, highlighting the diacritic in OSX

3.2 Understandable design

In his design principles, Rams states that ‘good design makes a product understandable. [Good design] clarifies the product’s structure. Better still, it can make the product talk. At best, it is self-explanatory’ (Vitsœ 2012). The product needs to be intuitive. Cross-platform implementations and keystroke ordering therefore need to be considered.

It used to be the norm that most endangered language writers had little to no exposure to computers. With globalization and the digitization of communication, this is no longer the case. The mismatch between what is printed on the keys of a physical keyboard and what the virtual keyboard layout returns as output to the screen causes confusion for those who are new to typing. An additional consideration is the effect of visual feedback to the user; this concerns both what the user sees on the screen and what they are observing on the keyboard. The pedagogy of typing usually advocates hiding the keys from the eyes in order to increase one’s typing speed. However, the universal benefit of this principle has been questioned (Byers et al. 2004). Writers of endangered languages have at least two reasons for needing to see the correct characters on the keys: first and foremost, this is how typists learn what to hit in order to achieve the desired results; second, it is the only way to complete the visual spectrum feedback loop. A user knows that if they strike a given button it means they will get a given result to display on the screen. The goal is to set the user’s expectations and then to meet them. Visual confirmation forms a vital part of this. One practical and inexpensive approach is to make a custom plastic overlay. This can be done for under $10 USD.

Figure 2 Visual feedback loop

It is important to note that visual feedback can be confusing to typists (see Figure 2). The Me'phaa text produced a situation where the typist intended to use guillemets, but actually used 'greater-than' and 'less-than' glyphs instead. In other words, the visual presence on the keyboard stimulated a character-key association that did not produce the intended character. Although an important point to bear in mind, this kind of error is notable by its absence from the literature on typing errors (Kano 2008; Kano and Read 2009:294). Errors of this kind might, however, be classified as a 'hardware' mistake as classified by Kano et al. (2007) and Read et al. (2001).

If the guillemet error were to prove common among typists in a given speech community, designers could make the angle bracket key output a guillemet when hit twice consecutively. However, in this particular instance, this method is divergent from the Spanish ISO keyboard, which is the most commonly available layout in Mexico. What do Me'phaa typists have to do, then, when writing a Me'phaa text on a Spanish keyboard layout? When using a Spanish keyboard layout, the characters needed for typing Me'phaa are simply not available, and even if Me'phaa typists are aware of the differences that exist between the two layouts, they may struggle when switching between these layouts. Education about the writing system and its implementation is key, as this will assist the multilingual typist using a particular layout to better understand the differences between two languages, their orthographies, and the capabilities of a given keyboard solution. Education is of course necessarily bi-directional: designers need to understand the user's orthographic environment and be able to receive feedback from the speech community in order to consider possible alternatives.

3.2.1 Cross-platform design

Layouts are more intuitive when they behave the same way across multiple operating systems. A typist should not need to relearn how to type on each new device. Cross-platform design provides continuity to users when they switch computers or operating systems and maximizes opportunities for social, peer-based learning. Both continuity (the status of previously understood analogies in graphical user interface design) and learnability are important factors that affect the adoption of software. Adoption of software is essential to the success of language revitalization when language use is in the digital medium.

In implementing the Me'phaa keyboard layout the following question was asked: 'Could the vendor key be used as a dead key?' The hope was to avoid dedicating a key in the grapheme production area of the keyboard to the sole purpose of becoming a dead key. The result was: 'No, the vendor key could not be used.' The design motivation was to respect and comply with device- and platform-oriented user interface guidelines. Many applications running on OS X use the vendor key, also known as the command (⌘) key, as an application-level shortcut key. In Windows and Linux, the control key is used in place of the vendor key. In Windows, the vendor key is used to access the Windows menu. Therefore, using the vendor key as a modifier key becomes problematic, as it changes the way the machine behaves overall rather than simply modifying the orthographic characters available to the typist. As designers of keyboarding experiences, our designs need to fit intuitively into the computing experience on each platform.

