PUST Computer Science and International Standards

Mike Izbicki

22 March 2022

1 Introduction

This document describes three recent historical technical incidents important for the PUST CS curriculum. In each incident, the DPRK tried to integrate into the international community, but they failed because they lacked citizens trained with technical skills who could speak English. For each incident, I provide technical and historical background on the incident, and a list of recommendations about how the PUST CS department can adjust our classes and research to prevent these problems in the future.

2 Incident 1: The Unicode Consortium

Unicode[1] is the standard method for representing multilingual text on computers, and all major programming languages have easy-to-use libraries for working with Unicode text. The Unicode Consortium is the organization responsible for maintaining the Unicode standard, and their goal is to be an apolitical body that ensures that all languages/cultures are equally supported.

The Unicode standard, however, is biased in favor of the ROK-Korean dialect over the DPRK-Korean dialect. This is due to a historical accident where the DPRK did not have the technical capacity to participate in the standards making process, and so their needs were not considered. This section first describes the technical problems with the Unicode standard for the DPRK, then describes the historical causes.

2.1 Technical Problems

The Committee for Standardization of the DPRK (CSK)[2] submitted a memo[3][4] to the Unicode Consortium in 1997 that lists three difficulties in working with Unicode in the DPRK. None of these problems have been fixed in the last 25 years. The problems are:

1. The official name of the Korean language script in Unicode is "Hangul" (see Section 18.6 of the Unicode 14.0 standard[5]). Hangul is the ROK's name for their script, and the DPRK prefers the name "Choseongul". The DPRK suggested that the name "Korean characters" be adopted as a politically neutral term.
2. The DPRK and ROK use a different sorting order for their alphabets[6]. The ROK order for consonants is

   ㄱ ㄲ ㄴ ㄷ ㄸ ㄹ ㅁ ㅂ ㅃ ㅅ ㅆ ㅇ ㅈ ㅉ ㅊ ㅋ ㅌ ㅍ ㅎ

[1] https://unicode.org
[2] https://www.iso.org/member/1657.html
[3] https://unicode.org/wg2/docs/n2056.pdf
[4] https://www.unicode.org/L2/L1999/99380.htm
[5] https://www.unicode.org/versions/Unicode14.0.0/
[6] https://en.wikipedia.org/wiki/North%E2%80%93South_differences_in_the_Korean_language

and the DPRK order is

ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ ㄲ ㄸ ㅃ ㅆ ㅉ ㅇ

For example, in the ROK, the word 까치 (magpie) comes alphabetically before the word 나비 (butterfly), but in the DPRK the word 나비 comes alphabetically before 까치.

The Unicode standard orders Korean characters according to the ROK-ordering, and so by default all sorting done in any programming language will sort Korean words in the ROK-preferred way. A special extension called a collation algorithm[7] is required to sort according to the DPRK-ordering.

As of 2022, the current list of collation algorithms[8] does not have an entry for the DPRK-dialect of Korean, and so it is currently impossible in any programming language to sort text alphabetically accoding to the DPRK-ordering.

3. The DPRK internally uses the KPS9566 character set[9]. This character set contains several characters that the Unicode Consortium does not want to support. For example, it contains political characters representing the Workers Party of Korea, and 4 distinct versions of the character 김 (one for normal text, and one each for Kim Il Sung, Kim Jong Il, and Kim Jong Un).

   This lack of support for certain characters used by the DPRK prevents documents produced in the DPRK from being opened in tools like Microsoft Word, and even programming languages like Python and R cannot work with these documents. This lack of compatibility adds considerable friction to negotiations, since diplomats between the DPRK and the United States cannot easily exchange documents.

There is at least one more problem with the Unicode standard for the DPRK not listed above:

1. The current Unicode standard does not support transliteration of Korean into Latin characters using the DPRK's preferred Romanization system[10], and instead only supports the McCune–Reischauer[11] system. Furthermore, transliterations into non-Latin alphabets are not supported at all, despite the importance of transliterating into Cyrillic. A 2018 UN report on romanization[12] describes a good history of the many Romanization systems for Korean.

2.2 Historical Basis

The ROK has been actively and publicly developing their systems for encoding Korean text since the earliest days of the internet. KAIST first developed the KSC5601 encoding method in 1974, and actively worked with companies like IBM and Microsoft, and standards organizations in the US and Europe to ensure widespread support for this standard. The ROK issued an official Request for Comments (RFC) on the encoding in 1993 via RFC1557[13] to suggest that KSC5601 be the standard format for exchanging Korean emails. When the Unicode Consortium was first founded in 1991, ROK programmers were well positioned to contribute to the developing standard. They had the detailed technical knowledge of developing many of their own internal encodings, they had experience interacting with diverse technical committees, and they had the English communication skills for communicating in the Unicode Consortium's working language.

