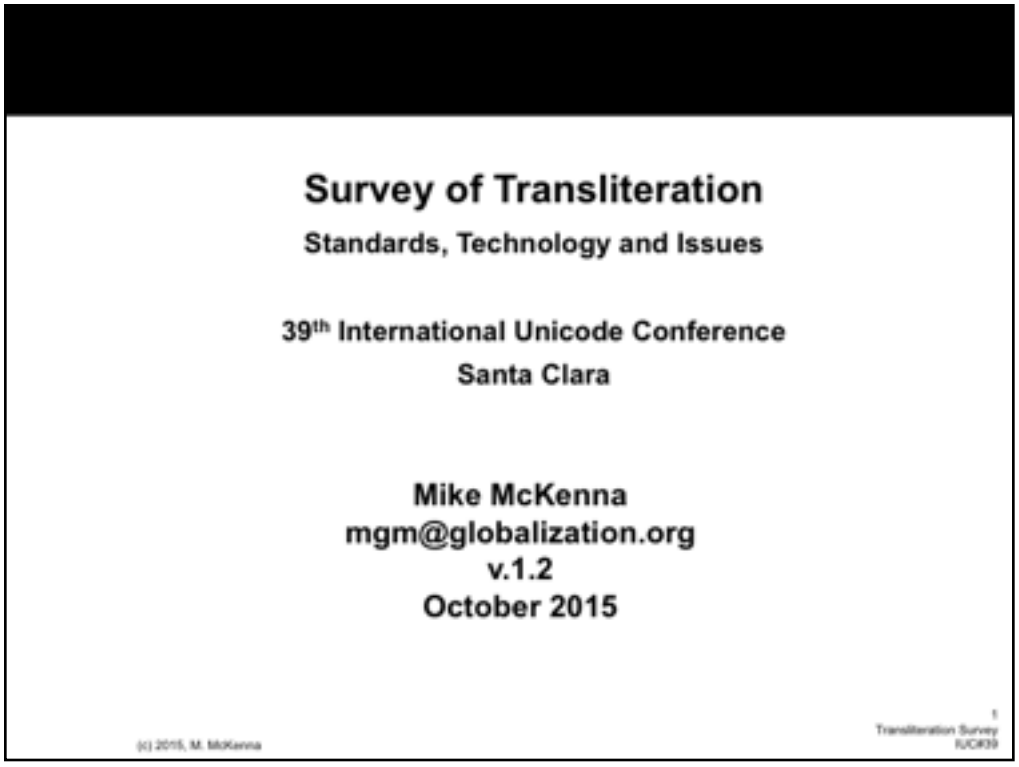
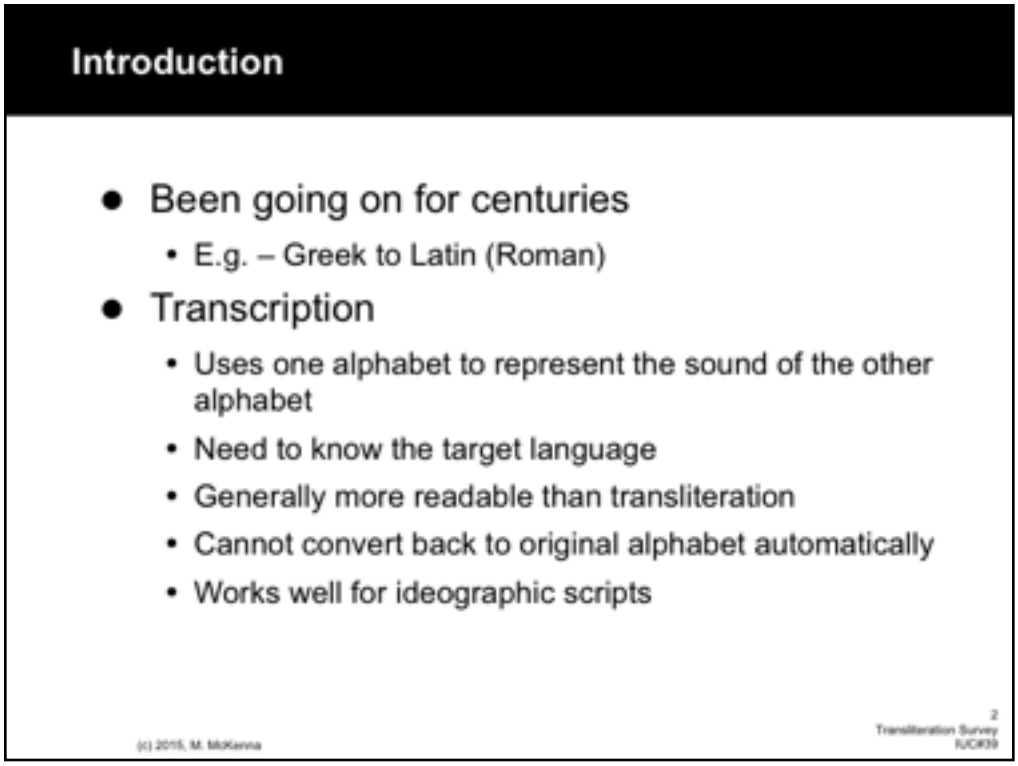
Transliteration Survey



Abstract

In the expanding European Union, a majority of people speak at least two languages. In the United States, English is becoming a second language to a larger and larger segment of the population. Even so, official documents, phone books, bibliographic records, and other digital repositories may need to present native language source information (e.g., Russian or Greek) in a transliterated form to allow it to be interpreted by someone who does not speak or read that language. This paper will present a survey of issues confronting the language engineer along with solutions and available technologies. We will look at historical standards and contexts in which the standards become useless. We will then look at early work funded by the Directorate General for Research of the European, then on to solutions provided by Java and software libraries. We'll finish by taking a look at some commercial universal names databases and the technology behind them.



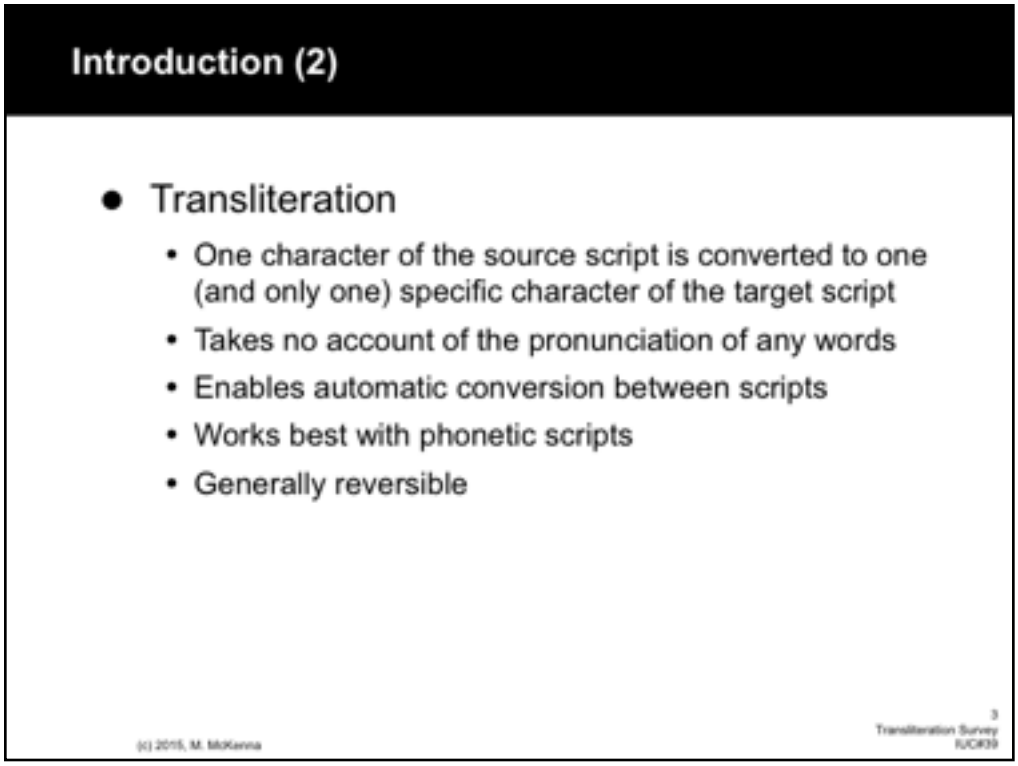
Introduction

Transliteration and transcription has been going on for centuries, even millennia. Classic examples are the transliteration of Greek *alpha* or *beta* to the Roman letters 'A' or 'B'. In South India, they have been moving from one Sanskrit-based language represented in an Indic alphabet to another for centuries.

