UNIVERSITY OF MANITOBA

DEPARTMENT OF COMPUTER SCIENCE

MASTER'S THESIS

Identity Awareness on Tabletop Computers

Grant Partridge

May 6, 2011

Abstract

Most multi-user horizontal interactive surfaces, or tabletop *computers, cannot determine which user has performed a given action. These tabletops are less capable than* identity-aware (IA) *tabletops, which can. However, current research on IA is scarce and speculative. Notably, no one has rigorously compared the power of IA and non-IA devices, so evidence that IA enables groups to work better together is lacking.*

My thesis establishes an identity-aware perspective for interactive surface design. First, I have constructed an experiment to determine that IA can improve the effectiveness of small collaborative groups. A second experiment compares several emulation *techniques designed to bring the benefits of IA to non-IA devices. I explore IA in detail through examples, present some open problems involving IA, and discuss promising solutions. Taken as a whole, this document serves as a comprehensive introduction to the study of identity awareness and a springboard for future research on the topic.*

ACKNOWLEDGMENTS

I graciously acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

IdenTTop was implemented in part by Jonathan Cullen, Sean Gustafson, Edward Mak, David McCallum, and Mat Tate.

I appreciate the contributions, suggestions and corrections from my labmates, colleagues, and particularly the members of my thesis committee, namely Drs. Dean Jin and Gordon Fitzell.

My advisor was Dr. Pourang Irani.

dedicated to Denys

CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Figures

1

INTRODUCTION

1.1 What is identity awareness?

The study of interactive surfaces like tabletop computers is an active and exciting area of research. At least one notable conference — namely, the ACM International Conference on Interactive Tabletops and Surfaces [9] — exists solely for the dissemination of research about interactive surfaces, especially tabletops. Researchers and marketers have both emphasized the great potential of tabletops and other interactive surfaces for transforming the way groups of people use computers together [34] [26] [20]. Interest in tabletops and related devices will only increase as the technology becomes more mature and accessible. However, large interactive surfaces have yet to meaningfully affect the day-to-day lives of the public at large. Since tabletops are novel, the population of potential users have few preconceptions about how they ought to work. This puts designers in the enviable position of being able to entertain their creative whims without the constraints experienced by desktop application developers, for instance.

As a result, developers for tabletops face additional choices. To begin, developers of tabletop hardware must choose from a wide variety of fundamental enabling technologies, each with its own benefits and shortcomings. The majority of popular touch surface technologies, including frustrated total internal reflection (FTIR) [12] and diffused illumination (DI) [33], involve a computer vision system that recognizes blobs formed by the outlines of fingers and other objects placed upon the surface. These design strategies have many compelling characteristics that tend to spark the imagination of conference goers and YouTube lurkers alike (including low cost, simple construction and ease of use), yet they all suffer a tremendous disadvantage which is easy to overlook.

Most collaborative digital systems (often known as *groupware*) offer the ability to attribute each user *action* within the system to a particular *actor*. This power can be called *identity awareness*, and we can refer to environments that support this property as being *identity-aware (IA)*[1]. Examples include multiplayer games where each

1 Depending on context, IA either can stand for identity-aware (an adjective) or identity awareness, the associated noun phrase.

player is given her own joystick or chat rooms where every user chooses a handle to represent his identity.

There has never been much of a need to discuss the concept of identity awareness before. A brief review of history reveals that the ability to distinguish users tends to emerge automatically as a consequence of typical multi-user interactive strategies.

1.1.1. A Brief History of Multi-user Systems

The earliest applications to be used by multiple people were probably not multi-user in any special sense. Any single user application can facilitate multiple users, however clumsily. One common approach is to nominate one user as the "driver", who physically interacts with the machine through the available interfaces (e.g., keyboard or mouse) while the other users discuss future courses of action.[2] Applications used in this manner need not address the issue of identity awareness, since only one user is directly interacting with the system at any given time.

The first application that supported multiple simultaneous co-located users was a game, Spacewar [36]. Identity awareness is essential to the traditional game mechanic of controlling opposite forces; it also enables other facilities that we now take for granted, like the ability for the computer to keep score between players

2 Naturally, these social structures are not rigid and a new driver can be appointed – or may appoint himself – at any time.

and select a winner. Besides computer games, electronic mail and other messaging systems were the only significant applications of multi-user systems for decades [17].

The 1980s led to the advent of research in the field of computer-supported cooperative work [28], though the technology to support groupware affordably and effectively did not begin to mature until at least a decade later. In the meantime, Ellis et al. [7] presented a taxonomy that has been used extensively to categorize groupware systems along two axes. The first of these, the space axis, describes whether groupware applications support co-located work (where users are physically near one another) or if they are distributed (connecting users over a network). The second axis, time, distinguishes synchronous applications, where multiple participants use the system at the same time, from asynchronous ones, where users access the system sequentially instead.

Upon considering this two-dimensional schema, it becomes clear that identity awareness is quite often automatic. Purely distributed applications, where each user has her own computer, offer a trivial means to identify each user's actions. Co-located applications, on the other hand, are only able to distinguish between multiple users if each person involved is assigned some sort of *token*. For example, in the case of multiplayer video games, each player is given her own controller, like a joystick or a region of the keyboard. Similarly, along the time axis, asynchronous applications support only a single user at a time, so in many cases, identity awareness need not even be considered. However, synchronous applications, like chat rooms, are much more likely to require the explicit assignment of identity tokens to track the actor responsible for each action. The key observation is that groupware systems featuring high degrees of spatial and temporal sharing—in other words, co-located, synchronous systems, like tabletops and interactive surfaces—tend to require deliberate attention and extra work to support identity awareness.

1.1.2. The Tabletops of Today

Vision-based approaches to tabletop design, including technologies like FTIR and DI, allow multiple users to interact with the system at the same time; however, they offer no token for identification. One advantage of such systems is the walk-up-and-play dynamic that they afford, but this comes at the expense of identity awareness. The recent emergence of these affordable and reasonably effective tabletops is the most important reason to investigate the distinction between IA and non-IA systems.

Identity-aware tabletops do exist, and are somewhat common in the human-computer interaction research community. Around 2001, Mitsubishi Electric Research Laboratories (MERL) created the DiamondTouch, a small tabletop that supports identity-aware interaction for four concurrent users. Every user is seated on a special pad that completes a capacitively coupled circuit when she touches the table. Each seat pad acts as a token for user identification, like the controllers in a multiplayer video game. The paper that introduced the DiamondTouch [6] described a simple reflex game where players compete to pop bubbles of assigned colours. Popping a bubble of the wrong colour incurs a penalty. To properly assign penalties, this game must determine who popped which bubble.

Pape et al. [23] also use multiplayer games as an example to demonstrate how IA tabletops like the DiamondTouch enable interactive applications that are impossible to implement without identity awareness. They suggest that software approaches, such as gesture recognition, could be used to *emulate* IA functionality on devices without native identity awareness. Equally important is their observation that any power gained with an IA tabletop comes at a cost, typically in the form of more complex and expensive technology. Given this, certain applications might not be worth the overhead that would be incurred by choosing an IA device. For example, non-IA systems would be sufficient for the implementation of turn-based games, since only one player should be acting at any given time. Conversely, competitive games that demand simultaneous interaction from all players tend to require IA in order to track the score and prevent cheating.

1.1.3. Benefits of Identity Awareness

A wide variety of capabilities are exclusively available on identity-aware systems. The example of a shared whiteboard drawing task effectively illustrates some of IA's key benefits [32]. In the real world, each member of a group collaborating on a whiteboard can use a different pen colour or tool (like erasers). However, if a computer system is unable to distinguish one user's input from another, it would be impossible to implement the ability for each user to choose a colour and have it applied only to his or her own future actions. This type of functionality is only possible on systems that are identity-aware.

The capabilities enabled by IA are diverse and appealing. An important paper by Ryall et al. [30] outlines the breadth of functionality made available by designing user interface components (or widgets) especially for IA devices. One of the most useful features they discuss is user-level undo, or the ability for each user to revert his or her most recent actions, independent of the actions of others. Other possibilities include the application of fine-grained access control, accommodating variable interaction preferences for different users, or logging the behaviour of each actor for later per-user analysis.

To classify the sorts of functionality enabled by IA, the authors introduce four categories of identity-based enhancements to widgets:

function (changing the way a widget behaves), content (changing the data shown within a widget), appearance (changing the look of a widget), and group input (reacting specially to input from multiple users). They also present a diverse collection of samples for each category, which are valuable demonstrations of the potential benefits of IA platforms.

1.1.4. Research on Identity Awareness

Identity awareness is a powerful capability harnessed by all sorts of multi-user applications, but since it arises automatically in the designs of many groupware systems, it is often taken for granted. On a related note, there is very little literature available that directly addresses the topic of IA. The discussion essentially begins with the documents created by MERL, the developers of the DiamondTouch, which focus on the technological aspects of the platform [6][35][8]. They describe a few toy applications that exploit the sorts of features later described in more detail by Ryall et al. [30]

Much research conducted upon the DiamondTouch platform considers identity awareness only in passing. An example of this is Piper et al.'s work on SIDES, a game designed especially to help children with Asperger's syndrome develop their social skills [26]. IA is elemental in implementing the core rules of the game, especially the constraint that each player may only move his or her own pieces. Though essential, IA is almost invisible in the end product of their research. Another example that demonstrates the typical role of IA in research is the UbiTable, again, from MERL [35]. UbiTable was designed to support the sharing of documents from personal devices like smartphones or notebooks which further reinforces the importance of effective and fluid privacy support.

Morris et al.'s [22] TeamTag study on the virtues of replicated versus shared widgets is similar to work on SIDES and UbiTable, particularly in the sense that IA is an enabling technology of all three projects. However, the work by Morris et al. distinguishes itself by providing guidelines that serve to inform the role of IA in general and how it can be applied in useful ways. They had groups assign descriptive labels to pictures using an application built upon the DiamondTouch platform. Users added labels by touching a widget first, then the photo to be marked. Their study compared a shared widget condition (presenting a single set of widgets to be used by all four participants) with another a replicated widget condition (offering several copies of identical widgets). Replicated widgets are meant to minimize disruption between users, though at a cost of increased clutter. In addition to subjective preference, participants in their study also completed their collaborative tasks more quickly in the replicated condition. This work, which generated theory that is both empirically derived and practically useful, should serve as a model for future research on this and related topics, especially identity awareness.[3]

1.2 MY CONTRIBUTION

In this document, I aim to unite several complementary aspects of identity awareness into a coherent field of study. This begins with the formation of a consistent terminology, including the term *identity awareness* itself. Upon establishing a succinct introduction to the topic of IA, I shall proceed to discuss a number of fundamental issues.

First, I introduce a variety of hardware and software solutions for identity awareness, including IdenTTop, a software library developed by my colleagues and I to greatly simplify the development of identity-aware multi-user applications for an extensible assortment of devices (Chapter 2). Then, I present an experiment designed to evaluate the effectiveness of IA systems versus those without identity awareness. Through this, I produce quantitative, empirical evidence that IA can offer significant small-group tabletop perfor-

3 Technically, the application described in Morris et al. could be implemented without IA, but it would be significantly less convenient, requiring a continuous gesture connecting label to photo. This is an obvious yet particularly effective application of IA, and the results are likely to be applicable to diverse IA systems based on the same interactive principles.

mance benefits in terms of reduced collaborative task completion time (Chapter 3). Next, I evaluate several strategies for emulation of IA on platforms like DI and FTIR tabletops. I derive guidelines that can immediately benefit designers of applications for a wide variety of interactive surfaces (Chapter 4). Following that, I explore the properties of identity awareness in greater detail by comparing and contrasting a diverse collection of both IA and non-IA tabletop applications (Chapter 5). Finally, I describe how the concepts that I introduce in this thesis form a cohesive whole. I illustrate the value of identity-sensitive perspective to application design by introducing a number of open problems and using my work as a basis to speculate about possible solutions (Chapter 6).

