IST BASIC RESEARCH PROJECT
SHARED COST RTD PROJECT
THEME: FET DISAPPEARING COMPUTER
COMMISSION OF THE EUROPEAN COMMUNITIES
DIRECTORATE GENERAL INFSO
PROJECT OFFICER: THOMAS SKORDAS

Global Smart Spaces

Initial Design of Interaction Techniques Using Multiple Interaction Surfaces

D18

27/10/2003, UNIV. JOSEPH FOURIER/WP7/VERSION 1.0

J. COUTAZ, C. LACHENAL, N. BARRALON, G. REY

IST Project Number	IST-2000-26070	**Acronym**	GLOSS
Full title	Global Smart Spaces		
EU Project officer	Thomas Skordas		

Deliverable	**Number**	D18	**Name**	Initial Design of Interaction techniques Using Multiple Interaction Surfaces			
Task	**Number**	T	**Name**	n/a			
Work Package	**Number**	WP7	**Name**	Interaction Techniques			
Date of delivery	**Contractual**		July 2003		**Actual**	Oct. 2003	
Code name	n/a				**Version** 1.0 draft ☐	final ☑	
Nature	Prototype ☑ Report ☑ Specification ☐ Tool ☐ Other:						
Distribution Type	Public ☑ Restricted ☐ to:						
Authors (Partner)	J. Coutaz, C. Lachenal, N. Barralon, G. Rey						
Contact Person	J. Coutaz						
	Email	Joelle.coutaz@imag.fr	**Phone**	+33 4 76 51 48 54	**Fax**	+33 4 76 44 66 75	
Abstract (for dissemination)	This document completes D19 where we describe I-AM, a middleware that extends windowing systems to support the dynamic composition of interaction resources into a unified space. In this document, we show the technical generality of I-AM in terms of surfaces composition. We then analyse one particular aspect of I-AM in relation to the visual discontinuities that may arise from the composition of surfaces: the mapping function between the digital and the physical spaces.						
Keywords	Multi-surface interaction, distributed user interface, migratory user interface, Interaction Abstract Machine, mapping digital and physical spaces.						

Table of Content

1 INTRODUCTION

This document completes Deliverable D19 "Final Reference Framework for Multi-Surface Interaction". It assumes that the reader is familiar with the concepts and techniques presented in D19.

D19 includes the description of:

1. An ontology that makes explicit the concepts of multi-surface interaction,
2. I-AM[1], a software infrastructure that implements this ontology.

I-AM supports the dynamic composition of heterogeneous interaction resources into a unified space. In this space, users can distribute and migrate whole or parts of the user interface as if this user interface were handled by a unique computer. This illusion of a unified space is maintained at no extra cost for the developer. I-AM is a middleware that can be viewed as an extension of current windowing systems to support the development of multi-surface, multi-instrument interaction in a unified way. It is an enabling technology intended to facilitate the development of user interfaces for smart spaces.

I-AM advances the state of the art by addressing all of the following problems:

1. Platforms heterogeneity (e.g., clusters of machines running a mix of MacOs X, Windows NT and Windows XP),
2. Interaction resources heterogeneity (e.g., screens with different sizes and resolutions),
3. Platforms and interaction resources discovery based on a fabric of contextors,
4. Multi-surface interaction grounded on the dynamic composition of hinged display surfaces whose spatial relationships are automatically modeled and maintained,
5. Multi-keyboard, multi-pointer capabilities (so that a user can use the mouse of a PC to manipulate a window displayed on a MacOS screen and drag the window across screens boundaries as if there were a single screen).

In this document, we discuss a particular aspect of I-AM: the mapping problem between the digital space and the physical space. The digital space is an homogeneous infinite Cartesian plane, whereas the physical space is a reconfigurable finite set of heterogeneous interaction resources. We will primarily concentrate the discussion on screen displays with the problem of visual discontinuity that the composition of multiple surfaces may induce. The document is structured as follows:

. In the next section, we briefly recall the technical principles of I-AM and its mapping function between the digital and physical spaces.

[1] I-AM stands for Interaction Abstract Machine.

- In Section 3, we illustrate the generality of I-AM in terms of the types of surfaces composition it is able to support.
- In Section 4, we concentrate the analysis on the mapping problem between the digital and the physical spaces.

2 TECHNICAL PRINCIPLES OF I-AM

Figure 1 illustrates the technical principles of I-AM.

