

Internationalisation and Localisation of the Internet

James Seng Ching Hong
Chief Technology Officer, i-DNS.net International

Mr Seng is also the Co-Chair of the Internationalised Domain Name Working Group, Internet Engineering Task Force (IETF).

Abstract:

The Internet is primarily an American invention. Therefore, it is natural that Internet Protocols have little or no consideration beyond the English language. Nevertheless, with over 60% of the world population being non-native English-users, it is necessary for the Internet to consider the use of non-English languages in its protocols in order for it to become a truly global network.

1. Introduction

When the Internet was gaining popularity five or six years ago, it was hailed as the “Global Village”. While the term “Global Village” is hardly used nowadays, it is still something worth pondering.

A village is a community of people within a region. The people living in the village have many similarities such as culture, language, religion, technology, lifestyle etc. Thus, a “Global Village” is, by this definition, a village that spans the whole world. In this case, we refer to a digital village bounded only by the network infrastructure. Every machine on the Internet uses the same technology and speaks the same protocol, namely Internet Protocol (IP).

But, alas, humans don't speak the same language.

Ever since humans attempted to build the Tower of Babel, we have been cursed and divided to speak a different language. And for many of us, we embrace the curse proudly, if not eagerly, for our language defines our culture and who we are.

To bridge the digital divide and facilitate universal Internet access, a more user-friendly Internet needs to be established for non-native English-users. The process to allow natives to use their own language is known as Internationalisation and Localisation.

2. Difficulties of I18N and L10N

Internationalisation (I18N) is a blanket term referring to the process of preparing software so that it can be used by more than one culture, region or locale. I18N is an acronym for "Internationalisation" ("I" + 18 letters + "N"). Localisation (L10N) is the process of adapting software to one specific culture, region or locale [2]. L10N is an acronym for "Localisation" ("L" + 10 letters + "N").

There is a common, widespread misunderstanding that I18N and L10N is basically a user interface problem. As long as the software is “8-bit clean”, doing localisation (or Internationalisation) is not a problem. Unfortunately, things are not as simple.

2.1. Coded Character Set (CCS)

A coded character set (CCS) is a set of unambiguous rules that establishes a character set and the relationship between the characters of the set and their coded representation. [9].

In doing L10N for Japanese (also known as Japanisation or J10N), we could use Japanese characters that are defined in JISx201, JISx208 and JISx212 CCS. Alternatively, we could use a universal character set, which contains all the characters used in (almost) all scripts, such as ISO10646, which is a flat 31-bits CCS.

The decision to use CCS is dependent very much on the requirements. In particular, the key difference between JIS and ISO10646 is that the former is a language-based CCS and the latter is a script-based CCS.

Some languages use more than one script. For example, the Japanese use the hiragana, the katakana and kanji (Han Ideograph). Some scripts are used by more than one language, e.g. Chinese, Japanese and Korean all use the Han Ideograph. [10]

ISO10646 also does CJK (Chinese-Japanese-Korean) Han Ideograph unification. This is to reduce the numbers of duplicated (or similar looking) Han Ideograph used across Chinese (*hanzi*), Japanese (*kanji*) or Korean (*hanja*) scripts by combining them into a single code-point.

Thus, by using ISO10646, you would lose some ability to differentiate languages from the codepoints. This gives rise to the need for out-of-band language tagging. On the other hand, by using localised CCS such as JIS for Japanese, you would end up using multiple CCS for different languages.

2.2. Character Encoding Scheme (CES)

A character encoding scheme (CES) is a character encoding form plus byte serialisation. [24]

Continuing with our example of J10N, if we decide to use JIS standard, there are multiple CES we could choose for encoding the CCS such as ISO2022-JP, EUC-JP or Shift-JIS. And if we decide to use ISO10646, there are also multiple CES we could use, such as UTF-8, UTF-16 or UTF-32.

