SecSpace: Prototyping Usable Privacy and Security for Mixed Reality Collaborative Environments

Derek Reilly[1], Mohamad Salimian[1], Bonnie MacKay[1], Niels Mathiasen[2],W. Keith Edwards[3], Juliano Franz[1]

[1]Faculty of Computer Science,
Dalhousie University
Halifax, Canada B3H 3P8

[2] Trifork
Margrethepladsen 4, 8000,
Aarhus, Denmark

[3]GVU Center, Georgia Institute
of Technology
Atlanta GA 30308 USA

{reilly, rizi, bmackay, franz}@cs.dal.ca nm@nielsmathiasen.dk keith@cc.gatech.edu

ABSTRACT

Privacy mechanisms are important in mixed-presence (collocated and remote) collaborative systems. These systems try to achieve a sense of co-presence in order to promote fluid collaboration, yet it can be unclear how actions made in one location are manifested in the other. This ambiguity makes it difficult to share sensitive information with confidence, impacting the fluidity of the shared experience. In this paper, we focus on mixed reality approaches (blending physical and virtual spaces) for mixed presence collaboration. We present SecSpace, our software toolkit for usable privacy and security research in mixed reality collaborative environments. SecSpace permits privacy-related actions in either physical or virtual space to generate effects simultaneously in both spaces. These effects will be the same in terms of their impact on privacy but they may be functionally tailored to suit the requirements of each space. We detail the architecture of SecSpace and present three prototypes that illustrate the flexibility and capabilities of our approach.

Author Keywords

Usable privacy and security, mixed reality, mixed presence, software toolkit, smart room, framework

ACM Classification Keywords

H.1.2; K.6.5

INTRODUCTION

Mixed presence collaboration, or collaboration between collocated and remote individuals, is becoming commonplace. In healthcare, telepresence has long been a topic of research [1], and is now supported by a variety of specialized systems. Virtual classrooms link groups of students and educators across distances, sometimes using immersive video or virtual worlds [2]. In the office, meetings often include remote participants connected via videoconference and/or shared desktop software.

During co-located collaboration, many physical privacy-related actions occur. These include managing the visibility (and sharing) of documents with others in a room, and using one's position and orientation relative to others to glance at personal information in private. During extended collaborations, smaller groups may also wish to achieve visible and/or audible privacy, or signal that they wish privacy based on their position relative to those in the larger group [3]. We are also cognizant of the current and likely future locations and actions of our collaborators and of others. For example, we maintain an awareness of who else is in a meeting room to manage sharing of sensitive information, and rely on social norms inherent in the collaborative activity and/or the environment: for example strangers in a cafe may eavesdrop on our conversation, but they are less likely to walk up to our table and peer intently at the work we are doing. Beyond individual meetings, physical information security policies are often in place in institutions (such as hospitals) that share large amounts of sensitive paper materials, and architecture also considers ways to support the need for privacy and security.

In mixed presence collaboration we have to manage privacy and security across two "channels"—the physical or co-located, and the virtual or remote—simultaneously. The privacy mechanisms used in each channel often differ. For example, once content is shared on the network we become concerned with encryption and access permissions. We are also challenged to maintain situational awareness across both channels. While we may know who is in the room with us, we can often be unaware of who is in the room with our remote collaborator(s). Technologies providing security policy specification and enforcement are often too brittle to apply during synchronous collaboration due to the negotiated and situated nature of privacy in these situations.

Recently, the technology for co-presence has advanced to the point where physical (and possibly virtual) collaborative spaces can be combined into a spatially fused environment (a shared or *blended* space as defined by O'Hara et al. [4]). Mixed reality environments blend physical and virtual spaces, such that they together form a hybrid space [5][6][7]. Mixed reality has been used quite extensively in locative games [35] , and increasingly as a mechanism to encourage a sense of co-presence during mixed presence collaboration [8][9][10]. Mixed reality holds several

EICS'14, June 17 - 20 2014, Rome, Italy

Copyright 2014 ACM 978-1-4503-2725-1/14/06…$15.00.

http://dx.doi.org/10.1145/2607023.2607039

potential benefits as an approach for mixed presence collaboration, including: not requiring expensive and identical technical setups at each linked physical location, not requiring a video presence while still providing a visual representation of collaborators, providing a shared repository for work that can be spatially meaningful, giving a sense of a shared “place” for collaboration that can exist beyond a single meeting, and supporting both asynchronous and synchronous collaboration.

Further, mixed reality offers interesting potential solutions to the multiple channel problems when managing privacy. Specifically, since the virtual/online channel is manifested as a space—and more specifically, the virtual is connected spatially to the physical place where co-located work happens—we might be able to transfer the physical privacy/security mechanisms so that they exist in physical and virtual simultaneously. Consistency in how privacy and security are achieved may make it easier for remote collaborators to understand what is happening in the physical space, and for local collaborators to understand what is happening in the virtual space. It may also reduce the overhead for privacy and security: those collaborating locally act within a single channel (the physical) and remote collaborators act solely in the virtual, knowing that their actions will affect both spaces. Mixed reality collaborative environments pose specific challenges to collaboration, however: actions and representations in a virtual space can be misinterpreted due to a literal interpretation of the spatial metaphor. Even though the physical and virtual may look the same and/or are linked together spatially, the expressive capacities of physical and virtual spaces are very different. Therefore, it is not a straightforward matter of replicating physical privacy mechanisms in the virtual. For example, in the physical world one can selectively show a portion of a document by folding it or holding one’s hand over a sensitive portion of the document, while doing the same in a virtual world would likely require a number of user interactions to secure a portion of a document’s content before displaying what one wishes to share.

