

タンジブルデバイスへのアプリケーション機能割り当て手法

松井 香純^{†1} 岸野 泰恵^{†2}
寺田 努^{†3} 西尾 章治郎^{†1}

TUI (Tangible User Interface) を用いてアプリケーションを操作する場合、アプリケーション機能と TUI のデバイスを関連付ける必要があるが、従来の TUI のためのツールキットでは関連付けが面倒であるという問題があり、ユーザが様々なアプリケーションを TUI を用いて気軽に操作することは困難であった。そこで本研究では、TUI のデバイスとアプリケーション機能を自動的に関連付ける手法の実現を目的とする。提案システムでは、機能に対するデバイスの種類、機能の配置、関連する機能の扱い方などに関するユーザの嗜好に基づいて自動的に関連付けを行うことで関連付けの煩雑さを解消し、さらに、追加学習を行うことでユーザに特化した関連付けを可能にすることで、87%の精度でユーザの想定した機能が割り当てられることを確認した。

An Algorithm for Allocating Application Functions to Tangible Devices

KAZUMI MATSUI,^{†1} YASUE KISHINO,^{†2} TSUTOMU TERADA^{†3}
and SHOJIRO NISHIO^{†1}

In recent years, to control application easily and intuitively, there is an increasing demand on Tangible User Interfaces (TUI). For using TUI devices to control applications, the system needs to associate the functions with TUI devices. Though several TUI applications and toolkits have been proposed, it is difficult to use them for various applications since conventional toolkits impose annoying operations on the user. In this paper, we propose an automatic matching algorithm between TUI devices and application functions. When a user places TUI devices on a board freely, our system associates TUI devices with application functions by considering three types of user preferences: favorite type of device for each function, intuitive position for functions, and relationship among functions. The proposed method achieved 87% accuracy on automatic function assignments.

1. はじめに

近年、コンピュータ上のアプリケーションはますます高度化し、操作は複雑になっている。そこで最近では、実体をもつデバイスに手で触れてアプリケーションを操作する TUI (Tangible User Interface) の研究が盛んに行われている³⁾。既存の TUI として、瓶の蓋を開けると音楽が流れる musicBottle²⁾、砂の形状を変化させることで地形パターンの変化を表現できる Sandscape⁷⁾ ブロック型の入力装置を用いて三次元形状から立体のオブジェクトを画面上で操作することができる ActiveCube⁴⁾ など、数多くの TUI が提案さ

れている。実際のもの进行操作のと同じようにして入力インタフェースとなっているものを操作すると、実際のもの进行操作したときに得られた結果に近い感覚の結果が PC 上で得られるため、ユーザにとってわかりやすいインタフェースを実現できる。しかしこれらはデバイスの操作をユーザがカスタマイズできなかったり、専用のアプリケーションを操作する TUI であったため、最近では PC 上の既存のアプリケーションを TUI を用いて操作する研究が行われている。TUI を用いてアプリケーションの操作を行うと、マウスで画面上のボタンを押すのではなく実際にスライダを上下に動かすことで画面を上下にスクロールするなど、直観的な操作が可能になったり、よく使う機能を手で押すボタンに割り当てることで初心者にもわかりやすいインタフェースが実現できる。しかしこれまでに TUI のためのツールキットとして提案されている Phidget¹⁾ では、アプリケーションと連携させるために専門的なプログラミングの知識が必要であったり、VoodooIO⁸⁾ ではデバイスとアプリケーションの機能を手動で対応付けしなければならないという問題があった。

^{†1} 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
^{†2} NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories
^{†3} 神戸大学大学院工学研究科
Graduate School of Science and Technology, Kobe University

そこで本研究では、TUI を用いて PC 上のアプリケーションを操作する際に、TUI のデバイスとアプリケーションがもつ機能を自動的に関連付けることで、ユーザによる煩雑な操作なしに TUI を構築するシステムを実現する。本研究ではまずユーザのデバイス配置に関する嗜好を抽出するために予備実験を行い、得られた知見を基に機能の自動割当アルゴリズムを構築する。提案手法を用いることで、プログラム知識や面倒な対応付けの操作を必要とせず、しかもユーザの嗜好にあった関連付けが自動的に行われるため、ユーザが自分専用カスタマイズした TUI を簡単に構築できるようになる。