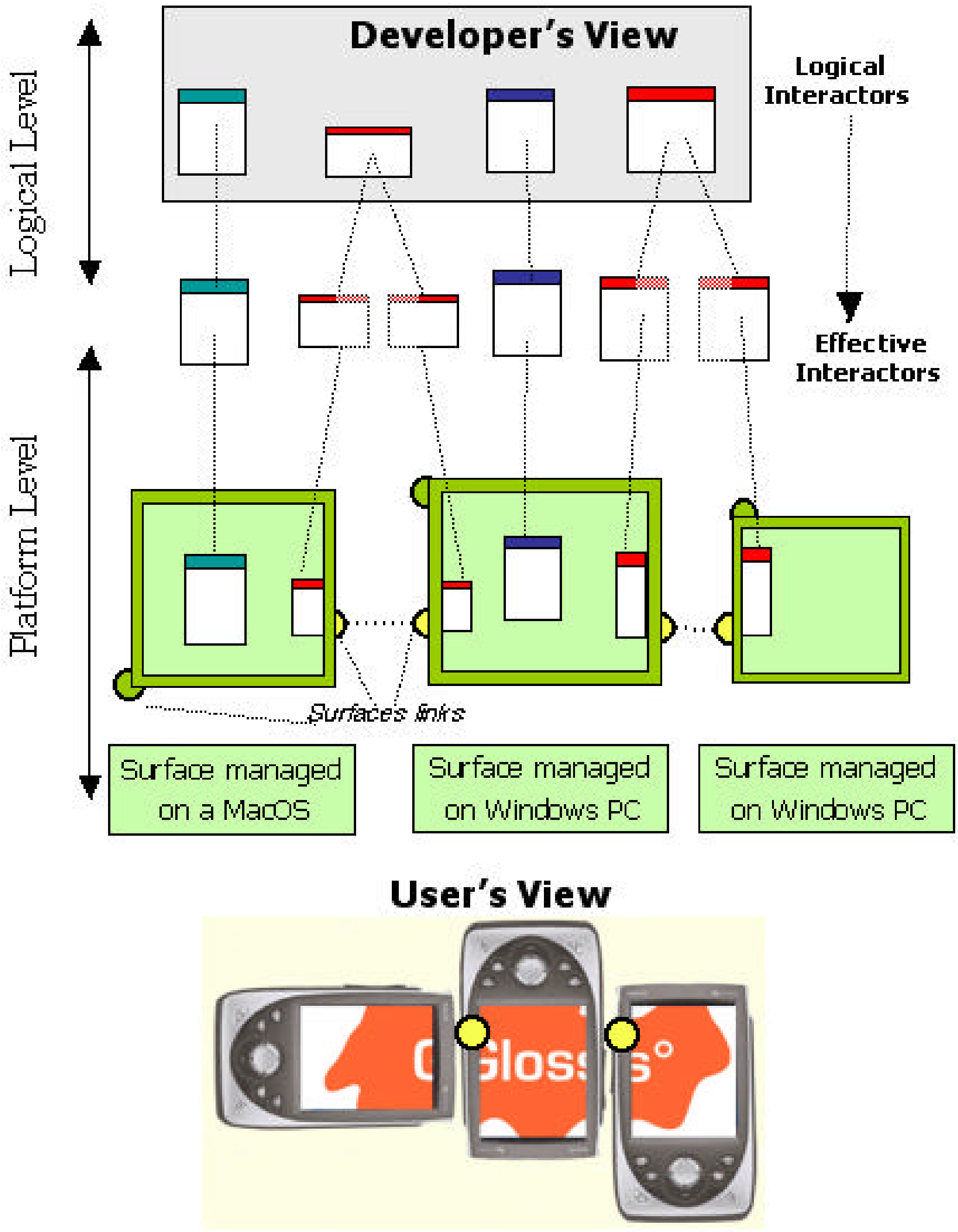

Figure 1. The principles of IAM [From D19].

As shown at the bottom of the figure, the platform is a cluster composed of three machines. Each one handles a unique surface and runs a different operating system (e.g., MacOS X, Windows XP, Windows NT). Through surfaces links, surfaces are composed in a plane using, possibly different, orientations in the plane. Surfaces links are reference points located on the edge of a surface. They can take the form of a physical sensor (e.g., infrared sensors, accelerometers as in Hinckley's example of synchronous gestures for connecting tablets [Hinckley 03]). They can also be painted dots tracked by a computer vision system. Surfaces links allow I-AM to dynamically compute the topology of the surfaces (i.e., their spatial relationships).

The bottom of the figure shows the distribution of the user interface across three surfaces. Some interactors such as the top left window of the developer's view, are fully rendered within a single surface whereas other interactors, such as the right most window of the developer's view, are split across two surfaces. In the latter case, the logical interactor of the developer's view is mapped into two effective interactors whose rendering is tightly coupled to entertain the illusion of a unified space: as the user moves one of the effective interactors using any pointing device of the cluster, the other "twin" effective interactor is moved and resized accordingly as if the twins were one single piece.

The role of I-AM is to continuously maintain the mapping between the logical view of interactors as handled by the developer, and the effective interactors as manipulated by the user. The next sections show examples of composition and their effect on user's visual perception.

