Proc. of the IADIS Interfaces and Human Computer Interaction (IHCI) 2007 conference, DVD-ROM (July 2007).

AN AUTOMATIC MATCHING ALGORITHM BETWEEN DEVICES AND APPLICATION FUNCTIONS FOR TANGIBLE USER INTERFACES

Kazumi Matsui
Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
matsui.kazumi@ist.osaka-u.ac.jp

Tsutomu Terada
Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
tsutomu@ist.osaka-u.ac.jp

Yasue Kishino
NTT Communication Science Laboratories
2-4, Hikaridai, Seika-cho, "Keihanna Science City", Kyoto 619-0237, Japan
yasue@cslab.kecl.ntt.co.jp

Shojiro Nishio
Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
nishio@ist.osaka-u.ac.jp

ABSTRACT

To enable easy and intuitive control of applications there is increasing demand for Tangible User Interfaces (TUIs). To allow use of TUI devices to control applications, a system needs to associate functions with the TUI devices. Several TUI applications and toolkits have been proposed, but it is difficult to use these for various applications since conventional toolkits require the user to perform several inconvenient operations. In this paper, we propose an automatic matching algorithm that can be used between TUI devices and application functions. A user can freely place TUI devices on a board and our system will associate the TUI devices with application functions by considering three types of user preferences: the favorite type of device for each function, an intuitive position for functions, and the relationships between functions. The proposed method achieved 87% accuracy in automatic function assignments.

KEYWORDS

Tangible User Interface, Configuration, Pin & Play

1. INTRODUCTION

Recently developed computer applications usually have advanced functions and provide convenient computing environments. However, the operations to control these applications have become increasingly complicated, and it takes a long time to learn to use such functions. The Tangible User Interface (TUI) is a promising solution to this problem[3]. The TUI enables us to operate computer applications intuitively by manipulating physical objects with our hands. Many TUI systems have been proposed. Musicbottle is a bottle-shaped device that plays a melody when it is opened[2], Sandscape is a new input interface for geographical features that uses real sand[7], and ActiveCube is a block-shaped input device for three-dimensional computer graphics (3DCG)[4]. Users can operate applications easily and intuitively with a TUI as if they were manipulating real objects. However, these systems are usually specialized for a particular application and cannot be widely applied to other applications. There is a need for TUIs that are more widely applicable and much research has been done on using general TUIs to control existing PC applications.

