

Betty Smith, Jude Yehling
CSCi 626 - Fall 2025
October 19 2025
Prof. Maoyuan Sun

Direct Manipulation

When doing something on a computer, I am able to see objects on an interface and interact with them with the key and mouse, and sometimes even my finger on a touch screen. I can click, drag, draw, and type and see it all happening in real time. Being able to do this has not always been possible. Early computers could only be controlled with complex commands line by line, and did not have the amazing visuals they have now. This type of human computer interaction is called Direct Manipulation.

Direct Manipulation is a term introduced by Ben Shneiderman. There are three main principles of Direct Manipulation, and are as follows: visibility of objects and actions of interest; the replacement of precise command syntax with physical actions such as clicking, dragging, etc; and rapid or reversal actions that have basically immediate visual feedback. This is an example of a model-world relationship, where the user is able to actually be engaged with the data or actions on a computer.

This topic still feels difficult to express because it is abstract in a way. A very simple way to understand Direct Manipulation is “What you see is what you get”, which was said by Don Hatfield from IBM (Shneiderman). This is a very high level take on Direct Manipulation, but also helps the topic be way more understandable.

For this assignment, we read two papers. The first one was by Ben Shneiderman in 1983 (Shneiderman paper) and the other was by Hutchins, Hollan, and Norman in 1985 (Hutchings paper).

Shneiderman’s paper really strongly focuses on usability and human satisfaction. His paper went through multiple examples of Direct Manipulation including video games, EMACS, VISICALC, and cars. Using these examples, Shneiderman tried to illustrate what Direct Manipulation is. The most understandable example in my opinion is the car example. When driving a car, the driver (user) can see the landscape through the window (interface). The driver uses the brake pedal (input) to slow the car down. The driver can immediately view the car slowing. This is exactly how Direct Manipulation works, the user does something that has an immediate visual reaction.

The Hutchins paper dives more into the why and how of Direct Manipulation. This paper introduces the idea of “cognitive distance” which has to do with the feeling of directedness. This directedness has two main obstacles: the gulf of execution, which is the gap between the user's visual goal and the physical actions the computer or system needs to achieve it; and the gulf of evaluation, which is the gap between the computer or system’s feedback and the user’s interpretation of it. The Hutchins paper makes the argument that Direct Manipulation feels direct because its features make these gulfs very small.

The concept of Direct Manipulation is the same concept in both papers, but the Shneiderman paper is about the design features and user satisfaction. The Hutchins paper is an extension of this and gives the reasoning behind the how of this. The Shneiderman paper is also more invested in the description of Direct Manipulation. Many examples are used to explain what Direct Manipulation is. The Hutchins paper is more of an analysis of Direct Manipulation. It really explains the semantics of the topic. Instead of describing an abstract topic, it explains the mechanics behind it.

A modern example of Direct Manipulation is a content creation platform like Canva. On Canva, users drag different objects to create projects like flyers. For example, if you want the color of the page to be blue, you are able to click the background color and change it to blue and it immediately changes. On the flip side, if Direct Manipulation was not an option, to change the color of the screen on a project, you would have to write the code to change the color of all the pixels to a specific hex color. You would then have to run the code to actually see the color change. That is the big difference here. Using Canva, visualizing change is, to the human eye, immediate, which not having access to Direct Manipulation would mean understanding complex commands and longer visual wait times.

Using the Hutchins paper's analysis of the *gulf of execution and evaluation* and *semantic and articulatory distance* the advantages and disadvantages of Direct Manipulation become more clear. The most obvious pro is that Direct Manipulation drastically narrows the cognitive gulfs for many actions on an interface. It reduces the semantic distance, or the gap between the intention that fuels the user's actions and the effect of it. For example, trashing something completes the action of deleting it. And it reduces articulatory distance, or the gap between the action's physical impression and its goal. Like dragging an object looks and feels like moving it.

A major con of Direct Manipulation is that it widens the gulfs for abstract, repetitive, and precise tasks that require higher technical understanding. For example, if a user had the goal to delete all files in specific folders that have not been accessed in over 30 days, using a Direct Manipulation interface would have a large semantic distance since there is no icon or button that can accomplish this with a single action. A user would have to go through every file, check its last access date, then delete or trash each matching file individually. Using command-line instructions would be a better way to accomplish this goal. Writing a code snippet that finds all accessed files 30 or more days ago and deleting them when run would accomplish this user's goal much more efficiently. In precision cases such as this, command or script is a more efficient and "direct" solution.

Hutchins, E. L., Hollan, J. D., & Norman, D. A. (1985). Direct manipulation interfaces. *Human-Computer Interaction*, 1(4), 311–338.

Shneiderman, B. (1983). Direct manipulation: A step beyond programming languages. *Computer*, 16(8), 57–69.