The result of my work is a comprehensive foundation that establishes IA's benefits and disadvantages with the aim of supporting further development of IA techniques and applications that best take advantage of IA's diverse, though subtle capabilities.

2

IMPLEMENTING IA ON MULTI-TOUCH SURFACES

2.1 HARDWARE SUPPORT FOR IA

Considering that tabletops are often touted as ideal collaborative tools, it is ironic that they typically lack the ability to distinguish between users. As previously mentioned, there are exceptions, like the DiamondTouch. However, current IA tabletops are less than ideal in several aspects. The DiamondTouch requires that users remain in contact with their identifying token, which is either a chair pad or a wrist strap. This compromises some of the key benefits that people associate with working around a table, like the ability of participants to freely move around the surface. Also, this system, like most IA systems, enforces a fixed user limit. Furthermore, the DiamondTouch is not a true multi-touch device. Each actor may either express a single touch point or define a bounding box, but if any user has more than one simultaneous touch upon the surface, the device cannot recognize the locations of each individual touch.

These compromises demonstrate an inherent tradeoff when considering identity awareness, particularly on systems where special consideration is required to gain IA, such as interactive surfaces. There are costs associated with the capabilities enabled by IA. Moreover, these costs are not just borne by the designers of the system and its applications, but by users as well. IA tabletops tend to be financially expensive, but there are also cognitive costs. The increased complexity of these systems requires a greater mental commitment from users. At a bare minimum, a group intending to use an IA system must assign tokens to users before such features can be used, which is simply not a concern for non-IA applications.

There are a variety of proposed approaches for identity awareness, but none are free of cost or effort. These costs are manifested in a number of ways.

Some solutions rely on special hardware. Roth et al. [29] take a particularly interesting approach to the problem, mounting an embedded system with an infrared LED upon a ring. The devices flash a cryptographic signal which acts as an identifying token for its owner when placed against the table. Any subsequent actions in the near vicinity of the ring will be ascribed to the ring's owner.

Small devices like the IR Ring (as it is called) address the IA problem elegantly, but they could be easily misplaced or damaged. They also must be specially registered with the tabletop before use, and while the device would get smaller and easier to use if widely adopted, the IR Ring is currently slightly too cumbersome to be used comfortably.

Schmidt [32] describes a system that assigns touches to actors using a camera mounted above the surface that recognizes the shapes of actor's hands. However, it is not robust when users' hands are presented at different angles or while performing varying gestures. Furthermore, other users' bodies, hands or arms might interfere with the camera's ability to accurately discern the required detail. Wang et al. [39] present a related approach that uses the orientation of users' hands to determine which side of the table a user is occupying. By associating each actor with a table edge, this technology could easily be extended to provide identity awareness information, although it would similarly suffer from robustness concerns. Walther-Franks et al. [38] introduce the idea of proximity sensors placed along the edge of a surface to determine when users are near. This could conceivably be applied in the same way as Wang's strategy to obtain identity awareness.

2.2 SOFTWARE SUPPORT FOR IA

The costs described so far relate to hardware, but software costs are also an inevitability when introducing identity awareness. In contrast with a non-IA application programming interface (API), every identity-sensitive event must be accompanied with at least one extra parameter to indicate the identity of the current user. Moreover, choosing an API invites a new set of restrictions, prescriptions and abstractions.

DiamondSpin [35] is probably the most visible identity-aware application development toolkit. This Java-based library is undeniably powerful, but its sophisticated capabilities come at an expense to developers, inflating the cost of application development time. Furthermore, it is not surprising that DiamondSpin is designed to take advantage of the power of the DiamondTouch tabletop in particular. Since DiamondTouch tables are neither commonplace nor affordable, this has a deleterious effect on the development of novel applications and interactive techniques. There is a clear value in a system that strives for generality, while still affording the sorts of interaction characterized by IA surfaces.

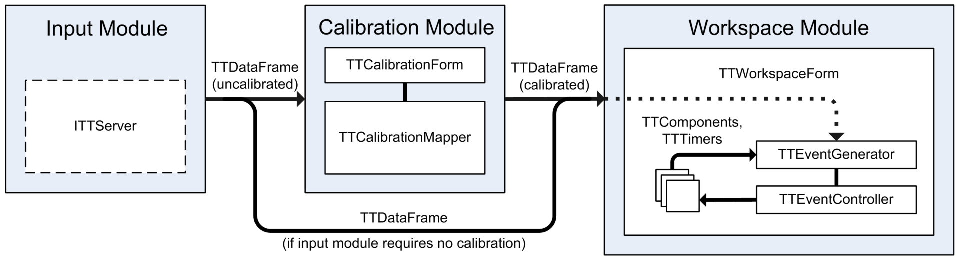

Figure 2.1.: The modules and main classes of IdenTTop.

2.2.1. IdenTTop

To address this concern among others, my colleagues and I have developed IdenTTop, a library for the rapid prototyping and development of identity-aware applications [24]. IdenTTop has many features that reduce the development time of IA software.

IdenTTop is built with the Microsoft .NET framework. By means of a modular architecture, it can support any multi-touch device connected to a Windows machine (see Figure 2.1). The simple API makes it easy to extend IdenTTop to support other devices.

IdenTTop provides two input modules, offering default support for two types of devices. The first input module enables the use of a Polhemus electromagnetic motion tracking device. These devices have been used before in academia to study techniques that require identity awareness [18]. The second module assigns a token to each USB mouse connected to the computer. IdenTTop uses the SDG library to support multiple mice attached to one computer [37]. This makes it possible to develop IA applications on ordinary Windows machines and then port them to a tabletop without any extra code.

A calibration module can be inserted into the signal chain if the data from the input module requires further processing to yield meaningful coordinates. Of the servers packaged with IdenTTop, the Polhemus input module requires calibration, but not the module that supports multiple mice.

The workspace module offers a familiar component-based interface development paradigm, much like Java or Windows Forms. The TTComponent base class offers much of the functionality that one would expect. Several events, like touch down, move, and lift, among others, are exposed as events to which delegate methods can be attached. Assigning a delegate to the Paint event allows free drawing on the canvas using the System.Drawing.NET namespace.

TTComponents have a few capabilities lacking from their typical desktop counterparts that address the unique concerns of tabletop computers. Since users tend to surround a tabletop, components should be orientable so that no users are at a disadvantage. TTComponents can be oriented to face any table edge with a single line of code. TTComponents can also be nested, which allows for the development of intricate reusable interface elements that can then be replicated and rotated to make them available to users all around the table.

Additionally, TTComponents can trivially be adapted to support drag-and-drop. A TTComponent is made draggable by implementing an event, then setting a flag at run-time. Similarly, a component can be designated as a container and given an action to be fired on the event of a dragged object being dropped upon it. Components can be dragged and dropped freely throughout the nesting hierarchy and every component can have a different response to being dropped upon, including rejecting the drop action and reverting the dropped object to its initial position. Combining drag-and-drop with nested components allows for powerful behaviours including automatically adjusting orientation of a dropped component to face the same way as neighbouring elements.

IdenTTop assigns each actor a distinct ID which is used by TTComponents to alter their behaviour based on the actor's identity. Every action event is tagged with that ID. Each ID is also associated with a distinct colour which can be used to consistently identify that user's selections and actions upon the interface. The colours I have chosen are appropriate for general use, but not particularly favourable to colourblind users. In this case, these colours can be globally changed by altering an IdenTTop setting. For devices like mice or the Polhemus that constantly express a value, a cursor is painted on the screen in each actor's colour, which reinforces the association between colour and actor.

Compared to the DiamondSpin library, IdenTTop code is written at a higher level of abstraction, making it possible to build applications with less code. At the same time, I provide the opportunity to access the underlying complexity when necessary to accomplish things that would otherwise be impossible. For example, graphical manipulations are usually accomplished with the familiar and simple GDI+ interface, but when performance and functionality dictates, the DirectX internals of the graphics display system are also available.

IdenTTop's extensibility, portability, and abstractions make it an ideal platform for the investigation of identity awareness. IdenTTop is also robust enough for the development of production applications. The musical instrument WallBalls [25] features an interface constructed with IdenTTop. WallBalls has proven itself stable enough to withstand public exhibition without fear of crashing or otherwise failing at runtime, which suggests the stability of the underlying toolkit.

Having demonstrated IdenTTop's utility through the development of WallBalls, among other applications, I then used IdenTTop's identity-aware features to learn more about the nature of IA itself through a pair of user experiments.

3

QUANTITATIVE BENEFITS OF IA

Identity awareness enables new styles of interaction that are simply impossible without it. It follows that the capabilities of IA systems are a proper superset of what is possible without using IA. I presented several especially illustrative examples of IA's capabilities in subsection 1.1.3, such as the ability for each user sharing a whiteboard to independently select and use tools without affecting the actions of others.[1]

The above example is a classic instance of what Ryall et al. [30] have termed *parallel interleaved modal input sequences*. To properly understand parallel interleaved modal input sequences, each constituent term of that phrase must be processed. Modal input sequences (stripped of the "parallel" and "interleaved" qualifiers) are

1 In some sense, this arises as a matter of definition. If a system is able to handle the preferences for independent users, it must be associating some actions with their actors, demonstrating identity awareness. These ideas are further developed in Chapter 5.

a common paradigm in single-user interfaces. In a modal input sequence, each user action in a series is interpreted in the context (or *mode*) defined by the actions that precede it. On a shared workspace, multiple users might establish modal input sequences in *parallel*. These parallel modal input sequences would then be *interleaved* with respect to one another.

Tool palettes, as seen in graphics editors and word processors, are particularly common and effective means of enabling modal input sequences in single-user environments. These handy widgets present many mutually exclusive options, which can be selected with a single click. Typically, choosing a tool (or colour, typeface, etc.) establishes a mode that frames the user's subsequent actions.

Tool palettes can also be easily extended to support multiple users. If each actor is associated with a colour, for instance, it is sufficient to mark each tool selection with that colour to indicate which actor has selected which option. As a result, tool palettes are an especially appealing approach to enabling parallel modal interleaved input sequence, and by extension, a promising means of demonstrating the potential of identity awareness.

However appealing these extra modes of interaction might be, they have not been shown to improve the effectiveness of groups working together (including parallel interleaved modal input sequences). It may be the case IA-based techniques are no more useful than non-

IA alternatives, despite the additional forms of interaction that IA enables. If and when it has been established that IA's apparent value can translate into real collective performance gains, we can examine which situations are particularly conducive to the application of IA and which stand to gain less benefit.

To gather evidence on whether and when IA can outperform non-IA alternatives, I conducted an experiment on groups of three users at once.

3.1 experiment 1: ia versus non-ia

The primary motivation of this experiment is to determine if IA can offer improved performance versus non-IA in a productive, collaborative task upon a shared workspace. Additionally, I hope to shed some light on the subtle and variable nature of IA in different scenarios.

The following hypotheses informed the design of our study:

1. IA offers measurable benefits versus non-IA alternatives for some tasks.

2. The effectiveness of IA is strongly dependent on the properties of the task to be completed.

To explore these hypotheses, I designed a timed task to be completed cooperatively by a group. Our task design was informed by previous work in the literature, especially the aforementioned TeamTag study by Morris et al. [22]

3.1.1. Experimental Design

Each trial of the experiment presents a of 1024×768 pixel workspace filled with eight different types of 64×64 pixel icons. The three participants share a goal of clearing these icons from the workspace as quickly and precisely as possible. To remove items from the workspace, users must perform a specific action corresponding to the icon. This task is identical to the photo tagging task used in the TeamTag study. Both tasks involve associating items in the workspace with choices from a predefined list. However, I have distilled the task to its basic interaction by removing any elements of subjectivity. By eliminating such potentially confounding factors, I hope to improve the validity and significance of this experiment's results. Additionally, I can vary the parameters of this abstract task to simulate a range of usage conditions.

