BCS issues in informatics

Text Encoding: from digital Babel to the multilingual Web

1. Birth of the byte-sized character

Telex machines and teleprinters came into being at the beginning of the 20th century. Text was sent over telephone wires using the 5-bit **Baudot Code**: this meant transmissions were limited to CAPITAL LETTERS.

Teleprinter, driven by punched paper tape.

Codes could also be transmitted over phone lines.

The **American Standard Code for Information Interchange** – ASCII – strangely enough has its origins in the UK. In 1956, Ivan Idelson at Ferranti developed a way of coding text onto seven-track paper tape, for a committee of the British Standards Institute. The first commercial use of this code was by AT&T, for their TWX teleprinter system. A number of control codes were added for machines other than teleprinters, and this seven-bit code became an American standard in 1963.

Industry agreement that there should be **8 bits in a byte** got a boost in the early sixties when Fred Brooks, in charge of developing the operating system for IBM's **System/360** mainframe family, insisted on a 7-bit character set to handle upper and lower case letters, numerals and punctuation. However, IBM didn't adopt ASCII, but 'stretched' their existing codeset to produce the **Extended Binary Coded Decimal Interchange Code** – EBCDIC.

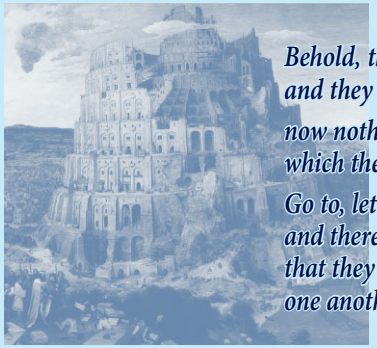
*Computers communicate **text** by exchanging blocks of binary **numbers**. As the foundation of data & information interchange, there has to be agreement about what numbers will be used to represent which characters in text…*

Photo: Jeanette Åkesson

7-bit ASCII

There are 95 printable characters in ASCII including the space, numbered 32 to 126.

```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWYYZ
[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
```



Painting: Tower of Babel by Peter Breughel, 1563

Behold, the people is one,
and they have all one language…
now nothing will be restrained from them,
which they have imagined to do.
Go to, let us go down,
and there confound their language,
that they may not understand
one another's speech…

2. 'Let us go down, and there confound their language'

ASCII provides a very limited character set.

No **£** or **€** sign, no proper printer's quote marks or other punctuation, and none of the accented letters required by languages other than English… **çàèœžñğǵű** &c

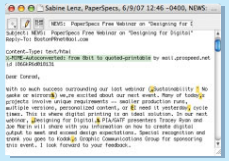
However, ASCII uses only 7 bits in the byte, and in the 1980s, companies and national standards committees set about exploiting the 'eighth bit' space to double the number of characters that could be represented.

Microsoft 'solved' the problem by adopting a range of 8-bit extended ASCII **code pages** for different language groups. Many of these code pages had been pioneered within IBM – examples are the Central European, Baltic, Turkish, Cyrillic, Greek and Vietnamese code pages.

Apple launched the Macintosh computer in 1984 with its own 8-bit *MacRoman* character set covering most of Western Europe.

The problem is, if a computer sending text uses a code page that's different from the receiving machine, the codes will be misinterpreted. Trouble!

This sample email shows the kind of transmission failures which are quite commonplace. The orange highlights mark where quote marks and dashes have been misrepresented; the green highlight gives a clue about a stage in the process where things may have gone wrong.



The second example is on a BBC News Web page. The encoding of the **é** character in the word 'protégés' has not been recognised by the Web browser.

Complicating search and retrieval

Consider the character **é** in the word *protégé* in the Web page example above. Within the HTML code, the character could be using the numeric value **142** if it's Macintosh text, or **233** if it's Windows text. Or it could be encoded as **&ecute;** or as **é** …

Any system for indexing and searching Web pages is going to have to take this into account – because if it doesn't, any search is going to miss a large number of instances of use of the word *protégé*.

3. Interchange encodings

In transferring texts across computer networks, it's often necessary to re-encode characters, for example to squeeze through 7-bit gateways.

Quoted Printable (QP) is a common ruse for sending high-ASCII characters through email, by converting them to the equals sign, used as an 'escape character', followed by a two-digit code. Thus **Allô bébé** would be turned into **All=F4 b=E9b=E9** (all ASCII characters).

Character entity encoding is often used to ensure that non-ASCII Latin-script characters display properly on Web pages, regardless of what browser or machine is being used to view them. The problematic character is replaced in the HTML file with a string that starts with an ampersand and ends with a semicolon:

£	£	£ sterling sign
é	é	e acute
		non-breaking wordspace

Numeric codes are also used:

ç	ç	c cedilla
—	—	em dash (long dash)

4. Unicode to the rescue?

Since 1991, the Unicode Consortium has been working on a 16-bit character encoding, with the aim of giving a unique identification code to each character in every script in the world.

Windows NT was the first operating system to use Unicode to represent text internally; with OS X, Macintosh now does the same.

The problem is that many applications do not support Unicode well, and some scripts have special rendering problems (Arabic runs right to left, Hindi combines characters in special ways). But as the example Web pages here show, it is now relatively easy and reliable to view information in many of the world's languages – thanks to Unicode.

The code samples show that the text is truly encoded in Arabic, Hindi and Chinese