For some background, we'll start with some definitions:

Transcription

Transcription is the process of using one alphabet to represent the sound of another alphabet. However, you need to know the source language, and you need to know the target language to match the phonemes correctly. Generally, transcription is perceived to be more readable than transliteration, but you cannot always convert back to original alphabet automatically. Transcription works well for ideographic scripts.



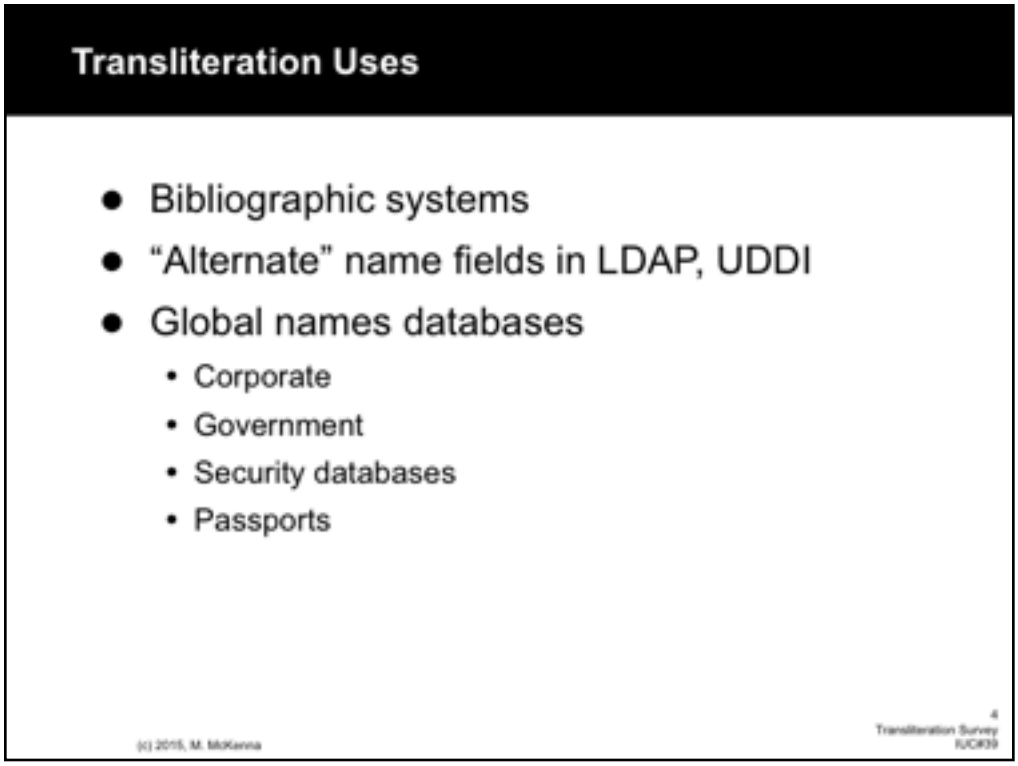
Transliteration

Transliteration is the process of taking one character of the source script and converting it to one (and only one) specific character of the target script. It takes no account of the pronunciation of any words. A key advantage to transliteration is that it enables automatic conversion between scripts. It works best with phonetic scripts such as Cyrillic, Greek, and Japanese kana. Transliteration, when done properly, is generally reversible, so you can get back the original source strings.

Transliteration Uses

Transliteration, as well as transcription, have several uses in the commercial and academic spheres. A few examples are highlighted below.

Transliteration Survey



Bibliographic systems

Libraries and bibliographic systems have relied on standardized transXtion (trans-{ litera | crip }-tion) for years to be able to store, catalog, and sort titles and authors of Russian, Greek, or Asian origin.

"Alternate" name fields in LDAP, UDDI

In Lightweight Directory Access Protocol (LDAP) or Universal Description, Discovery and Integration (UDDI), entities such as people, places, institutions, or applications are names and described. Local names should be entered and displayed in the local language and script. But remote applications or users may prefer to have a rendition in a script that is readable in their locale. Hence transcription or transliteration to Roman characters for global catalog access.

Global names databases

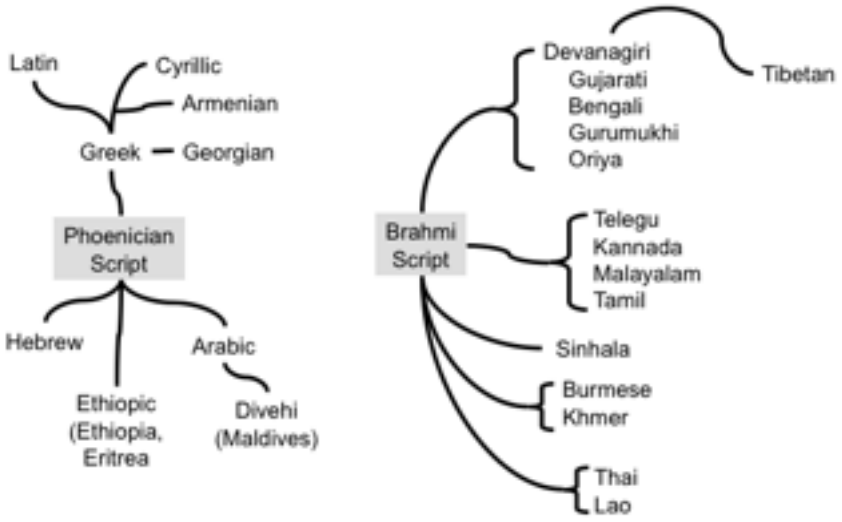
In the same way, other databases that store names for use in global environments may have a need for transXtion. Examples are

• Corporate – global corporations may need to have employee or distributor information globally accessible to speakers of various languages. 24x7 support databases may need to be able to read names of customers or even of other support personnel.

• Government – many governments must deal with names of persons and geographic places in multiple languages. By using standardized mechanisms, they can be presented in an intelligible manner that makes them accessible to readers on Latin or other scripts.

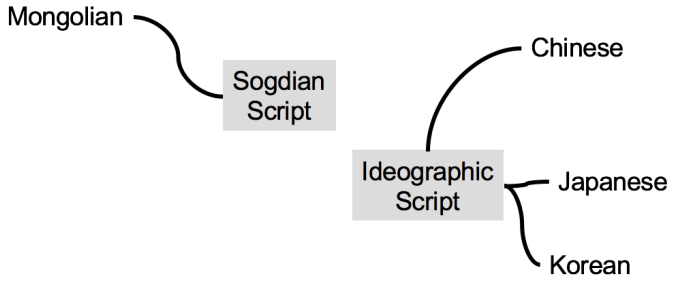
• Security databases and Passports – this is a fun area related to "Homeland Security" where people whose native name is in an Arabic or Asian script could have their name transliterated into any of several different permutations. If used inconsistently, a suspect name can easily pass through one system if transliterated by another.

Historical Script Families



5
Transliteration Survey
IUC#39

Historical Script Families (2)



6
Transliteration Survey
IUC#39

Transliteration Survey

Historical Script Families

From John Clews of the SESAME Project, we get a simple and concise overview of the etiology of various script families. Please note that there has been decades of research in this area, and many differing opinions. They are presented according to Clews views because these were used in the initial arguments to establish a set of ISO transliteration standards.

Phonecian

The Phoenician alphabet dates from around 1000 BC and is derived from the Proto-Canaanite alphabet. From Phoenician we get most of our Western phonetic alphabets, both printed and cursive. Modern alphabets thought to have descended from the Phoenician include Hebrew, Arabic, Greek, and Latin. Others include:

- Ethiopic
- Divehi
- Georgian
- Cyrillic
- Armenian

Brahmi

Brāhmī refers to the pre-modern members of the Brahmic family of scripts, attested from the 3rd century BC. The best known and earliest dated inscriptions in Brahmi are the rock-cut edicts of Ashoka. This script is ancestral to most of the scripts of India and Southeast Asia, Tibet, and perhaps even Korean Hangul. The Brahmi numeral system is the ancestor of the Hindu-Arabic numerals, which are now used world-wide.