According to Bødker [11] and McCarthy and Wright [12] , we should emphasize experience during user-centered design of ICT tools. Often privacy and security mechanisms do not clearly reflect this user-centric approach: instead they focus on establishing secure procedures that users should follow, specifying proper security policies, and providing end-user assistance with these procedures or specifications [13][14]. Our broader research goal is to explore potential designs to support privacy in heterogeneous, document-centric, mixed presence collaboration. In particular, we want to determine how people ‘naturally’ manage security and privacy while performing some of these tasks both in the digital and real world. Therefore we want to use an exploratory approach to look for physical patterns of security-related behavior and to generate and evaluate design ideas pertaining to user-centric privacy and security for mixed reality collaboration. In order to do this, we have developed a framework that allows the rapid development of privacy and security mechanisms that are manifested in both physical and virtual. The primary contribution of our work is the SecSpace framework, permitting rapid prototyping and evaluation of usable privacy and security mechanisms for collaborative mixed reality.

We also introduce several physical privacy and security mechanisms that might be useful in collaborative mixed reality. We illustrate these mechanisms through a set of three SecSpace prototypes. They serve to show the capabilities of the framework, and are not presented here as validated security mechanisms.

The first prototype considers mixed presence collaboration around a whiteboard. This carries a number of implicit security-related issues. For example, we can see who is using a whiteboard and decide whether or not to share information (orally or on the whiteboard). This changes when the whiteboard’s content is mapped onto to a whiteboard in a virtual world.

The remaining prototypes consider mixed presence collaboration around a table. While collaborating around a table participants manage what documents to share and when. When the document content on the table is mapped to a table in a virtual world these practices break down. For instance placing a document in the middle of a table normally implies that collaborators are allowed to view the document, while taking the paper back normally implies that the view permission is now expired. There is no guarantee that the same is achieved in the virtual space.

There are several key challenges to achieving a shared experience in these types of scenarios, in particular:

- how to manifest the physical cues employed by collocated collaborators in the views used by remote collaborators,
- how to enable remote collaborators to easily and naturally generate cues that are visible to and understood by the collocated group, and
- how to ensure that collaborators are aware of how their actions are manifested in the other space.

SecSpace allows researchers to explore ways to address each of these challenges.

BACKGROUND

Co-located Collaboration

A number of privacy and security approaches have been considered for co-located collaboration at a range of physical and temporal scales.

UbiTable [15] provides different levels of security and privacy when sharing documents. UbiTable defines three sharing semantics: private, personal, and public. Private documents will not be accessible or visible to others, personal documents (semi-private) are located on the side of the table close to the owner and can be shared if the owner chooses, and public data are accessible and visible equally to all users.

Semi-Public Display [16] promotes awareness and collaboration in small co-located group environments. Building on practices such as email status reports, shared calendars, and instant messenger status, the display is divided into a space for reminders, a collaboration space, a graphical representation of group activity over time, and an abstract visualization of planned attendance at shared events. The system protects the privacy of group members by using abstract visualizations and icons, such that casual viewers will not easily decipher its contents.

Virtual walls [17] provides a metaphor for user-defined privacy policies in sensor-rich pervasive environments. Users are given control over their digital footprints by defining how "visible" they will be in different regions of the physical space.

Shared Spaces

Shared space techniques seen in research prototypes like VideoArms [18], ShareTable [19], WaaZam! [20] and Carpeno [21] attempt to create the illusion of a single fused space, where interaction is identical in all connected locations; however, this is not possible when one or more parties do not have the required technical infrastructure. Furthermore, when a group of people are co-located and are working (or playing) with just one or two remote collaborators, it may be desirable to allow the collocated group continue to use the spaces and tools in their environment (like real playing cards, for example), while not requiring remote collaborators to do the same. This heterogeneous experience is not supported in many shared space tools, and while some permit alternative setups (VideoArms.[22], [12] for example), they often require significant resources such as calibrated cameras and large screens at each node.

Tools for remote collaboration on the desktop emphasize user-driven privacy and security through explicit sharing settings (e.g. Screen Sharing Item [23] for the Community Bar system). Shared spaces introduce new privacy and security concerns, and the absence of desktop-style interaction requires that we reimagine how to support privacy and security. For example, the ShareTable [16] system consists of video chat and a shared tabletop space. Targeted for communications between a separated parent and their children, it provides facilities for drawing, learning support, and physical document sharing. ShareTable raises some issues about the privacy of those around the users, and of the users themselves, with respect to what they are saying and doing.. To overcome these issues the authors suggested placing the system in the child's room and arranging the best time to make calls, but this kind of measure may not be feasible in all circumstances. While some guidelines exist for managing privacy in always-on media spaces [38], more research is required to identify privacy and security mechanisms for shared spaces and mixed presence collaboration.

Mixed Reality

Benford et al. [24] categorize shared spaces based on three attributes: *transportation, artificiality*, and *spatiality*. Transportation means the possibility of moving a group of objects and participants from their local space into a new remote space to meet and collaborate with others. Artificiality considers the degree to which the environment is synthetic or relies on the physical world. For example, video conferencing is seen as the physical extreme while Collaborative Virtual Environments (CVEs) are seen as the synthetic extreme. Spatiality is the degree of support for physical spatial properties such as containment, topology, distance, orientation, and movement [24]. Mixed Reality can be seen as a form of shared space that combines the local and remote, the physical and synthetic—merging real and virtual worlds to create an environment for physical and virtual objects to interact in real time. While privacy is identified as a concern in mixed reality collaboration, to our knowledge SecSpace is the only reported framework targeting research in this area.

Privacy Approaches in Mixed Reality Environments

To help promote the exploration of how people naturally manage privacy in mixed reality collaborative spaces, we derived five strategies, inspired by our own ethnographic research into privacy issues in office work and healthcare and the co-design of collaborative mixed reality concepts for these domains [32][37]. This is not meant to be a complete list, and we are not recommending that all approaches be present in a single usable privacy solution. Rather, they form a core set of requirements for SecSpace. In the interest of space we list the strategies here: 1. use privacy mechanisms that are appropriate to the physical and virtual worlds, 2. visually represent the current policies in both worlds, 3. build on social norms when negotiating privacy mechanisms between the worlds, 4. enforce privacy mechanisms based on context, and 5. provide simple authentication and permission controls.