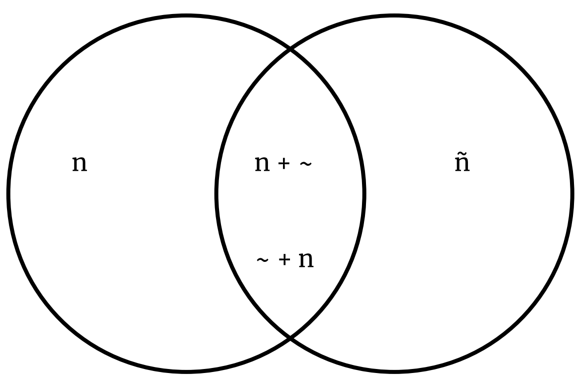
3.2.2 Conceptualization and symmetry of characters

A keyboard must also explain itself via the cognitive associations it suggests to its users. Should all characters be accessed the same way? Not all orthographies use the same characters to represent the same sounds or ideas. Many endangered and minority languages borrow visual elements from the languages of wider communication. Kutsch Lojenga (2011) offers an example where Yak [axk] borrows the circumflex from Sango [sag] as a visual cue for 'high tone'.

> Occasionally, different accents are used, e.g. when the circumflex is used for H tone, as is done in YAKA (Bantu C.10, spoken in C.A.R.), where the choice of tone marks had to conform to the system used in the widely known lingua franca Sango, by using a circumflex for H tone. It may not be elegant for a linguist, but it works.

In Mexico, endangered and minority languages make use of visual elements from Spanish. In Chinantec and Spanish, the use of ⟨ ñ ⟩ is in a relationship similar to that described above by Kutsch Lojenga. In Chinantec, ⟨ ñ ⟩ represents a velar nasal, whereas in Spanish, it represents a palatal nasal. In Me'phaa, the acute accent ⟨ ◌́ ⟩ represents tone, whereas in Spanish and Chinantec it represents stress. In terms of the character composition and tactile input of that character, the question becomes: 'How is the diacritic related to the base, and does that relationship parallel the semantically salient ideas about the sounds these glyphs represent?'

Other questions include: 'How do speakers conceptualize the graphical elements of the glyph?' 'Do indigenous typists think of ⟨ ñ ⟩ as a separate character from ⟨ n ⟩, or do they think of it as an altered ⟨ n ⟩?' As layout developers, we must consider when we are dealing with two separate ideas or the modification of one idea (see Figure 3).



How do speakers / writers conceptualize the graphical elements and relate it back to the sound it represents?

Figure 3 Conceptualization of a graphical element

Table 1 *Options available to encode ñ in the Me'phaa and Chinantec layouts*

Unicode composite and base characters with consonants	
ñ LATIN SMALL LETTER N WITH TILDE U+00F1	n + ˜ LATIN SMALL LETTER N U+006E + COMBINING TILDE U+0303

In designing the Me'phaa and Chinantec layouts, one option was to remove the ⟨ ñ ⟩ from having its own dedicated key and to make the tilde ⟨ ◌̃ ⟩ a diacritic that was then accessed through a dead key or even the same dead key through which other diacritics in the language were accessed (see Table 1).

Such a strategy would not only free up a key in the layout, which could then be used for a more common character, but also bring consistency to the input of characters with diacritics. However, for both Me'phaa and Chinantec, it was decided to leave the ⟨ ñ ⟩ key as it appears on the Spanish ISO keyboard layout. This example highlights the fact that such considerations should be made on a language-by-language basis.

If a keyboard layout is to be intuitive to its users, then there should be a parallel between the graphical representation of sounds and the way in which the glyphs are generated by the fingers. This points to a tactile element in keyboard layout design. There should also be some internal cohesion regarding how composite characters are created by a given keyboard layout. An example of internal cohesion can be seen in the Me'phaa layout in the way in which tone is marked. The orthography of Me'phaa indicates three levels of tone. High tones are marked with an acute accent above the vowel ⟨ ◌́ ⟩, mid tones are unmarked, and low tones are marked with a COMBINING MACRON BELOW U+0331 ⟨ ◌̱ ⟩. The use of the macron below gives the visual effect of an underline below the vowel. The Me'phaa keyboard layout dedicates one dead key for high tone and another dead key for low tone. By giving each tone mark its own dead key, the keyboard layout creates symmetry in the user experience for how a tone can be marked on each vowel.

In Me'phaa, the letter ⟨ a ⟩ can be used by itself, with a low tone mark or with a high tone mark. In every case that ⟨ a ⟩ is combined with a low tone mark, two Unicode characters are needed: the base character ⟨ a ⟩ and the combining macron below diacritic ⟨ ◌̱ ⟩. However, when a high tone is used, there are several ways these could be encoded: either as ⟨ a ⟩ plus ⟨ ◌́ ⟩ or as a single character ⟨ á ⟩ (see Table 2).