以下、2 章では想定環境について説明し、3 章では予備実験および提案手法の詳細について述べる。4 章ではシステムの実装について述べ、5 章では割当精度の評価を行う。最後に 6 章で本論文のまとめと今後の課題について述べる。

2. 想定環境

本研究の想定環境を図 1 に示す。既存のインターフェースであるキーボードやマウスだけでは操作が煩雑であったり操作が退屈な場合に、TUI を併用してアプリケーションを操作する。

本研究では TUI のデバイスとして Pin&Play を用いる⁶⁾。Pin&Play は図 2 に示すように、通信および電力供給を行うボードとそこに配置するピン型デバイス（図 3）からなる。ボードは 5 層からなり、第 2 層と第 4 層には導電性の繊維の布地、第 1 層、第 3 層、第 5 層は絶縁性の素材である。第 2 層は通信と電力供給のための層であり、第 4 層はグラウンドの層である。デバイスは 2 つの端子をもち、ボードに刺し込まれるとそれぞれの端子がボードの各導電層と接続される。また、デバイスは ID をもち、スイッチやスライドといった部品の他に LED を備えている。Pin&Play のボードは、定期的にボード上に存在するデバイスをチェックすることでデバイスの状態の変化や新たなデバイスの追加、デバイスが抜き取られたことを検出できる。また、デバイスの位置検出に関する研究も行われている。本研究では著者らの研究グループで提案した、カメラによる位置検出手法を用いることで、ボード上のデバイスの位置を細かく取得できる⁵⁾。

ユーザはボードを PC に接続し、アプリケーションの機能を想定して、その機能を操作したいデバイスを自由に配置する。配置が終わると提案手法により自動的に機能が関連付けされ、ユーザは TUI を用いてアプリケーションを自由に制御できる。またユーザが操作しているアプリケーションが変わった場合は、自動的にそのアプリケーションに適切な関連付けが行われる。

アプリケーションがどのような機能をもっているかはアプリケーション製作者からメタデータとして提供

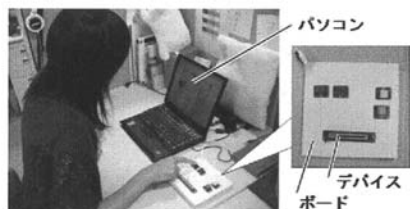


図 1 提案システムの利用例

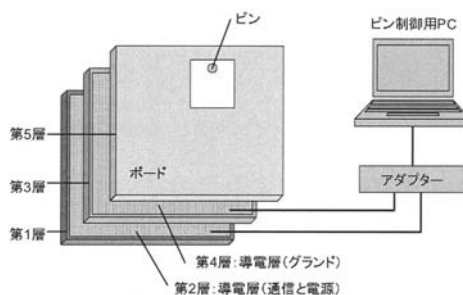


図 2 Pin&Play のシステム構成図

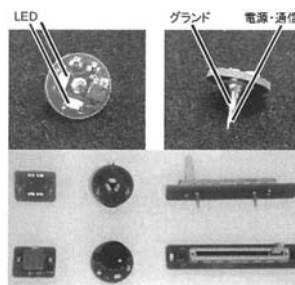


図 3 Pin&Play のデバイス (左からボタン、ダイヤル、スライド)

されるものとする。また、個々のユーザのデバイス配置に関する嗜好は、あらかじめユーザごとに記述されているものとする。GUI でのアプリケーションの表示に近づけようとしたり、操作の頻度の高いデバイスは一番操作しやすい位置に配置したり、普段良く使う他の機器の配置を参考に配置するといったように、デバイスの配置には様々な傾向があると考えられる。

これらの嗜好は、ユーザごとに異なったものになるため、嗜好はユーザごとに記述できるようにする。

3. 提案手法

本研究では PC 上のあらゆるアプリケーションを TUI で直観的に操作できる環境の実現を目的として、アプリケーション機能とデバイスの自動関連付け手法を実現する。