SYSTEM IMPLEMENTATION

Creating smart interactive spaces for collaboration has been a research topic in Ubiquitous Computing for some time; iRoom [26], NIST smart room [27],i-LAND [28] are examples of earlier projects in this area. However, these systems are useful for collaboration between people who are located in a single smart space. Connecting a virtual world to a smart room has been proposed as a way to bring mixed presence collaboration to these spaces. Virtual worlds can provide accurate, real time information about the location and orientation of participants and their actions in the virtual world, as a form of virtual sensing [5]. Recent advances in localized indoor tracking of both objects and humans make it possible for physical interactive smart rooms and virtual worlds to behave similarly, and in many respects give the feeling of being in one location to all participants.

SecSpace is an extension of the TwinSpace software framework for collaborative mixed reality applications [25]. TwinSpace provides a flexible mapping approach between objects and services in linked virtual and physical environments, allowing for example the movement of physical objects to cause linked virtual objects to move, or dynamically remapping how the virtual environment is manifested in a connected physical space based on the activity taking place. The architecture of TwinSpace is detailed in [25]. It is built using a document-centric collaborative virtual world called OpenWonderland [26], a blackboard model distributed messaging backbone (EventHeap [29]), and a context engine built using Semantic Web technologies (Apache Jena). In this section, we consider four core features of TwinSpace, and detail how each feature is exploited in SecSpace.

Distributed Communication

TwinSpace provides a distributed physical-virtual communication mechanism. This allows virtual entities to take part in distributed sensing and control, and includes a model of virtual Observers and Effectors that serve as counterparts to sensors and actuators in the physical world.

SecSpace defines Observers that detect events relevant to privacy and security and communicate these via the distributed communications channel. For example, the ProximityObserver detects when avatars come within a specified range of a location or entity. A set of Effectors is used to apply privacy and security policies in the virtual world in response to commands coming from the distributed communications channel. For example, the PermissionEffector can set global, group or individual permissions for a shared document. The set of Observers and Effectors currently available are listed in Table 1.

Table 1: SecSpace virtual Observors, Effectors and Proxies. Modified TwinSpace components are marked by *.

Category	Name	Description
Observers	Login	User logs on or off
	Proximity	User approaches object
	UICapture*	User interacts with object
	ObjectCreated*	Object is created
	NewlySecured	Objects is secured
	Permission	Object permission change
Effectors	AddSecurity	Secures target object
	AddUICapture*	Log all object interaction
	Permission	Change object permission
	Movement*	Move object in world
	Creation*	Create new object
	Destruction*	Destroy object
Proxies	Display*	Virtual display
	CardGame	Manages game events

Observers and Effectors combine to form privacy and security mechanisms in the virtual world. For example, the PermissionEffector can set which users can read a document; and the UICaptureObserver can then be used to determine whether a document will reveal its contents when clicked, or if a warning message appears instead. Similarly, a ProximityObserver and MovementEffector can be used to keep objects or avatars away from a given location.

Security-related messages coming from virtual Observers, physical sensors, and applications get placed on the distributed messaging backbone, as do security-related commands and policies coming from either the physical or virtual spaces, to be interpreted by corresponding virtual Effectors, physical actuators and applications. This model provides a great deal of flexibility in defining how entities communicate, share data, enforce and apply rules. For example, a dedicated server can manage all privacy and security by receiving all messages, determining relevant actions and communicating them via the backbone. A completely decentralized model is also possible, by letting each entity determine what messages it will listen for and how it will translate these into privacy and security-preserving actions. Mixed models are also possible, and developers can define and evolve specific approaches over time, facilitating prototyping and policy experimentation.

Importantly, SecSpace does *not* provide secure distributed communication. While it is possible to integrate SecSpace with middleware security technologies (e.g. the Event Heap iSecurity model [30]), this is not the goal of our work. SecSpace is a framework for exploring usable privacy and security approaches within the context of mixed reality collaboration.

Shared Ontology

TwinSpace [25] defines a common ontology for addressing, manipulating and linking physical and virtual objects, allowing a single set of rules to be defined that can be applied in both physical and virtual spaces. The ontology has evolved from a subset of the SOUPA ontology for pervasive computing applications [36], including classes for Location, Person, Document, among others. A set of proxy objects permit common concepts (such as Display, CardGame) to be used across physical and virtual environments when these concepts are not directly present in one environment, or where they are present in very different ways. Proxy objects typically wrap a set of Observers and Effectors that together provide the expected behavior for the object.

The ontology is also used for reasoning across objects and events in both spaces, for example to infer activity. An inferencing component called the Context Engine pulls relevant tuples from the backbone, adding them to the context state. Rules are evaluated which can generate commands to specific entities or classes of entity.

SecSpace uses this feature to define privacy rules once for both physical and virtual spaces, to link shared resources that have physical and virtual manifestations (paper and digital documents, for example), and to respond to contextual events (such as the approach of an unidentified person) that can occur in physical, virtual, or both spaces simultaneously. For example, both physical display and virtual Displays (proxy objects) share the same ontological class. We could define a rule such that when an unidentified Person enters either space, all Displays display a notification. Alternately, we could define a rule that displayed a notification only on those Displays with the MainDisplay attribute. We can then link a specific physical display to a specific virtual display by assigning this attribute to each of them. If we use the Context Engine component, rules are interpreted and applied dynamically. Because of this, it is possible to replace or update rules at runtime, which is useful for both *ad hoc* testing and controlled experiments. To continue with our example, one experimental condition may apply the global Display notification policy, while another condition applies the MainDisplay notification policy.

Interface Mechanisms

TwinSpace provides a set of lightweight interface mechanisms that link physical and virtual. These include virtual world clients that can be used as addressable portals in a physical environment, and mechanisms for dynamically linking input devices to these clients. The virtual world clients can listen for relevant messages on the distributed communication backbone, and can be connected to using a dedicated communications channel (typically OSC) where a high degree of control and responsiveness is required.