3 DYNAMIC COMPOSITION OF SURFACES

In Figure 2, two screens have been composed into a multi-surface space by bringing together the top edge of a PC-laptop screen with that of an Apple-laptop screen. A window, initially created on the PC, is currently overlapping the two screens. When crossing the top edge of the PC screen, this window would not be visible on the Apple screen if I-AM did not maintain an explicit model of the screens topology.

Figure 2. A window displayed on a multi-surface interaction space composed from a PC screen and of an Apple screen connected via their top edges.

The displays we are using are not yet equipped with physical sensors. Therefore, we simulate the physical composition of displays with a software application called the SurfaceConfigurator. The SurfaceConfigurator is used by a human wizard who mimics users'actions as they compose physical surfaces. This application may run on any computer of the local area network. This computer does not need to be a member of an IAM cluster.

Figure 3 shows the situation where the SurfaceConfigurator has discovered two surfaces. To discover the physical surfaces of a cluster, the SurfaceConfigurator uses the contextors infrastructure. As presented in D19, the existence of these surfaces as well as their ID and physical characteristics (size, resolution, borders width), are exported to the world by the ContextAdaptor of the PlatformManager that runs on each machine of the cluster. Any listener, including the SurfaceContextor, is automatically notified of the arrival/departure of surfaces.

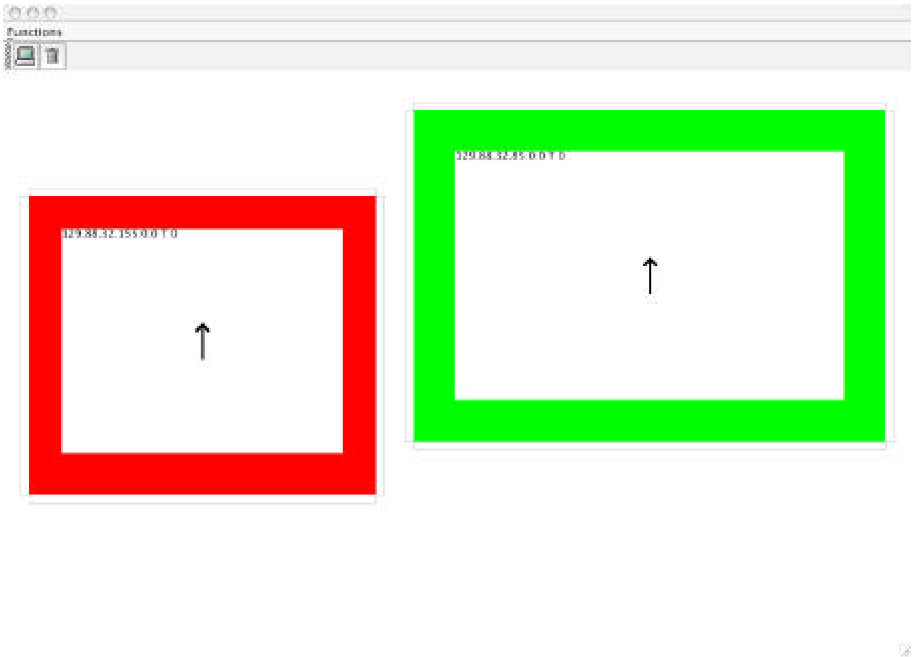

Figure 3. The user interface of the SurfaceConfigurator used by a human wizard to simulate the composition of surfaces via physical sensors. Here, two surfaces have been automatically discovered by the SurfaceConfigurator. They are not yet composed.

As shown in Figure 3, a surface is represented as a rectangle whose size and borders are proportional to that of the physical display. The orientation of the surface is represented by an arrow oriented towards the top edge of the surface. The ID of the surface is displayed on the top left border of the rectangle[2]. The rectangles can be rotated and assembled using the mouse. When the "computer" icon of the menu bar is selected, the SurfaceConfigurator creates the surfcaces links that physically bind the surfaces and, using the contextors infrastructure, publishes the appropriate "Arrival of a New link" events. From there, we leave the simulator and enter the "real code": All of the IAMApps that use these surfaces (i.e., those that have opened a communication port

[2] As presented in D19, the ID of a surface includes the IP address of the machine that handles it, the ID of the graphical port of the video card that handles it, and a unique integer.

with the surfaces) receive the events. Their logical space is automatically mapped onto the new physical space according to the new topology.

The following sequence of figures shows the different ways of composing two physical surfaces[3]. These surfaces are used by an IAMApp that has created one window interactor to render the GLOSS Logo.

3.1 EXAMPLE 1: TOP-TO-TOP COMPOSITION

When the user composes two surfaces as shown in Figures 2 and 4, the wizard user must position the rectangles as shown in Figure 5.

Figure 4. Top-to-Top composition: two surfaces are coupled via their top edges.