Sogdian

Sogdian is an extinct Middle Iranian language, known chiefly from texts and inscriptions dating from the second to the ninth centuries A.D. Modern Mongolian is believed to be derived from Sogdian, although Michael Everson may beg to differ.

Ideographic

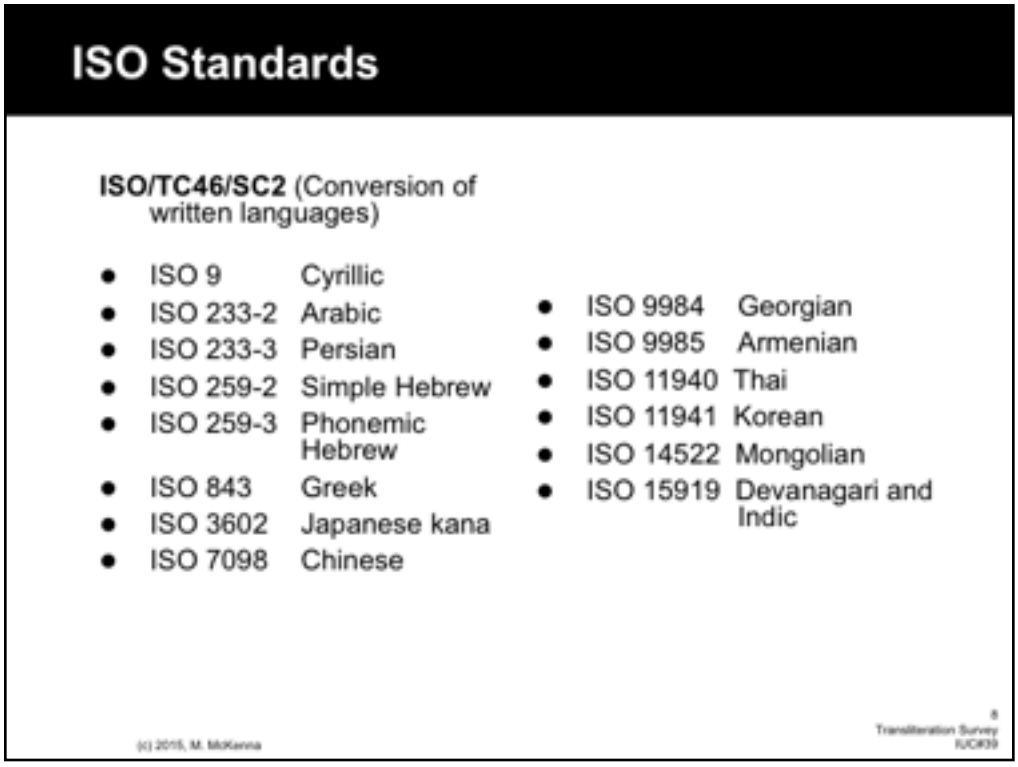
Ideographs (from Greek *ιδεα idea* "idea" + *γραφω grapho* "to write") are said to be graphical symbols that represent words or morphemes. They are composed of visual elements arranged in a variety of ways, rather than using the segmental phoneme principle of construction used in alphabetic languages. The effect is that while it is relatively easier to remember or guess the sound of alphabetic written words, it is relatively easier to remember or guess the meaning of ideographs. The other feature of ideographs is that they may be used by a plurality of languages which may pronounce them differently while using them in conformity to the same norms.

In internationalization and localization communities, we think of the acronym “CJK” to represent the major ideographic languages Chinese, Japanese, and Korean. The be more correct, it is often referred to as “CJKV” to include Vietnamese, which has extensive use of Chinese characters.

Transliteration Standards

There are multiple transliteration standards from international, national, and proprietary organizations. Many of them are misnamed and should, instead, be termed transcription standards, since they are keyed to specific source and target *languages*, not scripts.

The major sources of transliteration standards are from ISO, the United Nations Group of Experts of Geographic Names (UNGEGN), and the American Library Association, in conjunction with the Library of Congress (ALA/LC). In addition to these sources, there are mappings from British Standards (BS), Japanese, Russian, and others.



ISO Standards

The ISO transliteration standards have been designed by ISO/TC46/SC 2 – Technical Committee 46 (Information and Documentation), Subcommittee 2 (Conversion of Written Languages). The subcommittee is not currently active, as their work is considered complete for now.

The ISO standards relating to transliteration all map from a source script to Latin. They are not always reversible transformations, however. The ISO standards are:

ISO 9	Cyrillic
ISO 233-2	Arabic
ISO 233-3	Persian
ISO 259-2	Simple Hebrew
ISO 259-3	Phonemic Hebrew
ISO 843	Greek
ISO 3602	Japanese kana
ISO 7098	Chinese
ISO 9984	Georgian
ISO 9985	Armenian
ISO 11940	Thai
ISO 11941	Korean
ISO 14522	Mongolian
ISO 15919	Devanagari and Indic

Transliteration Survey

American Library Association / Library of Congress Standards

The two most comprehensive sets of Romanization standards are from the ALA-Library of Congress (LC) and the United Nations. These are best characterized as transcription standards as they are dependent of the source language, with Romanization assuming an English speaking reader. For instance, for the Cyrillic script, the ALA-LC standards support separate conversions for Russian, Church Slavic, Georgian, Ukranian, and representations of Non-Slavic languages in Cyrillic.

The transcription tables used are based primarily of the 1997 edition of the *ALA-LC Romanization Tables: Transliteration Schemes for Non-Roman Scripts*, approved by the Library of Congress and the American Library Association, with the following exceptions:

The table for Chinese is a revised table reflecting the Library's conversion to Pinyin romanization in 2000.

The table for Kurdish is a revised table that replaces both the Kurdish (in Perso-Arabic Script) table (p. 114-155) and the Kurdish (1946) section of the Non-Slavic Languages (in Cyrillic Script) table (p. 148)

The tables can be found at: http://www.loc.gov/catdir/cpso/roman.html

Transliteration Survey

United Nations Standards

The United Nations Working Group on Romanization Systems states:

> "Consistent use of accurate place names is an essential element of effective communication worldwide and supports socio-economic development, conservation and national infrastructure. That is why the United Nations established a Group of Experts on Geographical Names (UNGEGN). UNGEGN promotes consistent use of accurate place names and prepares documents for the United Nations Conferences on the Standardization of Geographical Names that are held every five years. …
>
> "Working groups are established by UNGEGN to deal with specific tasks, such as training courses in toponymy, toponymic data files and gazetteers, toponymic terminology, country names, publicity and funding, and romanization systems."

To date, there have been eight UN Conferences on the standardization of geographic names. Primary sources have come from the U.S. Board on Geographic Names (BGN), the Permanent Committee on Geographic Names for British Official Use (PCGN), and the *Institut Géographique National* in France (IGN). Various other national or regional standards are referred to from time to time, depending on the source languages and scripts being considered.

The UNGEGN group considers whether the romanization system is based on sound scientific principles, the system's degree of reversibility, and the extent of its implementation on cartographic products (maps and charts) by the proposing country. As a result ISO transliteration schemes are generally not used.