SecSpace can control how interface mechanisms function, as a way of enforcing privacy policies in the connected physical space. For example, a smartphone app reads touchscreen events and converts them into control commands for a virtual portal's camera. SecSpace can control which portal(s) are controlled by which phone(s), and can define allowable camera paths or ranges. In this way, we can experiment with policies that apply to collocated groups as a whole, and to define access permissions to individuals in collocated groups.

Decoupled Components

TwinSpace offers a great deal of flexibility when deciding how to prototype mixed reality interaction. For example, most early prototypes do not use the Context Engine, connecting physical and virtual entities more directly via the messaging backbone to save one level of indirection. While the ontology helps maintain consistency in messages across distributed code, a developer can decide not to use it when testing out an idea. When the interaction between physical and virtual is minimal, all communication can take place through a single virtual world client, rather than use the backbone to communicate with the virtual world server. OpenWonderland's module-based extension feature allows us to package a subset of TwinSpace's functionality to suit a specific application.

When using SecSpace, developers can choose to use the elements of the framework that best suit their purpose. If the research involves context inference (for example, determining when a group splits into subgroups) or adaptive privacy policies (either when evaluating candidate policies in a comparative study or as a feature of a prototype's design), the Context Engine is useful. The messaging backbone is useful if a prototype needs to respond to simple, discrete events (such as someone entering a room), or when non-VW visualizations and applications form part of a prototype (for example, maintaining a 2D abstract visualization of activity progress). If a prototype emphasizes providing a visual indication of what is happening remotely (for example when choosing to share the view of a specific remote collaborator), the addressable portals are most useful.

PROTOTYPE EXAMPLES

To demonstrate the capabilities of SecSpace we describe four prototypes. The prototypes were built for two specific mixed reality environments. One was designed in collaboration with Steelcase, Inc. [32], and is an example of a "project room" [33], a dedicated space for synchronous and asynchronous collaboration around a single project. The room features several distinct 'collaboration regions', including an area for brainstorming equipped with an interactive whiteboard (used in the "Cone of Engagement" prototype), and an area for active table work (used in the "Card Game" prototype). The second environment mimics a more public mixed reality setting, linking a virtual public space (a café), manifested on large displays surrounding an interactive tabletop in our lab. The Card Game prototype was ported to the café setting, and two additional tabletop prototypes were developed inspired by this configuration (a guessing game and a facility for sharing portions of paper documents). The Cone of Engagement, Card Game and Guessing Game prototypes are presented in turn below. We discuss the privacy mechanisms inherent in each prototype, and consider how SecSpace supports them.

Privacy around a physical/virtual whiteboard

We embedded a virtual whiteboard in a physical space by linking it with a physical interactive whiteboard, such that collocated and remote collaborators can edit and discuss whiteboard contents in real time. Projecting a straight-on view of the virtual whiteboard onto the physical whiteboard "links" the physical and virtual whiteboards. Because of this, in-room collaborators have a limited perspective on the virtual environment when using the whiteboard.

While it is clear who is working at the whiteboard in the physical team room, we considered several ways to advertise when a remote collaborator joins the group at the virtual whiteboard. Our ultimate design was to directly translate the physical act of approaching the whiteboard into the virtual environment, and augment this with visual aids and event triggers: the remote collaborator moves their avatar into a visible "cone of engagement", which fans out to the virtual whiteboard from a point in front of it. The cone of engagement is contained within the field of view of the 'camera' that presents the virtual whiteboard on the physical interactive whiteboard (see Figure 1). This provides a simple engagement cue: when an avatar enters or leaves the region they become visible on the periphery of the physical whiteboard. In the virtual environment, we can build further on the act of approaching the whiteboard to introduce access control. A collaborator in the physical environment must move close enough to a whiteboard in order to view all of its contents. Similarly, the cone of engagement can be used to control access to the content on the whiteboard for remote collaborators. Specifically, their avatar must enter the cone of engagement before being able to view and/or edit the whiteboard's content. This threshold mechanism can also be used to check access permissions and grant access only to those who have sufficient privileges, whether or not their avatar has entered the cone.

By inferring a relationship between an avatar's proximity to the virtual whiteboard and the visibility of the content of the whiteboard, we create a space in which the privacy of the discussion around the whiteboard can be negotiated. In this way, users do not have to understand and manage security policies or learn a secure procedure. They can derive and negotiate their own rules in an ad hoc manner. For instance, a group can decide to only discuss topic A while persons X and Y are not present, or decide not to speak of an especially controversial topic if persons who were not present during the entire meeting engage partway.

To disable the whiteboard for remote participants while their avatars are not near the whiteboard is indeed constraining the way activities can take place in the virtual world, and one may argue that it is not smart to introduce such constraints for dynamic group work. However, work at a physical whiteboard already has implicit constraints. A person standing several meters away may not be able to see and hear all details, and people even further away will likely not be able to participate at all. Thus collaboration around a physical whiteboard is also constrained, and importantly these constraints are used to negotiate and manage security and privacy.

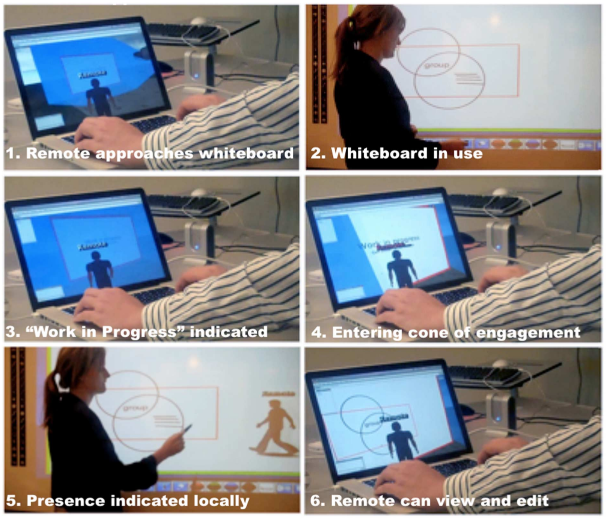
1. Remote approaches whiteboard
2. Whiteboard in use
3. "Work in Progress" indicated
4. Entering cone of engagement
5. Presence indicated locally
6. Remote can view and edit