I evaluated several candidate IA and non-IA techniques, then selected the most promising of each to represent its category. The primary independent variable in our study accounts for the change in technique. For the identity-aware condition, I decided to implement multi-user tool palettes, as they present the parallel interleaved modal input sequences paradigm in a simple, familiar, and effective way. To clear items in this condition, users select one of eight tools, then tap matching icons upon the workspace to remove them. I tested a number of techniques that do not require IA, including labeled bins into which matching items could be dropped and removed. The most effective technique was a modified context menu. Rather than presenting items in a linear fashion, as is the case with common menus, I use an adapted pie menu that minimizes the distance between the point of invocation and each menu item. Furthermore, each target's square shape is ideal for the imprecise nature of tabletop touches. I call it a ring menu because of how it surrounds the target. To clear an icon from the workspace in the non-IA condition, users touch the icon to open the ring menu, which surrounds the touched item. Choosing the menu item that matches the touched icon removes it from the workspace.

Besides interaction technique, I included a second independent variable in the experimental design to simulate a range of tasks. The *arrangement* condition systematically varies the way that icons are arranged within the workspace. The *grouped* setting presents clumps of the same icon type. For instance, one of the eight icons is a pair of scissors. In the grouped arrangement, all scissors icons will appear in the same region. This condition simulates a task with a

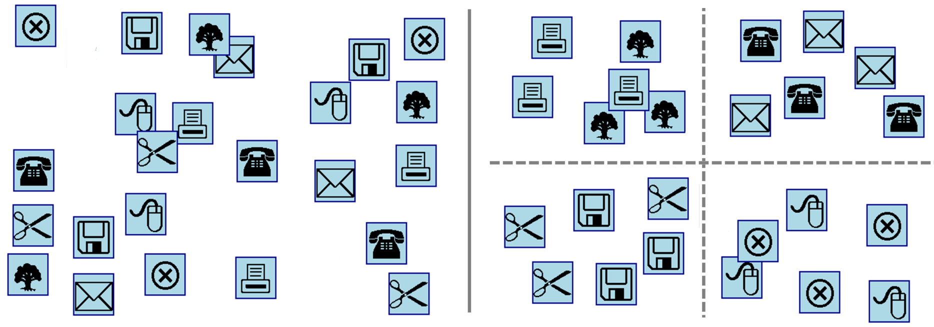

Figure 3.1.: A demonstration of the difference between Uniform (left) and Grouped (right) settings of the *arrangement* variable in Experiment 1. Graphics not to scale; dashed lines added for emphasis.

high degree of spatial differentiation, where different regions of the tabletop have different properties.[2] On the other hand, in the *uniform* setting, each type of icon is evenly distributed around the surface. This simulates compound tasks where subtasks are not spatially constrained. The effect of the arrangement variable is illustrated in Figure 3.1.

Given two independent variables that can both assume two values, my experiment features a 2×2 within-subjects factorial design. Across these two independent variables (technique and arrangement), I measure two dependent variables. First, I record the time it takes for each group to complete each trial. Time is the primary indicator of the effectiveness of a given technique in a particular

2 In real-world situations, semantics of tabletop regions may be prescribed by the application in use; alternatively, they may arise as a social entity.

arrangement. I also record the number of errors that occur in each trial. In the IA condition, an error occurs when a user selects an icon with a mismatched tool. In the non-IA condition, I label instances where users pick the wrong item from the ring menu as errors.

Each trial presents a workspace filled with 10 copies of the 8 icon types for a total of 80 items. I arranged the experiment as a series of blocks. Within a block, all trials shared a common technique and arrangement. Each block contained three trials, and every group of three participants completed two blocks of each combination, for a total of six trials of each. The data for the first two of these trials was excluded from our final analysis, as learning effects were evident. The order in which blocks are presented is counterbalanced with a Latin square. A total of eight groups (24 participants in all, 14 of them males) completed the experiment. All participants were Introductory Computer Usage students at the University of Manitoba who received course credit for their participation. Participants demonstrated various levels of technological aptitude; most used computers on a daily basis, though none considered themselves experts with computers. Since the experiment uses colours to encode identity information, I ensured that all participants were adequately able to distinguish the colours used in this study.

3.1.2. Apparatus

I conducted this experiment on the same platform that we used to develop IdenTTop. The system consists of a plain white tabletop made from particle board, measuring 6′ × 4′ (183 × 122 cm). A ceiling-mounted LCD projector displays an image upon the table, while a Polhemus electromagnetic motion tracker provides up to four users with identity-aware input capabilities. The experimental software was built in .NET upon IdenTTop.

3.1.3. Results

I analyzed the data for this first experiment with a repeated measures ANOVA, aggregated across repeated trials of each combination of arrangement and technique.

First, I discovered a significant main effect of technique upon trial completion time ($F_{1,7} = 26.8; p = 0.001$). Arrangement has a significant effect upon time as well ($F_{1,7} = 942; p < 0.001$). There is also a strong interaction effect upon completion time between technique and arrangement ($F_{1,7} = 68.9; p < 0.001$). Overall, users performed the task slightly faster in the IA setting versus non-IA. However, viewing the time data split by arrangement reveals a more telling story. Non-IA performs comparably to IA in the Uniform arrangement, but IA is much more effective when items are grouped by type. Figure 3.2 (left) demonstrates these three effects visually.

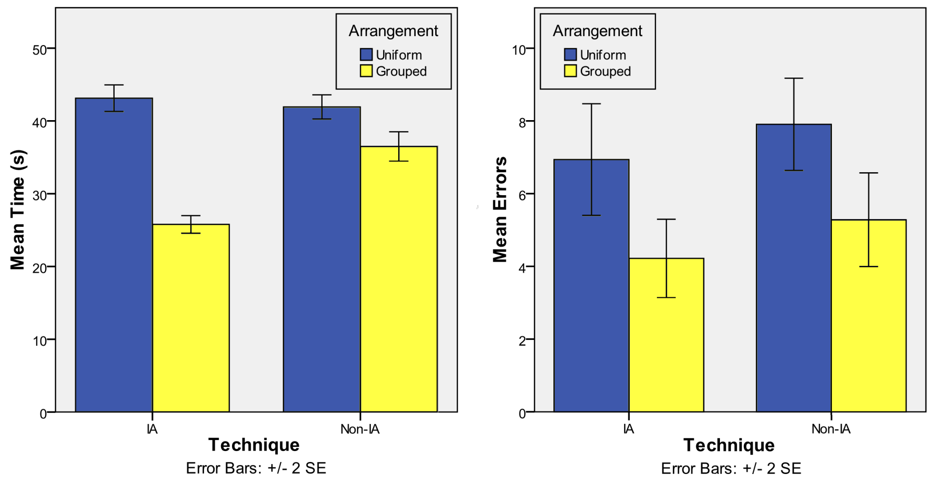

Figure 3.2.: Experiment 1 results: trial completion time (left) and error rate (right) across technique and arrangement.

Examining errors, I see a significant main effect of arrangement ($F_{1,7} = 32.2; p = 0.001$). The Uniform condition had a substantially larger error rate than Grouped. Participants committed slightly fewer errors in the tool palettes than with the ring menu, but this difference was not statistically significant ($F_{1,7} = 3.31; p = 0.112$). Tests revealed no arrangement-technique interaction upon errors.

3.1.4. Discussion

Since completion times for this task in the IA condition are significantly shorter than times in the non-IA condition, I conclude that identity awareness can improve group performance.[3]

3 Again, this assumes that the ring menu is among the best simple non-IA techniques for the task. I made every effort to represent non-IA fairly.

Aside from this finding, the next most important is probably the interaction between technique and arrangement upon trial completion time. The reason for this disparity becomes clear when considering the details of the task. Generally, users need to switch modes much less frequently in the Grouped setting. Regardless of technique, participants tended to focus on the area immediately in front of them, rather than moving around the table perimeter. Additionally, participants would typically remove all targets of one type before switching. In the IA condition, this corresponds to a reduced number of mode switches (i.e. interactions with the tool palette). I observed users employing this algorithm with every combination of technique and arrangement, but it was particularly effective in the Grouped condition with IA. In general, participants would only switch tools two or three times when items were grouped; in the Uniform condition, mode switching was more common. Sometimes, participants would remove all of one object before switching to another in the Uniform condition as well, but this would require a substantial amount of movement around the surface of the table to reach all such items. Both of these options incur an additional time penalty which is not sustained with IA in the Grouped condition, accounting for the especially effective performance with IA.

While interpreting the completion time results is rather straightforward, analyzing the change in error rate across arrangement conditions demands some subtlety. Participants consistently committed less errors in the Grouped condition, regardless of interaction technique. Casual observation of participants indicates that errors with the tool palettes usually occurred after a mode switch.[4] Since the Uniform condition requires more mode switching, this variation in error rate is easy to explain for the IA half of the study.

It is less obvious why the same pattern would be observed for non-IA trials, however. One explanation is that users must mentally "switch modes" regardless of any modality imposed by the software itself. Menu items appear in the same place every time, so users can improve performance with each successive repeated action. When using the tool palettes, users can to select a tool, then clear all nearby items of the same type. I observed that participants took advantage of a similar algorithm to complete the task in the non-IA condition as well — that is, choosing one item type and finishing with it before moving on to the next — which apparently offers advantages in the form of a lower error rate in the Grouped condition.

3.1.5. Conclusions

This experiment was primarily designed to demonstrate that identity awareness offers empirical benefits over equivalently powerful

4 Moreover, these errors would typically occur in clumps; participants would often activate the wrong tool and not realize for several subsequent interactions what they had done.

non-IA alternatives. It also revealed a number of other interesting findings. As suspected, IA gains are strongly task-sensitive. When using parallel interleaved modal input sequences, performance gains are likely to be greater if a task has some degree of spatial coherence, like the Grouped arrangement in this experiment. It follows that simply applying IA to a problem will not necessarily improve performance. Careful consideration of the task at hand is recommended to ensure that the advantages of IA outweigh the costs. In Chapter 5 we investigate the question of which applications stand to benefit the most from IA, among others.

4

EMULATING IA ON TYPICAL TABLETOPS

Now that I have offered evidence that IA can provide quantitative benefits over non-IA systems, a number of secondary questions emerge. Since there is a gap in functionality between IA and non-IA applications, it naturally follows that there is value in finding a way to enable identity-sensitive styles of interaction in an environment that offers no intrinsic IA support. Such a solution would be particularly valuable as the majority of interactive surfaces are not identity-aware. An effective strategy for *emulating* IA on non-IA surfaces would increase the power of all non-IA surfaces by providing developers with a new set of functionality to include in their applications. In addition to the capabilities presented by IA, these non-IA systems, ranging from smaller multi-touch devices like the Apple iPad to large DI and FTIR tabletops, could retain most of their unique advantages as well.

As previously mentioned, the benefits of identity awareness are inextricably linked to a variety of costs. These costs may be financial, in the case of costly hardware, but the development of applications is also made more complicated. Most importantly, users of IA systems (or *actors*) pay cognitive and social costs to satisfy the constraints required to use the IA paradigm. An emulated IA system is still subject to these same sorts of penalties. A successfully designed technique will minimize those costs experienced by the actors themselves. Emulated systems, however, are likely to suffer increased actor costs versus native IA systems (those with explicit support for IA at a fundamental level).

The ideal is clear: we want a system that offers identity awareness with the convenience and interaction style afforded by typical tabletops. Although the technology required to implement this style of interaction hangs tantalizingly close, it is still out of reach. Important progress is being made, however. Holz et al. [13] present work that relies upon a system with sufficient resolution to analyze the fingerprints of users. The impetus behind their system is primarily to increase the precision of users' touches, but the same technology would also enable identity awareness. Unfortunately, the state of the art is still years shy of bridging the gap between a proof of concept and a production-ready system; the prototype built by Holz et al. incurs a delay on the order of seconds, is far too small to be practical, and is prohibitively expensive. In contrast with these future-facing solutions, discovering and implementing a technique to achieve IA on typical multi-touch systems would positively affect tabletop development today. Moreover, since non-IA devices are likely to remain simpler and more common than IA devices, they will continue to be useful as time passes.

