# An Interaction Model for Touch-Aware Tangibles on Interactive Surfaces

**Simon Voelker**
RWTH Aachen University
Aachen, Germany
voelker@cs.rwth-aachen.de

**Kjell Ivar Øvergård**
Buskerud and Vestfold
University College
Tønsberg, Norway
koe@hive.no

**Christian Corsten**
RWTH Aachen University
Aachen, Germany
corsten@cs.rwth-aachen.de

**Jan Borchers**
RWTH Aachen University
Aachen, Germany
borchers@cs.rwth-aachen.de

**Nur Alhuda Hamdan**
RWTH Aachen University
Aachen, Germany
hamdan@cs.rwth-aachen.de


CHI 2014, Apr 26 - May 01 2014, Toronto, ON, Canada
ACM 978-1-4503-2474-8/14/04.
http://dx.doi.org/10.1145/2559206.2581273

## Abstract
Tangibles on interactive surfaces enable users to physically manipulate digital content by placing, moving, manipulating, or removing a tangible object. However, the information *whether* and *how* a user grasps these tangibles has not been exploited for input so far. Based on Buxton's Three-State Model for graphical input, we present an interaction model that describes input on tangibles that are aware of the user's touch and grasp. We present two examples showing how the user benefits from this extended interaction model.

## Author Keywords
tangibles; interaction model; interactive surfaces

## Introduction
Tangibles on interactive surfaces [3] bring physical interaction to the user by addressing the haptic sense, exploiting physical affordances, and enabling eyes-free interaction [12]. They can serve as direct physical representations of digital objects [9], or as generic tools to physically manipulate any content [12]. When interacting with a tangible, the user necessarily touches it— when placing the tangible object on the surface, or right before and while manipulating it. Research on tangibles beyond interactive surfaces shows that the information whether and how a tangible is touched can be used to enrich the

expressiveness of tangible interaction: Touché [8], for example, augments a door knob with a grasp password by analyzing how the user grasps the knob, and HandSense [13] uses capacitive sensors to recognize whether an object is being held in the left or right hand. However, for tangibles on interactive surfaces, the design space of similar touch information has not been mapped out yet.

Our contribution is an extension of Buxton's interaction model [2] to characterize the state of touch-aware tangibles on interactive surfaces, and to explore how touch information can be used to enrich the expressiveness of such tangibles.

## Related Work

In [2], Buxton presents a Three-State Model for input devices, such as mouse, touch screen, and puck or stylus on a tablet (Fig. 2). The model represents all possible interaction states of each device: an input device that has no effect on the system, like a puck that is not in contact with the tablet surface yet, is considered in State 0 (*Out of Range*). The system starts detecting the puck in State 1 (*Tracking*) when the puck touches the tablet. When physical manipulation is involved, such as pushing a button on the puck, the device enters State 2 (*Dragging*).

This model enables designers to choose a suitable input device for an application or task, and to compare different input states. For example, Patten and Ishii [4] used Buxton's model to compare mouse input in a graphical user interface (GUI) with non-manipulable tangibles in a tangible user interface (TUI). Similarly, Bricks [3] and URP [9] follow this model.

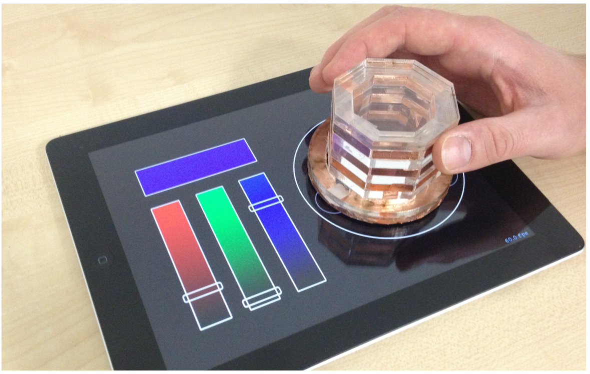

**Figure 1:** A touch-aware knob acting as an RGB color picker for an off-the-shelf iPad: rotating the knob by grasping it at the top adjusts red, at the middle green, and at the bottom blue. The knob consists of simple acrylic and copper foil.

Buxton's model has also been modified and extended to capture new strategies for touch input on interactive surfaces. Benko et al. [1] distinguished State 1 and 2 for finger input on touch screens by taking the size of a touch into consideration and by distinguishing single from double touch. Richter et al. [7] added a fourth state to Buxton's model to capture pressure input on a pressure-sensitive touch screen.