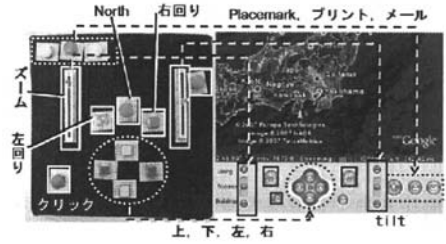
図4 予備実験の様子

3.1 予備実験

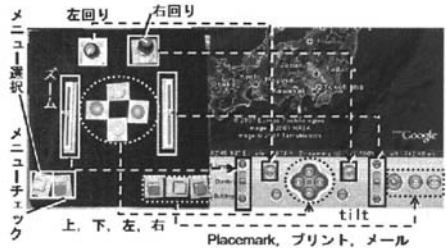
本システムは、ユーザがもつ嗜好をもとに TUI に対するアプリケーション機能の割当を行うシステムである。ユーザの嗜好はパラメータとして表現されるため、どのような種類の嗜好が機能配置に影響を与えるかを明らかにする必要がある。そこで、アプリケーション機能とデバイスの割当やデバイスの配置に関する嗜好を調査し、アルゴリズム設計の方針を決定するために予備実験を行った。実験は 20 代の男女 9 人を被験者として、まず被験者に Pin&Play のデバイスを使って簡単なゲームをさせることで、デバイスの操作感を把握させ、その後下記の手順で実験を行った。まず、被験者にブラウザ、音楽プレイヤー、地図ソフトという一般ユーザが頻繁に用いる 3 種類 8 個のアプリケーションを提示し、それをマウスやキーボードで使わせながら、そのアプリケーションの中で TUI で使いたい機能とその機能に割り当てたいデバイスを自由にボード上に配置させた。次にそれぞれのアプリケーションごとに数種類の機能に限定し、その機能を割り当てたいデバイスを自由に配置させた。

実験の結果、同じアプリケーションに対しても、被験者のデバイス配置にはさまざまなパターンがあることがわかった。図 5 に Google Earth に対して、機能を限定せずにデバイスを配置させた際の実験結果を示す。デバイスの配置に関して、被験者 A と被験者 B がズームとティルトのスライダを GUI での表示順に従ってボードの両端に分けて配置しているのに対して、被験者 C は 2 つのスライダを右端にまとめて配置するなど、機能に対する配置方針が異なることがわかる。一方で、三者とも地図を上下左右に移動させるボタンはその意味どおりに上下左右の位置に配置しており、地図の左回転と右回転のデバイスもそれぞれ左右に配置するなど、操作上の意味を重要視して配置していることがわかる。さらに、上下左右ボタンを隣接させて配置していることから、意味の近い機能は隣接させるという方針があることもわかる。

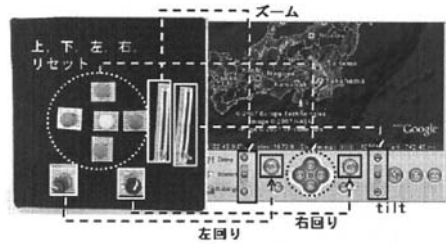
図 6 に Windows Media Player に対して、機能を限定せずにデバイスを配置した際の実験結果を示す。同じ音量を制御するのであっても、被験者 D は GUI



(a) 被験者 A



(b) 被験者 B

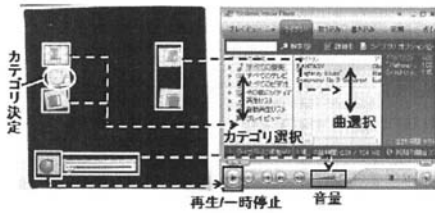


(c) 被験者 C

図 5 GoogleEarth を操作する場合のデバイスの配置

で表示されているボリュームのバーと同じ向きにスライダ配置しているが、被験者 E はダイヤルを利用して音量を調整しようとしている。このように GUI における配置に強く影響されるユーザもいれば、日ごろから使用している機器の形状に影響されるユーザも存在する。また、この例でも、プレイリストの上下を操作するボタンは上下に配置されるというように機能の意味を考慮して配置が行われていること、機能が近いデバイスは隣接して配置されていることがわかる。