4.1 EMULATING IA: A FRAMEWORK

Identity awareness emulation strategies all share a common purpose: to associate actions with the actors that performed them. The means of doing this can vary dramatically between one technique and another. We can classify and compare these techniques along a number of dimensions. One particularly important way to distinguish between types of emulation strategies is based on the degree of user involvement.

4.1.1. Automated IA Emulation

The first class of emulation strategies are the *automatic* emulation techniques. These are automatic in the sense that actors need not perform an additional action to identify themselves. Automatic techniques can either use *heuristics* to interpret various cues as indicators of identity, or impose *constraints* upon user activity such that each action's actor becomes obvious. A simple constraint-based automatic technique is what we have dubbed the "Odd Couple" strategy. Similar to the old comedy trope where sparring siblings or roommates split a shared space by drawing a line through the center, this automatic technique works by dividing the screen into several regions, one for each identity. Any action originating within the bounds of a particular region is then simply ascribed to that region's "owner". This strategy exploits the principle of space-multiplexing to facilitate IA, and takes advantage of people's territorial behaviours when sharing workspaces, as described by Scott et al. [34] Similarly, time-multiplexing could be used to achieve a similar effect; instead of splitting the workspace spatially as in the previous example, consider time slices assigned to each user.

Simple constraint-based techniques are trivial to implement, but more sophisticated techniques can soften the restrictions imposed on user input. These techniques form the class of automatic *heuristic* techniques. As an example, a technique introduced by Mohamed et al. [21] called *Dis*oriented Pen Gestures presents an automatic, alternative approach to constraint-based strategies. This method involves a mathematical analysis to determine which side of the table an actor was occupying when he or she made a gesture. By eliminating options considered to be "cognitively unpopular", it becomes possible to determine which side of the table the actor occupies with a high level of accuracy. Combined with the reasonable assumption that each edge of the table is associated with a user, this technique could be used as a heuristic automatic emulation strategy. Other related techniques include the work of Wang et al. [39] which describes a means to determine the orientation of the finger or hand performing a gesture. Again, it is simple to extend this to determine who is doing what with a high degree of accuracy, yet with more user freedom than constraint-based techniques allow.

4.1.2. Manual IA Emulation

In contrast with automatic strategies, *manual* techniques require actors to perform an additional identifying action. One class of manual techniques includes *signature-based* strategies, where actors supply an identifying gesture (or signature) alongside an action. This gesture might be something as simple as flicking towards the actor (again, assuming that users are anchored to a table edge). This signature-based category of techniques would include context menu or list-based strategies which present a local identifying list adjacent to actions; the only practical difference between menus and a pure signature technique is that menus present familiar visual feedback to coax users into performing the necessary gesture.

The interaction workspaces concept described by Kim et al. [16] should be considered a manual technique as well, although it does not rely upon a signature accompanying every action. With this strategy, users establish a context by defining a bounding box that covers some portion of the workspace. Future actions within that region are ascribed to the user that owns it. This technique cannot properly be considered automatic, since it still requires users to manually identify themselves. Instead, it represents another sub-class of manual strategies, which we shall call *contextual* emulation strategies. This class includes any technique that allows actors to identify themselves by exploiting spatial, temporal, or other phenomena as identifying tokens without explicitly associating a token with each individual action, as with signature-based strategies.

4.2 EXPERIMENT 2: EVALUATING EMULATION STRATEGIES

Several of the techniques listed above emerge as promising candidates for effective IA emulation strategies. Yet, no one technique stands out from the pack as a clear victor. Before application developers will be able to take advantage of IA on typical tabletops, it must be determined which of these strategies are the most viable. This will enable future discussion on the merits and shortcomings of emulated IA and hopefully lead to the discovery of techniques that offer all of IA's benefits while minimizing actor costs.

I have adapted my first experiment to shed some light on the nature of emulated identity awarness. To form a foundation for further examination in this area, I chose a representative technique from both the automatic and manual categories of emulation strategies: the Odd Couple strategy and a simple signature technique (a single stroke in one of the surface's cardinal directions). By focusing on these simple techniques (which can be implemented at the application level, if necessary), we hope to increase the extent to which our results can be generalized and applied to solving real problems in interaction design. Each of my chosen strategies is simple to implement on any multi-touch system, unlike some of the more complicated ones, which rely on extra information unavailable through the ubiquitous TUIO protocol [15] that underpins many DI and FTIR systems.

4.2.1. Experimental Design

This second experiment was based on the first one, with the following changes:

- The number of icon types was increased from eight to nine, to be evenly divisible by the number of participants (3).
- Two new *technique* conditions were introduced, bringing the total to four:

1. Non-IA with *ring* menu
2. Native IA with tool palettes (*tap* to activate)
3. Automatic emulated IA (*divide* the workspace)
4. Manual emulated IA (*pull* in one direction)

- The number of trials per block was reduced from three to two, for a total reduction per condition from six to four.

As before, I present two *arrangements* of items upon the workspace: Grouped and Uniform. In each trial, we record the completion time and error rate.

A total of twenty four people between the ages of 18 and 28 completed the experiment. Fifteen of the participants were female. All were enrolled in an introductory computer usage course at the University of Manitoba and received course credit for taking part. Notably, the participants were generally not students of computer science. Participants had a wide variety of experience with computers, but none had any experience with multi-user surfaces. No participants had any colour vision deficiencies that would affect the experiment.

To account for observed learning effects, the first three of the four trials in each condition were not included in our statistical analysis.

4.2.2. Qualitative Inquiry

Following the experiment, each subject completed a questionnaire to gauge his or her personal opinions about the techniques. Besides an open-ended question asking users to explain their preference for either of the emulated techniques (Divide or Pull), the questionnaire asked the participants to rate each technique upon three different 5-point Likert scales. The first scale measured ease of use in the eyes of the participants, ranging from "very easy" to "very hard". The second scale gauged the ultimate effectiveness or usefulness of each technique, similarly ranging from "very effective" at one extreme to "very ineffective" at the other. The purpose of the third scale was to measure participants' personal preferences, with a range of responses from "I strongly like this technique" to "I strongly dislike this technique".

4.2.3. Apparatus

Experiment 2 features the same hardware and software platform as the first experiment: namely, our custom-built tabletop in conjunction with the IdenTTop library. To support emulated IA, we extended IdenTTop's TTComponent and its event arguments to provide the ID assigned to each action by the emulation technique.

4.2.4. Results

I used a repeated measures ANOVA to probe for main effects. This test reveals an effect of technique upon the time it took groups to complete a trial ($F_{3,21} = 39.2; p < 0.001$). Arrangement also has a significant main effect upon time ($F_{1,7} = 248; p < 0.001$). The interaction between arrangement and technique is a significant factor as well ($F_{3,21} = 8.51; p = 0.001$).

Bonferroni-corrected pairwise tests identify that Pull, the manual emulation strategy, performed significantly slower than all other techniques ($p \leq 0.004$ for all). The difference in time between Ring and Tap (that is, native IA versus non-IA) is nearly significant here ($p = 0.057$), Ring being the slower of the two. This comparison would have been significant if not for the particularly conservative Bonferroni corrections, which is supported by the main finding in our first experiment.

Mauchly's test suggests that assumptions of sphericity hold for technique ($p = 0.061$) and technique-arrangement interactions ($p = 0.065$).[1]

With regards to errors, I see somewhat similar, though weaker, results when compared to completion time. Error rate varies significantly across technique ($F_{3,21} = 3.26; p = 0.042$) and arrangement

1 This test does not apply to the arrangement condition, which only assumes two values.

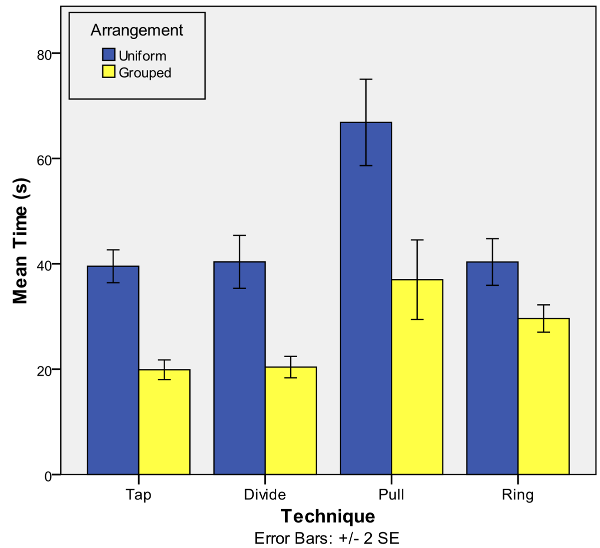

Figure 4.1.: Experiment 2 results: effects of technique and arrangement upon trial completion time.

($F_{1,7} = 8.88; p = 0.021$). There is not, however, evidence of an interaction between technique and arrangement upon error rate ($F_{3,21} = 0.814; p = 0.501$). Bonferroni-corrected pairwise tests of significance have insufficient strength to reveal significant differences in error rate between any pairs of techniques.

Figures 4.1 and 4.2 represent the data collected in this experiment.

Survey Results

I used a non-parametric test to analyze participants' survey responses, which support the quantitative data that I collected. Figure 4.3 presents the mean rank of participants' responses for every

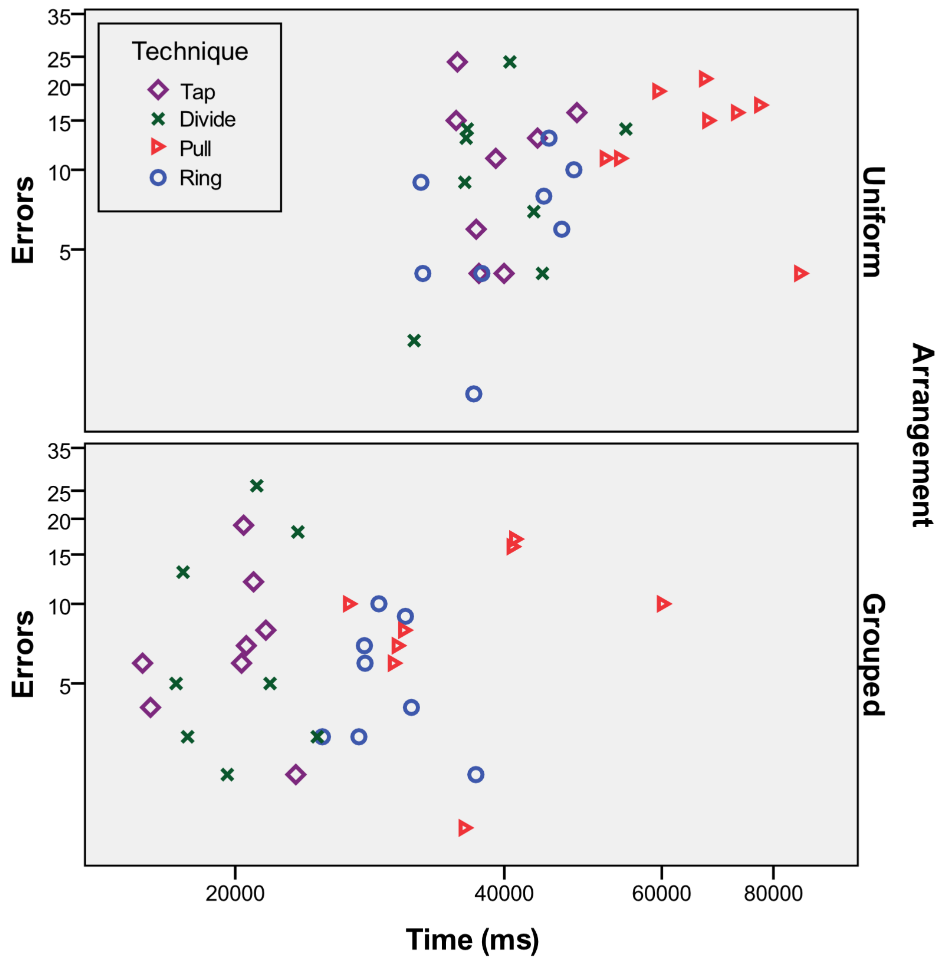

Figure 4.2.: Log-log plot of trial time and error rate across technique and arrangement. Axes have been adjusted to emphasize trend.

