

# UI の機能抽象化による GUI とハードウェア UI 透過性を実現するシステムの開発

久原 政彦<sup>†</sup>伊藤 誠<sup>†</sup><sup>†</sup> 中京大学

## 1. まえがき

計算機上で実行されるアプリケーション (AP) を操作するために用いるユーザインタフェース (UI) 装置は現在、マウス・キーボード・ディスプレイ等が標準である。

昨今これらに代わる次世代の UI として、ユーザの生体情報入力や、視聴覚以外の出力を用いるために外部ハードウェア機器を使った UI (HUI と略記) が多く研究されている。しかしながらこれら HUI は実験過程において独自の規格で制作され、AP も特注の場合がほとんどであり、HUI を普遍的に AP へ導入し利用するための枠組みづくりが為されていない場合が多いのが現状である。

本研究はこれら HUI を AP 上で普遍的に用いるための仕組みを策定し、標準 UI にはなかった新しい入出力の概念を取り込むための問題と解決策をまとめ、具体的実装として仲介規格 UICP (User Interface Control Protocol) を規定すると共に、開発例を基にその解説を行う。

## 2. HUI の現状・問題・解決策

### 2.1. HUI の研究事例

実世界指向 UI の研究には、Tangible User Interface<sup>\*2</sup> などがある。学会発表の例として Interaction2007<sup>\*3</sup> では、触覚<sup>\*3a</sup>・風力ディスプレイ<sup>\*3b</sup>等の出力 UI、タップによる入力<sup>\*3c</sup>や手書き入力<sup>\*3d</sup>等の入力 UI などが発表された。

障害者向け UI としては、マウスの代替として舌によるカーソル移動、体勢カーソル、呼吸ボタンなどが開発され、キーボードの代替として発話変換、視線キーボード、マクロ付きキーボードなどが開発されている。

### 2.2. HUI の利用における問題

HUI 群にはユーザ負荷を減らしたり仕事の効率を上げたりするものも数多くあるが、一般にまで普及する例は多くはない。問題となることに以下が挙げられる。

- (1) AP が先進的 UI を標準 UI の代替として受入できない
- (2) 先進的 UI ならではの特徴や概念を利用できない
- (3) 入手困難性や費用対効果

### 2.3. 期待される成果

本研究では上記(1)(2)を解決するため、HUI を容易に導入・代替できる環境を提供すること、および HUI ならではの特徴や概念を抽出し、これらを AP 開発者が手軽に利用できるような規格を策定することを目標とする。

その解決策として、HUI と AP の間で仲介役として働くプロトコル UICP を策定し、HUI が AP にとって汎用的 UI となることのできるような環境を目指す。

Development of System that Realization of Transparency of GUI and Hardware-UI using Functional Abstraction of UI

<sup>†</sup>Masahiko KUBARA, Chukyo-University

<sup>†</sup>Makoto ITO, Chukyo-University

## 3. HUI の導入に必要な概念

### 3.1. 代替性・抽象性

入出力を透過的に行うために、UI を特化して扱うのではなく抽象化して扱い、入出力の指示をデバイスベースではなく機能ベースで考慮することで、同じ機能を持つ UI を代替して取り扱える仕組みを考慮する。

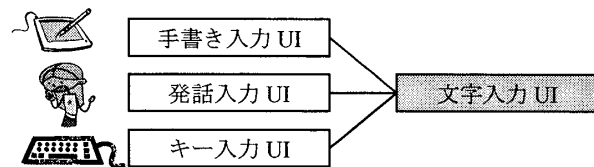


図 1. 代替性・抽象性の模式図

### 3.2. 多元同時入出力

HUI には、多指操作である多点入力や複数同時操作、データの同時多変量調節などが行えるものもあり、これら同時操作の概念を考慮する必要がある。

### 3.3. 連続量 (アナログ値)

計算機の UI はデジタル値を取り扱っているが、先進的 UI は実世界の UI が多数であり、それらは温度・圧力・動きなどの時系列に連続するアナログ値も取り扱っていることから、連続量の概念も考慮する必要がある。

## 4. UICP

### 4.1. UICP-API と UICP-DATA

AP が HUI を標準 UI と透過的に扱うためには、HUI と標準 UI を操作する API を統一する必要がある、また API によって授受されるデータも統一する必要がある。本研究ではこれらを UICP-API と UICP-DATA と定義し、プログラム上およびデータ上の透過性を実現する。

### 4.2. 意味的入出力の提供—代替性実現

HUI が AP とデータ授受を行う際、通常そのデータは機器によって異なり、例えばタクトスイッチと赤外線センサが同じデータを返すわけではない。しかしながら AP 側から見ればどちらもトリガに相当するものである。

ここで機器側に機械的入出力を意味的入出力に変換する機能を備えれば、機械信号をトリガとして取り出すことができる。同じ意味を持つ機器は同じ UICP-DATA を返すよう規定することで、自由な代替性を実現する。

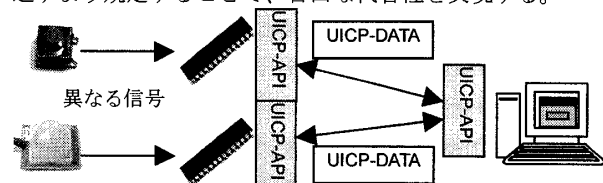


図 2. 意味的入出力による機能代替性

### 4.3. 時系列変数の導入—同時性・連続量実現

値	128	140	145	160	158	147
時刻	0	10	20	30	40	50

時系列 →

図 3. 時系列変数の模式図

HUI は GUI と異なり、同時多変量入出力や、アナログ値の入出力が発生する。これらを解決するため、デジタル値に時刻を付加したデータを連続して記憶する変数、時系列変数を導入する。