In the Me'phaa case, the available options in Unicode do not make a difference for the implementation of a symmetrical input method. However, for the Chinantec keyboard, the symmetry was not as simple to achieve owing to limitations in one of the operating systems on which the layout was being implemented and the way in which characters are encoded in Unicode. Understanding how Unicode allows for the target characters to be created will help us see where there is symmetry and where there is asymmetry. Chinantec, like Me'phaa, is a tonal language. However, its orthography does not mark the tone on the vowel, but rather with numbers at the end of the syllable (Foris 2000). Moreover, Chinantec marks a type of stress on vowels with an acute accent (Mugele 1982). Symmetry does not become a problem until one tries to implement a stressed barred i ⟨ ɨ́ ⟩. Unicode does not contain a composite character for LATIN SMALL LETTER I WITH STROKE AND ACUTE. This means that the character needs to be a series of at least two Unicode code points, and it could even potentially be coded as three code points (see Table 3).

Table 2 *Options available to encode* á *in the Me'phaa and Chinantec layouts*

Unicode composite and base characters with vowels		
a LATIN SMALL LETTER A U+0061	a + ̱ LATIN SMALL LETTER A U+0061+COMBINING MACRON BELOW U+0331	a + ́ or á LATIN SMALL LETTER A U+0061+COMBINING ACUTE ACCENT U+0301 LATIN SMALL LETTER A WITH ACUTE U+00E1

Table 3 *The conceptual construction of* ⟨ ɨ́ ⟩ *in Unicode*

The conceptual construction of a character		
ɨ LATIN SMALL LETTER I WITH STROKE U+0268	í + ̵ LATIN SMALL LETTER I WITH ACUTEU+00ED+ COMBINING SHORT STROKE OVERLAY U+0335	ı + ̵ + ́ LATIN SMALL LETTER DOTTLESS I U+0131+ COMBINING SHORT STROKE OVERLAY U+0335+ COMBINING ACUTE ACCENT U+0301

The keyboard layout editor from Microsoft for Windows (MSKLC) allows only one Unicode code point per keystroke. Triple code point input, as is required by some orthographies (Holton 2011:372) is therefore impossible. With OS X, a dead key can be used to enter another state of the keyboard wherein, when the correct key is struck, the desired series of Unicode code points is input. However with MSKLC, this multiple code point input behaviour cannot be replicated. A dead key must be used to insert the combining diacritic and then the next key is used to insert the base. In this manner, all of the necessary diacritics for Me'phaa were achieved. However, for Chinantec, no solution was found for the composite character of accented barred i ⟨ ɨ́ ⟩. To represent symmetry with the other characters in the orthography, it would be necessary to move from barred i, LATIN SMALL LETTER I WITH STROKE U+0268 ⟨ ɨ ⟩, to barred i with acute (accent). This is not possible with Unicode because barred i would need to be dot-less, rather than combining above the dot. Alternatively, it would be possible to add the diacritic COMBINING SHORT STROKE OVERLAY U+0335 ⟨ ̵ ⟩ to the base character LATIN SMALL LETTER I WITH ACUTE U+00ED ⟨ í ⟩. For Windows, this pattern would require a fourth dead key for the stroke overlay (a dead key already exists for acute, dieresis, and tone) and, if implemented as a stressed ⟨ í ⟩ plus a stroke overlay, this dead key would not match the behaviour of the layout for adding a stress mark to the other vowels. Furthermore, this implementation would not fit the way in which Chinantec typists perceive the vowel (namely, as being barred i plus stress ⟨ ɨ ⟩+⟨ ◌́ ⟩). Rather, it forces typists to perceive the glyph as stressed i ⟨ í ⟩+⟨ ̵ ⟩ plus stroke overlay.

3.2.3 Unobtrusive design

Rams' third point is that good design is unobtrusive. Up to this point, the discussion in this chapter has focused on tools used to create keyboard layout files which work and which are installed within the framework of the OS without requiring third-party software solutions. As seen above, however, the limitations of MSKLC have provided the opportunity for creative solutions on the part of developers. Third-party solutions add complexity to the computing experience and to the deployment of layout files.