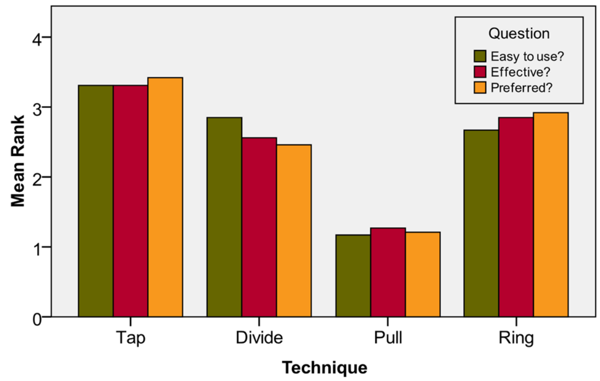

Figure 4.3.: Mean ranks of questionnaire responses by question and technique. Higher values indicate more favourable responses.

combination of question and technique. For each question, the most favourable technique is assigned a rank of 1 and the worst is assigned a 4 (since I am evaluating four techniques in total). In case of ties, ranks are averaged, such that two techniques tied for first will be assigned an effective rank of $1\frac{1}{2}$.

Friedman's test, which is used to analyze rankings as described above, indicated significant differences in response rankings across techniques for each of the three Likert scale questions ($\chi^2_3 > 42.0$, $p < 0.001$ for all). Tap was the highest-scoring technique in all three questions, and Pull was, unsurprisingly, the worst, by a wide margin. Ring (the non-IA technique) and Divide were essentially equivalent in the esteem of participants; measured differences between these two are statistically insignificant.

I also asked participants to freely discuss the differences between Pull and Divide. Their comments offer another dimension of insight to the results. Common themes include the idea that Pull requires more consistent effort. One participant said that Pull "involved a lot more thought" than Divide. Another mentioned that "if they're slower, you can't help your comrades." A third noted how Divide allows each person to focus on the work to be done in his or her region.

4.2.5. Discussion

I begin disseminating my results by examining the mean completion times broken down across the technique and arrangement dimensions. In the Uniform condition, none of the four techniques emerge as a surefire winner: Tap, Divide, and Ring are equally effective, while users struggled with Pull in comparison. Ring was less than a second faster than Tap on average, and the differences between these three techniques were not significant.

Inspecting the Grouped condition, on the other hand, I see the same pattern as demonstrated in the first experiment. Ring is no longer competitive with Tap and Divide, performing somewhere in between those techniques and Pull, which remains the least effective technique. Interestingly, all IA techniques demonstrate improved performance in the Grouped arrangement in contrast with Uniform.

One result worth discussing is the fact that the error rates for Ring are lower than the other three conditions. I failed to observe this in the first experiment between Tap (IA) and Ring (non-IA). This is despite the fact that clearing items with Ring requires two distinct actions: one to activate the relevant item, and a second to match it with a menu selection. The most likely explanation is that when using the tool palettes, participants would not immediately notice that they had the wrong tool selected, and they would commit multiple errors for every mistaken tool selection. The experiment did not offer any extra feedback when errors were committed. It seems, at least, that there is room to improve the error rates of the IA techniques and make them more effective than Ring in this regard. Since participants must return to one of the four corners of the table to change modes using the tool palette, there is a substantial disconnect between selecting a mode and acting on that change of context, which could plausibly account for the elevation of error count in the tool palette conditions versus Ring. One promising option is the use of contextual tool palettes that can be invoked anywhere upon the surface.

During the course of the experiment, I witnessed behaviours consistent across groups that help to explain some of the major quantitative findings that I report. Upon first glance of the scatter-plot in Figure 4.2, aside from the difference between arrangement conditions, the poor performance of Pull is the most striking difference, and is perhaps the most illuminating result of our study. Pull was always the least effective technique; in almost every case, each Pull trial was more than 10 seconds longer than the other three techniques. Pull trials also exhibited higher error rates than the others.

Several factors contributed to this. For one, participants could easily change the tools of others if they were imprecise with their signature. Some participants would occasionally drag towards themselves rather than their assigned cardinal direction, and if they moved from their initial position, their gestures could easily be misrecognized. If the erroneous gesture was performed while choosing a tool from the palette, another user's tool would be changed. In our experimental task, this generally led to a delay of no more than a few seconds, but in mission-critical tasks, such errors are unacceptable. In some situations, this factor alone would be enough to disqualify signature based techniques from consideration. In the Divide condition, errors analogous to the one described above — where one person inadvertently operated outside of his or her region, sending a signal associated with another actor — happened much less frequently, and usually with much less serious side effects. I attribute the instances of "wandering" in the Divide condition to the within-subject design, as Divide strongly resembles Tap; if the experiment did not switch techniques every few minutes, I expect these isolated incidents would disappear entirely.

Generally, each time one actor coded an action with the identity of another, it was done accidentally. However, in very rare circumstances, I witnessed participants deliberately *impersonating* other participants. In the case of Pull, this would manifest itself most frequently when one actor would reset a tool for another actor because someone had accidentally changed it. With Divide, participants could work together outside of their region, as long as there was a common understanding of the current mode (tool) in the active region. These acts of impersonation were entirely innocuous, from what I could tell; participants would do it without asking permission and without raising concern.

To summarize, this experiment revealed a number of interesting things about various emulation strategies (namely, the apparent insufficiencies with signature-based techniques and the effectiveness of automatic ones like Divide), while raising some interesting questions about emulation and IA in general. What does it mean that one user can impersonate another? In the next chapter, we examine questions like this in detail.

5

QUALIFYING THE IA EXPERIENCE

The initial chapters of this thesis have presented an introduction to concepts of IA and an exploration of some of the basic properties of IA systems. In simpler forms, these ideas and issues still provide a useful means to reason about identity awareness, but in reality, these fundamental concepts are more subtle than I have suggested. Thus far, I have posed contrasts such as native versus emulated IA, or even IA versus non-IA, as dichotomies; in truth, however, they are better represented by something like continua.

5.1 VARIABILITY OF IA SYSTEMS

First, it is important to distinguish between *platforms* that offer special support for IA and *applications* that are designed to take advantage of it. Neither of these dimensions is binary; the *degree* of

IA support for both platforms and applications can be expressed over a range of values.

With regard to the hardware and software that constitutes shared-surface platforms, one end of the IA spectrum represents machines like traditional console video games, which are explicitly designed to ascribe almost all input to particular actors. The opposite end consists of devices that offer no such support in any form. If implemented at a systems level, the emulated techniques described in Chapter 4 would fall somewhere in the middle of the continuum, as would systems that provide limited IA capabilities. For example, a heuristic strategy could be implemented on a DI or FTIR table that probabilistically assigns an actor to each action based on the distance of each gesture from the nearest table edge; such a technique falls between both extremes.

Applications are similarly diverse with respect to identity awareness. Some multi-user applications rely completely upon the properties of IA to present an interaction paradigm that could not exist without some form of identity awareness. Many more occupy the hazy middle stretch of the spectrum, in which some features may be distinctly identity-aware, yet possibly in a compromised or adapted form. Many other tabletop applications ignore IA altogether.

One Application, Two Paradigms

While applications and platforms vary in the degree to which they support IA, some applications (and by extension, the platforms that they run upon) can support multiple degrees of IA at once. Seth Sandler's AudioTouch project [31] consists of a set of musical tools and toys that demonstrate some of the various shades of loose identity awareness. The Musical Wong game is a two-player Pong variant that requires each player to implicitly adopt an identity by restricting his or her input. In this way, the application can interpret the state of the application in context of the two users in order to track score, for example. Users are also able to create keyboards that are so tiny as to reasonably accommodate no more than a single user; in this way, users can create an ad-hoc identity-sensitive context by denying other users access using physical means, like keeping objects out of sight or out of reach. The AudioTouch software was designed on a FTIR table, so it is compatible with non-IA shared surfaces, though it demonstrates subtle characteristics of IA.

Applications developed on identity-aware platforms, on the other hand, are able to depend on the reliability and consistency that these systems afford. Still, sometimes these applications can simultaneously demonstrate features common to either end of the spectrum. The collaborative musical instrument WallBalls [25] is a particularly illustrative case. As its name suggests, WallBalls consists primarily of virtual walls and balls; playing the instrument involves manipulating said components precisely and expressively (Figure 5.1). The walls are relatively static elements that present a complex language of interaction: walls can be added, deleted, moved, locked, or damaged. Each wall is also painted with a colour, which determines the sound that is played when the wall is struck by a ball.

To facilitate this wide array of functionality, WallBalls features two sets of tool palettes that permit each user to select a tool and colour/sound independent of the others. Doing so establishes a mode in which future user actions are interpreted; walls can be drawn, moved, or removed with a single click.

On the other hand, balls present a much more limited set of interactions. Balls can be added to the play area with the touch equivalent of drag-and-drop, and launched with a simple flicking gesture. Removing the ball is simply the converse of adding one. Moreover, balls in play are constantly moving; as a result, the interaction style must support rapid, convenient access for efficient use. Balls are implemented in a non-IA context; the semantics of gestures applied to balls are not interpreted within the context of a particular user. Since no mode is required, users can interact with balls with the spontaneity that the task requires.

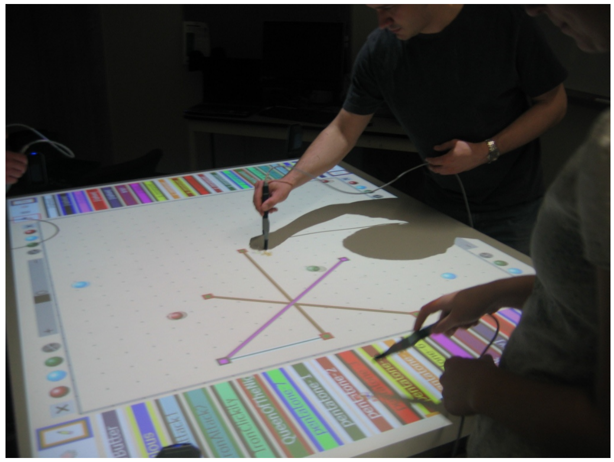

Figure 5.1.: WallBalls, a collaborative musical application implemented on an identity-aware (IA) platform, demonstrates how IA and non-IA features can effectively coexist.

In combination, the IA context presented by the walls is complemented well by the non-IA context embodied by the balls, providing an interface that supports the specific styles of action that both walls and balls demand. In effect, WallBalls simultaneously demonstrates features of IA and non-IA systems, and as such, makes it an interesting tool for examining the differences between the two types of systems.

The walls, which rely on IA to support a diverse variety of interactions, rely on pre-established parameters which are associated with the actor who draws the wall. Hence, there is a strong personal element to the creation of walls and that action's concomitant context. During a session, if each user were spontaneously asked to create a wall, each would likely be associated with a different colour and sound. Indeed, as a developer of WallBalls, I had the opportunity to witness musicians demonstrating some degree of ownership of the walls that they had created, resisting attempts of other group members to change the wall's properties. Furthermore, to realize complicated musical pieces, groups would occasionally plan a wall pattern ahead, then independently set to work creating the infrastructure of walls necessary to program the piece. This was not the case with balls, which tended to encourage a more transient quality which complements their non-IA style of interaction. Each WallBalls player is typically aware of every ball on the table, but balls usually required less planning and attention to detail.

