Internationaler Standard beseitigt Problematik inkompatibler Kodierungen

Unicode – Die Zeichen der Zeit ?

وُنمذىلمحولإحبنبجفىو

ほちせそすのみ

Ω¶∑ℓπ¢¥£ℭμ◊≥©ð¤§#‰

麁並遷欄鐮涼炎闇違牡映鯵逢鰯且俄間階襖

Bietet Ihre Billig-Airline das Flugziel „M□nchen" an? Werden Sie von Ihren ausländischen Geschäftspartnern als „J¿rg M€ller" angeschrieben? Dann gibt es ein Problem mit Zeichen. Wir benutzen Zeichen vor allem, um die Laute unserer Sprache in Form einer geregelten Schrift niederzuschreiben. Und der Computer, der keinen Bleistift führen kann, muss solche Zeichen notgedrungen elektronisch codieren, mit Hilfe eines so genannten Zeichensatzes. Jeder Code aus seinem Zeichensatz erzeugt dann auf dem Bildschirm ein festgelegtes Symbol, natürlich abhängig von den Kenntnissen und Einstellungen des Betriebssystems.

Nun sind aber Informatiker mit ihren Bits und Bytes seit je her sparsam gewesen – sie bieten uns sieben, höchstens acht Bits an, um die Zeichen unseres Kulturkreises zu codieren. Für die 26 Buchstaben unseres Alphabetes, denken Sie, müsste das doch wohl reichen! Was ist aber mit den 106 Zeichen der thailändischen Schrift? Was mit den 7000, die der intellektuelle Chinese beherrscht? Was ist, wenn man mehrere Schriften gleichzeitig darstellen möchte?

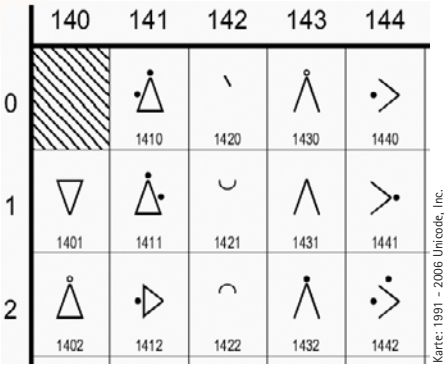
Lösung Unicode

Unicode ist hier die Lösung. Unicode macht Schluss mit der Sparsamkeit und der Abhängigkeit der Symboldarstellung vom Betriebssystem und seinen Einstellungen. Und zwar indem er mehr als ein Byte pro Zeichen anbietet.

Der internationale Standard des 1991 gegründeten „Unicode Konsortium" erlaubt die Verwendung von fast allen Zahlenbereichen bis zu 21 Bit, genauer gesagt maximal 1.114.112 Kombinationen. Etwa hunderttausend davon ist bereits ein wirkliches Zeichen zugeordnet. Damit ist gewährleistet, dass nicht mehr je nach Land und Zeichensatz ein und dieselbe Zahl für verschiedene Zeichen stehen kann.

Codepoints

Aber auch im Unicode müssen die Anwendungen und Betriebssysteme genau wissen, welches Symbol an welcher Position abgelegt ist. Die Codepunkte, also die Nummer des Zeichens im Unicode, heißen „Codepoints". Wie sich die Schriften dieser Erde auf die Codepoints verteilen, kann man sehr leicht unter http://www.unicode.org/charts/ erkennen. Ein Klick auf „Canadian Syllabics" offenbart uns zum Beispiel, dass die Schriften der kanadischen Ureinwohner im Bereich der hexadezimalen Codepunkte „1400" bis „167F" abgelegt sind. In der für uns verständlicheren dezimalen Bezeichnung sind das die Nummern 5120 bis 5759.



Ausschnitt aus den Unicode-Tabellen für „Unified Canadian Aboriginal Syllabics"

Man benutzt allerdings die hexadezimale Darstellung, weil diese naturgemäß näher an der Codierung im Computer liegt. Und selbst für ausgestorbene Schriften wie das Phönizische (Codepoints 10900 bis 1091F) oder Aramäische bleibt dann noch genügend Platz.

Codierung

Das scheint zunächst mal eine recht einfach Lösung zu sein. Leider haben sich mehrere Möglichkeiten entwickelt, die Codepunkte dieses Zeichensatzes in Bits und Bytes zu packen, also zu „codieren". Das ist ein Problem, welches bei den klassischen Zeichensätzen wie zum Beispiel Latin-1 nicht auftaucht, weil dort ja in der Regel immer nur ein Byte,

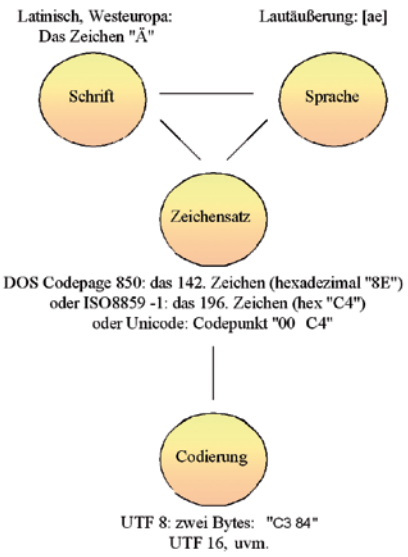
also 255 Positionen, zur Verfügung stehen und deshalb der Druck zur Platzeinsparung gering ist.