Transliteration Survey

UNGEGN Romanization systems

Languages/scripts covered by systems recommended by the United Nations

Amharic | Arabic | Assamese | Bengali | Bulgarian | Chinese | Greek | Gujarati | Hebrew | Hindi | Kannada | Khmer | Macedonian Cyrillic | Malayalam | Marathi | Mongolian (in China) | Nepali | Oriya | Persian | Punjabi | Russian | Serbian | Tamil | Telugu | Thai* | Tibetan | Uighur | Urdu

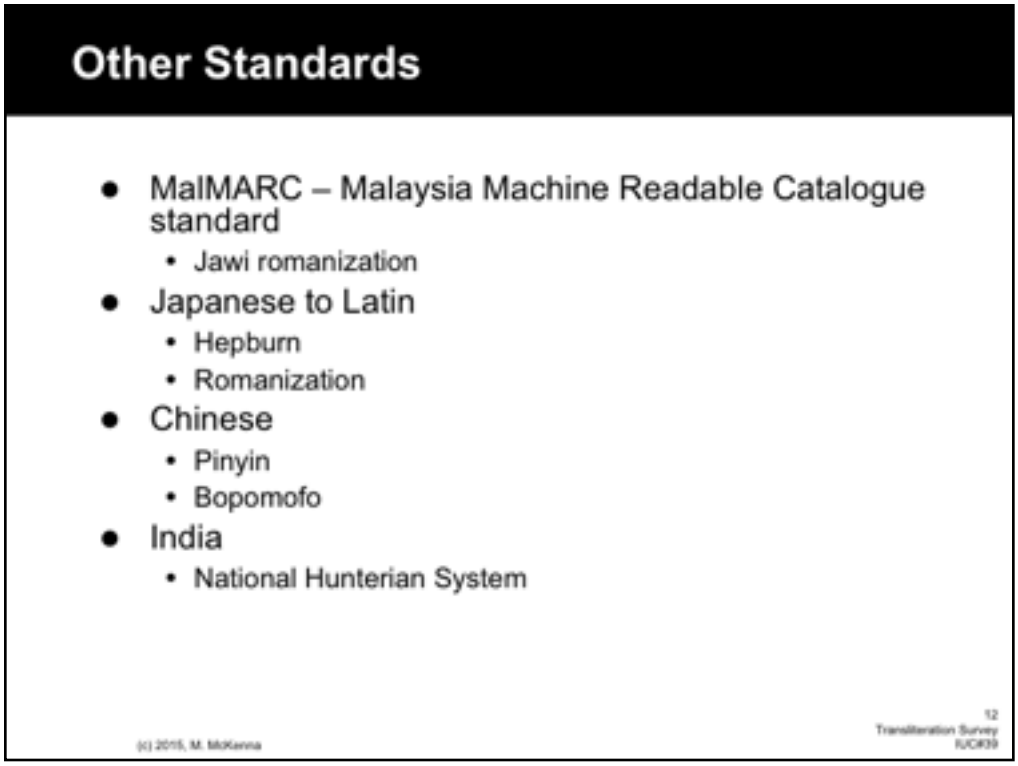
Other languages/scripts

Armenian | Burmese | Byelorussian | Dzongkha | Georgian | Japanese | Kazakh | Kirghiz | Korean | Laotian | Maldivian | Mongolian (Cyrillic) | Pashto | Sinhalese | Tajik | Tigrinya | Ukrainian |

Annex. Languages that have recently adopted Roman alphabets

Azerbaijani | Turkmen | Uzbek

* Note: Following comments from the Royal Thai Survey Department in May 2003, this section was updated and published, with two errors corrected, on 5 May, 2004.



Other Standards

There are several other standards and conventions in use, many converting from various scripts or languages to other non-Latin scripts and non-English languages. A small smattering of examples are included here.

MalMARC

The Malaysian Machine Readable Catalogue romanization standard was created to convert Jawi script to Latin letters for library cataloging purposes. Jawi script originated from Arabic with particular adaptations and additions. It was introduced into the Malay world, especially the Malay Peninsular soon after the arrival of Islam. There are differences in opinion of the exact date, probably as early as 440 H (1104 A.D.). However it is believed that the Arabic script was adopted into Jawi script after 7 Hijrah/671 A.D. Information on Jawi Romaization can be found at: http://www.ifla.org/IV/ifla65/papers/150-155e.htm

Japanese Romanization

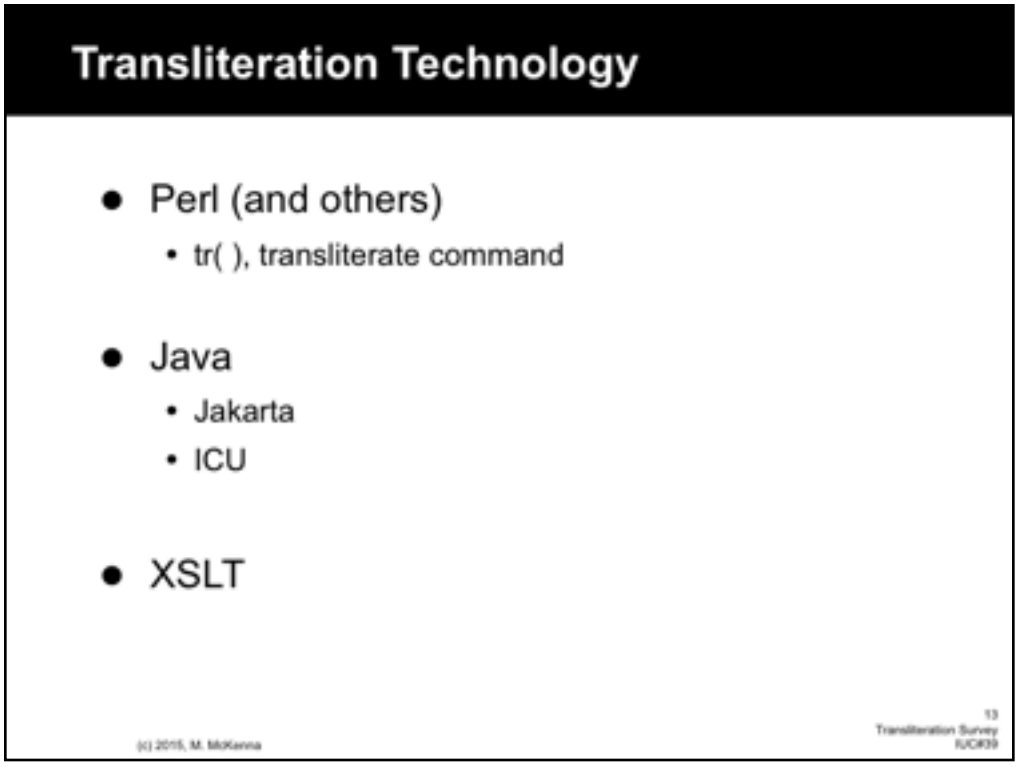
(from: http://www.glocom.org/tech_reviews/jt_review/20020212_s33/)

Even government entities in the same country sometimes cannot agree.

> "The Roman alphabet indication, the Hepburn and romanization systems exist, and the domestic indication system is not standardized. There is confusion as for whether to use Mt. Fuji or Fujisan, and ABC Avenue or ABC Dori. One reason is the fact that Ministry of Foreign Affairs uses the Hepburn system and Ministry of Education, Culture, Sports, Science and Technology uses romanization system. The Geographical Survey Institute uses Fujisan and Ministry of Land, Infrastructure and Transport uses Mt. Fuji on the map. "

Others

Other systems of note are Pinyin and Bopomofo for Chinese, and the National Hunterian System in India used to convert between various Indic scripts.

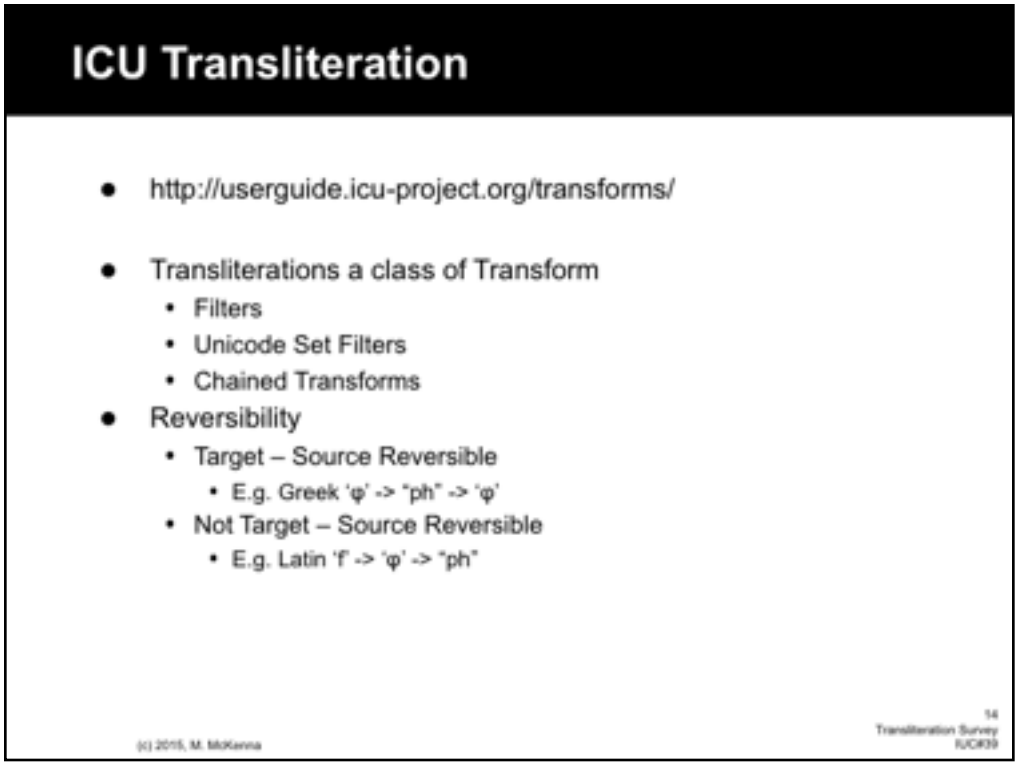


Transliteration Technology

Most all programming and scripting languages provide at least a rudimentary mechanism to achieve letter-for-letter transliteration. A classic example is the Perl *tr( )* "transliterate" function. Java provides basic transliteration capabilities in the Jakarta extensions.

