CS160, Spring 09
Nicholas Kong

Notes on the gulfs of execution and evaluation from "Direct Manipulation Interfaces", Hutchins et al.

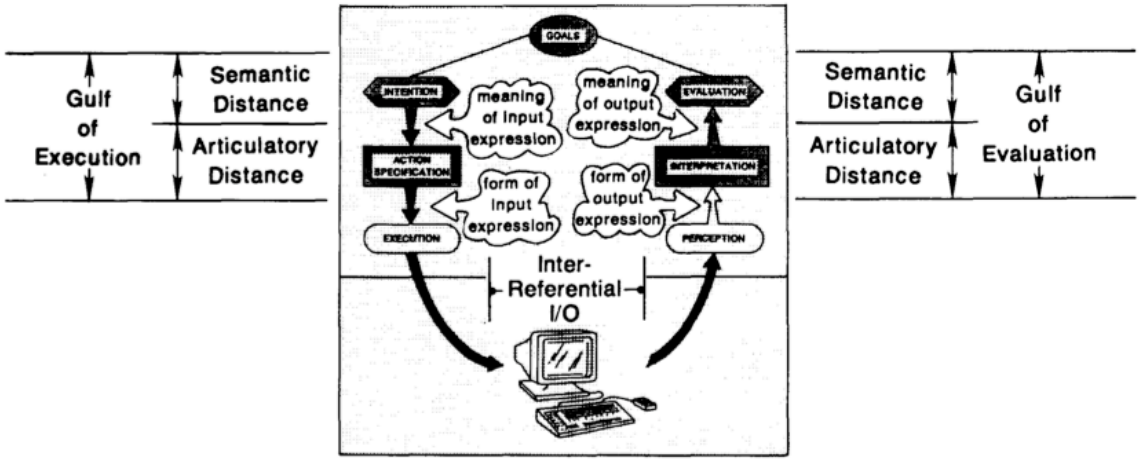

Figure 6 of Hutchins et al. [1] (shown above) is important and provides a pretty good distillation of the concepts. The paper can be a little confusing because some examples blur the lines between semantic and articulatory distances. I think the key contribution of the paper is to seed thought about what "intuitive" or "obvious" means in an interface. The exact pigeonholing of certain actions into their little bins is not as important.

As an interaction designer you will need to make decisions that help the user bridge both gulfs: your interactions will assist in bridging the gulf of execution, whereas your choice of output or feedback will assist in bridging the gulf of evaluation. However, it can be difficult to bridge semantic distance in all cases, as the user will inevitably have goals the designer will not have planned for. Some of the following examples of bridging the gulf are designer-related, while others are user-related.

Definitions

First, let's look at a few quotes from the paper:

> The **gulf of execution** is bridged by making the *commands and mechanisms* of the system match the *thoughts and goals* of the user.

In other words, to bridge the gulf of execution, you must translate your ideas or goals into the language of the input.

> The **gulf of evaluation** is bridged by making the output displays present a good conceptual model of the system that is readily perceived, interpreted, and evaluated.

To bridge the gulf of evaluation, you must translate the system's output language into your own internal language.

Both of these gulfs are decomposed into two types of distances, semantic and articulatory distances.

Semantic distance

> *Semantic distance* reflects the relationship between the *user intentions* and the *meaning of expressions* in the interface languages.

There are two interface "languages", one for input and one for output. The input language is comprised of the possible actions and commands; the output language is comprised of the feedback of the system. To bridge semantic distance, we must therefore provide interface commands which require little translation to or from the goals of the user.

Hutchins et al. pose two main questions about semantic distance:

1. Is it possible to say what one wants to say in this language [i.e., input or output language]?
2. Can the things of interest be said concisely [in this language]?

In other words, are the user's goals *expressible* in the interface language, and can the user *transform* their goals easily into the interface language? For example, assume the user wants to find the sum of a list of numbers. Let's look at three cases:

- If the user were to use **assembly language**, the user would have to deal with the intricacies of memory register storage and perhaps a limited instruction set. This results in a large semantic distance; it may take the user some time to discover the capabilities of the language as well as how to transfer his or her goals into the strict syntax.
- In a higher level programming language, say, **Python**, the user need only use a library function. While concise, the user may still experience a difficulty in determining whether the sum is possible, if they are not familiar with the math library.
- In a spreadsheet program (**Excel, Calc**), the user needs to input the data, then select and click a "Sum" button. If the Sum button is clearly displayed the answer to the first question comes quickly. "Saying" the expression "sum these numbers" is also concisely said, as there is a direct mapping from the goal of summing numbers to the vocabulary (i.e., a button) of the interface.

The paper also emphasizes that there is a difference between *"automated behaviour"* (memorization of the system's interface vocabulary) and *semantic distance*. In the first case, the user has adapted to the system to such an extent that their intents are already formed in terms of the system vocabulary, where in the second the system's vocabulary has been designed to match the user's intent.

- An example of **automated behaviour**, given in the paper, is how a regular user of vim (a Unix text editor) had internalized the command to delete a word (`dw`). While the user likely still forms the goal "delete this word", he has internalized the commands to the extent that the translation between goal and input language is instantaneous. However, to a beginner user of vim, this translation is far from obvious; hence, the semantic distance for this operation is still large.

- A system may reduce **semantic distance** by having its input vocabulary move closer to the user's goals. This may be achieved by making use by understanding users' internal models, many of which come about through metaphor. For example, let's say the user wants to move a file from one folder to another in your favorite graphical OS. This is achievable by drag-and-drop, where the user selects an icon, then physically moves it (via the mouse) to another folder. The semantic distance is small because the system has provided an input language which matches the user's goals.