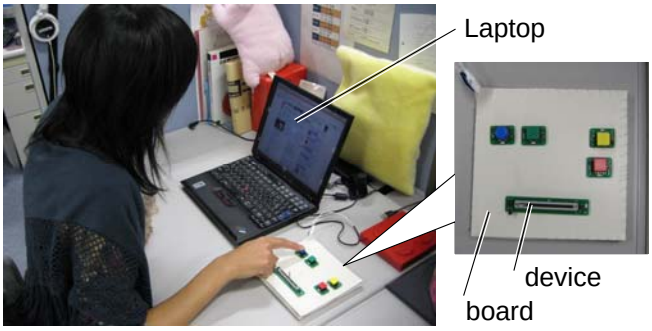


Figure 1. A snapshot of our system being used

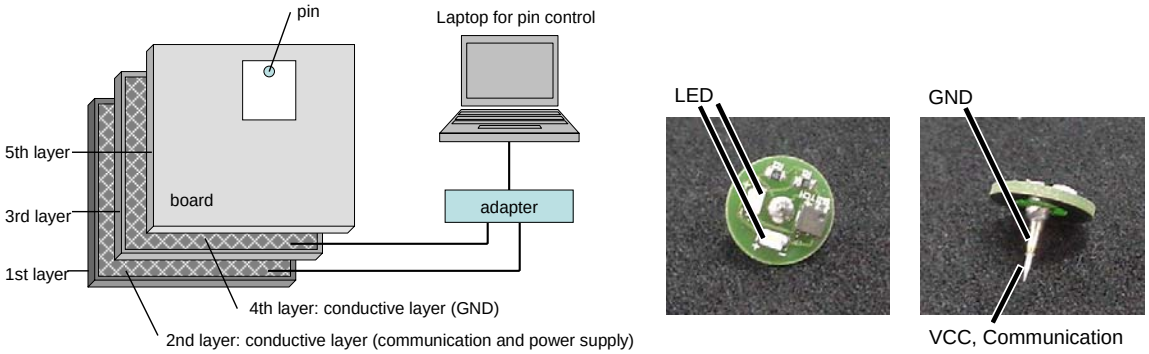


Figure 2. Pin & Play system

The benefit of a TUI is that it enables users to operate applications intuitively; for example, scrolling a screen by moving a real slider up and down is more intuitive than pushing a button on the screen through a mouse operation. Moreover, allocating frequently used functions to real buttons increases an application's intuitiveness and ease of use for beginners because they can carry out an intended function by simply pushing the appropriate button. Several products of this type are available and research is being done to develop new ones, but conventional TUI toolkits have some problems that prevent them being easy to use for various applications. For example, Phidget[1] requires users to have technical knowledge of programming, and VoodooIO[8] requires manual operations to associate devices with functions each time they are used.

We have designed and implemented a system that creates TUIs without requiring the user to perform several annoying operations. The system does this by automatically associating devices with PC application functions that the TUI will be used to operate. In this paper, we discuss a pilot study done to extract the user characteristics of physical device layout and the assignment of functions for each device. We propose an automatic matching algorithm based on the results of the pilot study. The system automatically associates functions with TUI devices according to user preferences and the device alignment. As well as reducing the number of assignment operations, the system enables users to construct a TUI without requiring them to have any programming knowledge. Thus, our system allows users to easily construct, customize, and reuse TUIs.

2. ENVIRONMENTAL ASSUMPTIONS

Figure 1 shows our assumed environment. Users can use a TUI to control PC applications when they want to operate an application intuitively or find the application functions are too complicated or tedious with only a traditional interface such as a keyboard or mouse. In our system, we employ Pin & Play[6,9] TUI devices, which enable users to allocate devices freely on a board according to the application functions and each user's device preferences. The proposed system automatically associates the functions with the TUI devices

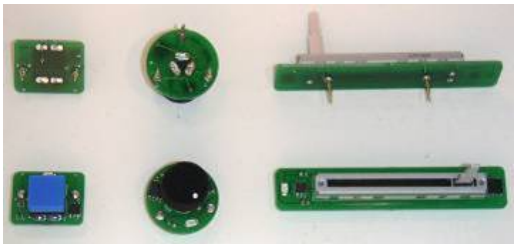


Figure 3. Pin & Play devices (button, dial, and slider)

and the user can then control the application with a TUI. When the assignment result differs from the user's intention, the user can update the user preferences in the system by informing the system of the undesired assignments. When the user changes application, the assignment is recalculated automatically.

Users describe their preferences regarding the function assignments in advance. Each user has his or her own assignment preferences which may be completely different from those of other users. For example, one user might place TUI devices in the same layout as the application GUI, while another would place the most frequently used devices in the most convenient position for operation. In one case, we have observed a user placing devices by considering the actual artifacts having the same functions as the application. Since these preferences vary according to users, the system adopts each user's preferences independently.

