Object-Oriented Interaction: Enabling Direct Physical Manipulation of Abstract Content via Objectification

Haijun Xia
University of Toronto
haijunxia@dgp.toronto.edu

ABSTRACT
Touch input promises intuitive interactions with digital content as it employs our experience of manipulating physical objects: digital content can be rotated, scaled, and translated using direct manipulation gestures. However, the reliance on analog also confines the scope of direct physical manipulation: the physical world provides no mechanism to interact with digital abstract content. As such, applications on touchscreen devices either only include limited functionalities or fallback on the traditional form-filling paradigm, which is tedious, slow, and error prone for touch input. My research focuses on designing a new UI framework to enable complex functionalities on touch screen devices by expanding direct physical manipulation to abstract content via objectification. I present two research projects, objectification of attributes [10] and selection, which demonstrate considerable promises.

INTRODUCTION
Direct-touch input offers several advantages over traditional indirect input, including mobility and the replacement of dedicated control surfaces with larger screens [8]. The use of direct physical manipulation of content, which can often be rotated, translated, and scaled with simple gestures, rather than manipulating offset controls, is now nearly universal in direct touch systems. Such manipulation focuses on employing knowledge of the physical world to interact with the digital one [7]. While this leads to interfaces which can be easily guessed, an apparent limitation is that such interaction techniques are limited by their reliance on analogs: the physical world provides no mechanism to directly physically manipulate the opacity of a photo, nor the transition between video clips. This results in a great variety of gestures which are guessed by users to try to perform each action, giving clear evidence that there is no universal set of "natural" gestures for HCI [9].

It is perhaps for this reason that applications for touchscreen devices which include even modest levels of functionality often fallback on the traditional form-filling paradigm which acts as the core of WIMP UI. While suitable for mouse and keyboard, on touchscreen devices, form filling is tedious, slow, and error prone [4]. As such, touchscreen devices are mainly used for content consumption instead of creation. My dissertation focuses on enabling direct physical manipulation of abstract content, with the hope of supporting applications with higher levels of complexity on touch screen devices as well as enhancing the functionalities of a UI to achieve tasks which were previously tedious or even impossible.

UIST'16 Adjunct, October 16-19, 2016, Tokyo, Japan
ACM 978-1-4503-4531-6/16/10.
http://dx.doi.org/10.1145/2984751.2984784

OBJECTIFYING UI ELEMENTS– THE BACKGROUND
Extensive works have investigated objectifying UI elements to employ knowledge of the physical world to interact with the digital one. However, it is perhaps because of the profound influence of usage of tools in human history that previous works have mostly focused on the objectification of tools or commands. Bier et al. [5] embodied tools as transparent widgets in a virtual sheet. Local tools [3] and HabilisDraw [1] focus on tools as first-class artifacts by importing the physical ecological properties of tools.

OBJECT-ORIENTED INTERACTION
Different from pervious works, my dissertation focuses on objectifying abstract content, which has no physical reference, to fully extend the use of direct physical manipulation. My previous and ongoing works have explored the objectification of the attributes of content [10] and the selection mechanism.

Attribute Object – The Micro Manipulation of Objects
In WIMP UI, manipulation of the many attributes has to be delegated to basic control widgets [2], which affords very limited interactions with attributes. Attribute Object extends direct physical manipulation gestures to the interaction of attributes without mediating user's input through form filling tools. We see attributes as more than parameters that define an object's appearance, layout, or behavior; they are treated as components of virtual objects; beyond seeing attributes as abstract numerical values, they are themselves objects which can be directly physically manipulated. In our drawing application, each attribute is assigned an independent identity and is represented as a paper-like card. As such, a drawing object is decomposed into a set of Attribute Objects organized in a tree structure (Figure 1). Attribute Objects may be attached to objects, and thus set the attribute for that object, or they may be detached from any object and sit independently on the screen.