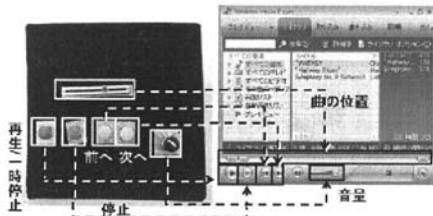
図 7 に iTunes に対して、機能を限定せずにデバイスを配置した際の実験結果を示す。同じ音楽プレイヤーである図 6 と比べて、アプリケーションが変わっても被験者 D は、プレイリストの上下を操作する機能や決



(a) 被験者 D



(a) 被験者 D



(b) 被験者 E



(b) 被験者 E

図 6 WindowsMediaPlayer を操作する場合のデバイスの配置

図 7 iTunes を操作する場合のデバイスの配置

定する機能など、同じ意味の機能を同じ種類のデバイスで操作しようとし、同じ位置関係で配置している。また被験者 E も、ボリュームや再生位置を同じ種類のデバイスで操作しようとしている。一方で被験者 E は、再生ボタンと前の曲を選択するボタンの位置を、GUI 表示での配置に影響されて入れ替えている。このように、同じ意味をもつ機能を操作するデバイスの種類や配置に関して、アプリケーション間で共通であるものと、そうでないものが存在することがわかる。

すべての被験者の評価結果より、被験者が考慮したと思われる配置に関する嗜好を抽出した結果とその適合率を表 1 に示す。結果より、すべての被験者が、機能間の意味が明確な場合はそれによってデバイスを配置し、さらにほとんどの被験者は GUI の配置に近い位置にデバイスを配置することがわかった。また、GUI とは関係なく、個人の嗜好によってデバイスを選択して配置する被験者もいることがわかった。これは「ボリューム＝つまみ」といったように、日常生活における機器の特徴をもとに機能配置しているものと考えられる。

3.2 関連付けアルゴリズムの設計方針

前節で述べた予備実験の結果から、アプリケーション機能のデバイスへの割当においては、(1) 機能に対するデバイスの適性、(2) 機能間の位置関係、(3) 機能間の関連性、の 3 つが重要であり、それぞれの要素はアプリケーション固有のもの、アプリケーションによらず汎用的なものに分けられることがわかった。

それぞれに関して以下に説明する。

機能に対するデバイスの適性 (嗜好 1, 5, 6, 7)

ユーザがアプリケーションの機能を操作するとき、操作に適したデバイスの種類や、デバイスに適した位置は、その機能の特徴に対してユーザがもつ嗜好による。予備実験では、スクロールバーはスライダに割り当てられることが多い、GUI 上の配置に従ってスクロールバーに対応するデバイスは右側や下側に配置する、被験者が左手で操作したい機能は常に左側に配置される、といった傾向があった。

機能間の位置関係 (嗜好 2)

意味的に 2 次元的な位置関係をもつ機能は、その位置関係に従って配置される。予備実験では、地図を上下左右に移動させるボタンは、どのユーザも上下左右の位置関係で配置していた。

機能間の関連性 (嗜好 3, 4)

意味的に近い機能や、対になる機能は、割り当てるデバイスを近くに配置する傾向がある。予備実験では、消音ボタンと音量ダイヤルを隣接して配置する例があった。

上で挙げた 3 つのユーザの嗜好にはそれぞれ、アプリケーション固有のもの、アプリケーションによらず汎用的なものがある。予備実験では、あるユーザはどの音楽再生ソフトでも、停止ボタンと再生ボタンを横に隣接して配置する傾向が見られたが、アプリケーションの GUI 上のボタン配置に影響され、2 つのボ

表 1 実験結果からの被験者の嗜好

| | 被験者の嗜好 | 適合率 |
|---|--|-----|
| 1 | ウィンドウ上の GUI の配置に近い配置にデバイスを置く | 7/9 |
| 2 | 順番に意味がある場合や、並ぶ順番が概念的に決まっているような場合は、その順にデバイスを並べる | 9/9 |
| 3 | 関連性の高い機能を隣接して配置する。 | 9/9 |
| 4 | 関連性の高い機能には同じ形状のデバイスを割り当て、そうでない場合は違う形状のデバイスを割り当てる | 9/9 |
| 5 | 同じ意味をもつ機能には、アプリケーションが変わっても同じ形状のデバイスを割り当てる | 6/9 |
| 6 | デバイスに向きがある場合、その方向とウィンドウ上での変化方向が一致するようにデバイスを置く | 8/9 |
| 7 | GUI の表示とは関係なく、特定の機能は特定の位置に配置したり、特定のデバイスを選択する | 4/9 |