Im Unicode aber gibt es eine Hintertür für Nutzer des ASCII-Zeichensatzes, also der lateinischen Schrift ohne Umlaute und Sonderzeichen. Diese Codepunkte sind so angelegt, dass man sie trotzdem mit nur einem Byte darstellen kann. Sie liegen nämlich zwischen 0000 und 007F, also zwischen 0 und 127, wie bisher auch. Auf diese Weise ist ASCII kompatibel zu Unicode.

Erst für andere Schriften muss man dann weitere Bits und zusätzlich auch Zählmarker, Startbits und Stopbits hinzunehmen, der Codepunkt benötigt dann zwei oder sogar drei oder vier Bytes. Diese Technik der Formatierung nennt man UTF-8 (8-bit unicode transformation format). Und sie ist auch am weitesten verbreitet.

Eine weitere, und ältere Möglichkeit ist die UTF-16 Codierung: Hier werden grundsätzlich immer zwei Bytes benutzt, so dass die Codepoints von 0000 bis FFFF vollständig abgedeckt sind. Für darüberliegende Codepoints muss man dann allerdings zwei „2-Byte"-Zeichen kombinieren.

Die Codierung kann auch als Formatierung oder „Encoding" bezeichnet werden. Der Zusammenhang zwischen Schrift, Zeichensatz und Codierung ist in folgendem Schaubild dargestellt, und zwar für das Zeichen „Ä", welches nicht zu den ersten 127 Codepunkten gehört.



Der Zusammenhang zwischen Sprache, Schrift, Zeichensatz und Codierung

Probleme

Damit sind aber bereits die ersten Nachteile des Unicode offenbar: Da man normalerweise mehr als ein Byte zur Codierung benötigt, nimmt die Menge an Speicherplatz zu, die man für einen Text verbraucht. Zum Beispiel kommt man bei einem herkömmlichen Zeichensatz wie ISO 8859-5 mit einem Byte für kyrillische Buchstaben aus, weil man ja alle anderen Schriften in solchen Zeichensätzen ausblendet. Im Unicode erreicht man hingegen leicht eine Verdoppelung der Datenmenge. Für größere Datenbanken wie zum Beispiel landesweite Straßennetze mit Namen kann das schnell eine kritische Schwelle überschreiten.

Infolgedessen kann auch die Leistungsfähigkeit von Anwendungen leiden. Es muss ja unter Umständen die doppelte Datenmenge gelesen werden, um dieselben Zeichen auf dem Bildschirm darzustellen. Man sollte also gut überlegen, ob für die benötigten Daten der Einsatz von Unicode überhaupt gerechtfertigt ist.

Schließlich muss das Betriebssystem auch eine Schriftart bereitstellen, die den Unicode für uns lesbar machen kann. Wer mit einem neueren Windows-Rechner arbeitet, findet in den Anwendungen in der Regel Schriftarten wie „Lucida Sans Unicode" oder „Arial Unicode", die einen großen Teil des Problems aus der Welt schaffen und für normale Anwendungsfälle recht gut brauchbar sind.

Auf anderen Betriebssystemen oder älteren Rechnern muss das aber nicht der Fall sein. Und wer auf weniger triviale Schriftarten zurückgreifen möchte, hat dazu in der Regel keine Unicode-fähige Variante und muss dann ohnehin einen passenden anderen Font kaufen.

Internationalisierung

Dennoch – die Zeichen der Zeit stehen auf Internationalisierung. Vorbei ist die Vorherrschaft der herkömmlichen IT-Welt, die mit 26 Buchstaben auskommt. Längst mischen die Menschen aus Asien und Osteuropa kräftig mit in den Märkten, bauen und nutzen Software und Internet. Und sie möchten natürlich nicht auf die Besonderheiten ihrer Schrift verzichten. Wer sich einmal über „M□nchen" oder „Munchen" auf ausländischen Internetseiten amüsiert hat, der weiß, wie sich Osteuropäer fühlen, die mit Schreibfehlern in ihren Ortsnamen konfrontiert werden. Mit Unicode hingegen können verschiedene Schriften gleichzeitig dargestellt werden.

Und damit wird das Beherrschen der Schriften und Symbole dieser Welt zum Erfolgsfaktor auf dem Markt. Denn wer „Łodz" in seinen Daten richtig schreibt, dem traut der Kunde mehr zu als dem Urheber von „Lodz". Die identitätsstiftende Sprach- und Schriftkultur ist ein so grundlegender Teil von uns selbst, dass sie oft unbemerkt kaufentscheidend sein kann. Der internationale Datenaustausch ist heute ohne die Vorteile von Unicode kaum noch denkbar.

Mehrere Schriftarten in derselben Karte - mit Unicode kein Problem

In Kürze

Um Zeichen zu sparen, kann man es in den Worten des Unicode-Konsortiums kurz auf den Punkt bringen: „Unicode bietet eine eindeutige Nummer für jedes Zeichen, egal welche Plattform, egal welches Programm, egal welche Sprache." (http://www.unicode.org/).

Dipl.-Geoökol. Thomas Engel, Jahrgang 1975, studierte an den Universitäten Karlsruhe und Hamburg Geoökologie mit Schwerpunkt Geoinformatik. Seit acht Jahren arbeitet er bei der PTV AG und betreut zahlreiche Datenversorgungsprojekte für alle geographischen Applikationen des Konzerns. Schwerpunkte sind das Standardformat GDF, die Umsetzung internationaler Geodaten und die Konzeptionierung der Mautberechnung.