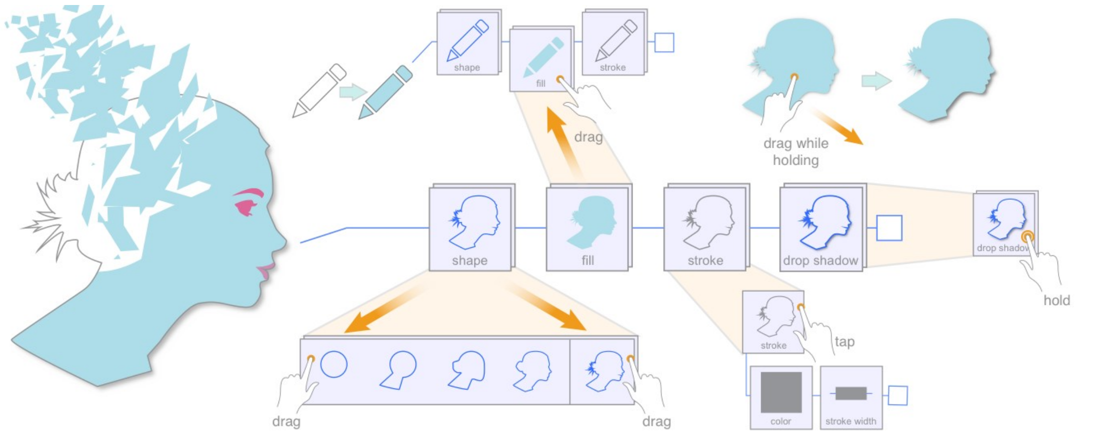

Figure 1Object-Oriented Drawing replaces most traditional WIMP UI with Attribute Objects which may be directly manipulated with traditional direct-touch gestures. This enables powerful and fluid interaction on touchscreen-based devices.

The Attribute Object concept combined with direct touch and pen input enables rich interactions that can be applied to any attribute of an application.

Cloning an Attribute

Attribute Objects may be directly cloned. Holding an AO with one hand and then tapping on the screen with the other clones it to the tapped position. A cloned card can be attached to another drawing object or be saved for later reuse. Cards can also be cloned directly into a drawing object's collection, simply by holding an AO and tapping the object the user wishes to clone it to. This allows for the quick and easy copying of style between drawing objects.

Linking Attributes

Though a simple clone is useful, additional power of style sharing comes from linking AOs, so that changes are instantly propagated. Attribute linking allows a many-to-many relationship between several objects' attributes. AOs are linked at the time of making a clone: immediately after the tap is performed, a link graphic is briefly displayed at the tap position that, if tapped, will create a persistent connection between the source and destination AOs.

Adjusting Attribute Value

Holding an Attribute Object places the system into a quasi-mode, in which manipulation gestures change the value of the selected Attribute Object. For example, holding the opacity card while moving one's finger on the canvas changes the alpha value; holding a drop shadow card while moving the finger directly manipulates the position of the shadow, as shown in Figure 1. Such direct manipulation eliminates most traditional UIs for attribute value selection.

Attributes as Drawing Templates

Holding an AO places manipulation gestures into a quasi-mode, so too the pen enters a mode related to a held AO. By default, the pen draws a path with its current fill and stroke attributes. If the user wishes to copy the style of an existing on-screen object to a current drawing, they can hold that object (or its desired attribute(s)) with the one hand and draw with the pen. This enables users to quickly create a large number of objects with the desired styles.

Restoring Attributes

By promoting attributes to Attribute Objects, there is a natural opportunity to provide a history for each attribute. By preserving each state of an attribute, a user may retrieve a previous state of any attribute without affecting other objects. A pinch gesture on the card reveals previous states (Figure 1). Tapping on a previous state rolls the AO back to the tapped state. In keeping with the theme of embodying attributes, each of the previous states of an Attribute Object is itself an object, with all of the capabilities of other AO cards. It is also worth pointing out that objects of any kind and any level maintain their own history, which enables flexible undo across the application.

Expert Evaluation Results

We conducted an expert evaluation to validate the belief that further objectifying the UI could add significant value to touch-based system. The results showed that experts were particularly fond of the ability to directly access each attribute, to share attributes among objects, and to manipulate attributes in a distributed manner without affecting others. Besides, experts also found the interface *uniform* and *discoverable* in that attributes can be easily navigated to according to the data structure, whereas in the WIMP UI, manipulation of objects has to be delegated through tools, which adds more indirectness to navigation.

Collection Object – The Macro Manipulation of Objects

Attribute Object discloses the inner world of objects to users and offers them more control of the micro level of objects. In dealing with complex drawings, users often have to collectively manipulate more than one objects. In WIMP UI, this is done by first selecting multiple targets and then applying the command. However, such typical operations suffer from the inherent limitations of direct manipulation paradigm and the ambiguity of the form filling paradigm.