Figure 1. The Cone Of Engagement

Using SecSpace, a ProximityObserver detects when an avatar enters or leaves the cone, and then a PermissionEffector updates access to the whiteboard for that individual accordingly. In our prototype, two objects are used for the whiteboard: an image that those outside the cone can see (displaying a "whiteboard in use" message), and the real whiteboard for those inside the cone. Note that all of this takes place within the virtual world: if desired, we could move control over authentication and response out to the Context Engine. This would be particularly useful if we wanted to combine whiteboard proximity triggers in both the physical and virtual spaces, for example to evaluate rules of engagement based on who else was collaborating and on the content being shared.

Privacy around a physical/virtual table

We are exploring three privacy scenarios involving sharing documents to people located around the physical table and with remote people who are connected using a virtual environment. We are focused on how to ensure privacy and security when using both physical and digital documents during mixed presence collaboration. The collocated participants around the physical table mainly use paper documents while the remote players share their documents digitally. Within this general setup we are exploring three types of privacy and interactions between people: 1) individual privacy and sharing with the world; 2) sharing with one individual/subset within a group; and 3) sharing partial information with others in a group. We discuss the first two of these in this section due to space constraints.

Individual Privacy and Sharing with the World

This involves keeping some information private while sharing other information globally. For example, a card game embodies this kind of privacy. While playing cards, a player hides cards from others until they decide what to share. We built a card game prototype, where people seated at the table could play the game using physical cards, while a remote participant uses a virtual world client.

An advantage of SecSpace is that we can quickly implement different versions of a concept as it is explored. In the first version of the card game, the virtual world tabletop was top-projected onto the physical game table. In this approach, we attached fiducials (visual markers that can be tracked by cameras) to the picture side of a deck of cards. An overhead camera recognized cards thrown on the table facing upwards. When this happened, the corresponding card was displayed on the top of the virtual table so that the remote participant could see the card. Permissions were set so that only the remote player saw the digital copy, to avoid projecting on top of the physical card. A second camera was placed in a box with a transparent top, making a virtual scanner. When the physical cards were dealt, the remote participant's cards were placed on

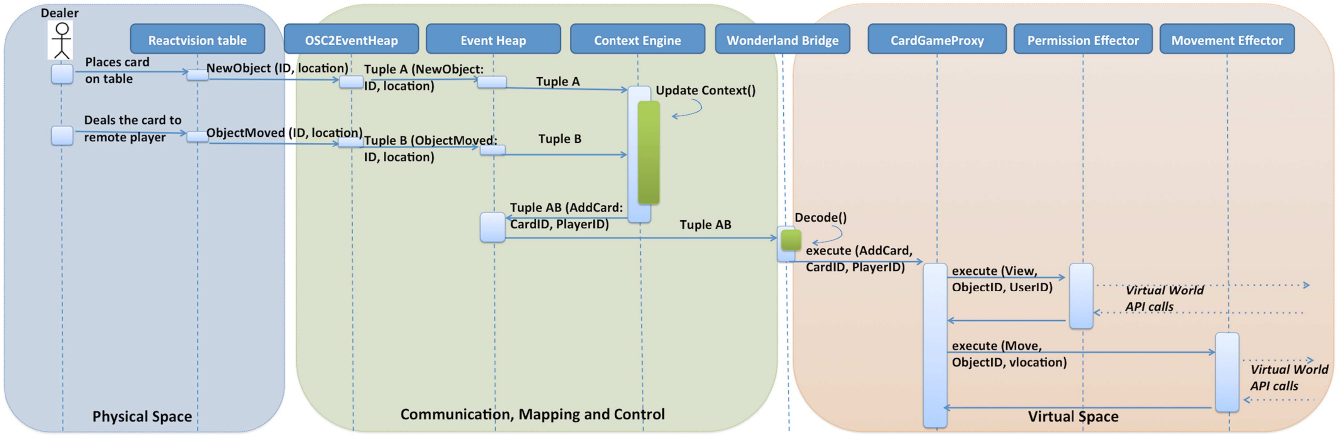
Dealer
Reactvision table
OSC2EventHeap
Event Heap
Context Engine
Wonderland Bridge
CardGameProxy
Permission Effector
Movement Effector
Places card on table
NewObject (ID, location)
Tuple A (NewObject: ID, location)
Tuple A
Update Context()
Deals the card to remote player
ObjectMoved (ID, location)
Tuple B (ObjectMoved: ID, location)
Tuple B
Tuple AB (AddCard: CardID, PlayerID)
Tuple AB
Decode()
execute (AddCard, CardID, PlayerID)
execute (View, ObjectID, UserID)
Virtual World API calls
execute (Move, ObjectID, vlocation)
Virtual World API calls
Physical Space
Communication, Mapping and Control
Virtual Space

Figure 2. Sequence diagram for card game implementation. The Dealer deals a physical in the table region designated for a remote player, generating a virtual world command. In the virtual world, the card becomes accessible to the remote player, and is placed in a dedicated region showing their hand. Here, a physical privacy action corresponds to a different virtual one.

the box facing downwards, recognized by the camera in the box, and displayed to the remote participant in a dedicated region. Cards dealt to the remote participant were visible only to them until they were played. When the remote participant clicked one of the cards it was "thrown on the table" and set to be visible to all, making it visible on the physical tabletop as well.