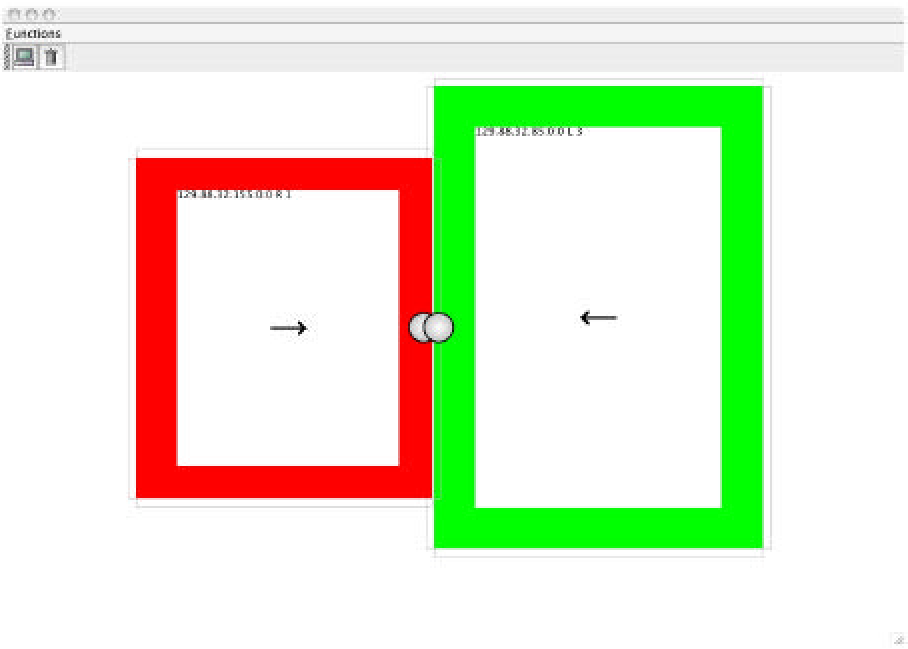

Figure 5. Top-to-Top composition simulated by the wizard. The surfaces links are represented by gray circles.

---

[3] I-AM can theoretically support any number of surfaces. As reported by Johanson et al., performance on the network is the actual limiting factor for distributed user interfaces [Johanson 02].

3.2 EXAMPLE 2: BOTTOM-TO-LEFT COMPOSITION

Figure 6. Bottom-to-Left composition: The bottom edge of one surface is coupled to the left side of the other.

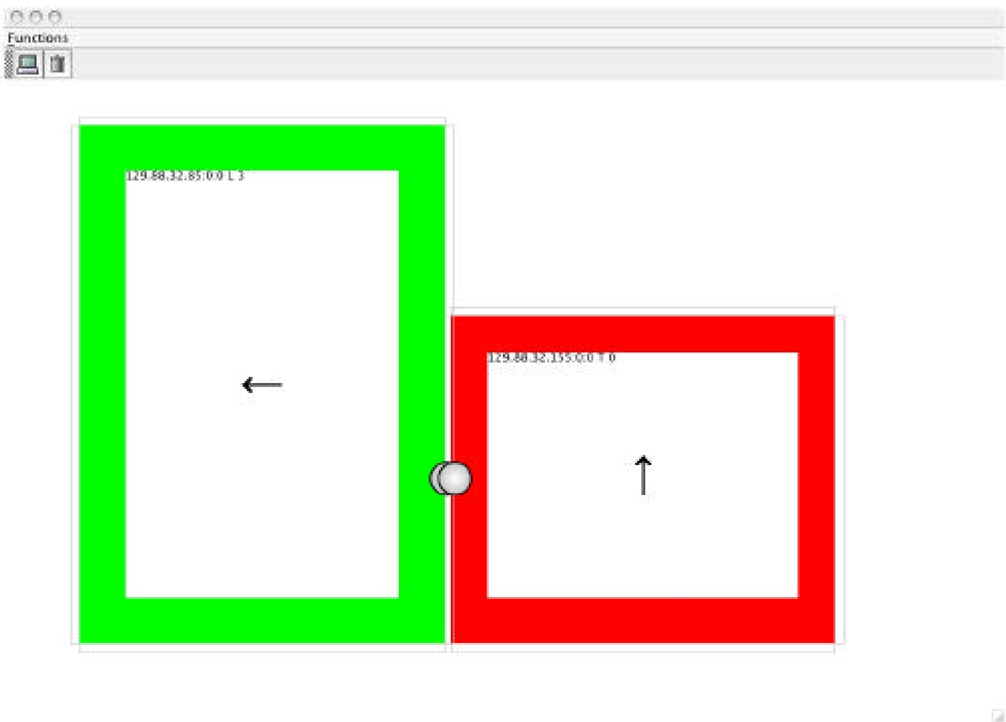

Figure 7. Bottom-to-Left composition simulated by the wizard.

3.3 EXAMPLE 3: TOP-TO-LEFT COMPOSITION

Figure 8. Top-to-Left composition: The top edge of one surface is coupled to the left side of the other.