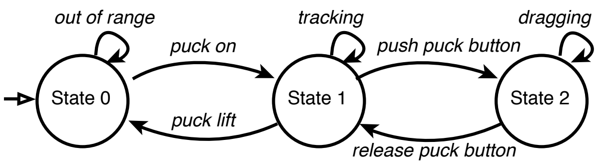


**Figure 2:** Buxton's Three-State Model for a puck with a pushbutton on a tablet [2].

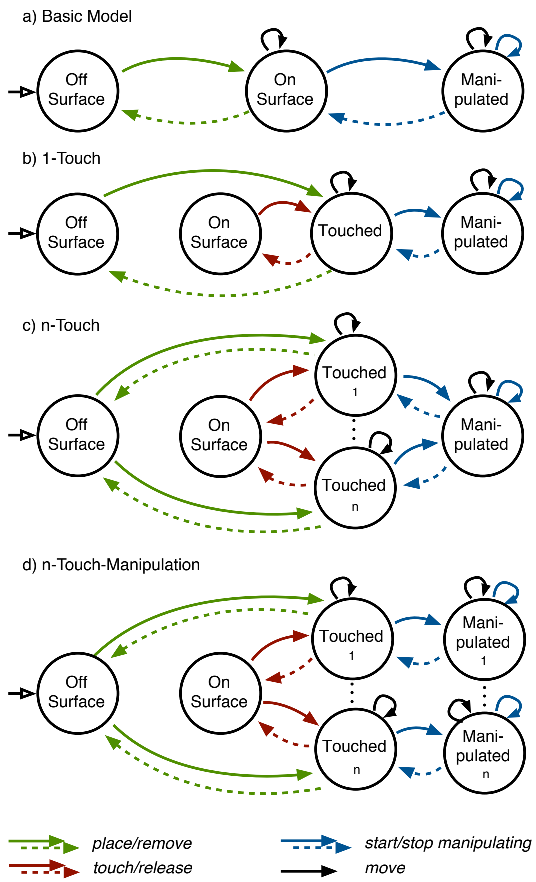


**Figure 3:** Consecutive tangible interaction models. (a) Generalization from Buxton's Three-State Model for a puck on a tablet to tangibles on interactive surfaces, (b) including *whether* a tangible is touched, (c) *how* it is touched, and (d) *how* it is *manipulated*.

## Interaction Model

Fig. 3a shows our basic model for tangibles on interactive surfaces, an adaptation of Buxton's Three-State Model for a puck on a tablet. When the tangible is *Off Surface*, it has no effect on the system until the user places it *On Surface*. As long as the tangible rests on the surface, the *On Surface* state is retained. When the user manipulates the tangible, for example, to push the puck button, the tangible reaches the third state: *Manipulated*. When manipulation stops, the tangible returns to the *On Surface* state until the user removes the tangible from the surface or manipulates it again.

In this model, a user can place, move, and manipulate a tangible on an interactive surface or remove it. All these actions, however, require another preceding action, namely *touching* the tangible. Wimmer [13] and Boring have shown that such touch information can be used as additional input, but this information has not been included in models for tangibles on interactive surfaces so far. We will explain how touch information can be included into the basic model and how this, in turn, extends the interaction model of tangibles on interactive surfaces.

### *Touching the Tangible*

Using our model (Fig. 3a) as a starting point for tangible interaction on interactive surfaces, we add a separate *Touched* state that is reached when the user touches the tangible—prior to placing, moving, manipulating, or removing it from the surface. By adding this *Touched* state, we derive three consecutive models, which we refer to as *1-Touch*, *n-Touch*, and *n-Touch-Manipulation*.

### *1-Touch*

The first extension is a single *Touched* state that models whether the user is touching a tangible or not (Fig. 3b).

This binary distinction can be used as a quasimode [5] or to trigger a transcendent action that stops as soon as the tangible is not touched anymore. For example, a tangible may display additional "tooltip" information around it while it is being touched, as we show in our sample applications.