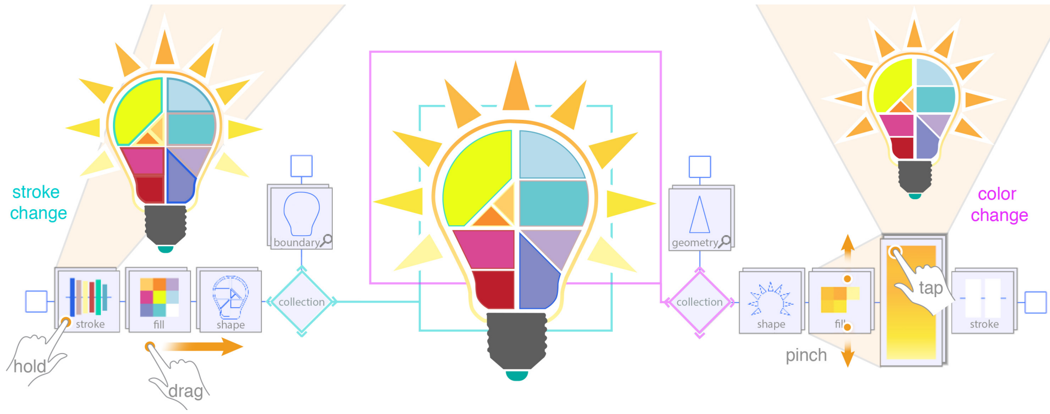

Figure 2 A drawing of a lightbulb, comprised by of a number of drawing objects, is manipulated using Collection Objects. Left: the thickness of all strokes is adjusted simultaneously. Right: The color of the "rays" is harmonized with a single gesture.

Difficulty of making selections can be exacerbated by a number of factors [6]. Selecting objects in a crowded or overlapping environment or ones simply out of the viewport requires a significant amount of time spent searching. Correctly identifying all the objects to be edited with a given value can also be challenging due to the existence of slightly different values. Relative value editing operations might not be supported and will therefore, require additional incremental selections. When a set of objects are selected, the WIMP UI only displays attributes with the same value and simply leaves the form blank if the selected objects have different values, which causes ambiguous interpretation of the selection and offers limited inteactions.

Beyond the problems of both selecting and editing multiple objects, in a typical workflow, users are forced to constantly deselect/reselect and group/ungroup the same objects. This results in a great number of redundant selections and frequent switches of context. Imagine a user wants to assign the fill color of one light ray triangle to all the triangles, after editing the outline of them (Figure 2). She will have to 1) deselect all the triangles, 2) select the triangle she wants to copy the color from, 3) look up and copy its value, 4) reselect all the triangles, and 5) finally apply the value.

My work on *Collection Objects* aims to provide fluid and rapid macro management of a number of objects by 1) providing a mechanism to ease the difficulties of multi selection, 2) enabling advanced manipulation of a collection of objects, and 3) supporting a fluid workflow of selecting and editing multiple objects

Replacing the Transient Selection and Rigid Group

Collection Object (CO) is the embodiment of selection, represented as a diamond shape (Figure 2). A user can hold his finger on the canvas to generate a CO. Holding the CO and tapping other objects adds them into the collection. A complex selection can be preserved simply by leaving the CO on the canvas. Different from *group*, CO can be set to a many-to-many relationship to selected objects. As such, one drawing object can be included in several COs.

Attribute Objects as Selection Filters

The top corner of the diamond connects to a set of attributes used as selection filters. Any attribute can be inserted into this set to enable selection by attributes. As such, a complex selection can be executed as a search operation. For example, one can simple copy the geometry of a triangle into the set to select all objects with the same triangle shape (Figure 2). Small objects can be selected by a size attribute; adjusting the value of the attribute dynamically changes the selection scope. Existing selection techniques such as lassoing and crossing are integrated by using the selection paths as boundary filters. Multiple selection filters can be applied together with an "and" relationship. As an example, one can select all the red objects in a region by combining a boundary filter and a color filter.

Enhanced Manipulation of a Collection of Objects

The right corner of the diamond connects to the aggregated attributes of the selected objects, which enables meaningful visualization and advanced manipulation of multiple objects. As shown in Figure 2, different values of an attribute are visualized in the card. A vertical pinch gesture revels useful operations to adjust the attributes of multiple objects simultaneously. For example, pinching on the color attribute reveals a gradient palette. Similarly, the position attribute provides different layouts (e.g. circular) of the selected objects and rotation attribute allows an interpolated set of rotations of the triangles. Such tasks are very tedious and difficult to do in current drawing applications.