Supporting one CES is not going to be sufficient in most cases. For example, in Japanese Windows 2000, the operating system is based on UTF-16 but some data stream and APIs are based on UTF-8 and legacy applications use Shift-JIS (Win95/98) or EUC-JP (Unix). Hence, transcoding between CES is necessary which involves usually large mapping tables.

2.3. Matching and Searching

In a typical English-based software, we could do matching by looking at $A = B$ (for case sensitive matching or $\text{upper}(A) = \text{upper}(B)$ for non-case sensitive matching). This would not work for I18N and L10N software.

For example, in ISO10646, there are two ways “å” can be represented, namely (i) composed form U+00E5; or (ii) decomposed form U+0061 U+030A.

Unicode defines an equivalency rule between characters in the Unicode Technical Report #15 [25] as illustrated in Figure 1. Basically, there are 4 normalisation forms, namely, Normalisation Form Composite (NFC), Normalisation Form Compatibility Composite (NFKC), Normalisation Form Decomposite (NFD) and Normalised Form Compatibility Decomposite (NFKD).

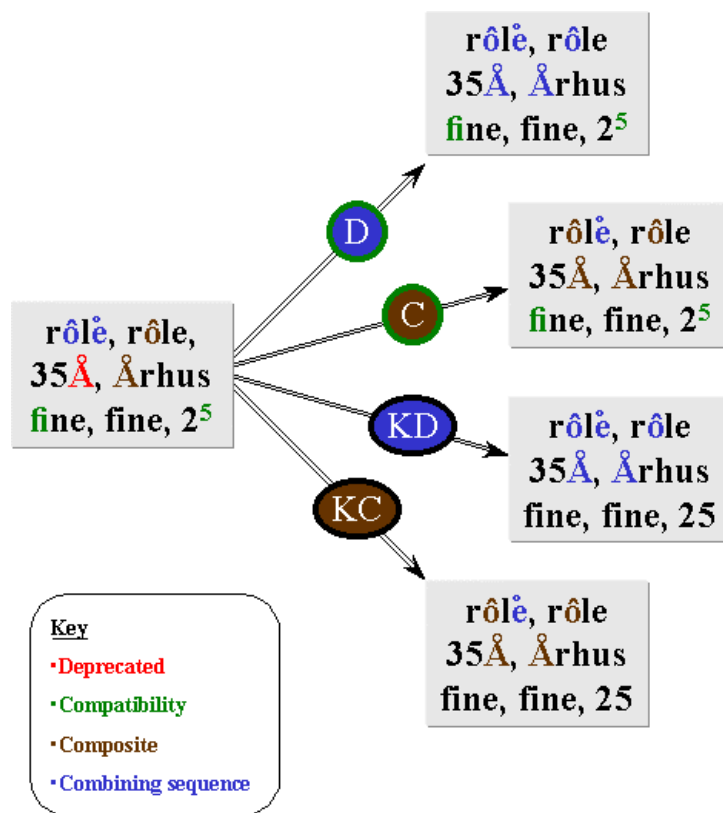


Figure 1. UNICODE equivalency rule between characters [25]

Matching could also be a strong match, a weak match or exact match. This would depend on the ability of the software in determining the normalisation forms to use. Normalisation forms may not suffice by themselves. For example, Chinese software which requires traditional and simplified Chinese equivalences-matching would require additional rules, making it into a new science of its own. Other languages like Hebrew or Arabic have vowels that are optional in normal use.

Searching in text adds more difficulties to the I18N problem. No longer it is possible to locate text boundaries of a phrase or sentence by looking for “space” characters. For example, Chinese sentence, written continuously without space, would require

“Morphological Analysis” to break up “? ? ? ? ” into lexemic component of “? ? ” (Taxi) and “? ? ” (Driver) before sub-string searching could be performed.