Some of the available third-party solutions include: *Keyman* (Durdin and Durdin 2011); *InKey* (Chenoweth 2012); *AutoHotKey* (Mallet 2012); *KeyTweak* (Krumsick 2009); *Sharp Keys* (Santossio 2011) and *Map Keyboard* (InchWest 2012). *Keyman* and *InKey* allow custom keyboard layout files to be edited but require their software to be active and running on the computer in order to use the custom layouts. Both these solutions are created with multilingual typists and endangered language typists in mind. *AutoHotKey* can be configured so that a script converts each keystroke into the desired character(s). *KeyTweak*, *SharpKeys* and *Map Keyboard* are essentially graphical interfaces on registry editors for the Windows-based keyboard registries. They can be viewed as MSKLC alternatives with one exception – namely, that if a user edits a registry file, then the changes are global; if a user creates a keyboard layout file with MSKLC, the user can choose when to use a given keyboard layout on a per programme basis.

These third-party solutions require the installation of both additional software and the particular keyboard desired by an endangered language typist. MSKLC-based solutions require only the installation of a keyboard file via an `.exe` script. Therefore, in terms of design, and in terms of creating a solution that can readily be adopted and used by an endangered language community, these third-party solutions are non-optimal.

As language documenters and advocates of endangered languages, when we introduce a solution to a community, we must always consider the solution's longevity and sustainability. What is the future capacity within the community to develop or modify this solution? Does this software have any dependencies (other than the OS) that might disrupt communication for this particular speech community? Closed-source, third-party solutions and third-party solutions that have only one maintainer are prone to become obsolete more rapidly. Solutions with a significant financial return on time invested are more likely to remain usable.

3.2.4 Detailed design

Rams' fourth point—that good design is thorough down to the last detail—has several implications for language documenters. As designers consider what is needed in layouts, they must also reflect on how these digital products can be disseminated through the speech community's social networks. What impact will the solution have on language vitality? What level of complexity will the solution bring to digital interactions within a given speech community?

Details matter if we hope to improve the digital interface for endangered language typists and to see our recommended (and collaboratively developed) digital solutions embraced by speech communities. Is character position within the keyboard layout congruent with the national language keyboard layout? How is the character formed in the mind of the typist? What is the relationship between a diacritic and its base? For tonal languages, it may mean thinking through whether the conceptual unit is a toneme, a vowel with a tone or a high-tone vowel. Is it a grammatical tone attached to the tone-bearing unit of the word or a tonal melody (Snider 1999) superimposed on a word or morpheme? What is involved in the actual composition of characters and the input of Unicode code points of the data?

Good design is not designing a (physical) usable keyboard. Good design is all about creating a keyboard layout that ergonomically, psychologically and intuitively meets the needs of native speaker-writers.

4 Designed distribution

When they license and release products, designers should keep free and open source software (FOSS) principles in mind and should choose stable, reliable and sustainable distribution mechanisms (Wong and Sayo 2004). FOSS principles allow for digital products to be accessed and shared for free throughout the speech community. Sustainability is encouraged by allowing anyone to alter and redistribute the software. FOSS principles include: first, the freedom to run a programme, for any purpose; second, the freedom to study how a programme works and to adapt it to a person's needs; third, the freedom to redistribute copies in order to help others; and fourth, the freedom to improve a programme and to release these improvements to the public, so that the whole speech community may benefit.

By adding a reliable distribution point to the FOSS principles, the designer builds trust with the endangered language community. Trust is built as community members have a stable place to obtain the keyboard layout product and a trusted copy of the source code. *ScriptSource* is one such example of a place that could be developed to meet this need at a global scale (Raymond 2012). Clearly, the places chosen for this purpose need to be accessible locally and to endangered language typists in diaspora.

In keyboard layout design, the goal is to develop technology that implements an orthography. While some may view the process of developing typing options for speech communities as a situation where both the keyboard layout and the orthography are variables, it is far easier to establish the orthography first and then afterwards to develop technology that meets the demands of this orthography. It has been argued that, if the characters of the endangered language orthography are restricted to those used by the majority language orthography, then typists of endangered languages may as well use the majority language keyboard layout. Jany (2010:235–6), for example, presents this position with respect to the Chuxnabán Mixe [pxm].

> [An] important non-linguistic factor in the development of an orthography for an oral language is ease of use with computers and new media. With the world-wide web reaching even the remotest areas of the world and expanding in use, it becomes clear that a new orthography should be designed in a way so that its graphemes are readily available on standard keyboards. This will not only facilitate the language documentation process, it will also encourage its use with new media and possibly in new domains.

In the Mixe example, the keyboard that is available is viewed as a potential repository for orthographic characters. However, this position is not accepted universally. In South Africa, rather than being subject to this type of 'mechanical imperialism', translate.org.za helped create the South African Keyboard (Bailey 2007), whose designers chose to not become subject to the confines of technology and created a keyboard layout that addressed the specific needs of one language, Venda [ven], but which also addressed the needs of multiple languages in South Africa. In this way, they brought the characters needed by typists to the keys that lay beneath their fingertips.