When examining other musical tabletop tools, the importance and relative merits of IA versus non-IA become clearer. The reacTable* [14] is undoubtedly one of the most intriguing digital musical instruments in recent years, not to mention one of the most captivating applications of tabletop technology. Though the instrument presents a radically novel interface, at its heart lies a conventional modular synthesizer. Such synthesizers are notable for exposing a great many parameters, each of which can commonly be controlled in real time independently of the others. The demands of a highly technical performance can exceed the physical capability of a single performer. Successfully realizing a musical piece upon the reacTable*, therefore, might require the tightly-coupled collaboration of as many performers that can fit around the circumference of the surface. The multi-touch interface of the SoundScape Renderer [4] demonstrates similar qualities of control, presenting a large number of control points upon its surface. Each of these parameters can be controlled towards the aim of a goal that transcends the individual. In situations such as these, it makes great sense to avoid user-dependent contexts as exemplified by the *walls* of WallBalls, rather favouring a system like that of the *balls*.

Thusly, a distinction arises between two types of interaction metaphors. On one hand, we have *user contexts*. Such an interaction style allows each user of an application to configure parameters that maximize the effectiveness of subsequent activity (selecting a tool or enabling a mode that applies only to a single user, for instance). On the other hand, we have a *group context* that favours visibility and shared knowledge over individual customizations. This echoes the dilemma presented by Gutwin and Greenberg [11], who emphasize that each concession to improve the capabilities of individual users tends to come at an expense to the group as a whole. In scenarios where high levels of group awareness are important, users should probably not be encouraged to concentrate on working independently. A non-IA context implies that an action has the same semantics regardless of the actor, so the state of non-IA applications can be more readily apparent at a glance. It also means that users can switch roles and incorporate new members into the group freely without having to convey or establish a complicated user context. Here we see one of the inherent compromises of multi-user design in action.

Both user and group contexts offer distinct advantages and suffer from particular drawbacks. The characteristics of both are summarized in Table 5.1.

User Context	**Group Context**
Powerful interaction	Flexible organization
Parallelized workflow	Coordinated workflow
Compatible with IA	Compatible with non-IA

Table 5.1.: Comparing the essential qualities of user and group contexts. Applications can and do exhibit characteristics of both contexts simultaneously.

Generalizations such as these are bound to exclude some cases, but they serve as valuable heuristics that can help developers of tabletop applications to make wise choices with limited information.

5.2 exceptional/unusual ia systems

The spectrum of identity awareness is further clouded by a wide variety of idiosyncratic tabletop applications. Many tabletop games are modeled on the traditional turn-taking paradigm of classic table-top board games like chess [19]. These computerized versions rely on social protocols mediated by an implicit transaction of identity; it is typically assumed that whichever player is currently taking his or her turn is the actor responsible for any action at that time.

Some of these games are then augmented with idiosyncratic hardware, like the Save aMazed Princess game described by Mahmud et al. [1] Players of this game attach electrodes to their ear to measure heart rate and galvanic skin response as indicators of bluffing. While their prototype lacks a sufficient number of electrodes to avoid sharing between players, a production-ready implementation of their game would certainly avoid this limitation. Regardless, this application clearly takes advantage of identity awareness to some degree, in the sense that there is an association between these biometric readings and a particular actor. In the case of this game, only the active player - the one who is taking his or her turn - is being measured to estimate whether or not the player is lying about the value of his or her roll of the dice. However, a game that is not turn-based, or perhaps, allows limited interaction off-turn (such as Monopoly, where players can organize deals, mortgages or trades between turns) could rely on these untraditional channels of information, such as biometric data, to present an identity-rich experience throughout the duration of the game. Imagine a poker game where all players were constantly monitored for markers of deception!

Examples such as these serve to demonstrate how various technologies and modalities of information can support identity awareness to various degrees, which potentially leads to improved collaborative performance, exciting new ways to interact, and totally novel computer experiences. Moreover, they exhibit the considerable variability among applications that support IA in one form or another.

5.3 emergent identity awareness

Finally, with regard to the varying degree to which systems can be considered to support IA, we should note that even systems offering no intrinsic support for identity awareness — which, outside of the context of shared surfaces, are few and far between — still potentially permit *emergent* identity awareness. Emergent IA is witnessed when users adjust their own input to provide a context for their actions. Anonymous web forums present an interesting example. The imageboard 4chan [2] (and others like it) thrive upon their anonymity, often giving people a platform to voice opinions they would not like to associate with themselves.[1] This anonymity can be subverted to various degrees if users decide to self-identify. On an imageboard, this might be accomplished by posting a series of related images (e.g. of the same character). This technique effectively exploits a natural mapping between image and identity and is useful for users who wish to carry on a conversation or an argument.

5.4 FLUIDITY OF IDENTITY

As I observed in Chapter 4, there is an inherent fluidity or looseness associated with emulated IA. One logical identity is assigned to each human actor, but this association between actor (logical) and user (physical) is adaptable. We saw both accidental and intentional impersonation, where one user would perform an action as another actor.

I do not fully grasp the ramifications of these sorts of behaviour, although I can speculate. For instance, it seems entirely possible that a group of two (or more) working finely in sync could share a *logical identity*. Consider the act of surgery, which involves a team of highly coordinated specialists performing a time- and performance-critical

1 Although 4chan provides a field to allow users to specify a handle or alias, convention dictates that the field be left blank so most messages are attributed to Anonymous.

task. The surgeon herself switches between instruments (analogous to modal input sequences), but often does so with the assistance of one or more operating room team members. We can imagine a similar degree of coordination upon the tabletop between a small group of expert users, where one person might switch modes for another. Such activity can conceivably occur within the context of a single logical identity.

This impersonation behaviour illuminates how identity awareness can be adapted to be useful in additional situations. We can imagine a system that goes beyond allowing impersonation and encourages it. The practical result would be a role-based system where each actor (identity) is associated with a set of parameters that are ideal for that actor's role. Users are expected to assume the role of one or more different actors during a session. In this way, we can fully embrace and exploit the divide between physical and logical identity.

5.5 SUMMARY

Simple questions often have complicated answers. Even though identity awareness has a concise, useful definition, in this chapter, I have illustrated that questions about IA which might appear superficially trivial are worthy of additional consideration. Even describing whether a given platform or application supports IA is a task demanding nuance and care. By carefully identifying where IA is used in less obvious ways, it becomes possible to take advantage of IA's special benefits — and to be cautious of its less advantageous consequences, as well.

6

A FUTURE OF IDENTITY-SENSITIVE DESIGN

6.1 IA AND THE DESIGN PROCESS

The capabilities enabled by identity awareness are wide-ranging, as are the drawbacks and subtleties. Although the definition of IA is trivial and its benefits can be easy to appreciate, the previous five chapters have demonstrated that there are many special cases and rules with exceptions that merit consideration when designing an application to best take advantage of IA's multifarious benefits. The fact that IA can offer benefits to group performance on tabletops means that all tabletop developers should pay attention to IA. Moreover, the fact that its benefits are strongly influenced by task-specific factors (as also demonstrated in Chapter 3) should be reason enough to pay deliberate attention to IA early and often in the design process.

While this thesis offers a number of unique and useful results on the topic of IA, its greatest contribution is to provide a fundamental collection of concepts that enable coherent discussion of IA's intricacies. With this set of tools, we can adapt a mindset that fosters what I like to call *identity-sensitive design*.

Identity-sensitive design is hard to define, but it arises naturally from careful consideration of IA principles from the onset of an application or platform's construction. As tabletop programs evolve from the toys used in the academic HCI community to sophisticated solutions to real world problems, the importance of proper identity-sensitive design will dramatically increase.

Just to give one example, security is an especially important concern for multi-user systems. As discussed in Chapter 5, the association of identity to actors is inherently unreliable. In an emulated IA system, it is trivial for one user to perform an action as if he or she were another actor, though it is probably possible on any shared surface.[1] Therefore, special strategies are required in situations like these to protect sensitive data and restrict access to potentially harmful actions. One possible solution is to allow users to lock their identities with a password, biometrics, or some other common security scheme.

1 If one user can physically acquire the identifying token associated with another, he or she can act accordingly. In situations where members of a group cannot be trusted, social protocols are insufficient for preventing malicious behaviour.

Were I developing an application involving sensitive data that invokes the situation outlined above, I could have come to such a conclusion incidentally. However, a proper framework that encapsulates identity-sensitive concepts provides us with the tools to think about this in a more rigorous and abstract way. Having identified an issue involving identity awareness and security, we could (hypothetically) proceed to compare a number of techniques for securing data and identify the benefits and drawbacks of each in much the same way that I compared emulation strategies in Chapter 4. Such is the real value of identity-sensitive design. It is not a replacement for traditional ways of thinking about software development; rather, it is an alternative perspective on the process that complements these other essential elements.

Chapter 5's discussion on user and group contexts is another great example of how a perspective informed by IA can be useful when developing an application. Even casual exposure to the concepts of IA can help users decide whether or not their application is best served by a focus on user and/or group contexts, and thusly, IA and/or non-IA interaction paradigms. Taken to the next level, a deep devotion to the principles of IA can be used to develop theory that all developers of multi-user applications can use to their benefit.

6.2 areas of special concern involving ia

Adopting an identity-sensitive perspective reveals a wide variety of issues that warrant particular attention. Some of these are entirely unique to the domain of identity-aware systems; others are familiar to tabletop application developers, yet are complicated by the introduction of this powerful interaction paradigm. In this document, I make no attempt to offer definitive solutions for these open problems, but I aim to use the principles that I have uncovered thus far to illuminate possible paths to solve them in the future.

6.2.1. Sharing

Techniques such as flicking [27] provide a convenient and intuitive way to move content around the surface of a tabletop. The flicking gesture leverages real-world interaction metaphors; flicking allows users to manipulate virtual objects much like marbles or playing cards. However, flicking suffers from a lack of precision, necessitating the introduction of more complicated strategies like SuperFlick to compensate [27].

Rather than making the flicking gesture more complicated, flicking could be improved by the simple addition of identity awareness. Since the system knows the location of each actor's most recent move, flicks could be "snapped" to the nearest user or target. This is similar to the way that Bubble Cursor works by transfering input activity to the nearest on-screen target [10], except that flicking targets could be either users or objects. In both cases, actions upon invalid targets are applied instead to the nearest valid target.

When using such a strategy, it would be important to clearly suggest the result of each action with visual feedback. The drag-and-pop technique provides inspiration for how to link a distant target with a localized action. With drag-and-pop, a proxy of faraway targets is displayed near the actor, visually connected with a strip evoking a rubber band [3]. In this way, each region of the near input space is cleanly associated with a remote target. To implement an enhanced variant of flick in an identity-aware environment, proxies of each user could be placed near an actor initiating a flick. By gesturing towards these proxies, actors can ensure that flicked objects will reach their target, instead of bouncing aimlessly around the surface of the table.

6.2.2. Security: Permissions and Delegation

In many situations, social mores are sufficient to mediate security concerns amongst a group of active users. This is not always the case, though. Tabletop computers are often deployed in public places, and it cannot always be assumed that all concurrent users of a given application are willing to respect the group dynamic.

Non-IA systems are unable to enforce security policies on a per-user basis. However, with the ability to differentiate between users comes the ability to prevent certain users from performing certain actions.[2]

In classical computing contexts, security is inflexible by design. In applications where security is a major concern, then practically every design decision must be made with this idea in mind, and processes must be diligently followed throughout design, implementation and use. Even so, compromising the security of such systems is often as simple as impersonating a secure identity by taking control of the machine in question, even if by force. Issues like these are greatly magnified on tabletop computers, where a single input device is shared amongst a number of users. Even with a complicated IA system like the DiamondTouch [6], a malicious user can impersonate a powerful actor simply by touching them. In this way, the identifying signal is transmitted through both users' bodies simultaneously and the system is unable to differentiate between the two. Even systems like the IR Ring, designed specifically to bring user-level security to tabletops, suffer from the the same weakness [29].