2.4. Sorting and Collating

Collating or sorting is the process of ordering units of textual information. [24]

Sorting order varies from culture to culture, and many specific applications require variations. Sort order can also be by word or sentence, case sensitive or insensitive, ignoring accents or not; it can also be either phonetic or based on the appearance of the character, such as ordering by stroke and radical for ideographs. [24]

As a result, it is neither possible to arrange characters in an encoding and in an order such that simple binary string comparison produces the desired sorted order nor is it possible to provide a single-level sort-weight table. This implies the sorting only has an indirect influence on a culturally expected sorting. [23]

2.5. Rendering Engine

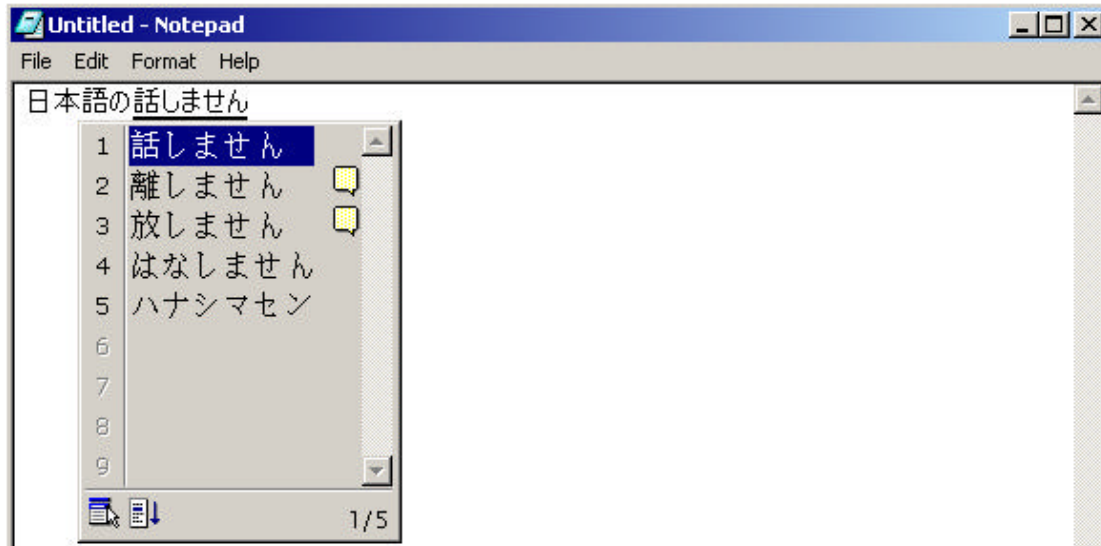
The rendering engine uses the fonts to display the characters to the users. Without a proper rendering engine, users would see a string of gibberish on the screen. It is important that the software understands the behaviour of the rendering and the font type used by the rendering engine.

There are rendering bidi, or bi-direction, languages like Arabic and Hebrew. Arabic alphabets are also rendered differently depending how they are sequenced and vowels marks may be displaced or pushed depending on other vowels marks or location of the alphabets. To date, there is no rendering engine that can display Arabic perfectly.

2.6. Input Method Engine

Input Method Engine (IME) is a mechanism for a user to enter text of their language.

Text can be entered into a computer in many ways. The keyboard is by far the most common device used, but many characters cannot be entered on a typical 101-keyboard. Thus, an IME may be a complex software system that allows the users to enter characters using phonetic or stroke-based input, and then selecting character from a list. Input methods are also required for languages that have many diacritics, such as European characters that have two or more diacritics to a single character. [22]



Not all OS are equipped with proper IME. Having I18N support in the software would be pointless if users cannot interact with the applications.

2.7. Others

L10N is not just about changing the languages used. An American accounting software based upon US currency, using US accounting regulations and tax laws would not sell in Singapore even though there is no language barrier.

Thus, L10N requires understanding of local community norms, usage behaviour, regulation and would require some level of customisation to achieve this.

3. I18N and L10N of the Internet

How do we bring I18N to a network that is already global in nature? And how do we do L10N without 'Balkanising the Internet'?

With more and more non-English speakers getting online and using the Internet, there is a strong demand to allow non-English characters in the Internet Protocol.