We include this *Touched* state in the model based on the observation that each action in the basic model requires the user to touch the tangible. While *placing* the tangible on the surface, the user stays in contact with it. Therefore, *placing* results in the *Touched* state. When *releasing* the tangible, the state changes to *On Surface* until the user grasps the tangible again, which necessarily involves *touching* it, therefore leading back to the *Touched* state. Continuously *moving* the tangible on the surface does not exit the *Touched* state until the user *manipulates* the tangible. When manipulation stops, the user is still in contact with the tangible, therefore the *Touched* state is re-entered. As *removing* the tangible from the surface is inverse to *placing* it, the *Off Surface* state is only reached via the preceding *Touched* state.

**Figure 4:** The SLAP knob [12] is a rotary button that can be pushed to confirm input.

*n-Touch*
So far, the model is limited to the binary information of *whether* a tangible is touched or not. To include information about *how* the user is touching the tangible, we extend our 1-Touch model by expanding the *Touched* state to several distinct *Touched*$_i$ states (Fig. 3c). Knowing how a tangible is touched can be used to provide different tooltips for tangibles that react to various user grasps. A rotary knob like the SLAP knob [12] (Fig. 4) could display different information when the user is about to turn it (by grasping the side) vs. when she wants to push it (by touching the top).

In the n-Touch model, each different way to grasp a tangible is represented by its own *Touched*$_i$ state (Fig. 3c). For example, if the door knob presented in Touché [8] can identify 10 different grasps, the model would have 10 different *Touched*$_i$ states. Basically, a widget with $n$ binary touch areas supports $(2^n - 1)$ *Touched* states. For our color picker from Fig. 1, which has three binary touch areas, a corresponding model would have seven discernible *Touched*$_i$ states.

As in the 1-Touch model, each *Touched*$_i$ state is connected to the *Off Surface*, *On Surface*, and *Manipulated* states by the corresponding actions.

*n-Touch-Manipulation*
The n-Touch model describes how the user is touching an object, but not how she manipulates it. Recalling the rotary knob, this model cannot tell whether the user is turning the knob with, e.g., two or three fingers. To combine the information of touch and manipulation, each *Touched*$_i$ state needs a *manipulation* transition to its own *Manipulated*$_i$ state (Fig. 3d). Hence, the information of where the user is touching the tangible influences the manipulation result. For example, a rotary knob that is used as an RGB color picker could change each color channel selectively depending on *where* the user touches the knob while rotating it (Fig. 1). Besides the added $n$ *Manipulated*$_i$ states, this model is identical to the n-Touch model.

In both the n-Touch and the n-Touch-Manipulation model, transitions between the *Touched*$_i$ states or *Manipulated*$_i$ states are possible if the user is able to change *how* she is touching the tangible without releasing the tangible. However, for visual simplicity, we omitted these transitions in Figures 3c-d.

## Sample Applications

In this section we explore the feasibility of our model by providing two interaction design examples. We developed both examples on capacitive touch displays. To detect if a tangible is placed on the display we used the PUCs marker setup introduced by Voelker et al. [10]. To detect *whether* and *how* a tangible is touched we attached conductive material such as copper foil to the tangible's surface, such that touches on the tangible are redirected to and recognized by the touchscreen [6].

*1-Touch: Tutorial on Demand*
In GUIs, tooltips provide help on demand without cluttering the screen. Using the new *Touched* state, we can bring this principle to TUIs.

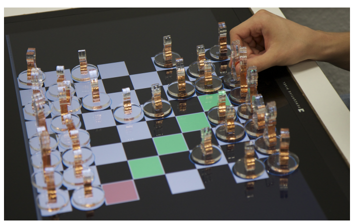

**Figure 5:** *1-Touch* example: By touching a chess piece, the user is shown where he can move the playing piece (green squares) and where the piece can be hit by his opponent (red squares).

One example is a chess tutorial for a novice chess player who may not know what movements a specific playing piece allows. When he grasps, e.g., the bishop, the underlying virtual checkerboard displays all possible paths that he can move the bishop to (Fig. 5). Instead of permanently displaying possible moves for all tangibles, information is only displayed on demand per playing piece. This reduces clutter and occlusion.

In general, having the tangible react to being touched can be exploited for quasimodes [5] that are activated prior to when the user manipulates a tangible. The user can safely explore the TUI before actually manipulating objects.