If orthography developers approach their task with the mindset that they also need to create a keyboard layout, then the orthography is unnecessarily limited by its digital implementation. In other words, orthographies become restricted in order to 'fit' known technologies. Addressing the larger and more complex question about the orthography and maximal differentiation, Cahill and Karan (2008:10) have stated that designing orthographies is all about matching the orthography to the social attitudes of the language speakers. In many respects, this is very much like corporate logo design or typeface design. Keyboard designers should follow orthography development rather than leading it. As Lüpke (2011:316–17) describes, the Baïnouk [bcz] orthography was changed overnight to comply with government codification policy so that the language would receive recognition and status. During the process of standardization, linguists often become spectators at a tug-of-war for social, economical or political clout, hoping that everyone involved feels like they have won something; even if it means reprinting language teaching materials. Of course, this has long-term implications for endangered language groups who are trying to establish a culture of writing. If the orthography is changed every five to ten years, then old materials, and potentially skills become redundant very quickly.

5 Conclusion

The advance of technology is changing the dynamics of communicative settings. Human beings are more than willing to adopt these advances as long as they can embrace the benefits. Languages become endangered partly because of the very fact that human beings are so highly pragmatic in the adoption of communicative frameworks. Speakers of endangered languages do not stop communicating, but they do start communicating in other languages. Well-designed keyboard layouts do not represent a technological panacea. By themselves, they will not reverse language shift (Dauenhauer and Dauenhauer 1998:70; Holton 2011:397). In the twenty-first century, well-designed layouts give a speech community the ability to use their language in the medium of choice. However, just because a speech community has the ability to type in their language does not, of course, mean that they will. Nonetheless, having that ability gives a speech community a choice. According to Pavlov (2011:241) 'The problem [of online Yakut use] is aggravated by the paucity of native speakers generally and … that young people are used to socializing in cyberspace in the majority, functionally stronger, language.' Pavlov goes on to say that a keyboard layout is a necessary component of the strategy for encouraging the use of Yakut [sah].

Typists of endangered languages will only be successful when there is desire, ability and opportunity to function in a digital medium. Technology in and of itself is not the saviour of an endangered language. Speakers and users of the language must also exist. Even with a keyboard layout and technology to support an endangered language in a digital medium, the social pressures that encourage writers of endangered languages not to use their language still exist. It is the designer's task to ensure that complex, confusing and unintuitive keyboard experiences do not provide an additional discouragement.

A. List of Abbreviations

C.A.R.	=	Central African Republic
FOSS	=	Free and Open Source Software
ISO	=	International Standards Organization
MSKLC	=	Microsoft Keyboard Layout Creator
NEH	=	National Endowment for the Humanities
OS	=	Operating System
USD	=	United States Dollar

B. List of Languages Mentioned

Table 4 *Languages mentioned, their ISO 639-3 codes.*

Language Name	**ISO 639-3 Code**
Baïnouk	bcz
Chuxnabán Mixe	pxm or mis
Kalsha	kls
Malinaltepec Me'phaa	tcf
Sango	sag
Sochiapam Chinantec	cso
Spanish	spa
Venda	ven
Yak	axk
Yakut	sah

References

Anderson, Stephen P. 2006. Creating Pleasurable Interfaces: Getting From Tasks to Experiences. Paper presented at: Refresh '06. Orlando, Florida. http://www.slideshare.net/stephenpa/creating-pleasurable-interfaces-getting-from-tasks-to-experiences

Anderson, Stephen P. 2009. Seductive Interactions. Paper presented at: Idea '09. http://www.slideshare.net/stephenpa/seductive-interactions-idea-09-version

Anderson, Stephen P. 2011a. Long After the Thrill: Sustaining Passionate Users. Paper presented at: #IxD11. http://www.slideshare.net/stephenpa/long-after-the-thrill-sustaining-passionate-users

Anderson, Stephen P. 2011b. *Seductive interaction design: creating playful, fun, and effective user experiences.* Berkeley, CA: New Riders.