In contrast, the DPRK has severely lagged the ROK in this area. It's not known when the DPRK first developed their own Korean encoding, but the DPRK's KPS9566 encoding was first published internationally in 1997 and officially registered with the Internaional Standards Organization (ISO) in 1998. It wasn't until August 1999 that the DPRK began discussions for enabling Unicode compatibility. The DPRK submitted an official statement[14][15] to the Unicode Consortium outlining their difficulties adopting the Unicode standard

[7] https://cldr.unicode.org/index/cldr-spec/collation-guidelines
[8] https://github.com/unicode-org/cldr/tree/main/common/collation
[9] https://en.wikipedia.org/wiki/KPS_9566
[10] https://en.wikipedia.org/wiki/Romanization_of_Korean_(North)
[11] https://en.wikipedia.org/wiki/McCune%E2%80%93Reischauer
[12] https://unstats.un.org/Unsd/geoinfo/UNGEGN/docs/22-GEGN-Docs/wp/gegn22wp48.pdf
[13] https://datatracker.ietf.org/doc/html/rfc1557
[14] https://unicode.org/wg2/docs/n2056.pdf
[15] https://www.unicode.org/L2/L1999/99380.htm

(summarized above), but since they entered this discussion 8 years after it began, the technical decisions had already been made. In order to not break backwards compatibility, the Unicode Consortium issued a statement[16] that they could not implement the changes requested by the DPRK.

> **Fun Fact:** There are 7 emojis[17] in the current Unicode standard that were added at the request of the DPRK. The DPRK originally suggested that the HOT BEVERAGE emoji should be called the HOT TEA emoji, but an American suggested the emoji be renamed so that Americans could use it to represent coffee. The DPRK delegation agreed, and so the emoji was renamed. This is an example of technical experts working on narrow technical problems being able to work together in a way that diplomats can't.

2.3 How PUST Can Help

Classroom: Proper use of Unicode should be included in both the undergrad and graduate curriculum. We should also include related concepts like internationalization and localization of software[18].

At the undergraduate level, these concepts should be included in the following courses:

1. data structures, and
2. natural language processing.

At the masters level, it might make sense to have a special seminar series for this topic.

Research: PUST students can write senior and masters theses that directly solve the problems listed above. For example, a PUST student could:

1. Develop a new collation algorithm for the DPRK-ordering of Korean text.
2. Add KPS9566 support to a programming language like Python or R that does not currently have support.
3. Implement the transliteration schemes and add them to the Unicode CLDR database.

The forein faculty can help the PUST students interface with the appropriate standards committee to get the developments approved.

3 Incident 2: Internet Networking Organizations

DPR Korean webpages online are important for diplomatic communication between the DPRK and outside world. For example, the Korean Central News Agency (KCNA) is the DPRK's primary venue for distributing official government policies, and they post these policies online at http://kcna.kp. Western political analysts rely on access to this website in order to understand the DPRK's official policy positions, and if this website goes offline, then the DPRK and Western policy makers cannot communicate. Both the DPRK and the West therefore have a shared interest in removing the technical obstacles that can prevent access to DPR Korean webpages.

3.1 Technical Problems

Every country in the world is allocated a country-code top level domain (ccTLD)[19] which identifies webpages for that country on the internet. DPR Korean webpages use the `.kp` ccTLD[20], which currently hosts about

[16] https://unicode.org/wg2/docs/n2392.pdf
[17] https://en.wikipedia.org/wiki/KPS_9566#Impact_on_Unicode_today
[18] https://en.wikipedia.org/wiki/Internationalization_and_localization
[19] https://en.wikipedia.org/wiki/Country_code_top-level_domain
[20] https://en.wikipedia.org/wiki/.kp

30 different webpages. (The full list is available from http://dprkportal.kp.) The Internet Assigned Numbers Authority (IANA) is the organization responsible for managing these ccTLDs. According to the IANA archives[21], the DPRK first attempted to register their `.kp` ccTLD with IANA in 2004. But they didn't have anyone who both spoke English and knew about computer networking, and so they couldn't correctly fill out the forms and configure their servers to get their domains online. They had to resubmit the forms many times, and it wasn't until 3 years later in 2007 that the `.kp` ccTLD was finally online.

Originally, a German company called the Korea Computer Center Europe (KCCE) was responsible for managing the `.kp` ccTLD. In 2009-2010, however, the `.kp` ccTLD stopped working for unknown reasons. After months of downtime, responsibility was transferred to the DPR Korean company Joint Star Ventures[22]. Again, a combination of better English and technical skills would have prevented this downtime.

3.2 Potential Future Problems

While the problems above have all been fixed, the DPRK will likely face new technical challenges relating to their internet infrastructure in the coming years.

IP addresses are how computers on the internet communicate with each other. The DPRK registered their first (and currently only) IP addresses in 2009[23]. APNIC is the organization responsible for assigning IP addresses to the Asian region, and they have a blog post about the DPRK's IP address allotment[24].

Currently, the DPRK has 1024 IP addresses allocated to them, and all of these IPs use the IPv4 protocol. IPv4 addresses are nearly exhausted, however, and so there is an ongoing effort to move to the IPv6 protocol. The DPRK will likely encounter communication difficulties with APNIC when they try to register their first block of IPv6 addresses.