Of course, we must not forget XSLT, which is used to transform XML data elements. XSLT can be written in such a way to transliterate text strings, although it would be easier, and more powerful to integrate with another system, such as Java or .NET.

The discussion following will concentrate on the International Components for Unicode (ICU) which now has a robust transliteration engine that can be used for both reversible true script-to-script transliterations as well as language to language transcriptions.



ICU Transliteration

The International Components for Unicode (ICU) were first derived as a set of Unicode-enabled classes and methods for Java. But, since Java had not yet been fully developed, they were prototyped in C++. The original development team was acquired by IBM and the code expanded its functionality beyond the core Java offerings. It was converted to Open Source, and now has contributors from all over the World. The library is available from

http://site.icu-project.org/ as both C++ and Java libraries

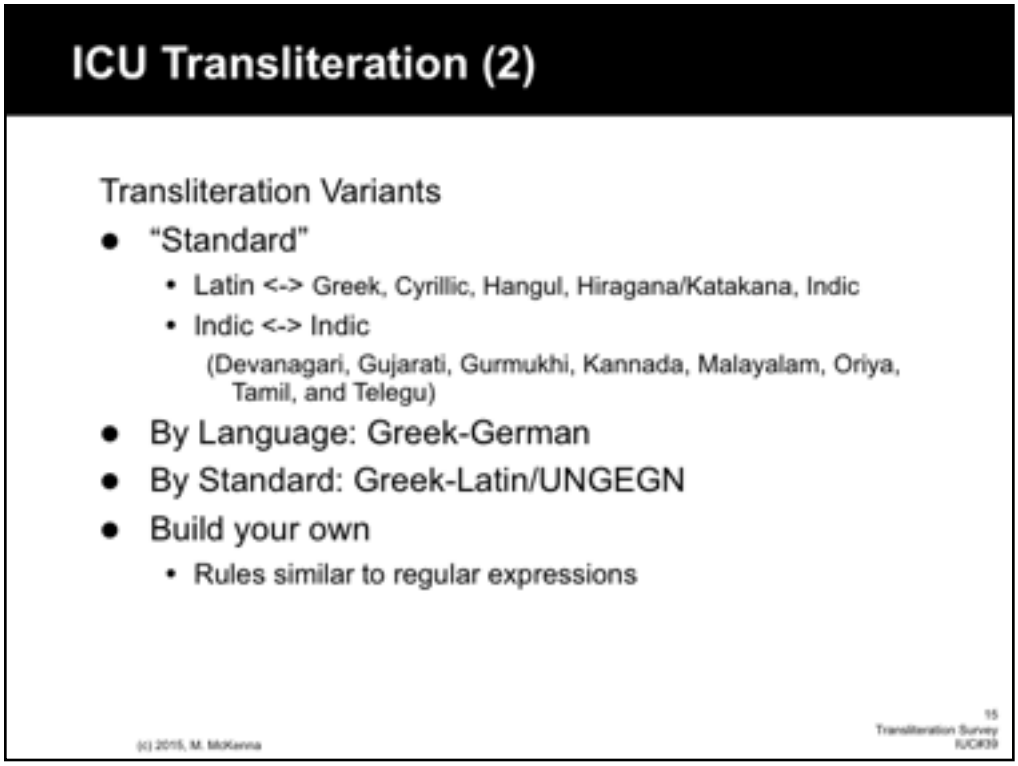
Transliterations

In ICU, transliterations are an extension of transform classes. By creating context or locale-sensitive filters or using the set of supplied Unicode filters, along with chained transforms, one can create complex and robust transliteration or transcription methods.

In ICU, transliteration is a more flexible mechanism that has pre-built transformations for case conversions, normalization conversions, the removal of given characters, and also for a variety of language and script transliterations. Transliterations can be chained together to perform a series of operations and each step of the process can use a UnicodeSet to restrict the characters that are affected. Most natural language transliterators (such as Greek-Latin) are written a rule-based transliterators. Transliterators can be written as text files using a simple language that is similar to regular expression syntax.

Note that transliteration from random Latin characters to other scripts may not be reversible, e.g., 'f' -> Greek *phi* which would be transliterated back to Latin 'ph'.

Complete and exact reversal **is** possible if the transform has been explicitly designed to support this. Examples of transforms that support this are "Any-Hex" and "SCRIPT-Latin", where SCRIPT is a supported transliteration script, e.g. Greek-Latin. The "SCRIPT-Latin" transforms support exact reversal of well-formed text in SCRIPT to Latin (via "SCRIPT-Latin") and back to SCRIPT (via "Latin-SCRIPT"). This is called **round-trip integrity**. They do not, however, support round-trip integrity from Latin to SCRIPT and back to Latin.



Guidelines

• complete: every well-formed sequence of characters in the source script should transliterate to a sequence of characters from the target script.

• predictable: the letters themselves (without any knowledge of the languages written in that script) should be sufficient for the transliteration, based on a relatively small number of rules. This allows the transliteration to be performed mechanically.

• pronounceable: transliteration is not as useful if the process simply maps the characters without any regard to their pronunciation. Simply mapping "αβγδεζηθ…" to "abcdefgh…" would yield strings that might be complete and unambiguous, but cannot be pronounced.

• unambiguous: it is always possible to recover the text in the source script from the transliteration in the target script.

Standard Transliterations

The built-in script transforms in ICU are:

Latin <-> Greek, Cyrillic, Hangul, Hiragana, Katakana, Indic

Indic <-> Indic

Indic includes Devanagari, Gujarati, Gurmukhi, Kannada, Malayalam, Oriya, Tamil, and Telegu. ICU can transliterate from Latin to any of these dialects and back, and from Indic script to any other Indic script. For example, you can transliterate from Kannada to Gujarati, or from Latin to Oriya.

Transliteration Variants

Transcriptions can be built to go from language-to-language, with transforms able to be chosen by Script, Language, or Standard.

Or, you can build your own compound transforms to accomplish very specific mappings.