Bailey, Dwayne. 2007. Creating a Single South African Keyboard Layout to Promote Language. *LEXIKOS* 17(1). 212–225. doi:10.5788/17-0-587

Bel, Bernard & Médéric Gasquet-Cyrus. 2015. Digital curation and event-driven methods at the service of endangered languages. In Mari C. Jones, *Endangered Languages and New Technologies,* 113–126. Cambridge, UK: Cambridge University Press. doi:10.1017/CBO9781107279063.011

Bernard, H. Russell. 1992. Preserving Language Diversity. *Human Organization* 51(1). 82–89. http://www.metapress.com/content/BP4765G377Q32761

Bird, Steven and Gary F. Simons. 2003. Seven dimension of portability for language documentation and description. *Language* 79(3). 557–582. http://www.jstor.org/stable/4489465

Brownie, John. 2012. Ukelele: Mac OS X Keyboard Layout Editor. 2.1.9 edn. Dallas, Texas: SIL International. http://scripts.sil.org/ukelele (2012-05-15.)

Byers, Emily M., Andrew Brovey & Jane Zahner. 2004. Teaching and Learning Keyboarding. *Action Research Exchange* 3(1). http://teach.valdosta.edu/are/vol3no1/pdf/embyers-article.pdf

Cahill, Michael C. and Karan, Elke. 2008. Factors in Designing Effective Orthographies for Unwritten Languages. *SIL Electronic Working Papers* 2008(001). 1–15. Dallas, Texas: SIL International. http://www.sil.org/silewp/abstract.asp?ref=2008-001

Chenoweth, Ben. 2012. Inkey Keyboard Creator. http://code.google.com/p/inkey-keyboard-creator

Csató, Éva Á. and David Nathan. 2007. Multiliteracy, past and present, in the Karaim communities. In Peter K Austin., eds. *Language Documentation and Description,* 4, 207–230. London: The Hans Rausing Endangered Languages Project, School of Oriental and African Studies. http://eprints.soas.ac.uk/6083

Dauenhauer, Nora Marks and Richard Dauenhauer. 1998. Technical, emotional, and ideological issues in reversing language shift: examples from Southeast Alaska. In Lenore A. Grenoble & Lindsay J. Whaley, *Endangered languages: language loss and community response,* 57–98. Cambridge: Cambridge university press. doi:10.1017/CBO9781139166959.004

de Graaf, Tjeerd, Cor van der Meer and Lysbeth Jongbloed-Faber. 2015. The use of new technologies in the preservation of an endangered language: The case of Frisian. In Mari C. Jones, *Endangered Languages and New Technologies,* 141–149. Cambridge, UK: Cambridge University Press. doi:10.1017/CBO9781107279063.013

Diki-Kidiri, Marcel. 2011. How to Guarantee the Presence and the Life of a Language in Cyberspace. In Evgeny Kuzmin, Ekaterina Plys & Anastasia Parshakova, *Linguistic and Cultural Diversity in Cyberspace. Proceedings of the International Conference (Yakutsk, Russian Federation, 2-4 July, 2008),* 230–232. Moscow: Russia: Interregional Library Cooperation Centre. http://www.ifapcom.ru/files/Documents/multiling_eng.pdf

Durdin, Marc and John Durdin. 2011. Tavultesoft Keyman. Tavultesoft Pty Ltd. http://www.tavultesoft.com

Eisenlohr, Patrick. 2004. Language Revitalization and New Technologies: Cultures of Electronic Mediation and the Refiguring of Communities. *Annual Review of Anthropology* 33(1). 21–45. Annual Reviews. http://www.jstor.org/stable/25064844 doi:10.2307/25064844

Foris, David Paul. 2000. Developing a Sochiapan Chinantec orthography: linguistic factors and sociolinguistic results. Paper presented at: Bilingualism at the Ends of the Earth. Department of General and Applied Linguistics, University of Waikato, Hamilton, New Zealand.

Galla, Candace K. 2009. Indigenous Language Revitalization and Technology From Traditional to Contemporary Domains. In Reyhner, Jon Allan, Lockard, Louise, *Indigenous Language Revitalization: Encouragement, Guidance & Lessons Learned,* 167–182. Flagstaff, AZ: Northern Arizona University. http://jan.ucc.nau.edu/~jar/ILR/ILR-13.pdf

Guérin, Valérie. 2008. Writing an endangered language. *Language Documentation & Conservation* 2(1). 47–67. http://hdl.handle.net/10125/1804

Harvey, Chris. 2013. Languagegeek.com. 23 January 2013 edn. http://www.languagegeek.com/keyboard_general/all_keyboards.html

Himmelmann, Nikolaus P. 1998. Documentary and Descriptive Linguistics. *Linguistics* 36(1). 161-95. doi:10.1515/ling.1998.36.1.161