3.3 How PUST Can Help

Classroom:

1. The networking class should emphasize the IPv6 protocol and the many international bodies that work together to make internet networking possible.
2. The architecture class should emphasize the physical components of networking infrastructure to prepare students for the networking class.
3. We should introduce a class dedicated to wireless networking (possibly at the masters level).

Research:

1. A good thesis project would be to convert the internal PUST network to use the IPv6 protocol. This would be closely related to the many IoT projects students have worked on, since IPv6 is often used with IoT.
2. Other research projects could look into applications of the latest HTTP protocols, or protocols designed specifically for wireless/mobile communication.

4 Incident 3: Web Development

It is important that webpages not just be online, but that the webpages themselves be well designed. There are many international standards for webpage design, and these standards change at a rapid pace. These

[21] https://www.iana.org/reports/2007/kp-report-11sep2007.html
[22] https://www.iana.org/reports/2011/kp-report-20110401.html
[23] https://bgpview.io/asn/131279
[24] https://blog.apnic.net/2015/08/31/the-internet-in-north-korea-hanging-by-a-single-thread/

changing standards have in the past caused the DPRK's websites to become essentially unusable for the Western audience. PUST can help teach the latest standards so that DPR Korean developers can keep their webpages highly usable.

4.1 Technical Problems

There are too many problems with the DPRK's webpages to provide an exhaustive list here, so I provide a representative sample of three specific problems with the KCNA webpage http://kcna.kp.

1. If you do a google search for "`united states site:kcna.kp`" (basically try to find all KCNA articles mentioning the United States), you won't get any results. This isn't because of censorship by anyone, but because of technical problems with the way kcna.kp uses javascript that doesn't work with Google's indexer. This makes it hard for US diplomats/academics to understand the DPRK because they can't search the historical record about what the DPRK has said/done in the past. A bespoke, paid search engine https://kcnawatch.org has been specially developed by the folks at https://nknews.org to crawl the kcna.kp webpage (demonstrating a high demand for access to this information), but because the search engine requires payment, it is not widely used.

   > **Edit:** Sometime within the last month, this has changed. KCNA updated their webpage so that it is now indexable on google!

2. Even worse, KCNA's improper use of javascript has resulted in security vulnerabilities[25] that have allowed foreigners to access the website's administrative panels. Foreign (i.e. non-DPR Korean) hackers have likely used these vulnerabilities to distribute malware through the kcna.kp webpage[26]. Since this website is the primary way that the DPRK communicates with the outside world, and US researchers and diplomats access it on a regular basis, we do not want it to be spreading malware. So we should help DPR Korean engineers keep the malware off.

3. If you visit httpS://kcna.kp (note the https and not plain http), you'll get a scary message that the webpage is a security threat. This is not due to malware, but because they are using old encryption protocols and self-signed certificates[27] that would let a 3rd party perform a man-in-the-middle attack[28]. We want these webpages to use cryptography correctly so that both foreigners and DPR Korean citizens can access them securely without being monitored.

4.2 How PUST Can Help

Classroom: The most obvious solution to this problem is in PUST's existing web development course. Ideally I think we would divide this up into 2 courses, one for frontend development and one for backend development. The problem with this, however, is that it will be hard to find teachers. (This class hasn't been taught by a foreign professor since Wes Brewer left PUST >5 years ago.)

Less obviously, but at least as important, I propose that we ensure that all our CS classes contain a web development component somehow.

1. Security:
   1. Should cover secure internet communication protocols like HTTPS, DNSSEC, Tor
   2. Should cover Public Key Infrastructure (PKI) and how this is necessary for the secure protocols to work
   3. Should cover how to build proper login systems for webpages
2. Algorithms/Data structures:

[25] https://web.archive.org/web/20131124224906/http://pp19dd.com/2013/11/north-korean-central-news-agency-kcna-server-logs-were-protected-b
[26] https://securelist.com/whos-really-spreading-through-the-bright-star/68978/
[27] https://www.thesslstore.com/blog/ssl-north-korea/
[28] https://en.wikipedia.org/wiki/Man-in-the-middle_attack

    1. Should cover "how Google works": Dijkstra's algorithm, B-Trees, Inverted Indexes

3. Compilers:

    1. Should cover compiling Markdown into HTML
    2. Should cover the many languages that compile down to javascript/web assembly

4. Databases:

    1. Should cover how databases serve as the backend for a webpage
    2. Maybe this could take the place of the backend web development class?

5. Image processing:

    1. Should cover new image formats designed specifically for web pages, like WEBP and JPEGXL

6. Software Engineering:

    1. Should cover web development techniques like: git, continuous integration, microservices
    2. Should cover case studies of web-based companies like: Facebook, Instagram, Twitter
    3. Should cover open source software development

Research:

1. Reimplement portions of the DPRK websites locally using more modern web development techniques.
2. Develop a custom Content Management System (CMS) that meets the specific needs of the DPRK's internet environment.