This first version was limited in that it was cumbersome to pass cards back and forth between digital and physical, and for the remote player to swap unwanted cards for new ones, limiting the kinds of games that could be played. We began developing a new version on a touchscreen tabletop. We designed an Arduino LED app for the scanner box so that played or selected digital cards would trigger a flashing LED below the corresponding physical card. SecSpace facilitated this approach by requiring that the scanner listen for CardPlayed events already coming from the CardGame proxy object – these events were generated by remote players clicking on the card or by local players touching the digital card on the touch table. Similarly, if a remote player selects the digital version of another player's card, the area around the physical card glows on the tabletop. After initial testing we found that the scanner surface was too small and wanted a way to place all cards directly on the table. We then simply attached fiducials to both sides of the deck of cards. While cards might be recognized by the fiducial marker, we were only interested in this implementation for a short controlled study—a final implementation could use a tracking camera beneath the tabletop surface. Instead of a dedicated device, the remote player was assigned a dedicated region of the table; the dealer placed the remote player's cards face down in that region. The same system of indicators (glowing regions around cards of interest) was used to allow players to communicate and play together with physical and digital cards (see Figure 3). Figure 2 shows a sequence diagram of the event sequence that takes place when the dealer deals a physical card to the remote player in this current implementation.

Just like in card games in the physical world, the game rules were player-enforced (i.e., the system did not enforce any rules). Players manage their own security and privacy and negotiate it with the other players. A player can show one of his cards to a local player by showing it physically or to a remote player by placing it face down in the remote player's region of the table temporarily. For instance if a player is amused by how fortunate she is to end up with an ace, she can show the ace to selected players in order to share her amusement and still not ruin the game.

Share with certain individuals within a group.

When people are in a group there may be times when they only want to share information with one person or a subset within a group. In particular, how would someone share information with both a real person and a virtual person and at the same time while hiding it from others and the world? To explore this scenario, we created a guessing game prototype. Two players share categories (each has a set of category cards they can share) and decide on an item (e.g., a movie title) for another player to try to guess by asking questions up to a maximum number.

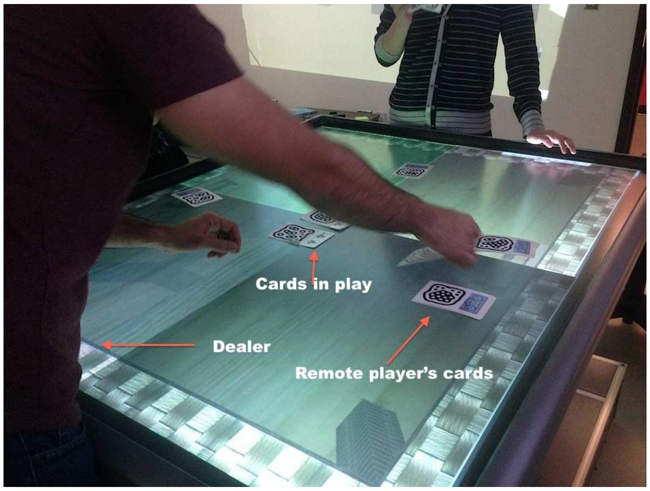

Figure 3. The final Card Game implementation.

SecSpace facilitates this by placing ProximityObservers around virtual table regions. Remote players move a category card into one of the regions to make it legible on the physical table. We place physical dividers around these table regions to prevent eavesdropping (Figure 4). Local players share their category cards with remote players by placing the category card face down in the remote player's dedicated table region (a fiducial is placed on the back of the category card as with the card game)

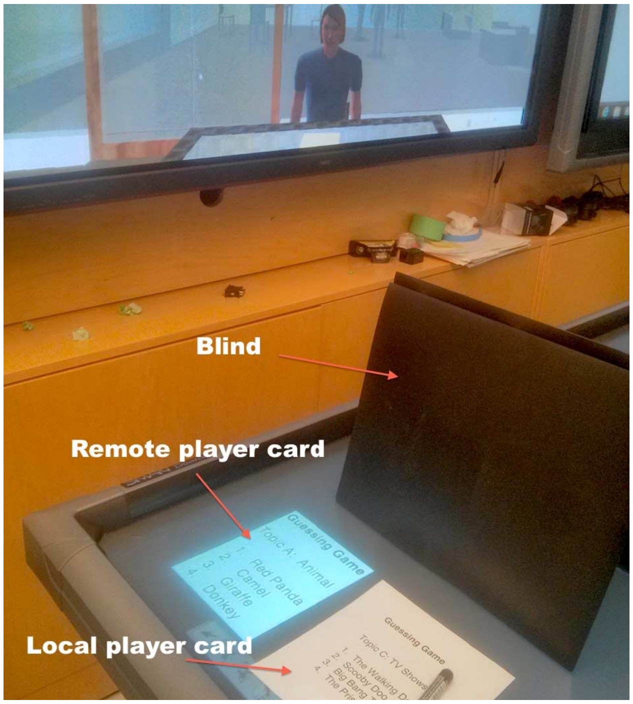

Figure 4. The Guessing Game

DISCUSSION

The use of SecSpace has allowed us to rapidly develop a range of prototypes exploring privacy in mixed presence collaboration. This has helped to build hypotheses that can be explored in future controlled studies and other evaluations. First, we are interested in how physical privacy behaviours (around paper documents, for example) can be sensed and translated into counterpart virtual privacy actions.

Second, we believe that design for management of privacy (rather than enforcing policies) may make security and privacy dependent artifacts more usable. Using SecSpace, we were able to design and prototype mechanisms that made it possible for users to negotiate and thereby manage their own privacy in a tangible (in the card game example) or embedded (in the whiteboard example) way. The two implementations do not guarantee security or privacy, however—they place that responsibility in the hands of the users, who can determine the actual need for security or privacy.

While we believe SecSpace provides a number of benefits in its current form, there are a number of technical limitations still to address. We outline these here.

