

これまでのアプローチ

この問題を解決するため、失敗した試みも含め、これまで多くのことがなされてきた。その一端を紹介する。

○ OS で使える文字数を増やす

- ◆ Windows 9,000 → 65,536/1,114,112 Unix 1,7000 → 2,147,483,648

○ 文字レパートリを増やしていく

- ◆ ISO 10646 UCS = Unicode の漢字に関しては
1993: 約 2 万字
2000: 約 2 万 6 千字
2003: 約 7 万字

✕ 文字コードを標準化するのではなく、誰でも登録できるような制度にしよう

- ◆ 字体の問題にぶつかる
- ◆ 一点しんによつと二点しんによつ、戸戸戸や、高高高など、字体としては異なるが(ほとんどの人にとって)同一文字と考えられるケースで、別の文字として登録されることになる。

✕ それでは、文字のレベルと、字体のレベルで階層化をしよう

- ◆ 文字レベル「戸」の 1 番が戸、2 番が戸、3 番が戸というようにコード化する
- ◆ 少なくとも、これまでのところこの方式で成功した例はない。
(一組織内のみで使用されるような局所的なアプリケーションをのぞいて)
- ◆ 字体の数が多すぎてコントロールが難しくなることが原因かと推測される。

✕ 要素組合せ型コード法

- ◆ 漢字をその要素にわけてコード付けし、文字をそのシーケンスとしてコード化する方法
- ◆ たとえば、「例」= 人+列 など
- ◆ 要素自体の階層性(列はさらに歹+刀になるなど)、要素字形のバリエーションの多様性(刀とりなど)、要素として特定しにくい例外的字体などの問題がある
- ◆ また、ひとつの文字に対し、複数の分解、複数の要素順が対応するのを防ぐことが難しい
- ◆ これらの理由で、文字コードとして成功したことはない。(入力方式としては使われている。)

それでは

今後どのようにすべきか、分析と経験とをふまえて提案する

■ 文字セット管理

- ◆ 現在の文字コードはすでに、字体に関して一貫性のないものとなっている。
- ◆ その中で字体に関して問題を起こさないようにするには、使用する側で使い方を工夫するしかない。
- ◆ たとえば、戸戸戸を同じ文字として取り扱うように決め、アプリケーションで正規化するなどの対応をする。

■ 字体に対する寛大な態度

- ◆ 文字の形の枝葉末節にとらわれることは、学問的な態度でも文化的な態度でもない。
- ◆ 江戸時代の漢学に造詣の深い日本人学者や文化人は、むしろ各種の字体を併用する傾向さえあった。
- ◆ 「熙」と「熙」の区別を強要するのは、知的能力の無駄遣いと考える。
- ◆ むしろ、おおらかな態度を取る方が、社会的コストの節約になる。

■ 文字化けやソフトウェア問題に関し

- ◆ 文字コードのグローバルな統一の方向に対し無駄な抵抗をしないこと
- ◆ 各言語文化の文字や伝統的書法を尊重することは文化にとって大切であるが、文字コードが各言語別になっていたからと言って、文化の保持や発展には寄与しない。
- ◆ ソフトウェア作成に関わるものであれば、言語ごとの文字コードを前提とした開発を行わないこと。
- ◆ 一般ユーザであれば、文字コード環境が落ち着き、ソフトウェアが改良されて文字化け等が起こらなくなるのを待つ、
- ◆ あるいは、ソフトウェアメーカーに対し、積極的に文句を言って、改善を促す。