タンの左右が入れ替わる場合があった。

3.3 関連付けアルゴリズム

提案手法では、前節で抽出した方針に従い、下記の手順でアプリケーション機能とデバイスの関連付けを行う。

STEP0 関連付け情報の記述

関連付けを行う前準備として、ユーザの嗜好を、各機能に対するデバイスの適切さを表すスコア (S^d)、デバイスの相対位置に関するスコア (S^p)、機能間の関連の強さ (S^r) としてあらかじめ記述する。 S^d はある機能に対して、絶対位置やデバイスの種類に基づくユーザの嗜好を表すスコアで、 S^p はデバイス間の相対位置に関する嗜好を表すスコア、 S^r は関連する機能に割り当てたデバイス間の、距離に関する嗜好を表すスコアである。絶対位置や距離に関する嗜好は、ボードの各辺を 10 分割し、それぞれの段階ごとにスコアを付加する。スコアはユーザが好むものほど高い値を設定する。またこれらの嗜好を、アプリケーションによらないものと、アプリケーションに特化したものと 2 種類用意する。システムは、操作しようとするアプリケーションに特化した嗜好がある場合はそれを優先して用いることになる。

STEP1 グループ列挙

縦スクロールバーは、スライダやダイヤルのように 1 つのデバイスによる操作だけでなく、2 つのボタンで上下にスクロールさせるという方法も考えられる。そこで、機能が割り当てられる可能性のある、複数デバイスのすべての組合せを列挙する。

STEP2 機能とデバイスの関連付け

アプリケーション機能とデバイス (および複数デバイスのグループ) の全ての組合せを列挙し、以下の式を用いて各組合せの評価値を求める。

$$V_i = \sum_j (S_{ij}^d + S_{ij}^p) + \sum_j \sum_k^{j-1} S_{ijk}^r \quad (1)$$

ここで V_i は組合せ i に対する評価値を意味し、 S_{ij}^d は組合せ i における機能 j を割り当てたデバイスに対するスコア、 S_{ij}^p は組合せ i における機能 j を割り当てたデバイスの位置関係に対するスコア、 S_{ijk}^r は組合せ i における機能 j と k を割り当てたデバイスの距離に関するスコアである。列挙されたすべての組合せに

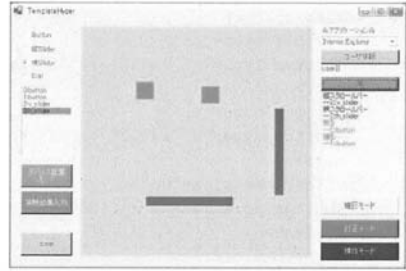


図 8 シミュレータの表示例

対して V を計算し、最大のスコアをもつ組合せを採用する。

STEP3 パラメータ修正

STEP2 で提示した関連付けがユーザの想定と異なる場合、ユーザは間違い訂正モードに入り、誤った関連付けが行われたデバイスをシステムに通知する。システムは、通知されたデバイスの履歴も考慮しながら、正しい関連付けに寄与した全ての嗜好のスコアを一定量増やし、間違いに寄与した嗜好のスコアを一定量減少させることでパラメータを修正する。正しい嗜好情報の中でも S_{ij}^d の絶対位置や S_{ijk}^r の距離に関するスコアについては、段階の近いスコアにも修正値を伝播させることで汎用性を持たせる。修正後、再度関連付けを行い、ユーザの嗜好に合う関連付けになるまでこの操作を繰り返す。

4. 実 装

システムのプロトタイプを実装した。プロトタイプは、想定環境に示したように Pin&Play のボード上にデバイスを配置すると、自動的にフォアグラウンドのアプリケーションの機能を配置したデバイスに割り当て、実際にユーザがデバイスを操作することでアプリケーションが制御できる。各アプリケーションがもつ機能情報に関しては、筆者らが XML 形式であらかじめ記述しておいた。また、図 8 に示すシミュレータも実装し、実際に TUI を用いた場合と同様の機能を PC 内で実現できるようにした。