Transliteration Survey

Supported ICU Script-to-Script Transliterations

Any-Arabic
Any-Bengali
Any-Cyrillic
Any-Devanagari
Any-Greek
Any-Greek/UNGEGN
Any-Gujarati
Any-Gurmukhi
Any-Han
Any-Hangul
Any-Hebrew
Any-Hiragana
Any-Kannada
Any-Katakana
Any-Latin
Any-Malayalam
Any-Oriya
Any-Tamil
Any-Telugu
Any-Thai

Arabic-Latin, Latin-Arabic

Bengali-Devanagari, Devanagari-Bengali
Bengali-Gujarati, Gujarati-Bengali
Bengali-Gurmukhi, Gurmukhi-Bengali
Bengali-Kannada, Kannada-Bengali
Bengali-Latin, Latin-Bengali
Bengali-Malayalam, Malayalam-Bengali
Bengali-Oriya, Oriya-Bengali
Bengali-Tamil, Tamil-Bengali
Bengali-Telugu, Telugu-Bengali

Cyrillic-Latin, Latin-Cyrillic

Devanagari-Gujarati, Gujarati-Devanagari
Devanagari-Gurmukhi, Gurmukhi-Devanagari
Devanagari-Kannada, Kannada-Devanagari
Devanagari-Latin, Latin-Devanagari
Devanagari-Malayalam, Malayalam-Devanagari
Devanagari-Oriya, Oriya-Devanagari
Devanagari-Tamil, Tamil-Devanagari
Devanagari-Telugu, Telugu-Devanagari

Fullwidth-Halfwidth, Halfwidth-Fullwidth

Greek-Latin, Latin-Greek
Greek-Latin/UNGEGN, Latin-Greek/UNGEGN

Gujarati-Gurmukhi, Gurmukhi-Gujarati
Gujarati-Kannada, Kannada-Gujarati
Gujarati-Latin, Latin-Gujarati
Gujarati-Malayalam, Malayalam-Gujarati
Gujarati-Oriya, Oriya-Gujarati
Gujarati-Tamil, Tamil-Gujarati
Gujarati-Telugu, Telugu-Gujarati

Gurmukhi-Kannada, Kannada-Gurmukhi
Gurmukhi-Latin, Latin-Gurmukhi
Gurmukhi-Malayalam, Malayalam-Gurmukhi
Gurmukhi-Oriya, Oriya-Gurmukhi
Gurmukhi-Tamil, Tamil-Gurmukhi
Gurmukhi-Telugu, Telugu-Gurmukhi

Han-Latin, Latin-Han

Hangul-Latin, Latin-Hangul

Hebrew-Latin, Latin-Hebrew

Hiragana-Katakana, Katakana-Hiragana
Hiragana-Latin, Latin-Hiragana

Jamo-Latin, Latin-Jamo

Kannada-Latin, Latin-Kannada
Kannada-Malayalam, Malayalam-Kannada
Kannada-Oriya, Oriya-Kannada
Kannada-Tamil, Tamil-Kannada
Kannada-Telugu, Telugu-Kannada

Katakana-Latin, Latin-Katakana

Latin-Malayalam, Malayalam-Latin
Latin-Oriya, Oriya-Latin
Latin-Tamil, Tamil-Latin
Latin-Telugu, Telugu-Latin
Latin-Thai, Thai-Latin.

Malayalam-Oriya, Oriya-Malayalam
Malayalam-Tamil, Tamil-Malayalam
Malayalam-Telugu, Telugu-Malayalam

Oriya-Tamil, Tamil-Oriya
Oriya-Telugu, Telugu-Oriya

Tamil-Telugu, Telugu-Tamil

Fallback mechanism

The ICU transform methods can massage text data based on language pairs, if the appropriate transforms are registered with the system. ICU may supply transliterations that are specific to language pairs, or between a language and a script. For example, ICU could have a ru-en (Russian-English) transform.

As with locales, there is a fallback mechanism. If the Russian-English transform is requested and is not available, then ICU will search for a Russian-Latin transform. If the Russian-Latin transform is not available, ICU will search for a Cyrillic-Latin transliteration.

Transliteration Survey

ICU Script Transliteration Sources or Standards

Currently ICU offers script transliterations between Latin and certain other scripts (such script transliterations are called romanizations), plus transliterations between the Indic scripts (excluding Urdu). Additional romanizations and other script transliterations will be added in the future. In general, ICU follows the UNGEGN: Working Group on Romanization Systems where possible.

Even using standards such as UNGEGN and those below, most had to be tweaked to fill holes and make as reversible as possible.

Korean

There are many romanizations of Korean. The default transliteration follows the Korean Ministry of Culture & Tourism Transliteration regulations using a specific clause variant for reversibility.

Japanese

The default transliteration for Japanese uses the a slight variant of the Hepburn system. With Hepburn system, both ZI (ジ) and DI (ヂ) are represented by "ji" and both ZU (ズ) and DU (ヅ) are represented by "zu". This is amended slightly for reversibility by using "dji" for DI and "dzu" for DU.

The Katakana transliteration is reversible. Hiragana-Katakana transliteration is not completely reversible since there are several Katakana letters that do not have corresponding Hiragana equivalents. Also, the length mark is not used with Hiragana. The Hiragana-Latin transliteration is also not reversible since internally it is a combination of Katakana-Hiragana and Hiragana-Latin.

Greek

The default transliteration uses a standard transcription for Greek. The transliterations is one that is aimed at preserving etymology. There is an ISO 843 variant with minor differences.

Cyrillic

Cyrillic generally follows ISO 9 for the base Cyrillic set. There are tentative plans to add extended Cyrillic characters in the future, plus variants for GOST and other national standards.

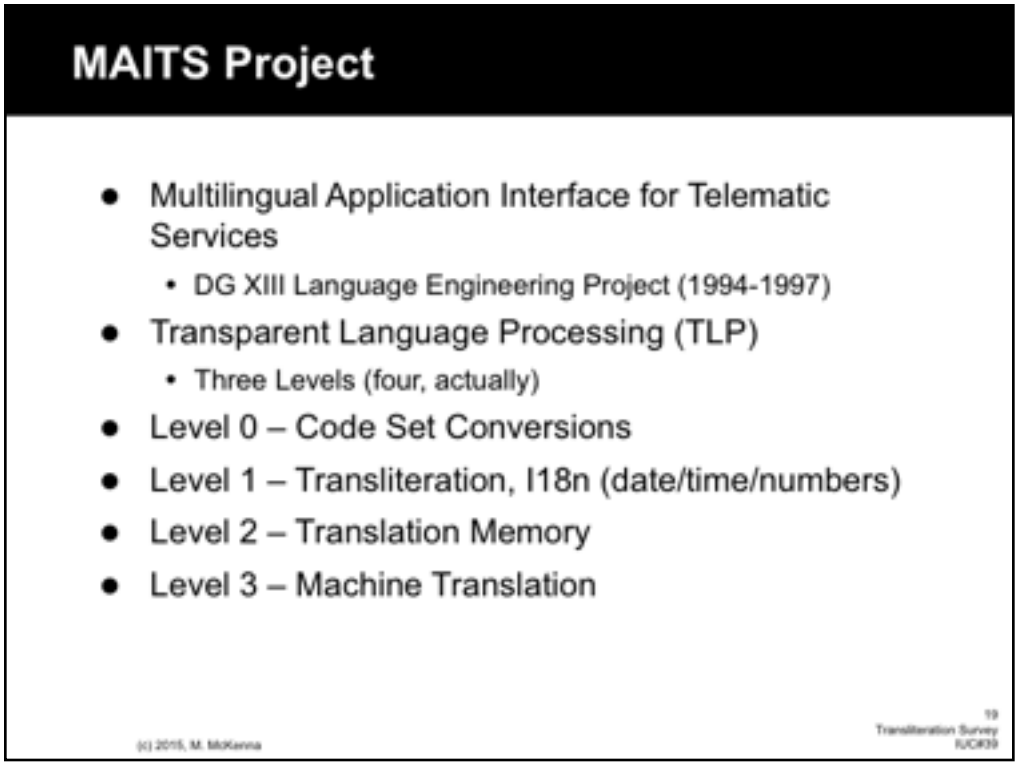
Indic

The default romanization uses the ISCII standard with some minor modifications for reversibility. Internally, all Indic scripts are transliterated by converting first to an internal form, called Interindic, then from Interindic to the target script.

Transliteration of Indic scripts in ICU follows the ISO 15919 standard for Romanization of Indic scripts using diacritics. Internally, all Indic scripts are transliterated by converting first to an internal form, called Inter-Indic, then from Inter-Indic to the target script. ISO 15919 differs from ISCII 91 in application of diacritics for certain characters.

Transliteration rules in Indic are reversible with the exception of the ZWJ and ZWNJ used to request explicit rendering effects.

There are two particular instances where transliterations may produce unexpected results: (1) where a halant after a consonant is implied by the romanization (in such cases the vowel needs to be explicitly written out), and (2) with the transliteration of 'c‘, which could be interpreted as ‘k’.



The MAITS Project

A silent precursor to the robust transform and transliterations provided by ICU was a dream of the MAITS project, a European Union funded language engineering project from the mid-1990's.

MAITS was conceived as a project for developing an Applications Program Interface (API) to support access to software-based translation services which could be integrated into any network service application. The API was intended for uses such as messaging, and to provide in-line multilingual capability.