*n-Touch-Manipulation: Coarse-to-Fine Video Navigation*
As another apt example, we built a multi-granularity video navigation knob. Turning the knob clockwise forwards the video at a certain granularity, e.g., 30 frames per turn. To select different levels of granularity, distinct touch-sensitive rings are wrapped around the tangible from top to bottom, each representing a different level of granularity: Turning the knob using the lowest ring could result in the coarsest granularity; higher rings in finer granularities. Alternatively, the layout of the touch zones could semantically match different user grasps. Turning the knob from the top with thumb and index fingers (precision grip) results in fine motor adjustments and therefore could map to fine granularity, whereas grasping the tangible with the entire palm would map to coarser control.

This design does not require adding hinged mechanical parts to the tangible, and instead of using one tangible per granularity, it can offer different granularities using just one physical controller. This saves space on the interactive surface and allows the user to switch granularity levels without looking at the knob, e.g., by sliding the hand up and down along the touch rings.

## Limitations

Taking a closer look at existing research on tangible interaction on tabletops, our model still has some limitations. Madgets [11], e.g., are actuated widgets on interactive tabletops that can enter the *Manipulated* state without the user having to touch the tangible: electromagnets move the widget autonomously. Similarly, flicked tangibles do not require the user to grasp the tangible to manipulate it. Once flicked, the user is not touching the tangible anymore; it slides across the surface autonomously. We will extend our model by adding new states and transitions to include interaction with actuated and flicked tangibles. In this context, we will apply our model to other existing research on tangibles on interactive surfaces to investigate its applicability and generalizability.

## Conclusion

Detecting whether tangibles on interactive surfaces are being touched or not, and how they are being touched, significantly expands the design space of possible interactions with this class of user interfaces. We adapted and extended Buxton's well-established Three-State Model to represent these additional states, leading to our *1-Touch*, *n-Touch*, and *n-Touch-Manipulation* models. Two sample applications demonstrated the increased richness of interaction that our model can capture. We hope that our models facilitate future research and design in the exciting new space of multi-touch tangibles.

## Acknowledgments

This work was funded in part by the German B-IT Foundation and by the Norwegian Research Council through the MACS project (project number 217788).

## References

[1] Benko, H., Wilson, A. D., and Baudisch, P. Precise Selection Techniques for Multi-touch Screens. In *Proc. CHI '06*, 1263–1272.

[2] Buxton, W. A Three-State Model of Graphical Input. In *Proc. INTERACT '90*, 449–456.

[3] Fitzmaurice, G. W., Ishii, H., and Buxton, W. A. S. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proc. CHI '95*, 72–79.

[4] Patten, J., and Ishii, H. A Comparison of Spatial Organization Strategies in Graphical and Tangible User Interfaces. In *Proc. DARE '00*, 41–50.

[5] Raskin, J. *The Humane Interface: New Directions for Designing Interactive Systems*. Addison-Wesley Professional, 2000.

[6] Rekimoto, J. SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. In *Proc. CHI '02*, 113–120.

[7] Richter, H., Ecker, R., Deisler, C., and Butz, A. HapTouch and the 2+1 State Model: Potentials of Haptic Feedback on Touch Based In-vehicle Information Systems. In *Proc. AutomotiveUI '10*, 72–79.

[8] Sato, M., Poupyrev, I., and Harrison, C. Touché: Enhancing Touch Interaction on Humans, Screens, Liquids, and Everyday Objects. In *Proc. CHI '12*, 483–492.

[9] Underkoffler, J., and Ishii, H. Urp: a Luminous-Tangible Workbench for Urban Planning and Design. In *Proc. CHI '99*, 386–393.

[10] Voelker, S., Nakajima, K., Thoresen, C., Itoh, Y., Øvergård, K. I., and Borchers, J. PUCs: Detecting Transparent, Passive Untouched Capacitive Widgets on Unmodified Multi-touch Displays. In *Proc. ITS '13*, 101–104.

[11] Weiss, M., Schwarz, F., Jakubowski, S., and Borchers, J. Madgets: Actuating Widgets on Interactive Tabletops. In *Proc. UIST '10*, 293–302.

[12] Weiss, M., Wagner, J., Jansen, Y., Jennings, R., Khoshabeh, R., Hollan, J. D., and Borchers, J. SLAP Widgets: Bridging the Gap Between Virtual and Physical Controls on Tabletops. In *Proc. CHI '09*, 481–490.

[13] Wimmer, R., and Boring, S. HandSense: Discriminating Different Ways of Grasping and Holding a Tangible User Interface. In *Proc. TEI '09*, 359–362.