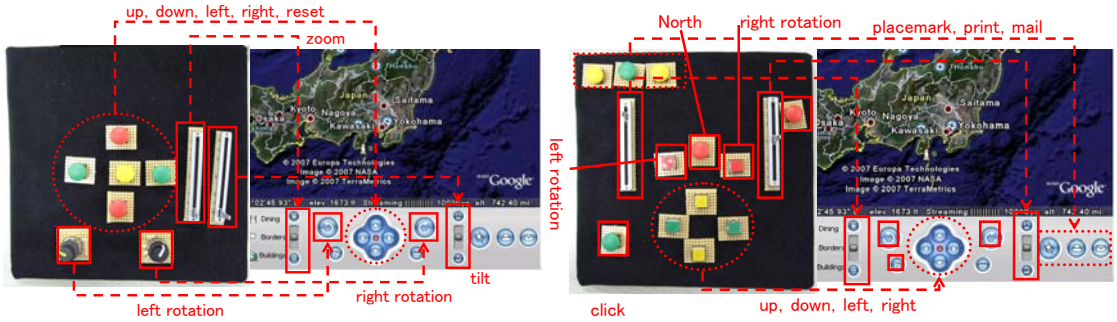
Figure 2 shows the structure of the Pin & Play system, which consists of a board to handle communication and the power supply, and pin-shaped devices (Figs. 2 and 3) that are inserted into the board. The board consists of five layers. The second and fourth layers are made of a conductive fabric while the others are made of an insulator. The second layer is connected to the communication and power supply ports of the adapter and the fourth layer is connected to the ground port. Each device has two ports, and each port is connected to each conductive layer when the device is inserted into the board. Each pin has its own ID and LEDs. The device types include pins, switches, buttons, and sliders (Fig. 3). We earlier proposed an image processing method to detect the position of inserted devices[5]. This is done by capturing camera images of the board. When a new pin is inserted, it blinks so that its position will be detected by the image processing system. Pin & Play devices enable flexible user interfaces since users can attach them anywhere on a board (Fig. 1).

3. PROPOSED ALGORITHM

The purpose of our current research is to construct an environment where we can intuitively and easily operate various PC applications through a TUI. We propose an algorithm for automatic matching between application functions and TUI devices since one of the most difficult and annoying tasks when using a TUI is manually matching functions and devices. Our system determines the linkage between application functions and TUI devices, based on the user's preferences, by considering parameters described in advance to express user preferences. Our method allows users to use applications with a variety of TUI devices without dealing with the time-consuming assignment task.

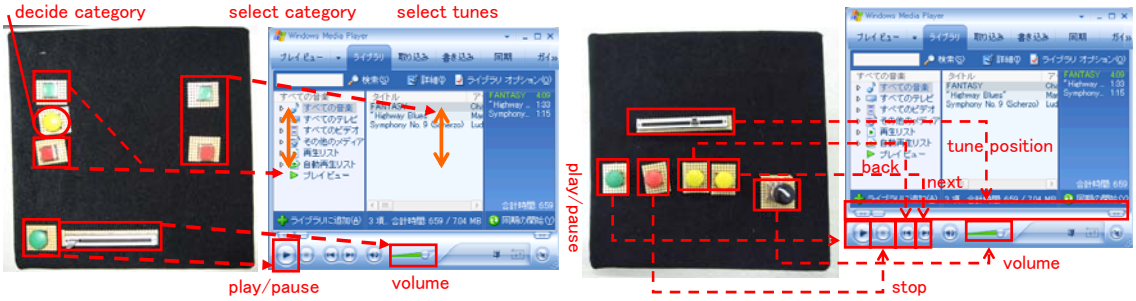
3.1 Pilot Study

Preferences vary among users regarding the allocation of devices and the linkage between devices and application functions, and the variety of these preferences strongly affects our matching algorithm. We performed a pilot study to determine the most important assignment decision factors. The participants were nine university students in their twenties. Each participant played games with Pin & Play to get used to TUI application before the experiment. The procedure of the experiment was as follows. First, we showed the participants several commonly used PC applications such as a WWW browser (e.g., Internet Explorer), a music player (e.g., Windows Media Player), and a mapping application (e.g., Google Earth). Each participant was then allowed to freely locate Pin & Play devices and decide the application function assignment for each



(a) Participant A (b) Participant B

Figure 4. Examples of the layouts for Google Earth



(a) Participant C (b) Participant D

Figure 5. Examples of the layouts for Windows Media Player

Table 1. Participants' preferences extracted from the pilot study

	Preference	Precision
1	**Locating devices according to the position in the GUI**	7/9
2	**Allocating devices in logical orders/layouts**	9/9
3	**Locating devices having similar purposes close together**	9/9
4	**Assigning relevant functions to same-shaped devices and irrelevant functions to different-shaped devices**	9/9
5	**Linking functions having similar purposes to same-shaped devices for every application**	6/9
6	**Aligning devices in the same direction if their purpose is related to direction**	8/9
7	**Locating devices based on personal preferences without considering their GUI position**	4/9

device. The same procedure was then repeated with the application functions to be assigned to devices limited to several of the most common functions.