A simple framework was developed by the MAITS consortium to break down the task of Transparent Language Processing (TLP) into four levels. Levels range from level 0, which is considered the bare minimum for non-English use of networked services, to level 3, which requires access to online machine-aided translation services:

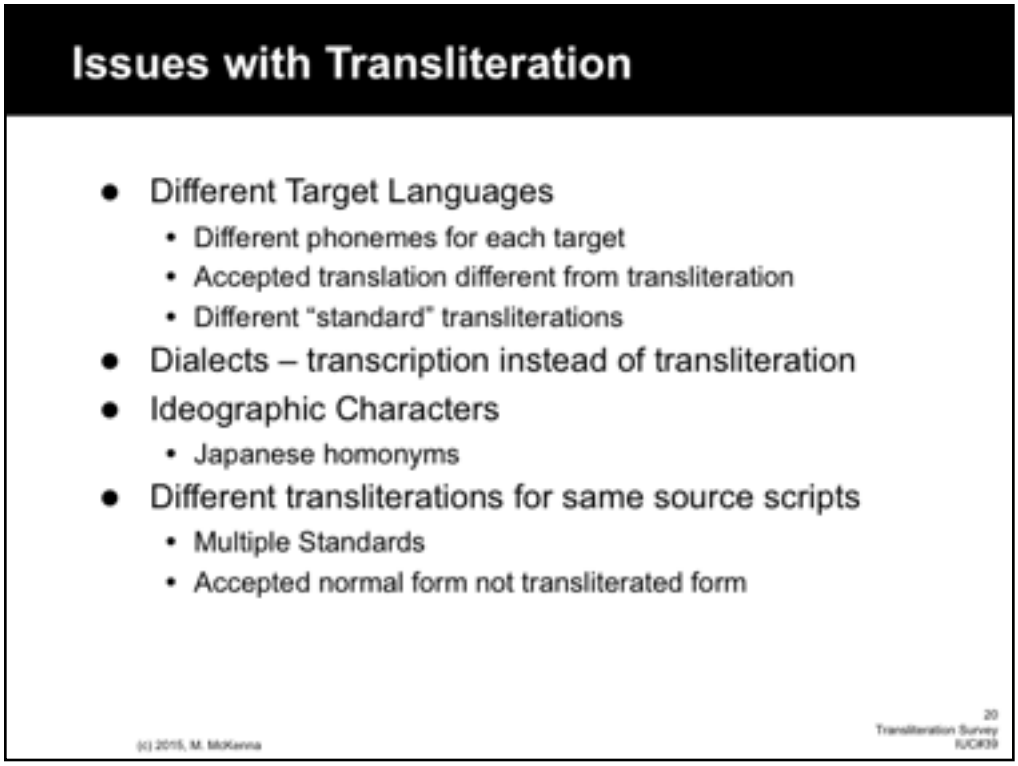
• **level 3 Machine Translation**: to be able to access textual information in one's native language even if it is written in another;

• **level 2 Translation Memory**: provides a mechanism where messages, attributes and phrases can be translated from one language to another using pre-packaged modules;

• **level 1 Transliteration and Internationalization**: enables the user to read text that was originally in a different writing system or script, and to have culturally expected formatting of such items as date, time and numbers;

• **level 0 Code Set Conversion**: converts human-readable text into a format that the user's operating environment can understand.

The prototype of the API supporting UNICODE, a demonstration of the API in the Apache Web server, and a friendly keyboard mapping utility were delivered to a European research agency, ELRA, for distribution in 1997.



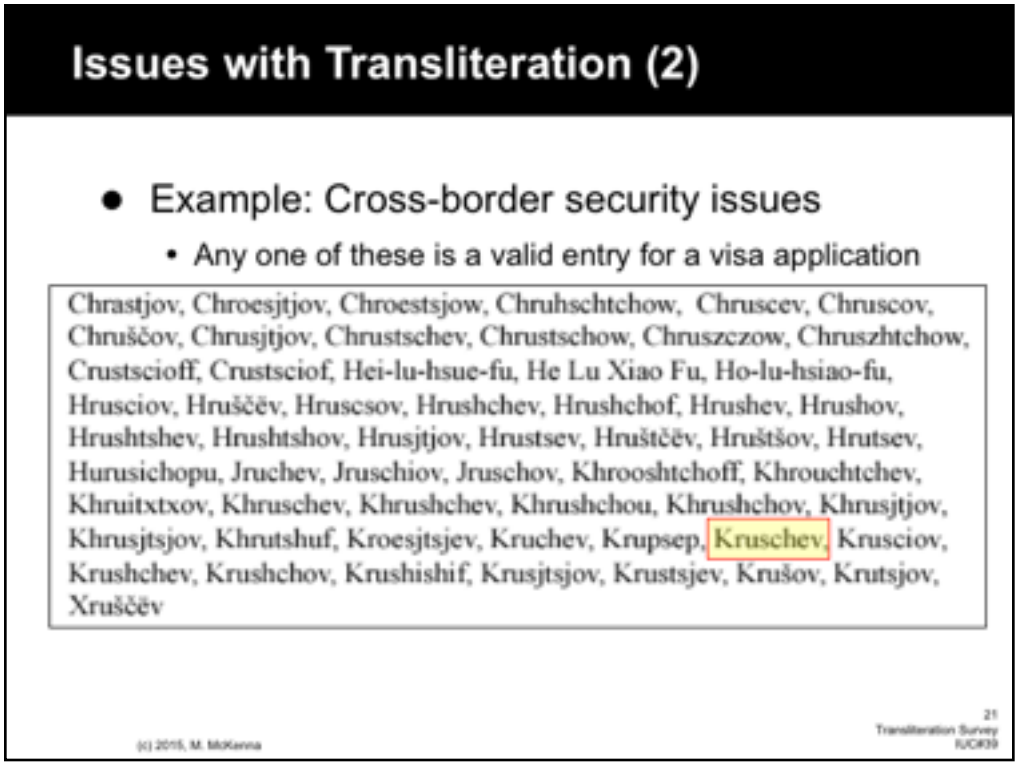
Issues with Transliteration

One of the issues with transcription is that, even with Latin, you have different target languages, with different expectations of the pronunciation of character sequences. With true transliteration, the transformation is merely from script to script. But the transliteration itself is oftentimes biased towards an English speaking reader and English phonetic conventions for interpreting the resulting Latin character string. The transliteration may be very different from the accepted translation. For instance, the Russian composer *"Tchaikovsky"* gets transliterated from the original Cyrillic as "*Chaĭkovskiĭ"* using the Library of Congress scheme. And *"Tchaikovsky"* is still not correct colloquially accepted version for German, Polish, Spanish and several other languages. At least with transliteration, you end up with a single readable string that is potentially wrong for everybody, so everyone is treated alike.

Asian ideographic characters need dictionary lookup for proper conversion. In Japanese, you may have multiple homonyms that would befuddle reverse transliteration into kanji. If a context sensitive dictionary lookup is made, why not just use the dictionary results? The transliteration from kana and hangul to and from Latin is fairly well defined and gives reasonable results.

Even with transliteration standards, there are multiple transliteration standards that may be used, so you may have several possible results for the same original source string. The discussion on the next slide illustrates this point and the possible dire consequences.