On the other hand, the Internet is designed to be a worldwide interoperable network. Hence, whether you are in Russia or you are in Brazil, the Internet will be based on the same Internet Protocol (IP) with the same Transmission Control Protocol (TCP) [20, 21]. Hence, the concept of L10N whereby each different country or region would customise the Internet to suit its own needs is in conflict with the basic principle of the Internet. L10N carries the risk of fragmenting the Internet into numerous Balkanised networks, each with its own sets of protocols and rules which would be an interoperability nightmare.

These are challenges faced by the Internet Engineering Task Force (IETF). [8] Moreover, these challenges are amplified alongside with the intricacy of I18N and L10N as discussed above.

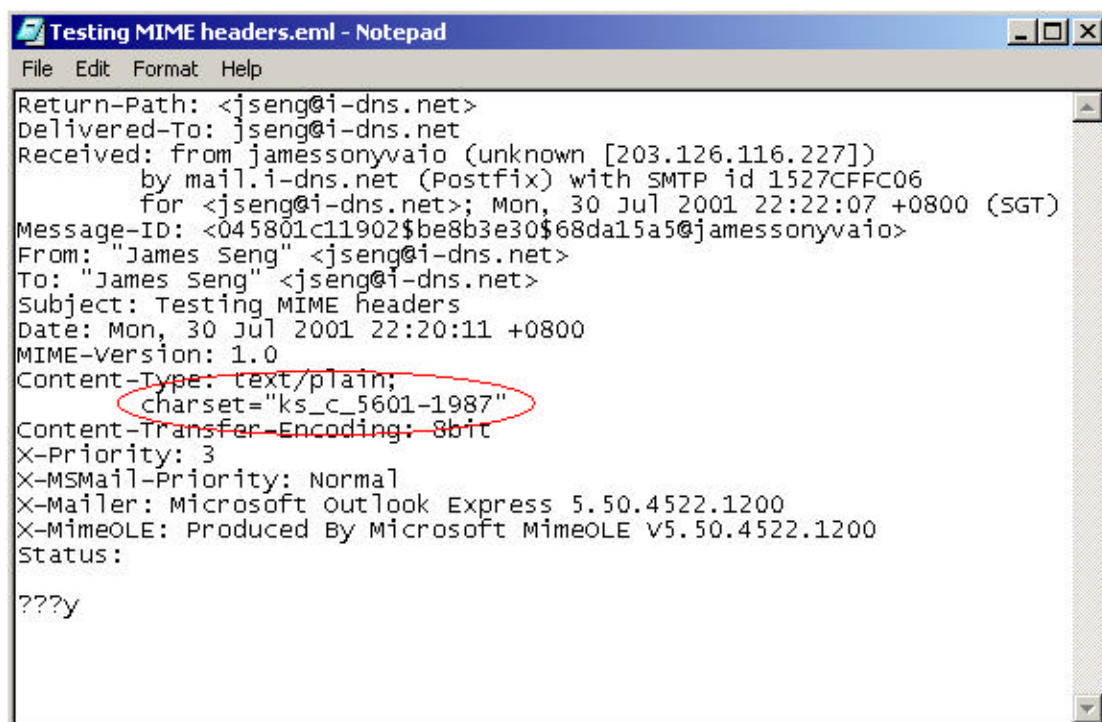
Despite this, the IETF have been putting a lot of effort into careful design and deployment of I18N or L10N solutions to some of its protocols and systems.

It is not possible to produce a complete list of all the work done on I18N and L10N for the Internet. Hence, the following are just some examples of the more established efforts undertaken to provide non-English speakers an opportunity to use their language on the Internet.