Fluid Workflow of Selection and Manipulation

CO aims to enable a fluid workflow to interact with multiple objects by:

Using landing position as mode of touch: As stated, the value of an Attribute Object could be directly manipulated by holding its card and sliding another finger on the screen (e.g. holding the "drop shadow" card and dragging on the background translates the shadow). CO further uses the landing position as a mode of touch input. Now, if a user holds the stroke attribute of the collection, and lands another finger on the canvas, moving her finger will change the stroke width of all the objects in the collection. If she lands her finger on a particular object, only that object's stroke width will be changed. This can significantly reduce the switch between single selection and multi-selection.

Fast sharing of attribute across a collection: With CO, assigning the fill color of one triangle to all the selected triangles can be trivially achieved by holding the fill attribute of the collection of objects, and then tapping any other object to retrieve its fill color.

Interaction between the two sets of attributes: The set of attributes which are used as selection filters and that of the selected objects are collocated on the diamond shape. As the selection-action sequence keeps going, an intermingled series of selection and editing operations may be rapidly executed on these two sets, without needing to context switch between different menus or tools.

Expert Evaluation Results

Experts responded positively to the concept of CO. They found selection by filters allows them to easily select a large amount of objects without spending effort searching the whole canvas. Additionally, being able to fluidly switch focus between objects of interest enables them to quickly execute a series of operations.

FUTURE WORK

X Object

Future work can keep exploring extending direct physical manipulation to other elements of an interface. To name a few, Input Objects, the embodiment of the user's actions, could enable easy re-use of complex user input; Workflow Objects, objectification of a series of user actions, could be used for skill acquisition and sharing.

Adaptability

The paper card metaphor and gestures are designed in the context of a 2D drawing application with pen + touch interaction. Future work can explore applying the concepts of AO and CO to other domains. For example, in data visualization, users have to interact with many attributes of both the data and its visual representation. It would be interesting to explore how the proposed techniques can assist data visualization authoring and exploration. The adaptability of the concept can be further investigated by evaluating the expressiveness and usefulness with other input technologies, such as a desktop setup with mouse and keyboard input and Virtual Reality with hand gestures.

Development of Object-Oriented User Interface

Consider modeling the content of an application and its change with a state machine: attributes describe the states, while commands and tools drive the transitions. Most existing systems are command-centered. Therefore, development of such applications focuses on implementing tools of the application. Our approach, on the other hand, enables direct access to states, with transitions driven by pen and touch manipulations. Since the structure of Attribute Object is a direct mapping to the underlying data format, future research can explore how the development effort can be reduced by automating the development by using the existing data format as input.

CONCLUSION

My works focus on extending the direct physical manipulation to abstract content of an interface via objectification. Proved useful in expert evaluations, AO and CO hold considerable promises to enable higher level of functionalities with direct physical manipulation. My future work will focus on applying the concept to other UI elements, application domains, and input technologies to further reveal its strength and weakness. I believe the doctoral symposium is a wonderful opportunity to deeply discuss my work and inspire new ideas of great breadth.

ACKNOWLEDGEMENT

I would like to thank the guidance and support from my supervisor Daniel Wigdor and the contribution from my collaborators Bruno Araujo and Tovi Grossman.

REFERENCES

1. Amant, R. and Horton, T. Characterizing tool use in an interactive drawing environment. *SMARTGRAPH '02*, 86-93.
2. Beaudouin-Lafon, M. Instrumental interaction: an interaction model for designing post-WIMP user interfaces. *CHI '00*, 446-453.
3. Bederson, B., Hollan, J., Druin, A., Stewart, J., Rogers, D., and Proft D. Local tools: an alternative to tool palettes. UIST '96, 169-170.
4. Benko, H. and Wigdor, D. Imprecision, Inaccuracy, and Frustration: the Tale of Touch Input. In Tabletops - Horizontal Interactive Displays, Springer London, 2010, 249–275.
5. Bier, E., Stone, M., Pier, K., Buxton, W., and DeRose, T. Toolglass and magic lenses: the see-through interface. *SIGGRAPH '93*, 73-80.
6. Frohlich, D. The history and future of direct manipulation. *Behaviour & Information Technology* 12.6 (1993): 315-329.
7. Jacob, R., Girouard, A., Hirshfield, L., Horn, M, Shaer, O., Solovey, E., and Zigelbaum, J.. Reality-based interaction: a framework for post-WIMP interfaces. *CHI '08*, 201–210.
8. Jobs, S. 2007. Mac World Expo 2007: Keynote.
9. Wobbrock, J., Morris, M., and Wilson, A. User-defined gestures for surface computing. *CHI '09*, 1083–1092.
10. Xia, H., Araujo, B., Grossman, T., and Wigdor, D. Object-Oriented Drawing. *CHI '16,* 4610-4621.