シミュレータは、スライダ、ダイヤル、ボタンの 3

```

<application>
  <InternetExplorer>
    <v_scrollbar position="right">
      <button position="up"
        key="{PGUP}" />
      <button position="bottom"
        key="{PGDN}" />
      <dial position="" motion="-"
        key="{PGUP}" />
      <dial position="" motion="+"
        key="{PGDN}" />
    </v_scrollbar>
    <prev position="up">
      <button type="m" x="35" y="63" />
    </prev>
    <next position="up">
      <button type="m" x="92" y="63" />
    </next>
    <refresh position="up">
      <button position="" x="166" y="63" />
    </refresh>
  </InternetExplorer>
</application>

```

図9 アプリケーション機能の記述例 (Internet Explorer の例)

```

<function>
  <v_scrollbar>
    <h_position value="same" score="5" />
    <v_slider num="1" color="" score="4" />
    <dial num="1" color="" score="3" />
  </v_scrollbar>
  <h_scrollbar>
    <v_position value="8" score="3" />
    <v_position value="9" score="5" />
    <v_position value="10" score="3" />
    <h_slider num="1" color="" score="4" />
    <dial num="1" color="" score="3" />
  </h_scrollbar>
  <prev>
    <v_position value="same" score="2" />
    <button num="1" color="" score="2" />
  </prev>
</function>

```

図10 機能のもつ特徴の記述例

種類のデバイスを自由に配置できる。ただし、スライダは、ユーザに対して垂直か平行な向きで配置されるものとする。実際に予備実験で、スライダを斜めに配置した被験者は1名のみであった。

シミュレータに入力するユーザの嗜好の記述方式について説明する。図9にアプリケーション機能の記述例を示す。アプリケーション機能を記述する際には、まず application タグの中に個々のアプリケーションの機能を記述する。さらに各機能のタグには GUI 上

```

<relations>
  <relation func1="prev" func2="next"
    h_position="right" v_position="same"
    score="2" />
  <relation func1="up" func2="down"
    h_position="same" v_position="bottom"
    score="2" />
  <relation func1="left" func2="right"
    h_position="right" v_position="same"
    score="2" />
</relations>

```

図11 機能間の位置関係の記述例

```

<links>
  <link func1="up" func2="down"
    v_distance="0" score="2" />
  <link func1="up" func2="down"
    v_distance="1" score="3" />
  <link func1="up" func2="down"
    v_distance="2" score="2" />
  <link func1="right" func2="left"
    h_distance="2" score="5" />
  <link func1="up" func2="right"
    v_distance="2" score="1" />
  <link func1="up" func2="left"
    v_distance="2" score="1" />
  <link func1="down" func2="right"
    h_distance="2" score="1" />
</links>

```

図12 機能間の関連性の記述例

での実際の配置と、そのデバイスが属性に記述された様な動作をした時にアプリケーションに送られるキー又はマウスイベントが記述されている。

機能のもつ特徴の記述例を図10に示す。この記述では、機能に対するユーザの嗜好情報を列挙する。まず function タグの中に個々の機能を記述する。さらに各機能のタグの中には、その機能の位置に対するスコアと、デバイスの種類に対するスコアを記述する。

図11に、機能間の相対位置に関する記述例を示す。ここでは relations タグの中に機能間の相対位置に対する要望を列挙する。func1 属性の機能からみた func2 属性の機能の、水平方向の相対位置を h_position 属性に、鉛直方向の相対位置を v_position 属性に記述する。func1 と func2 の機能間の相対位置が満たされているときに与えるスコアを score 属性に記述する。

機能間の関連性の記述例を図12に示す。2つの機能間の関連性を記述した link タグを列挙する。func1 属性と func2 属性には機能を記述し、機能間の距離を、水平方向なら h_distance 属性に、鉛直方向なら v_distance 属性に記述する。機能を割り当てたデバイス間の距離がこの記述を満たすときに与えるスコアを