Hinkle, Lauren, Miguel Lezcano & Jugal Kalita. 2010. Designing Soft Keyboards for Brahmic Scripts. *ICON-2010: 8th International Conference on Natural Language Processing.* 191–200. Kharagpur, India: Macmillan Publishers. http://ltrc.iiit.ac.in/icon_archives/ICON2010/10Dec2010/Paper11-File39-Paper186.pdf

Holton, Gary. 2011. The role of information technology in supporting minority and endangered languages. In Peter K. Austin & Julia Sallabank, *The Cambridge Handbook of Endangered Languages,* 371–399. Cambridge: Cambridge University Press. http://dx.doi.org/10.1017/CBO9780511975981.019 doi:10.1017/CBO9780511975981.019

Hosken, Martin. 2001. An introduction to keyboard design theory: What goes where? In Melinda Lyons, *Implementing Writing Systems: an Introduction,* 121–137. Dallas, Texas: SIL International. https://scripts.sil.org/KeybrdDesign

Hugo, Russell. 2015. Endangered languages, technology and learning: Immediate applications and long-term considerations. In Mari C. Jones, *Endangered Languages and New Technologies,* 95–111. Cambridge, UK: Cambridge University Press. doi:10.1017/CBO9781107279063.009

InchWest. 2012. MapKeyboard. 1.5 edn. InchWest. http://www.inchwest.com/mapkeyboard

Jany, Carmen. 2010. Orthography Design for Chuxnabán Mixe. *Language Documentation & Conservation* 4(1). 231–253. https://scholarspace.manoa.hawaii.edu/handle/10125/4481

Kano, Akiyo. 2008. MECE Method For Categorising Typing Errors. In David England, eds. *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction,* 2, 249–250. http://www.chici.org/references/MECE.pdf

Kano, Akiyo and Janet C. Read. 2009. Text input error categorisation: solving character level insertion ambiguities using Zero Time analysis. *People and Computers XXIII Celebrating People and Technology: Proceedings of the 2009 British Computer Society Conference on Human-Computer Interaction, BCS-HCI 2009, Cambridge, United Kingdom, 1-5 September 2009,* 293–302. ACM Press.

Kano, Akiyo, Janet C. Read, Alan Dix and I. Scott MacKenzie. 2007. ExpECT: An Expanded Error Categorisation Method for Text Input. In Linden J. Ball, M. Angela Sasse, Corina Sas, Thomas C. Ormerod, Alan Dix, Peter Bagnall & Tom McEwan, eds.

People and Computers XXI – HCI… but not as we know it: Proceedings of HCI 2007, 1, 147-56. ACM Press.

Krishna, Ashish, Rahul Ajmera, Sandesh Halarnkar and Prashant Pandit. 2005. Gesture Keyboard - User centered design of a unique input device for Indic Scripts. *HCII 2005 Proceedings.* Volume 7 - Universal Access in HCI: Exploring New Interaction Environments:Paper 59. Mahwah, NJ: Lawrence Erlbaum Associates.

Krumsick, Travis. 2009. KeyTweak. Redmond, Washington: Travis Krumsick. http://webpages.charter.net/krumsick/

Kutsch Lojenga, Constance. 2011. Orthography and Tone: Tone system typology and its implications for orthography development. Paper presented at: Linguistic Society of America Annual Meeting. Pittsburgh, Pennsylvania – Jan 6–9, 2011. http://www.sil.org/linguistics/2011LSASymposium/KutschLojenga.html

Lüpke, Friederike. 2011. Orthography development. In Peter K. Austin & Julia Sallabank, *Handbook of endangered languages,* 312–336. Cambridge, UK: Cambridge University Press. doi:10.1017/CBO9780511975981.016

Mallet, Chris. 2012. AutoHotKey. 1.1.09.02 edn. Indiana, USA: Chris Mallet. https://www.autohotkey.com

Marlett, Stephen A. 2010. *Documenting the Me'phaa Genus (Mexico).* Washington, D.C, USA: National Endowment for the Humanities - Documenting Endangered Languages - Fellowship: FN-50079-10 (11/1/2010 – 1/31/2012). https://securegrants.neh.gov/publicquery/main.aspx?f=1&gn=FN-50079-10

McLendon, William W. Jr. 2011. SIL Technology for Multilingualism in Cyberspace. In Evgeny Kuzmin, Ekaterina Plys & Anastasia Parshakova, *Proceedings of the International Conference: Linguistic and Cultural Diversity in Cyberspace (Yakutsk, Russian Federation, 2-4 July, 2008),* 97-104. Moscow: Interregional Library Cooperation Centre. http://www.ifapcom.ru/files/Documents/multiling_eng.pdf

Microsoft. 2012. The Microsoft Keyboard Layout Creator. Version 1.4 edn. Redmond, Washington: Microsoft. https://www.microsoft.com/en-us/download/details.aspx?id=102134

Mugele, Robert Louis. 1982. *Tone and Ballistic Syllable in Lalana Chinantec.* Austin, Texas: The University of Texas at Austin dissertation.