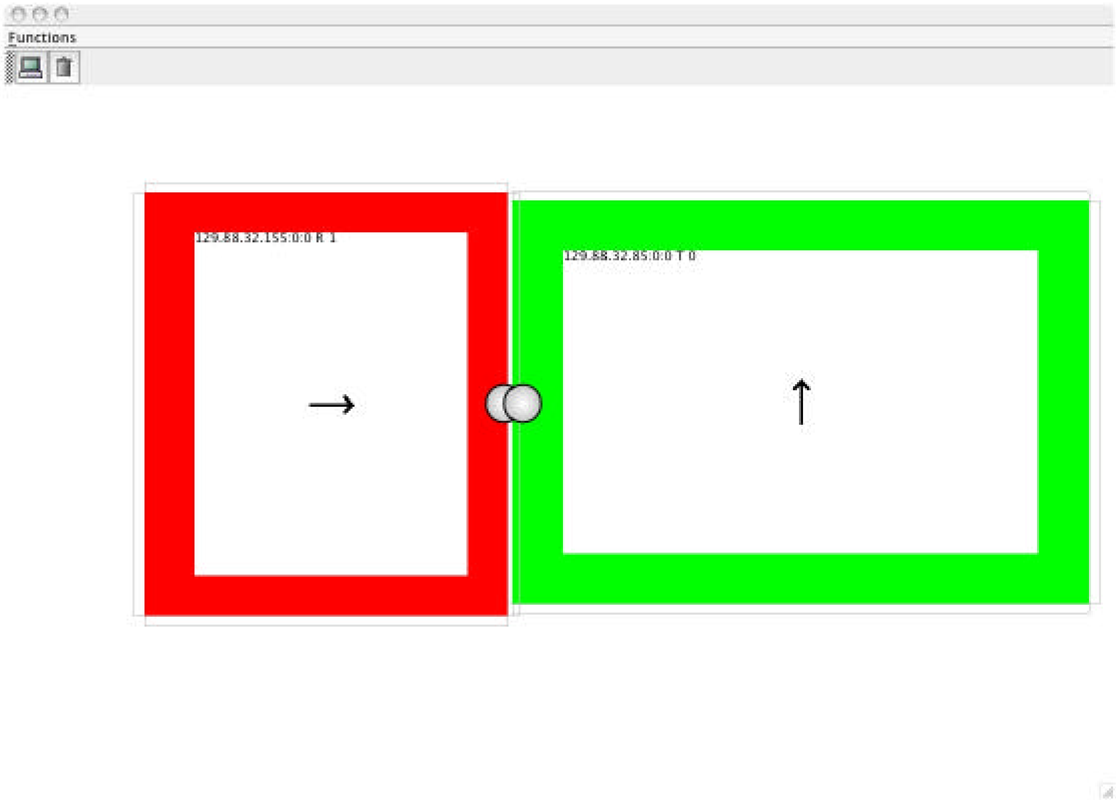

Figure 9. Top-to-Left composition: simulation of the coupling by the wizard.

To mimic the decoupling of a surface, the wizard moves the rectangles apart and asks the SurfaceConfigurator to generate the appropriate event: information that was displayed on the corresponding physical surface disappears.

We have shown how the SurfaceConfigurator can be used by a human wizard to generate the events that would be produced by effective sensors. Because the SurfaceConfigurator "talks" to the world via the contextors infrastructure, it will be easy to replace the SurfaceConfigurator with the appropriate contextors that encapsulate the physical sensors when they will be available.

4 MAPPING BETWEEN THE DIGITAL AND THE PHYSICAL SPACES

As reported by [Hinckley 03], mapping the digital space onto the set of composed surfaces can be performed in many ways. As shown in our examples above or in [Yee 03], one way is to consider each surface as a physical peephole on the digital world. When a new surface is connected, visual access to the digital space is expanded. Another metaphor is to interpret the arrival of a new surface as a way to transform the rendering of the digital content so that it can take full advantage of the new real screen estate. Typically, a city map rendered at a low resolution on a small screen, would be displayed at a high resolution with additional information such as areas of interest, when several screens are docked together. In a multi-user setting, surface contents may be swapped between two users, or joined as in Dynamo [Izadi 03] or the ConnectTable [Tandler 01]. In [Gorbet 98] and [Rekimoto 01], connecting triangles or Data Tiles allows users to construct a storyline or express a series of operations: the state of the digital world is modified.

Therefore, the capacity of composing surfaces opens the way to a large space of design decisions that depend on the application, users activities, and so on. This observation translates into the software design of I-AM by separating the mechanisms from the politics: mechanisms are general so that they can interpret as many application-dependent politics as possible. In the following discussion, we do not promote any politics, but we show how they impact the rendering of visual information across multiple surfaces.