表 2 評価実験の結果：割当ての精度

| アプリケーション | 機能数 | ユーザごとの割当てが一致した機能の数 | | | | | | | | | |
|--------------------|-----|--------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | | A | B | C | D | E | F | G | H | I | J |
| Internet Explorer | 6 | - | 6 | 6 | 6 | - | - | 6 | 6 | 6 | - |
| Lunaspape | 6 | - | 6 | 6 | 6 | - | - | 6 | 6 | 6 | - |
| Firefox | 6 | - | 6 | 6 | 6 | - | - | 6 | 6 | 6 | - |
| Google Earth | 5 | - | - | 4 | 5 | 5 | - | 5 | - | 5 | 5 |
| プロアトラス | 5 | - | - | 5 | 0 | 5 | - | 5 | - | 0 | 5 |
| SonicStage | 5 | 1 | - | 3 | 4 | - | 4 | 5 | - | 5 | 5 |
| iTunes | 4 | 4 | - | 2 | 4 | - | 4 | 4 | - | 4 | 2 |
| WindowsMediaPlayer | 5 | 1 | - | 5 | 5 | - | 4 | 5 | - | 5 | 5 |
| 割当成功率 | - | 0.429 | 1.000 | 0.881 | 0.857 | 1.000 | 0.857 | 1.000 | 1.000 | 0.881 | 0.833 |

score 属性に記述する。

これらの記述を、ユーザごとに、また、ユーザの中でもアプリケーションに関わらない記述とアプリケーションに特化した記述との2種類ずつ用意する。上下左右の機能には上下左右の位置に並んだボタンを関連付けるなど、どのアプリケーションにも共通するような嗜好をアプリケーションに関わらない記述に、再生ボタンが一番左に置きたいなどそのアプリケーションだけに對しての嗜好をアプリケーションに特化した記述に列挙する。

システムはこれらの嗜好に基づいてアプリケーション機能とデバイスの関連付けを行う。関連付けがユーザの想定したものと異なった場合は、ユーザはホットキーによりシステムを間違い訂正モードに移行させ、関連付けが誤っているデバイスを操作することにより誤りをシステムに通知する。再びホットキーによりシステムは自動的に再関連付けを行う。

5. 評価実験

PCの操作に習熟した20代の男女10名を被験者とした評価実験を行った。被験者は予備実験の被験者と同じであり、予備実験の結果をもとに筆者らが嗜好情報をあらかじめ用意した。表2に、対象としたアプリケーションとその割当精度を示す。本評価では、代表的なウェブブラウザ、音楽プレイヤー、地図ソフトをそれぞれ数種類用い、機能のある程度限定した上で被験者にデバイスを自由に配置させた。表ではユーザごとに、ユーザが想定した割当とシステムが実際に行った割当が一致した度合いを示している。平均すると、提案アルゴリズムは87%の一致率を実現しており、実用に耐え得る精度であることを確認した。

結果を詳細に見てみると、ユーザIはGUIの配置にデバイスの配置が影響されるユーザ、ユーザE、ユーザGはGUIの配置にデバイスの配置が影響されないユーザであったが、ユーザIにはGUIの影響を考慮した関連付け、ユーザE、ユーザGにはGUIの影響を考慮しない関連付けになるような嗜好情報ファイルになっていたため、ほぼ正しい割当が行われているこ

とがわかる。嗜好を適切に設定することで、割当成功率が100%となるユーザのデータが多く見られた一方で、ユーザDやIのプロアトラスの例のように、関連付けが全く一致しない結果もみられた。これは、現状の割当アルゴリズムが、複数の異なる機能を1つのデバイスに割り当てることを想定していなかったり、対象となるデバイスの動作特性とあまりにも異なる機能が割り当てられたためである。例えば、「戻る」と「次へ」の機能をスライダ1つへと割り当てたユーザがいたが、これは2つの機能を単一のデバイスに割り当てており、しかもアナログ値をとるスライダというデバイスにON/OFFを行う機能を割り当てようとしている。このような割当は本アルゴリズムでは考慮されておらず、したがって連鎖的に他の機能の割当へも影響を及ぼした結果、関連付けの精度が悪くなっていた。この問題を解決するためには、個別にユーザの特殊な要望を記述できる枠組みが必要となる。