Limitations

Learning curve

There are a lot of technologies (OpenWonderland, Event Heap, Jena, OSC) that prototype builders need to become familiar with in order to use SecSpace to its fullest extent. It is often difficult to know "where" the best place is to write logic, interface code, or privacy policies. Through the development of a number of TwinSpace and SecSpace prototypes (involving approximately 70 individuals with widely varying expertise) we have found that developers typically start with a very simple model at first and then (if necessary) move to more comprehensive use of SecSpace. Typically, a developer will capture a physical interaction, and generate an OSC message that a specific virtual world client will receive and respond to, or generate an EventHeap message that a specific Effector will handle in the virtual world (see Figure 5); alternately the developer will write simple code that responds to an EventHeap message coming from the virtual world via an Observer. We plan to define abstraction layers (for example, hiding the communication mechanism through a factory pattern) to simplify engagement with more of SecSpace earlier.

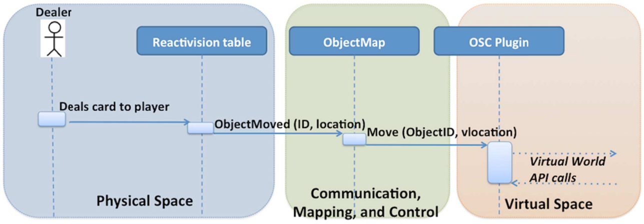

Figure 5. Sequence diagram showing dealer dealing a card in initial card game implementation. Here, moving physical cards maps directly to virtual card movements.

Difficulty transitioning between reduced and full feature set

When prototypes are developed using simple models, it is often difficult to encourage engagement with the larger feature set of SecSpace, and this may reduce the ability of the system to flexibly evaluate alternative privacy mechanisms. For example, the card game began using the model shown in Figure 5. While straightforward, it requires a fair bit of custom glue code between the physical and virtual pieces, and custom code on the virtual world client to enable interaction with cards. The first card game version used Observers and Effectors, but did not use the Context Engine, instead placing the mapping, privacy and game logic in the glue code. Moving this code to the Context Engine and a Card Game proxy object was tedious and challenging. It is useful that SecSpace decouples components to give flexibility, but well-defined, clear best practices and supports need to be available.

Distributed model: sometimes robust, often opaque

Developing a prototype using all of SecSpace requires configuring a number of distributed components. The decoupled message passing of the Event Heap and redirection capability of OSC permits robustness when some components aren't present, but this will depend on the way the prototype is designed. Complex prototypes with deeply interlocking components benefit less from these features, and can be difficult to set up correctly and to debug. Finally, while SecSpace (and TwinSpace) were designed to permit different rendering and virtual world engines, they were built using OpenWonderland. This means that certain features (specifically the mechanics and capabilities of particular Effectors and Observors) will be reliant on the availability of features (such as document editing and related permissions) on the base platform. We plan to port SecSpace to a Unity-based platform in future, and will be able to better assess the generality of our model.

CONCLUSION

We have presented SecSpace, a software framework for prototyping usable privacy and security mechanisms for mixed reality collaborative environments. Its key features are distributed communication, shared physical-virtual ontology and reasoning, a set of interface mechanisms for real-virtual interaction, and a high degree of feature decoupling permitting a range of development strategies. We demonstrated the value of SecSpace through the description of three prototypes, one focused on a shared whiteboard and the others a shared tabletop. Developing and modifying prototypes using SecSpace has contributed to our understanding of usable privacy in mixed presence collaboration, inspiring targeted research. We also identified a number of current limitations that we hope to address in future work: a high learning curve, difficulty transitioning between simple and more complete system models, and difficulties understanding system status when developing highly interconnected prototypes.

ACKNOWLEDGEMENTS

This work is funded in part by Boeing and GRAND. Earlier work was supported by NSF grant IIS-0705569, and a research collaboration grant from Steelcase Inc.

REFERENCES

[1] P. Garner, M. Collins, S. M. Webster, and D. A. D. Rose, "The application of telepresence in medicine," *BT Technol. J.*,154, 181–187, 1997.

[2] A. Petrakou, "Interacting through avatars: Virtual worlds as a context for online education," *Comput. Educ.*, vol. 54, no. 4, pp. 1020–1027, May 2010.

[3] J. Lang, "Privacy, Territoriality and Personal Space – Proxemic Thoery," in in *Creating Architectural Theory: The role of the behavioral sciences in design*, New York, 1987, pp. 145–156.

[4] K. O'hara, J. Kjeldskov, and J. Paay, "Blended interaction spaces for distributed team collaboration," *ACM Trans. Comput. Interact.*, 181, pp. 1–28, Apr. 2011.

[5] J. Lifton, M. Laibowitz, D. Harry, N.-W. Gong, M. Mittal, and J. A. Paradiso, "Metaphor and Manifestation Cross-Reality with Ubiquitous Sensor/Actuator Networks," *IEEE Pervasive Comput.*, 83, pp. 24–33, Jul. 2009.

[6] Z. Pan, A. D. Cheok, H. Yang, J. Zhu, and J. Shi, "Virtual reality and mixed reality for virtual learning environments," *Comput. Graph.* 0, pp. 20–28, Feb. 2006.

[7] C. Sandor, A. Olwal, B. Bell, and S. Feiner, "Immersive mixed-reality configuration of hybrid user interfaces," in *Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'05)*, 2005, pp. 110–113.

[8] I. Wagner, W. Broll, G. Jacucci, K. Kuutii, R. McCall, A. Morrison, D. Schmalstieg, and J.-J. Terrin, "On the Role of Presence in Mixed Reality," *Presence Teleoperators Virtual Environ.*, 184, pp. 249–276, Aug. 2009.

[9] O. Oyekoya, A. Steed, R. Stone, W. Steptoe, L. Alkurdi, S. Klare, A. Peer, T. Weyrich, B. Cohen, and F. Tecchia, "Supporting interoperability and presence awareness in collaborative mixed reality environments," in *19th ACM Symposium on Virtual Reality Software and Technology - VRST '13*, 2013, p. 165.

[10] P. Van Schaik, T. Turnbull, A. Van Wersch, and S. Drummond, "Presence Within a Mixed Reality Environment," *CyberPsychology Behav.*, 75, pp. 540–552, Oct. 2004.