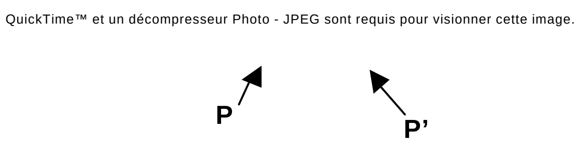

Figure 10. A snapshot of the Drap-and-Pop technique developed for the Dynawall for moving icons between remote screens [Baudisch 03]. The mapping technique does not take bezels into account.

From previous work in perceptual psychology, it is reasonable to expect that human performance be influenced by the surfaces topology and the bezels [Campbell 03, Tan 03a]. In [Tan 03b], Tan et al. report a study on the effects of visual separation and physical discontinuities when digital content related to the same task is distributed across multiple displays. Their experiment shows that the physical discontinuities introduced by bezels or differences in depth alone, do not seem to have a significant effect on subjects' performance for text comparison. However, in their setting, windows content are displayed on a single screen at a time. As discussed below, the mapping algorithms between the digital and the physical spaces produce different results depending on whether the bezels are ignored or taken into account.

The example shown in Figure 10, illustrates the effect of bezels. The picture shows the visual effect when the rubber band that joins the base and tip icons crosses the bezels of two contiguous SmartBoards. In this example, the mapping algorithm does not take the bezels into account.Figure 11 is a new version of Figure 10 that we have modified using Photoshop. Here, P', whose position is strongly coupled to that of P, has been translated along the Y axis to maintain visual continuity.

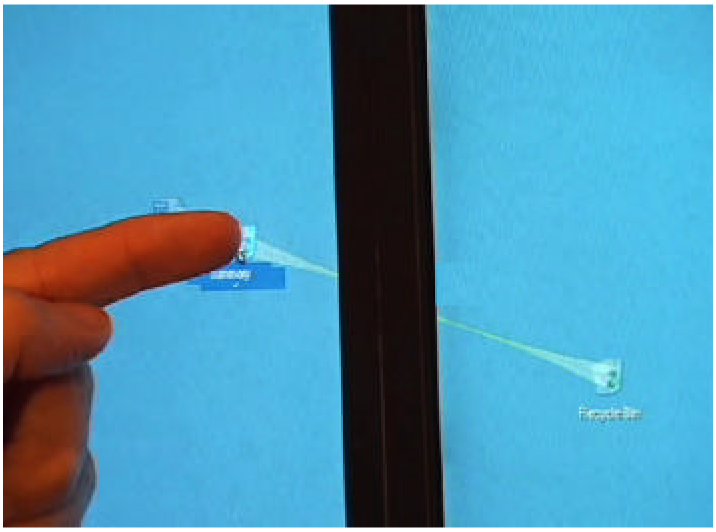

Figure 11. The snapshot of Figure 10 modified with Photoshop to show how visual continuity may be improved.

The following example illustrates the problem in a more systematic manner. Figure 12 shows a physical space composed of three tiled surfaces used to render a digital space that contains the picture of a graph composed of three nodes A, B, C. The top row of the figure shows the final result of the mapping as perceived by users for three different mapping politics. The bottom row shows how the physical surfaces are projected onto the digital content. On the left, bezels are ignored. The result looks like a broken graph (just like in Figure 10). In the middle, bezels are taken into account but they are treated as opaque surfaces: the graph looks correct but the pixels that fall under the bezels are lost. On the right, the digital content is processed so that no pixel is lost while preserving the shape of the original image.