From the result of the pilot study, we could see various patterns of device layout and function assignments even when functions were limited. Figure 4 shows an example of device layout and function assignment for mapping software without any limitation. As shown, Participant A located two sliders, one for the zoom function and one for the tilt function, together on the right side, while Participant B placed these sliders on opposite sides of the board as in the application GUI layout. In other words, there was a difference between the device-layout policies of the participants. In contrast, both participants adopted the same policy of placing buttons for the map-moving function relative to their meanings (up, down, left, and right) and buttons for right-rotation and left-rotation on the right and left sides, respectively. This shows that people tend to locate devices to emphasize each device's functional meaning. In addition, the participants' placement of the buttons for the map-moving function in a cluster suggests a policy of locating closely related devices within the same positional cluster. Figure 5 shows examples of placement for a music player without any limitation. Participant C assigned a slider for volume control, as in the GUI, while participant D used a dial. This suggests that Participant C was mainly influenced by the GUI layout, while Participant D was more strongly influenced by objects used in daily life. The policy of grouping closely related functions that was apparent in Fig. 4 also seemed to apply in this case.

Table 1 shows the obtained layout preferences of all participants and the frequency for each preference. We found that when there is a clear relationship between functions, people locate devices according to the relationship. Moreover, although most participants located devices in positions similar to the corresponding positions in the GUI, some participants laid out the devices based on personal preferences without considering the GUI layout.

3.2 Matching Algorithm

Based on the pilot study results, we designed our system to associate application functions with TUI devices by considering four factors: the characteristics of each function, the positional relationship among functions, the semantic relationship among functions, and application dependence.

Characteristics of each function: when users operate applications, the devices used and their most suitable position for operation depend on the function characteristics. This policy is based on preferences 1, 5, 6, and 7 from Table 1. In the pilot study, we found that the scrollbar function was always associated with a slider, the position of a slider with a scrollbar function was placed on the right side or at the bottom according to the corresponding GUI position, and functions to be operated by the left hand were placed on the left side.

Positional relationship: devices are intuitively placed by considering the positional relationship. This is based on preferences 2 and 7. In the pilot study, four buttons for map movement were located on the top, bottom, left, and right relative to their directions, and the page-forward button of WWW browsers was consistently located to the right of the page-back button.

Semantic relationship: users tend to cluster devices that have similar purposes or should be coupled semantically, as shown by preferences 3 and 4. In the pilot study, for example, participants tended to locate the mute button close to the volume control dial.

Application dependence: the above three factors include both application-dependent and application-independent preferences. In the pilot study, one user changed the placement of the stop and play buttons according to the type of music player.

3.3 Procedure

We designed an algorithm to match application functions and TUI devices based on the pilot study results. Matching is done through the following steps.

STEP0 Preference Description

Our proposed algorithm needs preference information. Therefore, each user must begin by describing three types of preference: the adequacy of devices for each function (S^d), a positional relationship score for devices (S^p), and the strength of relationship among functions (S^r). Each preference is divided into two categories: a common preference and an application-dependent preference. If a user has the latter preference, it overrides the former preference.

STEP1 Listing Device Groups

Some functions can be associated with multiple devices. For example, a scrollbar can be operated through a slider or through buttons at each end of the scroll bar. In such a case, all groups that can be used to construct a function are listed.

STEP2 Assignment of Functions

The system lists all possible combinations between functions and devices, and calculates each evaluation value for the assignments using the following formula:

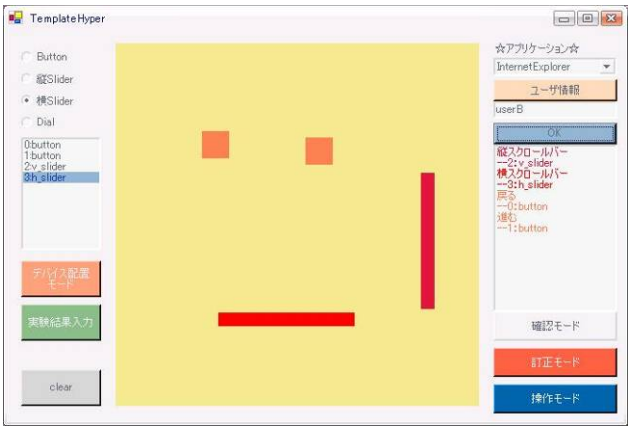


Figure 6. Screenshot of simulator

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

For combination i, V_i is the evaluation value, S_{ij}^d is the score of function j, S_{ij}^p is the positional relationship score for function j, S_{ijk}^r is the score of the semantic relationship between functions j and k, and $d(j,k)$ is the positional distance between devices for functions j and k. The system uses the combination that has the largest evaluation value.

STEP3 Error Correction

When the assignment from STEP 2 differs from the user's intention, the user can enter the error correction mode. This mode allows the user to inform the system of incorrect assignments by touching the TUI devices. The system then updates the preferences and reassigns the functions.

4. IMPLEMENTATION

We implemented a simulator and a prototype system. The simulator was implemented using Microsoft Visual C#.NET on Microsoft Windows XP Professional (a screenshot of the simulator is shown in Fig. 6). A user could place devices (sliders, dials, and buttons) freely in the simulator, and it calculated the assignment of functions by referring to the application function description and preference description. The user could then control applications using TUI devices. The application function description is the list of functions that are candidates for TUI device control. Figure 7 shows an example of an application function description. As an example of preferences, a description example for the characteristics of each function is shown in Fig. 8. It includes scoring information of the devices for each function.

We confirmed the proposed algorithm worked correctly and that we can comfortably control applications with a TUI as shown in Fig. 1. We also confirmed that any incorrect function assignment can be corrected using the error correction function of the proposed system.

5. EVALUATION EXPERIMENT

We evaluated the accuracy of the matching functions for three types of application: three WWW browsers, three music players, and two mapping applications. Participants in this experiment were nine university students in their twenties who were familiar with PC operation. The preferences were prepared beforehand based on the pilot study described in Section 5. In this experiment, the participants could freely locate TUI devices on a board for several specified application functions.

```
<application>
 <InternetExplorer>
  <v_scrollbar position="right">
   <button position="up" type="k" key="{PGUP}" />
   <button position="bottom" type="k" key="{PGDN}" />
   <dial position="" motion="-" type="k" key="{PGUP}" />
   <dial position="" motion="+" type="k" 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>
  <reflesh position="up">
   <button position="" type="m" x="166" y="63" />
  </reflesh>
 </InternetExplorer>
</application>
```

Figure 7. An example of application function description

```
<function>
 <v_scrollbar>
  <position value="same" score="5" />
  <v_slider num="1" color="" shape="" score="4" />
  <dial num="1" color="" shape="" score="3" />
 </v_scrollbar>
 <h_scrollbar>
  <position value="same" score="5" />
  <h_slider num="1" color="" shape="" score="4" />
  <dial num="1" color="" shape="" score="3" />
 </h_scrollbar>
 <prev>
  <position value="same" score="2" />
   <button num="1" color="" shape="" score="2" />
 </prev>
</function>
```

Figure 8. An example of the characteristics of functions

Table 2. Results of evaluation experiment

Application	Number of functions	Number of matching functions				
		Participant A	Participant B	Participant C	Participant D	Participant E
WWW browser A	6	-	6	6	-	-
WWW browser B	6	-	6	6	-	-
WWW browser C	6	-	6	6	-	-
Mapping application A	5	-	-	-	5	5
Mapping application B	5	-	-	-	0	5
Music player A	5	1	-	-	-	-
Music player B	4	4	-	-	-	-
Music player C	5	1	-	-	-	-
Success rate	-	0.43	1	1	0.5	1

Application	Number of functions	Number of matching functions			
		Participant F	Participant G	Participant H	Participant I
WWW browser A	6	-	-	6	6
WWW browser B	6	-	-	6	6
WWW browser C	6	-	-	6	6
Mapping application A	5	-	5	-	5
Mapping application B	5	-	5	-	0
Music player A	5	5	5	-	5
Music player B	4	4	4	-	4
Music player C	5	5	5	-	5
Success rate	-	1	1	1	0.88

The overall matching accuracy was 87%, indicating that the proposed algorithm is suitable for practical use. Table 2 shows detailed results: the number of correct function assignments with respect to the participant's intention, and the success rate for each user. Note that although Participant I located devices according to their position in the GUI, while Participants E and G located them without considering the GUI position, our system achieved high accuracy in most assignments because it reflected each participant's preferences.