[11] S. Bødker, "When second wave HCI meets third wave challenges," in *4th Nordic conference on Human-computer interaction changing roles - NordiCHI '06*, 2006, pp. 1–8.

[12] J. McCarthy and P. Wright, "Technology as Experience," Sep. 2004.

[13] E. Chin, A. P. Felt, V. Sekar, and D. Wagner, "Measuring user confidence in smartphone security and privacy," in *Eighth Symposium on Usable Privacy and Security - SOUPS '12*, 2012, pp. 1–16.

[14] J. Goecks, W. K. Edwards, and E. D. Mynatt, "Challenges in supporting end-user privacy and security management with social navigation," in *5th Symposium on Usable Privacy and Security - SOUPS '09*, 2009, p. 1.

[15] K. R. Chia Shen, Katherine Everitt, "UbiTable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces," in *UbiComp 2003*, 2003, pp. 281 – 288.

[16] E. M. Huang and E. D. Mynatt, "Semi-public displays for small, co-located groups," in *conference on*

Human factors in computing systems - CHI '03, 2003, pp. 49 – 56.

[17] A. Kapadia, T. Henderson, J. J. Fielding, and D. Kotz, "Virtual walls: protecting digital privacy in pervasive environments," pp. 162–179, May 2007.

[18] S. G. Anthony Tang, Carman Neustaedter, *VideoArms: embodiments in mixed presence groupware*. Springer London, 2007, pp. 85–102.

[19] S. Yarosh, A. Tang, S. Mokashi, and G. D. Abowd, "'almost touching'," in *2013 conference on Computer supported cooperative work - CSCW '13*, 2013, pp. 181 – 192.

[20] K. Hunter, S., Maes, P., Tang, A., and Inkpen, "WaaZam! Supporting Creative Play at a Distance in Customized Video Environments.," in *SIGCHI Conference on Human-factors in Computing Systems 2014*, 2014.

[21] H. Regenbrecht, M. Haller, J. Hauber, and M. Billinghurst, "Carpeno: interfacing remote collaborative virtual environments with table-top interaction," *Virtual Real.*,102 pp. 95–107, 2006.

[22] A. Tang, C. Neustaedter, and S. Greenberg, "VideoArms: Embodiments for Mixed Presence Groupware," *People Comput. XX — Engag.*, pp. 85 – 102, 2007.

[23] K. Tee, S. Greenberg, and C. Gutwin, "Providing artifact awareness to a distributed group through screen sharing," in *2006 conference on Computer supported cooperative work - CSCW '06*, 2006, pp. 99 – 108.

[24] Steve Benford, Chris Greenhalgh, Gail Reynard, Chris Brown, and Boriana Koleva. 1998. Understanding and constructing shared spaces with mixed-reality boundaries. *ACM Trans. Comput.-Hum. Interact.* 5, 3 (September 1998), 185-223.

[25] D. F. Reilly, H. Rouzati, A. Wu, J. Y. Hwang, J. Brudvik, and W. K. Edwards, "TwinSpace: an infrastructure for cross-reality team spaces," in *23nd annual ACM symposium on User interface software and technology - UIST '10*, 2010, pp. 119 – 128.

[26] B. Johanson, A. Fox, and T. Winograd, "The Interactive Workspaces project: experiences with ubiquitous computing rooms," *IEEE Pervasive Comput.*, vol. 1, no. 2, pp. 67–74, Apr. 2002.

[27] N. A. Streitz, J. Geißler, T. Holmer, S. Konomi, C. Müller-Tomfelde, W. Reischl, P. Rexroth, P. Seitz, and R. Steinmetz, "i-LAND: an interactive landscape for creativity and innovation," in *SIGCHI conference on Human factors in computing systems - CHI '99*, 1999, pp. 120–127.

[28] V. Stanford, J. Garofolo, O. Galibert, M. Michel, and C. Laprun, "The NIST Smart Space and Meeting Room projects: signals, acquisition annotation, and metrics," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, vol. 4, pp. IV–736–9.

[29] B. Johanson and A. Fox, "The Event Heap: A Coordination Infrastructure for Interactive Workspaces," In *Mobile Systems and Applications, 2002.* p. 83, Jun. 2002.

[30] O. F. Yee Jiun Song, Wendy Tobagus, Der Yao Leong, Brad Johanson, "isecurity: A security framework for interactive workspaces," 2003.

[31] "The OpenWonderland Project." [Online]. Available: http://www.openwonderland.org.

[32] D. Reilly, S. Voida, M. McKeon, C. Le Dantec, J. Bunde-Pedersen, W. K. Edwards, E. D. Mynatt, and A. Mazalek, "Space Matters: Physical-Digital and Physical-Virtual Co-Design in the Project." *IEEE Pervasive Comput 9(3)*, 2010, pp. 54–63.

[33] G. Olson and J. Olson, "Distance Matters," *Human-Computer Interact.*, 152, pp. 139–178, 2000.

[34] W. K. Edwards, E. S. Poole, and J. Stoll, "Security automation considered harmful?," in *2007 Workshop on New Security Paradigms - NSPW '07*, 2008, pp. 18 – 21.

[35] Benford, S., Magerkurth, C., & Ljungstrand, P. (2005). Bridging the physical and digital in pervasive gaming. *Communications of the ACM*, *48*(3), 54-57.

[36] Chen, H., Finin, T., & Joshi, A. (2005). The SOUPA ontology for pervasive computing. In *Ontologies for agents: Theory and experiences* (pp. 233-258). Birkhäuser Basel.

[37] Reilly, D, Salimian, M., and Brooks, S. (2013) Document-Centric Mixed Reality and Informal Communication in a Brazilian Neurological Institution. Beyond Formality: Informal Communication in Health Practices Workshop, CSCW 2013, San Antonio, TX, USA

[38] Neustaedter, C., & Greenberg, S. (2003, January). The design of a context-aware home media space for balancing privacy and awareness. In *UbiComp 2003: Ubiquitous Computing* (pp.297-314). Springer Berlin Heidelberg.