時系列変数は時刻を引数にして値を取得できる変数で、通常は値と時刻をペアにした構造体の配列で実装される。

アナログ値は任意時刻にサンプルされたデジタル値の時系列変数として記憶させることにより実現する。

多元同時入出力は各元の時系列変数に任意の時刻を引数として与えて得られた値を比較することで同時入出力の成否を判断できる。

## 5. UICP を適用した開発例

### 5.1. IDE を用いた HUI 制作

ソフトウェアの AP プログラムは統合開発環境 (IDE) 上で GUI を生成することで UI を構成する。これに対しハードウェア制作者はブレッドボード上などでテストした後基板へ実装する、という設計方法の違いがある。

そこでソフトウェア上で HUI を GUI のようにレイアウトできる IDE を制作し、これを用いて AP を設計することで、ソフトウェア GUI の延長線上で HUI を設計できるような形を考案した。

### 5.2. GUI を作る感覚で HUI をレイアウト

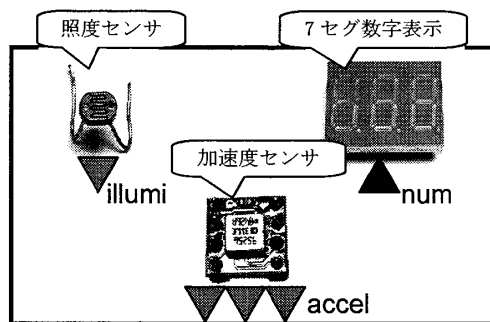


図 4. HUI レイアウト例

HUI-IDE は GUI-IDE のようにウィンドウへパーツを貼り付けるような形でパーツを貼り付けることができる。

各パーツは意味の入出力を持ち、例えば前述のタクトスイッチはボタン入力に、7セグ LED は数値出力に、加速度センサは 3 値アナログ入力に、それぞれ置き換わる。

設計者はこれらのパーツが具体的にどのような実装や電気的設計を持つかを考える必要はない。機械的の入出力を意味の入出力へ置き換えることは、IDE が自動で行う。

### 5.3. パーツを管轄しデータ授受を行う MCU

パーツの機械的入出力を解釈して意味の入出力へ変換し、計算機上の AP へ授受するための仲介役として、パーツの中心に MCU (Micro Controller Unit) を置く。MCU

へ UICP 用に最適化され自動生成されたプログラムを焼きこむことで、AP プログラムは UICP-API を通じて意味的入出力を引き出すことができる。

表 1. MCU 関連の対応表

項目	詳細
MCU	Microchip PIC24F
コンパイラ	Microchip MPLAB C30
PC 間通信	RS232C, USB-Serial, USB-HID

### 5.4. ハードウェア実装情報の自動生成

HUI を制作する上で障害になることに、MCU と各パーツの配線やレイアウトを行うことが難しいことが挙げられる。気軽に HUI を制作したいという要望に応えるため、HUI-IDE は配線や実装を自動で行い、実体配線図などを出力する機能を備える。

### 5.5. UI の代替エミュレーションとデバッグ

HUI が UICP-API 対応になると、HUI そのものが手元になくとも GUI や既存 HUI で代替でき、UI 開発と AP 開発の独立化が可能になる。またデバッグ時には UICP-DATA をフックすることで UI 挙動をデバッグすることや、アナログ入力値をファイル入力やマウス入力で置き換えてテストすることなども可能になる。

## 6. UICP を適用できる参考例

### a. 先進的 UI の汎用化

先進的 UI 研究者は、UI に対応した AP を併せて提供する必要があったが、UICP に対応すれば既存 UICP-AP に容易に導入でき、迅速な評価や普及が可能になる。

### b. 障害者向け UI の汎用的利用

障害の程度に応じてカスタマイズされた個々の UI を、AP は等価な UI として取り扱うことができる。AP 設計者は UI 毎の設計を行う必要がなくなり、ユーザも AP 毎に UI を替えるという費用負担や再学習負担がなくなる。

### c. ユーザに応じたカスタマイズ

前述の障害者向け UI の他、高齢者向け単純化 UI、子供向け機能制限型 UI、女性向け UI など、対象ユーザに特化した UI を必要に応じて提供することや、UI 開発を専門とする起業が可能になる。

## 7. まとめ

現計算機環境下の標準 UI を代替・拡張する存在として HUI を挙げ、これらの導入に必要な概念を整理し、概念を充足する仲介規格 UICP を提案した。併せて開発例や参考例も提示した。今後は規格のブラッシュアップとともに、HUI 研究者に向けた発信や、開発用の資料整備を行い、UI の発展へ資するものとしたい。

## 参考文献

- 1) 久原政彦, 伊藤誠: "UI の透過的利用を可能にする制御プロトコル UICP の開発と運用", IPSJ 第 70 回全国大会, 2008/03
- 2) Ishii, H., et al: "Tangible bits: towards seamless interfaces between people, bits and atoms", Proc. of CHI'97, 1997
- 3) 情報処理学会: Interaction2007 論文集, (a)pp.121-128 (b)pp.105-112 (c)pp.171-172 (d)pp.183-184, 2007/03