Given adequate preferences, our system achieved a 100% success rate in most cases. However, the result was completely different from a participant's intention in some cases, such as for Participant I and mapping application B. The main reason for this happening was a participant assigning undefined functions for the

devices. For example, one participant assigned a WWW browser's page-back and page-forward functions to a slider, but our algorithm does not allow these functions to be assigned to a slider.

6. CONCLUSION

We have designed and implemented an algorithm to automatically match between application functions and TUI devices. After investigating user preferences regarding device location with regard to function assignments through a pilot study, we were able to construct an algorithm that allows user preferences to be applied for function assignment. We then implemented a simulator and an actual system. The simulation study showed an assignment accuracy of 87%, indicating the effectiveness of the algorithm. Using our system, users can therefore comfortably operate PC applications through TUIs.

In our future work, we plan to evaluate use of our system by computer beginners, children, and elderly people. We expect the TUI to be especially useful for such users. Moreover, the current implementation requires description of preference information in XML files. Since it can be difficult to decide on the preferences and this step takes a long time, we plan to create a mechanism that allows the system to construct preferences without any description by users. Specifically, we plan to present several types of template for initial use of the system, and automatically update preferences through the error correction method described in Section 4. In addition, while our current system employs Pin & Play devices as TUI devices, our proposed algorithm is independent of the TUI devices so we will implement the system for other TUI devices and toolkits.

ACKNOWLEDGEMENT

This research was supported in part by a Grant-in-Aid for Scientific Research (A) (17200006) from the Japanese Ministry of Education, Culture, Sports, Science and Technology.

REFERENCES

[1] Greenberg, S., and Fitchett, C., 2001. Phidgets: easy development of physical interfaces through physical widgets. *Proceedings of the ACM symposium on User interface software and technology (UIST 2001)*. pp. 209-218.

[2] Ishii, H., Mazalek, A., and Lee, J., 2001. 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.

[3] Ishii, H., and Ullmer, B., 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. *Proceedings of Conference on Human Factors in Computing Systems (CHI 1997)*. pp.234-241.

[4] Kitamura, Y., Itoh, Y., and Kishino, F., 2001. Real-time 3D Interaction with ActiveCube. *in Extended Abstracts of Conference on Human Factors in Computing Systems (CHI '01)*. pp.355-356.

[5] Kishino, Y., Terada, T., Villar, N., Gellersen, W. H., and Nishio, S., 2005. Position Detection Mechanism using Camera Images for Pin & Play. *in Adjunct Proc. of 17th International Conference on Ubiquitous Computing (UbiComp 2005)*.

[6] Laerhoven, V. K., Villar, N., Schmidt, A., Gellersen, W. H., Hakansson, M., and Holmquist, E. L., 2003. Pin & Play: The Surface as Network Medium. *in IEEE Communications Magazine*, Vol.41, No.4, pp.90-96.

[7] Piper, B., Ratti, C., and Ishii, H., 2002. Illuminating Clay: A 3-D Tangible Interface for Landscape Analysis. *Proceedings of Conference on Human Factors in Computing Systems (CHI '02)*. pp.355-362.

[8] Villar, N., Gilleade, M., K., Ramduny-Ellis, D., and Gellersen, W. H., 2006. The VoodooIO gaming kit: a real-time adaptable gaming controller. *Proceedings of Advances in Computer Entertainment 2006 (ACE 2006)*.

[9] The Pin & Play Project; http://ubicomp.lancs.ac.uk/Pin & Play/.