Pavlov, Nikolai. 2011. Increasing the Numbers of Sakha-Speaking Internet Users. In Evgeny Kuzmin, Ekaterina Plys & Anastasia Parshakova, *Linguistic and Cultural Diversity in Cyberspace. Proceedings of the International Conference (Yakutsk, Russian Federation, 2-4 July, 2008),* 238–242. Moscow: Interregional Library Cooperation Centre. http://www.ifapcom.ru/files/Documents/multiling_eng.pdf

Raymond, Martin. 2012. ScriptSource: Making information on the world's scripts and languages accessible. Paper presented at: Charting Vanishing Voices, 29–30 June 2012. Cambridge, UK.

Read, Janet, Stuart MacFarlane and Chris Casey. 2001. Measuring the Usability of Text Input Methods for Children. In Blandford, Ann, Vanderdonckt, Jean, Gray, Phil, *People and Computers XV—Interaction without Frontiers,* 559-72. London: Springer. http://dx.doi.org/10.1007/978-1-4471-0353-0_35 doi:10.1007/978-1-4471-0353-0_35

Santossio, Randy. 2011. SharpKeys. Redmond, Washington: Randy Santossio. http://www.randyrants.com/2011/12/sharpkeys_35.html

Scott Warren, Anthony and Geraint Jennings. 2015. 'Allant contre vent et mathée': Jèrriais in the twenty-first century. In Mari C. Jones, *Endangered Languages and New Technologies,* 127–140. Cambridge, UK: Cambridge University Press. doi:10.1017/CBO9781107279063.012

Seifart, Frank. 2006. Orthography development. In Jost Gippert, Nikolaus Himmelmann & Ulrike Mosel, *Essentials of language documentation. (Trends in Linguistics. Studies and Monographs [TiLSM] 178),* 275–301. Berlin; New York: Mouton de Gruyter. doi:10.1515/9783110197730.275

Silva, Kalena and Keola Donaghy. 2004. Ke A'o Ho'okeleka'a'ike: Hawaiian Language Instruction On The Internet. In Yoshiko Saito-Abbott, Richard Donovan & Thomas Abbott, *Language on the Edge: Implications for Teaching Foreign Languages and Cultures, Proceedings of Digital Stream 2003 (Emerging Technologies in Teaching Language and Culture IV)*. San Diego, California: Montezuma Publishing. https://www2.hawaii.edu/~donaghy/eng/digitalstream2003.html

Snider, Keith L. 1999. *The geometry and features of tone.* Dallas, Texas: Summer Institute of Linguistics.

Tavultesoft. 2013. Keyboard Search. 23 January 2013 edn. tavultesoft.com. http://keymankeyboards.com

Vitsœ. 2012. Dieter Rams: Ten principles for good design. http://www.vitsoe.com/en/gb/about/dieterrams/gooddesign (27 June 2012.)

Wong, Kenneth and Phet Sayo. 2004. *Free / Open Source Software.* (e-Primers for the Information Economy, Society and Policy.) Kuala Lumpur, Malaysia: United Nations Development Programme's Asia-Pacific Development Information Programme (UNDP-APDIP). http://www.iosn.net/downloads/foss_primer_current.pdf

Zhozhikov, Anatoli, Yakov Aleksandrov and Alexander Varlamov. 2011. The Type Fonts of the Yakut Alphabet and Those of the Minority Peoples Residing in the Republic of Sakha (Yakutia): Challenges of Applying in Operating Systems. In Evgeny Kuzmin, Ekaterina Plys & Anastasia Parshakova, *Linguistic and Cultural Diversity in Cyberspace. Proceedings of the International Conference (Yakutsk, Russian Federation, 2-4 July, 2008),* 250–253. Moscow: Russia: Interregional Library Cooperation Centre. http://www.ifapcom.ru/files/Documents/multiling_eng.pdf