Co-Adaptive Instruments

Human beings interact with the world through instruments. Although you can write your name in the sand with your finger, most of the time, you write with a pen or pencil or whiteboard marker on the appropriate surface.

In each case, an instrument *mediates* the interaction between the user and an ‘object of interest’. When you write, your object of interest is the message: you apply the pen to the paper to create that message. Sometimes, the instrument itself becomes an object of interest. When you use a pencil sharpener to sharpen a pencil, the pencil becomes the object-of-interest and the pencil sharpener is the instrument. The relationship between instruments and objects-of-interest is thus dynamic, similar to Gibson’s notion of ‘affordances’. Instruments are not instruments in an absolute sense, but only with respect to the objects they affect.

The concept of instrumental interaction (M. Beaudouin-Lafon, 2000) offers a powerful way of thinking about interaction, enabling us to create simpler and yet more powerful ways to manipulate objects. The idea is to decouple instruments from the applications in which they are normally embedded, such that they can be applied to any type of appropriate data. Instruments are based on the following principles (Beaudouin-Lafon & Mackay, 2000):

1. **Reification:** Turns actions into persistent objects in the interface
   - Reification of a command into an instrument
     - scrolling -> scrollbar

2. **Polymorphism:** Instruments can operate on multiple object types
   - The same instrument operates on various objects
     - delete, move

3. **Reuse:** Captures interaction patterns for later reuse
   - Output: Reuse previously created objects
     - duplicate, copy/paste
   - Input: Reuse previously issued commands
     - redo, history, macros

To describe an instrument, begin with the user’s goal, here, to choose a color. To pick a color the user moves the magnifier (the instrument) to the desired color on the screen (the object of interest) and clicks. The instrument sends a command that updates the selected color. The user sees two types of feedback: the instrument magnifies the area with the color, and the color swatch changes color.

Some instruments, like thermometers, are designed only to detect information. Others, like thermostats, both detect information and change the environment accordingly. Here, the color picker can both detect colors and also apply them to various objects. (The diagram and description above only cover the selection of a color, not its application to an object.)
Some instruments require a great deal of practice and skill, such as learning to write correctly with a pen. However, users do not always use instruments in the ways that their designers intended. A pen may be used to write a letter, but also to prop open a door or poke a hole in a piece of paper. This phenomenon is called co-adaptation (Mackay, 1990): users both adapt their behavior in response to the design of a system and also adapt the system in novel, often unanticipated ways.

The challenge for designers is to create instruments that support both kinds of adaptation: learning and appropriating it. Providing feedforward helps make instruments incrementally learnable, telling users what is possible and how to express their intentions in a way the system can understand. Providing feedback, from the instrument as well as the object-of-interest, helps users know if their actions were successful. Creating appropriable instruments is more subtle: users should be able to redefine instruments and how they operate, through the principles of reification, polymorphism and reuse.

Refer back to your storyboard and consider an interaction point. What is the object-of-interest? What is the action that the user wants to perform? Is there a way to convert an action into an instrument that persists independently of that particular situation? Do users need to detect information in the world or modify that information? Could a single instrument do both? Can such an instrument be used in multiple settings, on multiple objects-of-interest? What happens when users need to perform the same or similar actions multiple times? Is it possible to reuse previous actions or previously created objects?

What is the interaction like? How do users express their intent? How do users find out what actions are possible and whether the instrument will work on a particular type of object? Is it possible to create incremental feedforward, such that novices receive the help they need while experts perform quickly? What kind of feedback is appropriate? Can this instrument be used at other interaction points in your scenario? What changes are needed to make it more flexible, simpler and/or more powerful?