Granted, in day-to-day use, it can probably be assumed that users will not escalate disputes to physical levels. However, in mission-

2 Any system that accomplishes security on a user level is, by definition, exploiting the benefits of identity awareness. Such applications could then conceivably be extended with other IA capabilities.

critical applications where security is to be strongly enforced, it stands to reason that tabletops might not be an appropriate platform, as they are more susceptible to "identity theft" (for lack of a better term) than most other platforms. Furthermore, since it is possible to elevate security privileges by physically overpowering another user and stealing his or her identifying token, it might very well be the case that tabletop computers encourage aggressive physical behaviour. This is likely to be a particularly serious concern among less restrained user populations, including children.

The inherent flexibility of security on tabletop computers is a crucial disadvantage in certain applications. In others, however, it can be a great feature. For example, consider a situation involving a teacher conducting a learning session upon a tabletop. Perhaps the group is working on mathematical problems involving symbolic manipulation. While the teacher is demonstrating the key concepts, it would be useful to restrict student input. When it is time to perform exercises, it would be useful for teachers to have the ability to delegate control to a single student while others remain unable to modify the workspace. One possible technique to enable the temporary delegation of permissions in a multi-user environment relies on an analogy with a real-world technique to coordinate meetings. In situations where multiple speakers could cause a meeting to fall into disarray, a token such as a stone may be passed around, and the group acknowledges that only the stone holder is free to speak. A similar strategy could be used to control activity among a rambunctious group of users. By preventing all users except the stone holder from manipulating virtual objects, a teacher or other authority figure can more effectively manage a group while maintaining the interactivity that tabletop computers afford.

6.2.3. Identity Labels

One especially overarching concern is how to associate on-screen objects and data with the actor that owns them, either visually or through other means. There are many possible solutions, each with its own virtues and failings.

IdenTTop encourages the use of colour to differentiate between actors. This was an appropriate choice given the characteristics of our particular system. For instance, our tabletop offers a fixed number of simultaneous users (four). Each user interacts with the system using a wired stylus, which allows the system to track user interactions above the surface of the table and constantly display coloured cursors projecting their positions upon the surface. This feedback is sometimes the only way that a user knows which colour is his or hers. If the platform itself does not provide such feedback, it would be necessary to incorporate it at an application level.

Using colour to label actors is less ideal in other situations. Humans lack the ability to readily discriminate between more than a handful of colours, so in situations involving a larger number of simultaneous users, an alternative is required. The limited bandwidth of colour as an information channel also makes it dangerous to permit users to choose their own colours to identify themselves (beyond selecting from a narrow range of options).

In situations where more precise identifying labels are required, text is an option. However, many tabletops lack sufficient resolution to accommodate text labels of any practical size, maintaining legibility while avoiding impractical amounts of on-screen clutter. Moreover, text entry itself is still an open problem on tabletop computers. Applications that otherwise involve no text entry should probably not require users to enter a name or label if an alternative approach will suffice. Finally, orientation issues become problematic on a tabletop where text is concerned, since no one direction can be rightfully referred to as "up".

In this way, orientation of components itself could be used as a scheme for identifying. This assumes each user is facing a different direction. Such a scheme is likely to be less effective in situations where users move around the table, although components owned by a given user could be automatically reoriented to follow them around the table. This may be more distracting than practical.

One promising option is to allow users to associate themselves with a small icon. Such a choice can be made quickly (or automatically, if need be) before a user joins a session. To reinforce identity at a glance, each user can also be assigned a colour.

6.2.4. Supporting Flexible Social Structures

Some social activities enforce a rigid group organization. Board games are good examples. Generally, if a player were to leave in the middle of a game, it will disrupt the activity for the remaining participants. Conversely, people cannot usually join a game in progress. Without careful consideration, IA applications are potentially susceptible to the same fate. For instance, WallBalls requires a consensus among all four actors to end a session and clear the screen, accomplished by having all users press close buttons at the same time. Even if only two people are playing, the system expects four simultaneous touches to clear the screen [25]. Shared systems that assign each user a resource (e.g. screen real estate, or a vote upon a consensus action) stand to benefit by allowing new users to acquire a share of such resources and to free them when users quit mid-session. It might be effective to detect user inactivity and suspend inactive profiles, perhaps allowing users to resume their previous sessions at a later time.

Ultimately, if accommodating a flexible social dynamic is a priority, it probably makes sense to avoid the IA paradigm and all of its associated complexities. Considerations such as these are compelling examples why it is crucial to integrate an identity-sensitive perspective from the earliest stages of the process.

Having acknowledged that groups can gain or lose members, there remains an additional means by which groups alter their structure: members within a group can trade roles. In a system designed to accommodate user contexts by means of identity awareness, it may be useful to facilitate users to exchange contexts with one another. If the software offers no support to allow group members to trade places, users might achieve these means by simply switching identifying tokens, impersonating one another from the perspective of the system. Alternatively, rather than exchanging roles, it might be useful for one user to copy the configuration of another, leaving the second users' settings unaltered rather than receiving the first's. Explicit software support is obviously necessary in this case.

This idea can be extended even further by detaching the concept of the individually identifiable input channels from the roles used to consolidate functionality within the application. In this way, more than one identity can be associated with a given role, and an unnecessary role can remain vacant until it is required. A hybrid approach can also be concocted where some properties are associated with (more rigid) identities and some are attached to (more fluid) roles instead.

6.2.5. Mediating Conflicts of Control

Aside from arrays, vectors and other lists, each variable that controls the operation of a computer program can typically assume a single value. This contrasts with the nature of applications where multiple users are able to simultaneously interact. Every multi-user application adopts a scheme to rectify this incongruity, whether implicitly or explicitly.

One category of approaches can be labeled *preemptive*. A preemptive approach to assigning control to a variable favours more recent gestures. In contrast, the opposite category of *exclusive* strategies use locking to offer precedence to earlier actions. Somewhere in between lie *compromise* or *consensus* strategies which consolidate input from multiple users into a single value.

To clarify these principles with a concrete example, WallBalls features a pair of linked sliders which control a single value — in this case, they are linked to a global speed control. All of the moving elements of the application are accelerated or decelerated in sync with the slider's position. A copy of the speed slider appears on the opposite side of the display to make it accessible to everyone, regardless of users' positions around the table. When one user is manipulating the speed, other users are unable to make similar changes until the first relinquishes control. When either slider is in use, both sliders change colour to indicate that neither can be used again immediately.

Alternatives such as preemptive control were unacceptable in this situation. It was important to preserve smooth adjustments of speed, since WallBalls is a musical instrument, and the elements of the application combine in precise, musical ways. A preemptive strategy would introduce discontinuities into the speed control, which provides an interesting alternative, but one that we considered undesirable in our design. It would have been interesting to incorporate a compromise technique, but WallBalls predates the notion of identity-sensitive design.

Even though WallBalls features IA as an integral part of its design, opportunities remain to improve it as a collaborative tool. I noticed that players of WallBalls would occasionally disagree about which colour (and hence, which sound) should be associated with a given wall in play. While the software was being used regularly, it offered no explicit support for resolving conflicts such as these. Ideally, we would have incorporated more flexible techniques to mediate these sorts of disagreement. One interesting strategy that a system could employ to choose between two options presented by conflicted users is to simply randomly select one of them. In case a losing user emphatically disagrees with the outcome, he or she could simply try again or again, with his or her increased effort corresponding with a mounting desire to win this particular battle.

6.3 Looking forward

While each of these problems are specific and unique, they are all woven together with common threads. At a very basic level, they are all enabled or recontextualized by the addition of identity awareness to multi-user systems.[3] The sorts of functionality enabled by identity awareness have a characteristic flavour. They tend to act in service of a similar goal: namely, helping users tailor their experience to suit their individual needs. As previously discussed in Chapter 5, promoting user contexts can be expected to invoke a cost in the form of a weakened group context. However, this tradeoff can be particularly advantageous for certain types of tasks.

Having a solid understanding of these elements promotes the adoption of an identity-sensitive perspective. Just as a musician needs to learn the basics before advancing to bigger and better things, developers of multi-user applications need to have a grasp

3 Here, we have selected problems with a particular focus on concerns related to tabletop computers. However, most of the issues and suggested solutions apply to many co-located synchronous groupware applications.

on the fundamental concepts of IA to make their applications as powerful and easy to use as possible.

7

CONCLUSION

In an article posted to his website called "Multi-Touch Systems that I Have Known and Loved", lauded HCI expert Bill Buxton reiterates one of his "primary axioms", as he has called them: "Everything is best for something and worst for something else." [5]

This statement applies not only to multi-touch systems, but identity-aware systems as well. Identity awareness is a simple concept with the potential to greatly improve user experience, but for a given application, any gains are far from certain. My first experiment has demonstrated that identity awareness can offer profound benefits in certain situations. On the other hand, I also observed that sometimes, it offers absolutely no benefits.

IA is in no way a panacea or magic bullet; it is simply another instrument to be employed in the development of applications intended for more than one user at a time. Much like any instrument, though, its full potential can only be realized in the hands of an experienced practitioner. Developers of multi-user applications are much more likely to witness the benefits of identity awareness if they understand the best ways to take advantage of IA. Conversely, designers of multi-user experiences must be aware that IA can be blatantly misapplied to great detriment.

While it is far from trivial to effectively leverage the benefits of IA, the technology and its associated effects are so appealing that it is useful to try and apply IA to systems like typical touch surfaces that cannot intrinsically distinguish between users. My second experiment demonstrates the potential for emulating identity awareness on non-IA systems using only the simplest imaginable techniques implemented entirely in software.

Considering its complexity and inevitable compromises, some may question the value of identity awareness. In response, I am inclined to illustrate the virtues of IA by analogy. Object-oriented programming (OOP) has proven itself as a valuable tenet of software architecture. However, strictly speaking, OOP is not necessary to achieve any desired functional outcome. Prior to the innovations of OOP, the paradigm of procedural programming was predominant, and according to the Turing-Church thesis, the two systems are equivalently powerful. However, in contrast with procedural programming, proper OOP greatly facilitates the development of complicated applications. Similar claims can be made of identity awareness. While OOP can improve the implementation of an application with respect to its code, identity awareness can similarly be leveraged to develop enhanced user experiences.

Even though it is possible to build all sorts of applications on a non-IA system, introducing identity awareness can help developers construct superior solutions. Simply introducing IA, however, is unlikely to improve the state of tabletop application development. To promote any real improvements in tabletop applications, it is crucial to properly educate developers so that they are aware of the capabilities of IA, in addition to techniques to leverage IA strategies on systems that offer no native IA support.

Currently, besides multiplayer games, there is little public demand for more systems that support multiple synchronous co-located users. If developers educate themselves about the capabilities of IA and adopt an identity-sensitive perspective on application design, the state of multi-user application development could improve to the point that the general public will demand these new technologies. From that point, tabletop development will flourish and these systems will realize their true capability. This, however, can not occur until typical tabletop systems are able to offer the powerful capabilities required to support sophisticated multi-user experiences. Identity awareness is the way.

Given the simplicity of the concept in its essential form, it is fascinating that identity awareness is such a rich and complex topic to study. I have been greatly rewarded with the opportunity to expound upon the subject and I excitedly anticipate future discourse on IA, its enabling technologies, and its successful applications. Mostly, however, given recent trends towards socialization of computer experiences, I look forward to the time when user experience designers will learn to adopt an identity-sensitive perspective to develop more rich multi-user experiences, and in doing so, elevate social computing to a higher level.

A

SAMPLE EXPERIMENT PARTICIPANT CONSENT FORM

The following page presents a reproduction of the consent form signed by all participants of my second experiment. Participants of the first experiment completed a form that was virtually identical.