Articulatory distance

> *Articulatory distance* reflects the relationship between the *physical form* of an expression in the interaction language and its *meaning*.

By "physical form" we mean the actual actions required to invoke an "expression" in the interaction language.

Let's take an example with the gulf of execution . Say we have a "Dismiss all" button in an error dialog. The meaning of this expression in the interaction language is to ignore all further errors. The physical form of this expression is the requirement of the user to move the mouse cursor to the button and click. The articulatory distance is the effort required of the user to connect the meaning of the "dismiss all" button to the physical action required to depress it. If the button is well designed and visually affords pressing, the articulatory distance is very small.

On the gulf of evaluation side, the articulatory distance is the translation between the physical form (e.g., string, picture, sound, etc.) into the meaning of that expression *in terms of the output language*. For example, take the example of a mail notifier that changes color when you have new mail. The articulatory distance is the effort required of the user to determine what the change in color means.

Examples

ESP game

Gulf of execution

The goal of the player is to match tags of an image with their partner. Let's split this gulf up into its component distances.

- **Semantic distance.** The player's idea/goal is to input a tag for an image. The semantic distance is composed of the "translation" of the player's goal into the interface's language. So what's allowed in this interface? We have a textbox in which we can input a tag. It is apparent that the player must type their chosen tag and then confirm, either by hitting enter or clicking submit. The textbox affords typing, so the effort the user needs to exert to connect their goal to the system vocabulary is small, and thus the semantic distance is small.

- **Articulatory distance.** There are at least two physical forms of the actions we have to underatke in the system, namely the input of the tag into the textbox and the confirmation that the player has matched tags with his/her partner. The user must type the tag, but can then either press "Enter" or move the mouse cursor and click "Submit". I argue that pressing "Enter" has a slightly smaller articulatory distance

than clicking the "Submit" button. Pressing "Enter" to submit is more physically direct than using the mouse.

Gulf of evaluation

The player wishes to know whether they've successfully matched a tag with their partner. Let's split this gulf into its component distances as well. Note that in the gulf of evaluation we pass through the articulatory distance first, then the semantic distance.

- **Articulatory distance.** The player learns of a successful match via a pop-up notice that notifies the player of which word was matched on. Thus, the articulatory distance is very small, in that the string "You matched on jet" directly reveals the system's expression "matched words". The articulatory distance might be somewhat increased if instead of a pop-up box, the word that the players matched on was *highlighted*. In this case, the player would first have to translate the output of a highlighted word into the system expression "matched word", which might be less clear than the current text feedback.
- **Semantic distance.** The semantic distance is also small, as the goal of the player is to match words, and the feedback string immediately provides this information to the user. The semantic distance in turn might be increased if the system instead returned a string like "Your partner typed 'jet'." The user would then need to do the extra word to determine if they too had typed 'jet' and thus matched.

First person shooter - increasing the articulatory distance in the gulf of execution

One way to increase the articulatory distance in the gulf of execution for a first person shooter like Halo is to provide an RPG-style menu-selection interface for firing a gun. That is, the player would have to select the "Action" menu, then the "Fire" menu, then the "Gun" menu, etc. In most first person shooters, the method to fire a gun is to depress a single button, in a sense *mimicking* the action of pulling a trigger. With a menu-based firing system, articulatory distance increases because there is more of a disconnect between the physical form and the meaning of "fire a gun". We have also made the articulatory language less direct, in this case.

Minimizing semantic distance in the gulf of execution

The question here is how we can go about minimizing semantic distance in the gulf of execution. This is possible by designing an interface whose vocabulary or language matches that of the goals/ideas of the user. Let's say we want to animate a character walking from point A to point B. In one interface, we may have to manually specify each keyframe, all the way from point A to point B. The semantic distance is quite large: we need to figure out how the actions of walking transfer to the smaller motions of a character, as the vocabulary of the system only allows for changes to individual frames. We could potentially reduce the semantic and articulatory distances by simply providing a "Walk" button. There would be virtually no translation from the user's idea of "make this character walk" to the language of the interface, thus reducing the semantic distance. Articulatory distance is also reduced because the physical form of the "walk" vocabulary item (a button displaying "walk") is

very indicative of its meaning. However, we would lose fine-grained control of the movement of the character.

Direct Engagement

An important concept from the paper is that of *direct engagement*. Hutchins et al. have this to say:

> Direct engagement occurs when a user experiences direct interaction with the objects in a domain. Here there is a feeling of involvement directly with a world of objects rather than of communication with an intermediary.

Hutchins et al. claim that to achieve direct engagement we must achieve the following:

- Minimize the semantic and articulatory distances, and thus close the gulfs of execution and evaluation. They also use the term "directness"; the more semantically or articulatory "direct" an interface is, the smaller the semantic/articulatory distance.
- The input and output languages of the interface should be "inter-referential", that is, the output should be able to be used as input. Compare viewing a directory listing in Windows versus viewing a directory listing using `ls` in the command line.
- The system should be responsive.
- The interface should be not be noticed as an interface.

It's worth thinking about if and how these principles apply to applications you use every day, or your favorite games.

References

[1] Edwin L. Hutchins, James D. Hollan, D. A. N. Direct manipulation interfaces. *Human-Computer Interaction 1* (1985), 311–338.