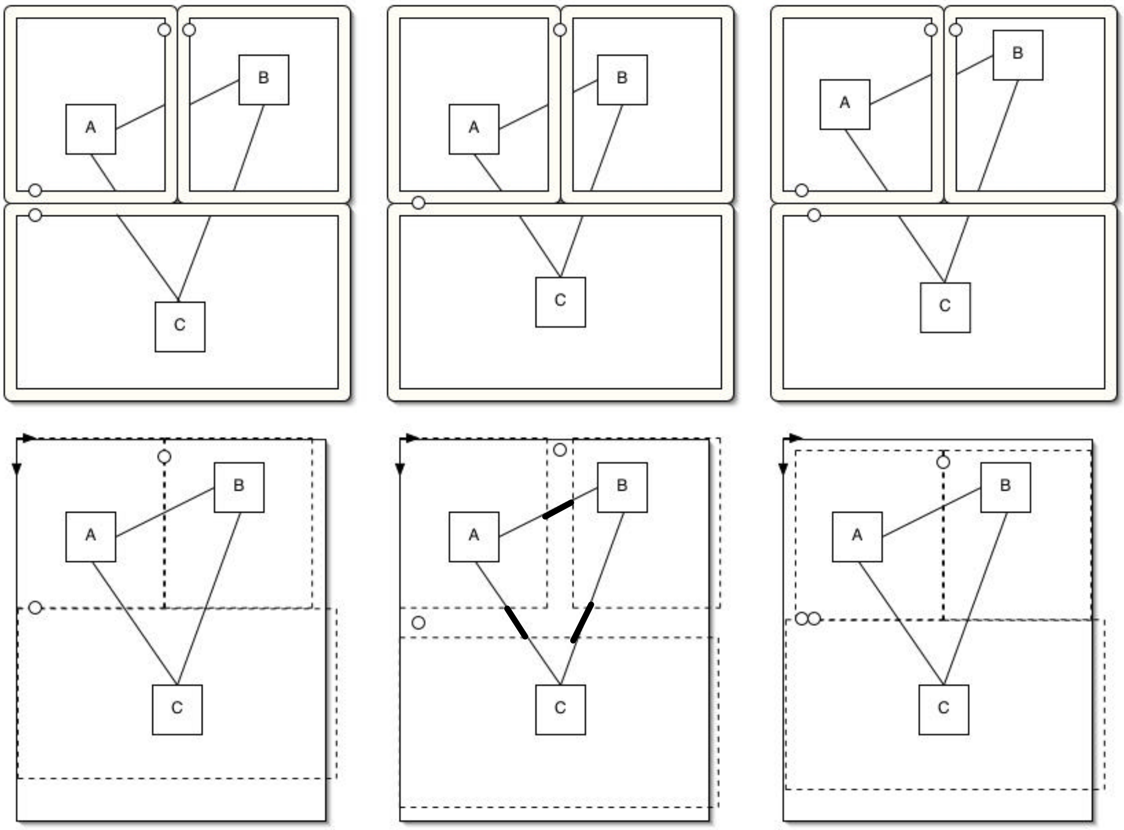

Figure 12. Mapping a geometric figure onto three tiled surfaces using different politics. On the left, bezels are ignored. In the middle bezels are modeled as opaque surfaces (at the cost of information loss shown as thick lines). On the right bezels are modeled and the image is modified to improve visual continuity without information loss.

Figure 13 illustrates the same problem for rendering mouse cursors. Here, the user is supposed to sit in front of the tiled surfaces. He moves the mouse forward from the bottom surface to the top surfaces. The top left surface is supposed to be the reference, i.e., the surface whose coordinates system is the reference for the topology manager (Cf. D19). On the left, bezels are ignored by the mapping politics. The arrows with dotted lines show the trajectories of the mouse cursor in the digital space. These trajectories are those that the user produces by moving the mouse forward. Thick arrows show the trajectories that the user perceives for each of the politics presented for Figure 12. As one can see, it is very hard to maintain the same trajectories in the digital and the physical spaces except for the politics where bezels are considered as opaque surfaces. However, with this politics, the cursor may disappear when it enters the opaque surfaces of the edges.