3.1. Internet Email

Internet mail was initially defined to be 7-bit US-ASCII only. Each message was made of extensible headers (metadata) associated with a single 7-bit US-ASCII message body traditionally pre-wrapped for display in a fixed width font on an 80-column screen. [18, 19]

At the time IETF attempted I18N or L10N (more accurately, multilingualism), there was no viable choice for an international CCS like ISO10646. Thus, the solution was to create a charset tagging mechanism within the mail headers. Since some of the localised charsets were 8-bit, an encoding scheme was also developed to transport non-ASCII characters. This method is known as Multipurpose Internet Mail Extension (MIME). [13, 14]



```
Testing MIME headers.eml - Notepad
File Edit Format Help
Return-Path: <jseng@i-dns.net>
Delivered-To: jseng@i-dns.net
Received: from jamessonyvaio (unknown [203.126.116.227])
    by mail.i-dns.net (Postfix) with SMTP id 1527CFFC06
    for <jseng@i-dns.net>; Mon, 30 Jul 2001 22:22:07 +0800 (SGT)
Message-ID: <045801c11902$be8b3e30$68da15a5@jamessonyvaio>
From: "James Seng" <jseng@i-dns.net>
To: "James Seng" <jseng@i-dns.net>
Subject: Testing MIME headers
Date: Mon, 30 Jul 2001 22:20:11 +0800
MIME-Version: 1.0
Content-Type: text/plain;
    charset="ks_c_5601-1987"
Content-Transfer-Encoding: 8bit
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 5.50.4522.1200
X-MimeOLE: Produced By Microsoft MimeOLE V5.50.4522.1200
Status:
???y
```

MIME provides a mechanism where it is theoretically possible to build international clients, or at least know when the charset used is unfamiliar or unavailable. Currently, most major email clients support the MIME content transfer encodings. Unfortunately, the result is usually clients that are localised for a limited set of languages and large mapping tables are required to do transliteration of the charsets in Internet mail.

3.2. World Wide Web

The Hypertext Transfer Protocol (HTTP) [15] is the basic protocol adopted in the World Wide Web. It is designed to be a simple protocol to transfer web pages from the server to the client.

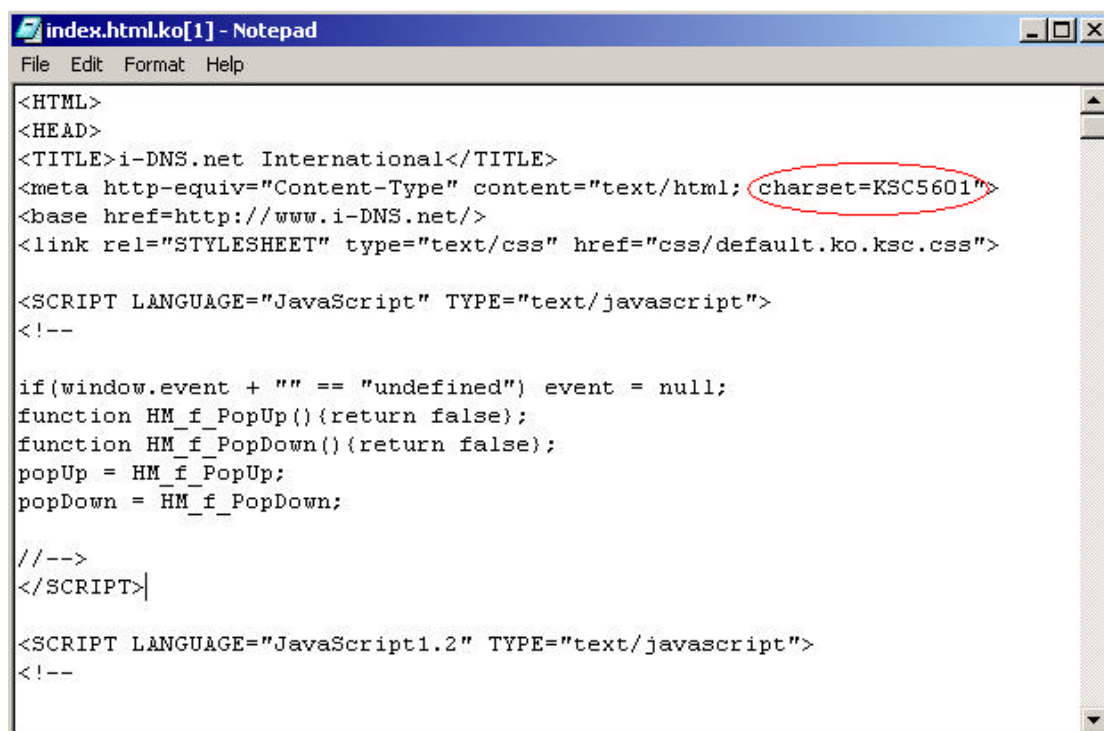
In 1999, the World Wide Web Consortium [26] and the IETF produced HTTP version 1.1 [15]. HTTP/1.1 introduced a series of I18N features:

- a. Indicating the character encoding of a page sent from the server to client (charset parameter)
- b. Indicating the character encodings understood by the client to the server
- c. Indicating the language(s) of a page sent from the server to the client (Content-language header)
- d. Indicating the language(s) understood by the user to the server, also known as language negotiation (Accept-language header).

With these features, particularly with (d), it is possible for the servers to determine the language preference of the users and return the web page in the preferred language(s).

HyperText Markup Language (HTML) is the *lingua franca* for publishing on the World Wide Web. Having gone through several stages of evolution, today's HTML has a wide range of features reflecting the needs of a very diverse and international community. [4]

Originally, HTML base CCS is ISO8859-1. This is obviously not able to serve the web page beyond the Latin-1 languages. Hence, multilingualism of the HTML started from the grass-root level where users attempt to put their localised encoding in HTML and serve it from there.



```
index.html.ko[1] - Notepad
File Edit Format Help
<HTML>
<HEAD>
<TITLE>i-DNS.net International</TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=KSC5601">
<base href=http://www.i-DNS.net/>
<link rel="stylesheet" type="text/css" href="css/default.ko.ksc.css">

<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
<!--
if(window.event + "" == "undefined") event = null;
function HM_f_PopUp(){return false};
function HM_f_PopDown(){return false};
popUp = HM_f_PopUp;
popDown = HM_f_PopDown;

//-->
</SCRIPT>

<SCRIPT LANGUAGE="JavaScript1.2" TYPE="text/javascript">
<!--
```

W3C soon started an I18N working group & interest group [5] to standardise the I18N efforts of HTML. The current HTML v4.0 now uses ISO10646 as the base CCS with

numerous of the CES of ISO10646, such as UTF-8, UTF-16 could be used. Other CCS and CES could also be used with appropriate language tagging.

While the W3C intention is to use ISO10646 for all HTML, the use of localised encoding is still very popular given the historical lack of support of ISO10646 in popular browsers. Support for ISO10646 was not complete for version 5 and 6 for Netscape and Internet Explorer respectively. Thus, it would be some time before ISO10646 would be fully adopted in HTML.

3.3. Domain Names

The Domain Names System provided a consistent name space that maps domain names to a network resource like IP address. It is a hierarchical distributed reliable lookup system. To provide a non-ambiguous simple lookup mechanism, domain names are restricted to a very limited number of characters, namely A to Z (case insensitive), 0 to 9, and a hyphen “-” (LDH – Letters, Digit and Hyphen). [16, 17]

With the proliferation of the Web and Email addresses in advertisements, domain names have taken on the stage of attention as the core component of any web or email address. The restriction to use only LDH is apparently a barrier to I18N of Internet Identifiers. Hence, in late 1999, IETF created a Working Group for Internationalised Domain Names (IDN) [6] to examine the requirements and to establish a standard for the use of non-English characters in domain names.

Domain name is a very integrated part of Internet, which is used widely across many protocols. For years, protocols and software have been designed and developed that assumed domain names to use only LDH. Hence, it is not easy to introduce characters beyond LDH without breaking existing protocols and software.

Moreover, as with any I18N effort, it is faced with the challenges of exact matching involving complex normalisation rules. In IDN terminology, the process to normalise names for the use of domain names is known as NAMEPREP [12].