[Pinawa] – March 2010
Page 1/1

Thank you for participating in the Pinawa experiment at the HCI Lab!
Please read and complete the form below.

Date / Time: ______________________________________

Participant Number: ______ Participant Name: _________________________________

Student No (**write clearly**): ____

Credit for which course? ________________ Section/Instructor: _________________________________

Informed Consent Agreement

Please read this consent agreement carefully before you decide to participate in the study.

Purpose of the research study: We are comparing two new tabletop interaction techniques with two controls.

Time required: The study will require about 60 minutes of your time.

Risks: The risks associated with this study are incidental and minimal. Users are required to stand for the duration.

Benefits: Students will receive course credit as outlined in agreement with the instructor of the course named above. Read your ROASS or talk to your instructor for more information.

Confidentiality: The information that you give in the study will be handled confidentially. It will be accessible only to researchers working on this project, which may involve faculty members and graduate research assistants.

Voluntary participation: Your participation in the study is completely voluntary.

Right to withdraw from the study: You have the right to withdraw from the study at any time without penalty.

How to withdraw from the study: If you want to withdraw from the study, please inform the experimenter and leave the room. There is no penalty for withdrawing. You will still receive full credit for the study. If you would like to withdraw after your materials have been submitted, please contact one of the researchers listed below.

If you have questions about the study, email *hcilab@cs.umanitoba.ca* or contact the following individuals:

Grant Partridge
Department of Computer Science, E2-560 EITC
University of Manitoba, Winnipeg, Manitoba R3T 2N2, Canada
Phone: +1 (204) 474-8995

Faculty Advisor: Dr. Pourang Irani
Department of Computer Science, E2-580 EITC
University of Manitoba, Winnipeg, Manitoba R3T 2N2, Canada
Phone: +1 (204) 474-8995

Agreement:
I agree to participate in the research study described above.

Signature: ____ **Date:** _____________

Revision Date: March 1, 2010

BIBLIOGRAPHY

[1] Abdullah Al Mahmud, Omar Mubin, Johanna Renny Octavia, Suleman Shahid, LeeChin Yeo, Panos Markopoulos, Jean-Bernard Martens, and Dima Aliakseyeu. Affective tabletop game: A new gaming experience for children. In *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer systems (Tabletop '07)*, pages 44–51, 2007.

[2] Anonymous. 4chan. `http://www.4chan.org`. Accessed on October 14, 2010.

[3] Patrick Baudisch, Edward Cutrell, Mary Czerwinski, Daniel C. Robbins, Peter Tandler, Benjamin B. Bederson, and A. Zierlinger. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. In *INTERACT*, 2003.

[4] Katharina Bredies, Nick Alexander Mann, Jens Ahrens, Matthias Geier, Sascha Spors, and Michael Nischt. The multi-touch SoundScape renderer. In *AVI '08: Proceedings of the*

Working Conference on Advanced Visual Interfaces, pages 466–469, Napoli, Italy, 2008. ACM.

[5] Bill Buxton. Multi-touch systems that I have known and loved. `http://www.billbuxton.com/multitouchOverview.html`, 2009. Accessed on December 15, 2010.

[6] Paul Dietz and Darren Leigh. DiamondTouch: a multi-user touch technology. In *UIST '01: Proceedings of the 14th annual ACM Symposium on User Interface Software and Technology*, pages 219–226, Orlando, Florida, USA, 2001. ACM.

[7] Clarence A. Ellis, Simon J. Gibbs, and Gail L. Rein. Groupware: Some issues and experiences. In Ronald M. Baecker, editor, *Readings in Groupware and Computer-Supported Cooperative Work: Assisting Human-Human Collaboration*, pages 9–49. Morgan Kaufmann, 1993.

[8] Alan Esenther, Clifton Forlines, Kathy Ryall, and Sam Shipman. Multi-touch surfaces: A technical guide. Technical report, Mitsubishi Electric Research Laboratories, November 2002. Technical Report TR2002-48. `http://www.merl.com/publications/TR2002-048`. Accessed on December 8, 2009.

[9] Association for Computing Machinery. Interactive tabletops and surfaces (ITS 2009). http://www.its2009.org, 2009. Accessed on May 21, 2010.

[10] Tovi Grossman and Ravin Balakrishnan. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the 2005 Conference on Human Factors in Computing Systems (CHI 2005)*, pages 281–290, 2005.

[11] Carl Gutwin and Saul Greenberg. Design for individuals, design for groups: tradeoffs between power and workspace awareness. In *CSCW '98: Proceedings of the 1998 ACM Conference on Computer Supported Cooperative work*, pages 207–216, Seattle, Washington, USA, 1998. ACM.

[12] Jefferson Y. Han. Multi-touch sensing through frustrated total internal reflection. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches*, Los Angeles, California, USA, 2005. ACM Press.

[13] Christian Holz and Patrick Baudisch. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *CHI '10: Proceedings of the 28th International Conference on Human Factors in Computing Systems*, pages 581–590, Atlanta, Georgia, USA, 2010. ACM.

[14] Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *TEI '07: Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, pages 139–146, Baton Rouge, Louisiana, USA, 2007. ACM.

[15] Martin Kaltenbrunner, Thomas Bovermann, Ross Bencina, and Enrico Costanza. TUIO: A protocol for table-top tangible user interfaces. In *Proc. of the The 6th International Workshop on Gesture in Human-Computer Interaction and Simulation*, 2005.

[16] KyungTae Kim, Tejas Kulkarni, and Niklas Elmqvist. Interaction workspaces: Identity tracking for multi-user collaboration on camera-based multi-touch tabletops. In *Workshop on Collaborative Visualization on Interactive Surfaces*, 2009.

[17] Keith A. Lantz. An experiment in integrated multimedia conferencing. In *CSCW '86: Proceedings of the 1986 ACM conference on Computer-Supported Cooperative Work*, pages 267–275, Austin, Texas, USA, 1986.

[18] Jun Liu, David Pinelle, Samer Sallam, Sriram Subramanian, and Carl Gutwin. TNT: improved rotation and translation on digital

tables. In *GI '06: Proceedings of Graphics Interface 2006*, pages 25–32, 2006.

[19] Carsten Magerkurth, Maral Memisoglu, Timo Engelke, and Norbert Streitz. Towards the next generation of tabletop gaming experiences. In *GI '04: Proceedings of Graphics Interface 2004*, pages 73–80, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.

[20] Microsoft. Microsoft Surface. `http://www.microsoft.com/surface`, 2010. Accessed on December 15, 2010.

[21] K. A. Mohamed, S. Haag, J. Peltason, F. Dal-Ri, and Th. Ottmann. *Dis*oriented pen-gestures for identifying users around the tabletop without cameras and motion sensors. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 43–52, Washington, DC, USA, 2006. IEEE Computer Society.

[22] Meredith Ringel Morris, Andreas Paepcke, Terry Winograd, and Jeannie Stamberger. TeamTag: Exploring centralized versus replicated controls for co-located tabletop groupware. In *CHI '06: Proceedings of the SIGCHI Conference on Human Fac-

tors in Computing Systems, pages 1273–1282, Montréal, Québec, Canada, 2006.

[23] Sebastian Pape, Laura Dietz, and Peter Tandler. Single display gaming: Examining collaborative games for multi-user tabletops. In *Workshop on Gaming Applications in Pervasive Computing Environments at Pervasive '04*, Vienna, Austria, 2004. `http://www.ipsi.fraunhofer.de/ambiente/pervasivegaming/papers/0402%20pape-dietz-tandler%20gaming%20applications%20-%20final.pdf`. Accessed on December 8, 2009.

[24] Grant Partridge and Pourang Irani. IdenTTop: a flexible platform for exploring identity-enabled surfaces. In *CHI Extended Abstracts*, pages 4411–4416, Boston, Massachusetts, USA, April 2009.

[25] Grant Partridge, Pourang Irani, and Gordon Fitzell. Let loose with WallBalls, a collaborative tabletop instrument for tomorrow. In *Proceedings of the 9th International Conference on New Interfaces for Musical Expression*, pages 78–81, Pittsburgh, Pennsylvania, USA, June 2009.

[26] Anne Marie Piper, Eileen O'Brien, Meredith Ringel Morris, and Terry Winograd. SIDES: a cooperative tabletop computer

game for social skills development. In *CSCW '06: Proceedings of the 2006 20th Anniversary Conference on Computer-Supported Cooperative Work*, pages 1–10, Banff, Alberta, Canada, 2006.

[27] Adrian Reetz, Carl Gutwin, Tadeusz Stach, Miguel Nacenta, and Sriram Subramanian. Superflick: a natural and efficient technique for long-distance object placement on digital tables. In *GI '06: Proceedings of Graphics Interface 2006*, pages 163–170, 2006.

[28] Mike Robinson. Computer supported co-operative work: cases and concepts. In Ronald M. Baecker, editor, *Readings in Groupware and Computer-Supported Cooperative Work: Assisting Human-Human Collaboration*, pages 29–49. Morgan Kaufmann, 1993.

[29] Volker Roth, Philipp Schmidt, and Benjamin Güldenring. The IR Ring: authenticating users' touches on a multi-touch display. In *UIST '10: Proceedings of the 23nd annual ACM Symposium on User Interface Software and Technology*, pages 259–262, New York, New York, USA, 2010. ACM.

[30] Kathy Ryall, Alan Esenther, Clifton For3es, Chia Shen, Sam Shipman, Meredith Ringel Morris, Kathering Everitt, and Frédéric D. Vernier. Identity-differentiating widgets for mul-

tiuser interactive surfaces. *IEEE Computer Graphics and Applications*, 26(5):56–64, 2006.

[31] Seth Sandler. AudioTouch OS by Seth Sandler. `http://sethsandler.com/audiotouch`, 2008. Accessed on May 21, 2010.

[32] Dominik Schmidt. Know thy toucher. CHI Workshop: Multi-touch and Surface Computing. Boston, Massachusetts, USA, 2010.

[33] Johannes Schöning, Peter Brandl, Florian Daiber, Florian Echtler, Otmar Hilliges, Jonathan Hook, Markus Löchtefeld, Nima Motamedi, Laurence Muller, Patrick Olivier, Tim Roth, and Ulrich von Zadow. Multi-touch surfaces: A technical guide. Technical report, Technical University of Munich, October 2008. Technical Report TUM-I0833(2008). `www.multigesture.net/wp-content/uploads/2008/09/bymultitouch.pdf`. Accessed on December 8, 2009.

[34] Stacey D. Scott, M. Sheelagh, T. Carpendale, and Kori M. Inkpen. Territoriality in collaborative tabletop workspaces. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, CSCW '04, pages 294–303, Chicago, Illinois, USA, 2004. ACM.

[35] Chia Shen, Frédéric D. Vernier, Clifton Forlines, and Meredith Ringel. DiamondSpin: An extensible toolkit for around-the-table interaction. In *CHI '04: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 167–174, Vienna, Austria, 2004. ACM.

[36] Scott V. Swearingen. The dawning of virtual environments: Discovery of identity through interaction. Master's thesis, Ohio State University, Columbus, Ohio, USA, 2004.

[37] Edward Tse and Saul Greenberg. Rapidly prototyping single display groupware through the SDGToolkit. In *Proceedings of the Fifth Australasian User Interface Conference*, volume 28, pages 101–110, Dunedin, New Zealand, 2004.

[38] Benjamin Walther-Franks, Lasse Schwarten, Jens Teichert, Markus Krause, and Marc Herrlich. User detection for a multi-touch table via proximity sensors. In *Proceedings of the IEEE Workshop on Tabletops and Interactive Surfaces*, Amsterdam, 2008. IEEE Press.

[39] Feng Wang, Xiang Cao, Xiangshi Ren, and Pourang Irani. Detecting and leveraging finger orientation for interaction with direct-touch surfaces. In *UIST '09: Proceedings of the 22nd annual*

ACM Symposium on User Interface Software and Technology, pages 23–32, 2009.