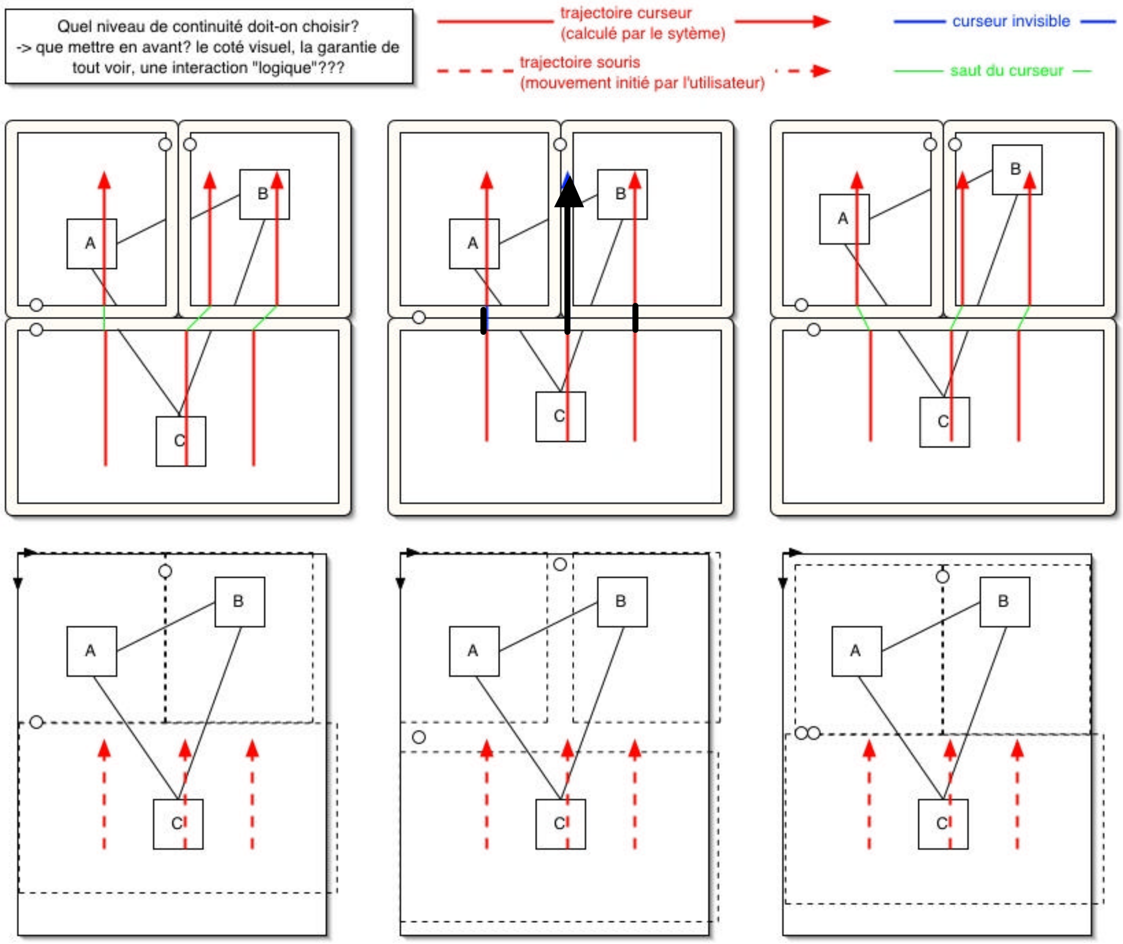

Figure 13. Mapping the mouse cursor onto three tiled surfaces using different politics. Thick lines denote cases where the cursor disappears although the user is still moving the mouse. With the politics used on the left and the right, the cursor never disappears. It is however jerky when it crosses the boundaries of surfaces.

5 CONCLUSION

In this document, we have reported our early analysis of the consequences of the digital-physical mapping function on the final result observable by users. We do not promote any type of mapping since it depends on the user's activities and the (yet to be invented) interaction metaphor. Instead, we stress the importance, for a middleware infrastructure like I-AM, to separate the mapping mechanisms from the mapping politics, to allow the politics to be defined by the application developers, and to provide defaults politics. In our current implementation, we offer two default politics: one that ignores the bezels, and one that takes the bezels into account with a possible loss of information.

6 REFERENCES

[Baudisch 03] Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., Zierlinger, A. Drag-and-Pop and Drap-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-Operated Systems. In Proc Human Computer Interaction –INTERACT'03, Rauterberg et al. Eds., IOS Press Publ., IFIP, 2003, pp. 57-64.

[Campbell 03] Campbell, C., Maglio, P. Segmentation of Display Space Interferes with Multitasking. In Proc Human Computer Interaction –INTERACT'03, Rauterberg et al. Eds., IOS Press Publ., IFIP, 2003, pp.575-582.

[Gorbet 98] Gorbet, M., Orth, M., Ishii, H. Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography. In Proc. CHI'98, Conf. On Human Factors in Computing Systsems, 1998, pp. 49-56.

[Rekimoto 01] Rekimoto Y., Ullmer B., Oba H. "DataTiles: A Modular Platform for Mixed Physical and Graphical Interactions". *In Proc. of the ACM conf. On Human Factors in Computer Human Interaction (CHI2001)*, ACM, Seattle, 2001, pp. 269-276.

[Tan 03a] Tan, D. Gergle, D., Scupelli, P., Pausch, R. With Similar Visual Angles, Larger Displays Improve Spatial Performance. In Proc. Computer Human Interaction –CHI 2003, ACM Publ., Fort Lauderdale, April 5-10, 2003, pp. 217-224.

[Tan 03b] Tan, D., Czerwinski, M. Effects of Visual Separation and Physical Discontinuities when Distributing Information across Multiple Displays. In Proc Human Computer Interaction –INTERACT'03, Rauterberg et al. Eds., IOS Press Publ., IFIP, 2003, pp. 252-255.

[Hinckley 03] Hinckley, K. Synchronous gestures for multiple persons and computers. Proc. UIST 2003, ACM, 2003.

[Izadi 03] Izadi, S., Brignull, H., Rodden, T., Rogers, Y., Underwood, M. Dynamo: A public interactive surface supporting the cooperative sharing and exchange of media. In Proc. UIST 2003, ACM Pub., 2003.

[Johanson 02] Johanson, B., Fox, A. The Event Heap: A Coordination Infrastructure for Interactive Workspaces. In Proc. of the 4h IEEE Workshop on Mobile Computer Systems and Applications (WMCSA 2002), June, 2002.

[Tandler 01] Tandler, P., Prante, T., Müller-Tomfelde, C., Streitz, N., Steinmetz, R. ConnecTables: Dynamic Coupling of Displays for the Flexible Creation of Shared Workspaces. In Proc. UIST 2001, ACM publ., 2001, pp. 11-20.

[Yee 03] Yee, K. Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers. In Proc. Computer Human Interaction –CHI 2003, ACM Publ., Fort Lauderdale, April 5-10, 2003, pp. 1-8.