6. まとめ

本研究では、TUIを用いてPC上のアプリケーションを簡単に操作するために、アプリケーションの機能とタンジブルデバイスの自動的に関連付けする手法を提案し、プロトタイプシステムの実装を行った。本稿では、予備実験を行うことでユーザのTUI配置に関する嗜好を抽出し、その結果に基づいてアルゴリズムを実装することで、87%の精度でユーザの意図通りのアプリケーション機能割当を自動的に行うことができた。また、割当が間違っている場合にも訂正モードを用意することで、ユーザの嗜好の更新を容易に行えるようにしている。提案システムを用いることで、ユーザはTUIを用いて快適にPC上のアプリケーションを操作できるようになる。

今後はユーザの嗜好を記述することが難しいという問題に対して、簡単な記述ツールの作成や、一般的なユーザ間で共通の嗜好や傾向をテンプレート化するなどユーザに嗜好を記述する負荷を与えない方法を提案する予定である。さらに現状では、ユーザの嗜好はXML形式で記述する際に、一般ユーザにとっては、

自分の嗜好がどのようなものであるかをスコアで表現することは難しく、また直接XML記述を作成することは手間も大きい。そこで、ユーザの嗜好の記述性を高めるための嗜好設定ツール等を用意することを考えており、より効率よくプロフィールを自動構築するアルゴリズムを検討している。また、ユーザが自分の嗜好がわからないという問題に対しては、一般的なユーザ間で共通の嗜好や傾向をテンプレートとして用意しておき、初期段階ではこのテンプレートを用いることでユーザに嗜好を記述する負荷を与えないことを考えている。さらに、本研究ではPin & Playシステムについてのみ実装および評価を行ったが、他のTUIデバイスを用いた場合でも、そのデバイスに対する位置や機能割当への嗜好を記述すれば、提案アルゴリズムはそのまま適用できる。したがって、今後は3次元TUIなど他のTUIデバイスへ提案手法を適用し、また、PC操作の初心者や子供などを対象とした運用評価を行い、提案アルゴリズムの有効性を評価する予定である。

謝辞 本研究の一部は文部科学省科学研究費補助金基盤研究(A)(17200006)、および中山隼雄科学技術文化財団の研究助成によるものである。ここに記して謝意を表す。

参 考 文 献

- 1) Greenberg, S. and Fitchett, C.: Phidgets: easy development of physical interfaces through physical widgets, in *Proc. of the ACM symposium on User interface software and technology(UIST 2001)*, pp.209-218 (2001).
- 2) Ishii, H., Mazalek, A., and Lee, J.: Bottles as a Minimal Interface to Access Digital Information, in *Extended Abstracts of Conference on Human factors in Computing Systems (CHI 2001)*, pp.187-188 (2001).
- 3) Ishii, H. and Ullmer, B.: Tangible bits: towards seamless interfaces between people, bits and atoms, in *Proc. of Conference on Human factors in Computing Systems (CHI 1997)*, pp.234-241 (1997).
- 4) Kitamura, Y., Itoh, Y., and Kishino, F.: Real-time 3D Interaction with ActiveCube, in *Extended Abstracts of Conference on Human factors in Computing Systems (CHI '01)*, pp.355-356 (2001).
- 5) Kishino, Y., Terada, T., Villar, N., Gellersen, W.H., and Nishio, S.: A Position Detection Mechanism enabling Location-aware Pin & Play, *International Journal of Smart Home*, Vol.1, No.1, pp.31-39 (2007).
- 6) Van Laerhoven, K., Villar, N., Schmidt, A., Gellersen, W.H., Hakansson, M., and Holmquist, E.L.: Pin&Play: The Surface as Network Medium, *IEEE Communications Magazine*, Vol.41, No.4, pp.90-96 (2003).
- 7) Piper, B., Ratti, C., and Ishii, H.: Illuminating Clay: A 3-D Tangible Interface for Landscape Analysis, in *Proc. of Conference on Human Factors in Computing Systems (CHI '02)*, pp.355-362 (2002).
- 8) Villar, N., Gilleade, M.K., Ramduny-Ellis, D., and Gellersen, W.H.: The VoodooIO gaming kit: a real-time adaptable gaming controller, in *Proc. of Advances in Computer Entertainment 2006 (ACE 2006)* (2006).