

第12回 ユーザーインターフェース、ウインドウシステム・国際化、インプットメソッド

12-1. ウインドウシステム

	CUI ()の略	GUI ()の略
どのようなものか (説明) (何ができる) (どう使う)		
身近な例		
比較		

ウインドウシステム(GUI)を実現するのに必要なもの

ディスプレイ ()

入力装置 ()

12-2. 国際化 インターナショナルリゼーション ～～ 多言語対応

多言語化の例を挙げよ (何が多言語化か?)

.....

.....

国際化と地域化の枠組み

- ① 設定を変えればどの言語にも対応できる(ように準備しておく)
- ② 設定のための情報やサブシステムを用意する(日本語表示、日本語入力、日本語メッセージなど)

ロケールとは何か

.....

.....

文字コード:

英数字: 何文字表したいか? () それを表すために必要なビット数は? ()

かな漢字: 何文字表したいか? () それを表すために必要なビット数は? ()

用語: コードポイントとは()

漢字コードの歴史的な問題・混乱 ← 授業で説明

Unicode の原理を自分で調べてみよう。(たとえばウィキペディアで unicode を調べてみよう)

12-3. インputメソッド

インputメソッドは、コンピュータに（ ）を（ ）する為の手段(主にソフトウェア)である。

英数字(アルファベットと数字)は、PCでは（ ）から直接に入力することができるが、かなや漢字はキーが不足するので直接入力することが難しい。たとえば日本語の場合は、（ ）変換など、ソフトウェアによる支援を受ける。この支援システム(支援ソフトウェア)を総称して、インputメソッドと呼んでいる。

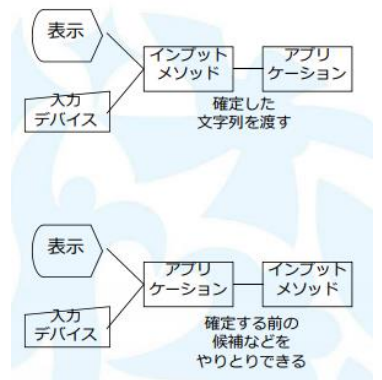
(注) コンピュータ以外、たとえばスマートフォンにおいては、入力ハードウェアが PC と異なる(タッチパネル)ため、支援の仕方も見かけ上異なるが、大筋の考え方は同じであろう。

2つの構成法

図に示すように、(人間が入力する入力装置+表示装置)と(入力文字を利用するアプリケーション)とインputメソッドとのつながり方には、2つの方式がある。

上段を（ ）方式と呼び、あらかじめ（ ）で入力を変換しておき、それをアプリに供給する方式である。利点として、アプリの構造を英文対応のまま変更しないで済むので、既存英文アプリを日本語に対応させる場合には具合が良い。欠点は、変換の候補選択や候補の表示の仕方はアプリがコントロールできないので、インputメソッドで決まった形式で決まった場所に変換候補が表示されることになり、アプリと融合しない。(極端な例として、ワープロの変換候補が、入力場所に表示されるのではなく、画面の左下隅に表示される、などという時代もあった。)

下段は（ ）方式と呼び、入力デバイスや表示の制御は（ ）が握る。アプリは（ ）にその情報を渡して変換してもらい、その変換結果を受け取って、アプリの入力として使う。この利点は、変換の候補選択や候補の表示方法などは（ ）が制御できるので、柔軟性が高い。他方で欠点は、アプリが変換の作業に(データの仲介や表示制御などの形で)関わるので、アプリを作るときにそのことを織り込まなければならないことが、上段に比べて余分な作業になる。



変換エンジンの共有

前提: インputメソッド(たとえばかな漢字変換)は、カスタマイズ(ユーザが自分に合わせて設定を変える)が可能。

更に、かな漢字変換は、ユーザのくせを「学習」して辞書に蓄積し、より使いやすくしようとする。

1 人のユーザが複数の端末を操作するとき、どこからでも同じ使い勝手で使いたい。つまり同じ変換エンジン、同じユーザ設定、同じ学習状況(辞書)で使いたい。このためにはどうすればよいか？

.....

.....

.....

(脱線) 豆知識 かな漢字変換の方法論

カナ漢字変換への要求は、①応答が() ②プログラムやデータが() (メモリ等を圧迫しない)

2つの変換原理がある。①辞書式(辞書に変換パターンを記憶しておく) ②ルール式(パターンでなくルールで書く) 一方だけに限定すると問題解決が難しいので、両方をうまくバランスさせた形で混在させる。

自然言語処理(たとえば構文解析や意味処理)を使うか？ これは処理が複雑になりすぎる。確率遷移モデル上で統計的に処理する方法が、広く使われている。ただし、誤変換ゼロは難しい。