Transliteration Survey



Gun Ban Expiration Renews Concern Over Aged Systems to Check Criminal Backgrounds, Air Passenger Names Against No-Fly List

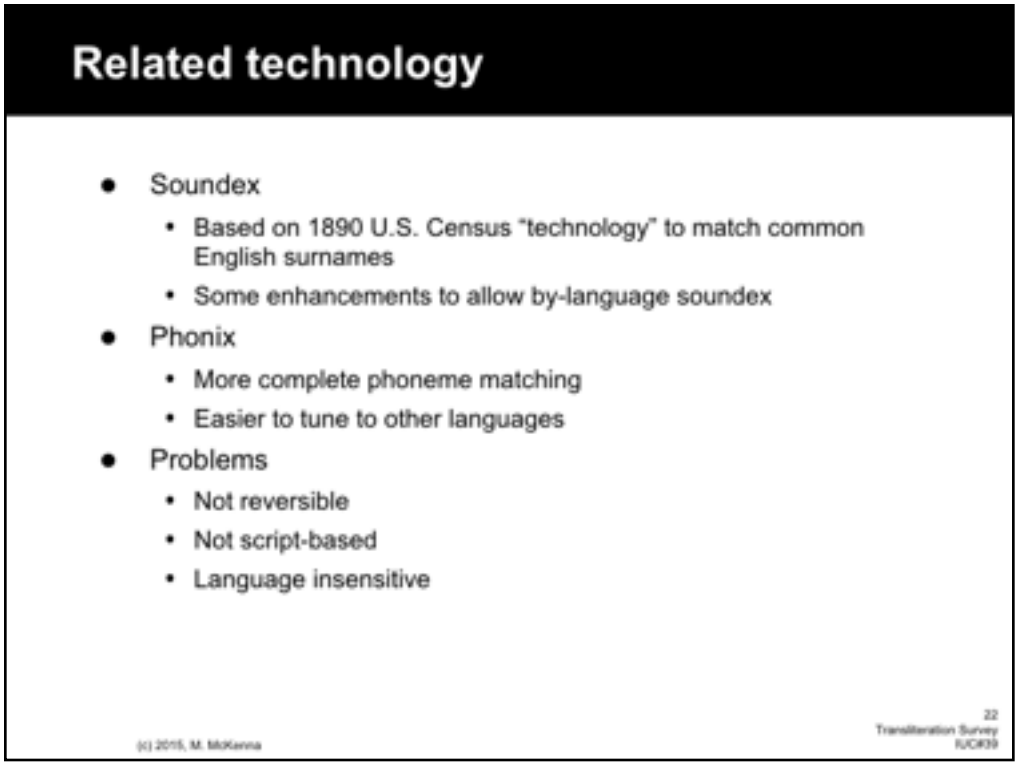
By Anthony L. Kimery

Senior Correspondent

WASHINGTON, DC, SEPT. 24, 2004

http://www.hstoday.net/HSTKimeryReport.htm

> Indeed. The final report of the 9/11 Commission emphasized that "among the more important problems to address is that of varying transliterations of the same name. For example, the current lack of a single convention for transliterating Arabic names enabled the 19 hijackers to vary the spelling of their names to defeat name-based watch list systems and confuse any potential efforts to locate them." Consequently, the Commission said the long-standing hole in our border security caused by the US government's ineffective name-handling software must be addressed if the nation is to have an effective border security program at all.
>
> …
>
> "The [9/11] Commission hits the nail on the head by identifying the lack of a single transliteration method for names," Hermansen told *HSToday*. "There will never be a single standard for name transliteration. Technology aside, it's just too political. There is no country I can think of that would allow us to dictate how their names must be Romanized. And, just within the US, it is very unlikely that such a standard could be forced on Americans with Romanized names. Even in Communist China, where such things are much more easily mandated, we have seen over a dozen ways in which the name Osama Bin Laden has been written in Chinese. Everyone, everywhere, would like to make this problem more tractable, but it is simply too complex for a solution by edict.
>
> The only effective and useful approach is to use linguistically-smart software that understands all of these transliteration standards at once."



Related Technology

There are several other related technologies to transliteration that can be used to create searchable strings that can be easily indexed. Soundex and Phonix are two phonetic based transformation algorithms that attempt to remove some of the spelling ambiguities in homonyms from single languages. The results are not necessarily user friendly for human readers, but they do aid in retrieving names that sound similar. To a degree.

Soundex

Soundex was created in 1890 as a means of filing and indexing English surnames from the U.S. Census. It maps Latin letters to soundex values so that strings such as "Smith" and "Smythe" end up in the same bucket. It has been implemented in various programming languages and databases basically unchanged from its 1890 debut. There have been a few attempts to tune it to various Western European Latin based languages such as French and German. There have been a few patents filed that transliterate from other scripts to Latin, then create a soundex value. But, since it is phonetically based, the transliterations are not too accurate.

Phonix

Phonix takes Soundex a step further to account for multi-letter sequences, such as "ph", "th", "st", etc. and is more easily tuned to other languages by assigning phonetic weighting factors.

The problems with Soundex and Phonix with respect to transliteration is that they are not script based, are not reversible, and are pretty much language insensitive.

Transliteration Survey

Language Engineering

Language Engineering is a field that concentrates on theory and technology related to manipulating and analyzing linguistic streams. It encompasses everything from speech detection and analysis to machine translation (MT). One company of note that seems to be mentioned on a regular basis in Homeland Security circles is *Language Analysis Systems*. They seem to have tackled the cross-language transliteration issues for name recognition quite well and seem to be making in-roads in getting contracts with the U.S. Government in this area.

As for Language Engineering, areas that are of particular concern with relation to transliteration would fall into two categories, understandability and machine matching.

Understandability

To make text understandable to speakers of other languages, language engineering uses a number of different disciplines, including

- Source-Target language context – this, of course considers what language the information originated from and what language the information will be interpreted in. This context helps to set up all the following.
- Translation Memory – this uses "fuzzy" matching to match up "translation units" from text, broken into segments by punctuation to previously translated text.
- Linguistic analysis – for free text strings, analyzing the parts of speech, the context of the words, etc.
- Dictionary look-up – the old stand-by, looking up word for word, or phrase by phrase, strings in a dictionary to aid in translation.
- Machine translation (MT) – does a complete grammatical and linguistic analysis, using several of the above techniques to recompose text streams into the target language. With proper training, MT can achieve over 95% accuracy. Without training, they can be as poor as 80%. --- would be like --- every fifth word (*That would be like missing every fifth word*).

Machine Matching

Another area of importance is machine matching of personal and geographic place names. Once they have been transliterated or transcribed to different scripts, it becomes quite an art to try and match two Latin strings for the same string in the original script. Areas that help out in this area include:

- Authority standards – standards exist for accepted translations of geographic place names and most famous or important dignitaries. These can be used to automatically translate names from passports and government documents.
- Reversibility – truly reversible transliteration standards could be used, in conjunction or alongside phonetic transcriptions of personal names to provide better matching of names.
- Unicode in "original" field – if the Unicode character stream, perhaps in hex format, of names and places were encoded in passports and government documents, it would aid in keeping track of who's who.

References

Language Analysis Systems, acquired by IBM
http://www-03.ibm.com/software/products/en/infosphere-global-name-management
Transliteration schemes used by European Libraries and Name Authority Issues, Project Helen, CEC Telematic Systems in Areas of General Interest - Libraries Programme, 1995
http://www.emeraldinsight.com/doi/abs/10.1108/eb047206
Clews, John. Digital Language Access : Scripts, Transliteration, and Computer Access. SESAME Computer Projects. D-Lib Magazine, March 1997
http://www.dlib.org/dlib/march97/sesame/03clews.html

References (2)

ALA/LC Transliteration Tables, Cataloging Policy and Support Office, Library of Congress, 2004-06-17
http://www.loc.gov/catdir/cpso/roman.html
United Nations Working Group on Romanization Systems,
http://unstats.un.org/unsd/geoinfo/UNGEGN/wg5.html
http://unstats.un.org/unsd/geoinfo/ungegnbrochure.htm
http://www.eki.ee/wgrs/rom1_2.pdf
http://www.eki.ee/wgrs/
Davis, Mark. Unicode Transforms in ICU, 21st International Unicode Conference. Dublin, Ireland, 2002-04-13.
http://www.macchiato.com
http://site.icu-project.org/

References

Language Analysis Systems, acquired by IBM:

http://www-03.ibm.com/software/products/en/infosphere-global-name-management

Transliteration schemes used by European Libraries and Name Authority Issues, Project Helen, CEC Telematic Systems in Areas of General Interest - Libraries Programme, 1995

http://www.emeraldinsight.com/doi/abs/10.1108/eb047206

Clews, John. Digital Language Access : Scripts, Transliteration, and Computer Access. SESAME Computer Projects. D-Lib Magazine, March 1997

http://www.dlib.org/dlib/march97/sesame/03clews.html

ALA/LC Transliteration Tables, Cataloging Policy and Support Office, Library of Congress, 6/17/2004

http://www.loc.gov/catdir/cpso/roman.html

Knight, K. and J. Graehl. 1997. Machine Transliteration. *Proceedings of the 35th Association of Computational Linguistics Conference*. Madrid, Spain, 1997, (128—135). Morgan Kaufmann