While there is no conclusion from the IDN Working Group, there has been an established support on a proposal to implement IDN using Application [7]. With IDNA, the user applications would be responsible for handling NAMEPREP and converting the input to an ASCII-Compatible Encoding [1, 3, 11], a transformation encoding scheme (TES) of ISO10646 that produces a resultant string of LDH only. With ACE, existing protocols, servers and applications will be able to continue functioning without the risk of non-interoperability.

4. Conclusion

Internationalisation and Localisation carries very complex problems that require cautious design and planning. Subsequent ease-of-use to the end users has to be balanced with the risks of Balkanisation.

Nevertheless, Internet architects at the IETF are aware of the needs of non-English speakers. Much effort and work have been put into examining the issues of I18N and L10N with regards their application and usage on the Internet.

One day, the hidden pre-requisite of having to possess some basic English knowledge to use the Internet will be broken down. Users would be able to use their language of choice online, as they would offline. Only then, would the vision of a “Global Village” with a multiverse of diversified societies and cultures be fully realised.

Acknowledgements

Thanks to Jerry Yap and Ang Wui Liang from i-DNS.net for proof-reading.

Section 3.1 on Internet Mail originated from a yet-to-be published Internet Draft by Chris Newman.

References

- [1] AMC-ACE-Z version 0.2.1, Adam M. Costello, draft-costello-idn-amc-ace-z-00.txt
- [2] CJKV Information Processing, Ken Lunde, ISBN 1-56592-224-7
- [3] Differential Unicode Domain Encoding (DUDE), Mark Welter, Brian W. Spolarich, Adam M. Costello, draft-ietf-idn-dude-02.txt
- [4] HyperText Markup Language, <http://www.w3.org/MarkUp/>
- [5] Internationalisation at W3C, Martin Duerst, <http://www.w3.org/International/>
- [6] Internationalised Domain Names Working Group, James Seng, Marc Blanchet, <http://www.i-d-n.net/>
- [7] Internationalised Host Names in Applications, Patrik Faltstrom, Paul Hoffman, draft-ietf-idn-idna-03.txt
- [8] Internet Engineering Task Force, <http://www.ietf.org/>
- [9] ISO/IEC10646-1:2000. International Standard – Information technology – Universal Multiple Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane
- [10] Language and Script, Unicode Consortium, <http://www.unicode.org/unicode/onlinedat/languages-scripts.html>
- [11] MACE: Modal ASCII Compatible Encoding for IDN, M. Ishisone, Y. Yoneya, draft-ietf-idn-mace-01.txt
- [12] Preparation of Internationalised Host Names, Paul Hoffman, Marc Blanchet, draft-ietf-idn-nameprep-05.txt
- [13] RFC 1521 MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies. N. Borenstein, N. Freed. September 1993.
- [14] RFC 1522 MIME (Multipurpose Internet Mail Extensions) Part Two: Message Header Extensions for Non-ASCII Text. K. Moore. September 1993.
- [15] RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. June 1999.

- [16] RFC1034 Domain names - concepts and facilities. P.V. Mockapetris.
Nov-01-1987.
- [17] RFC1035 Domain names - implementation and specification. P.V. Mockapetris.
Nov-01-1987.
- [18] RFC2821 Simple Mail Transfer Protocol. J. Klensin, Editor. April 2001.
- [19] RFC2822 Internet Message Format. P. Resnick, Editor. April 2001.
- [20] RFC759 Internet Message Protocol. J. Postel. Aug-01-1980.
- [21] RFC761 DoD standard Transmission Control Protocol. J. Postel. Jan-01-1980.
- [22] Terminology used in Internationalised in IETF, Paul Hoffman,
draft-hoffman-i18n-terms-01.txt
- [23] Unicode Collation Algorithm, Unicode Standard Annex #10, Mark Davis, Ken
Whistler
- [24] Unicode Consortium, <http://www.unicode.org/> The Unicode Standard Version
3.0, Unicode Consortium, ISBN 0-201-61633-5
- [25] Unicode Normalisation Forms, Unicode Standard Annex #15, Mark Davis,
Martin Duerst
- [26] World Wide Web Consortium, <http://www.w3.org/>