Kinds of Interfaces

Semester 2, 2009

1 Interface Categories and Styles

Basic Categories of Interaction

- key-modal
- direct manipulation
- linguistic

Notes

People can interact with computers in many different ways. The purpose of this lecture is to present a useful, if not definitive, taxonomy of interaction. At the top level, interaction is classified into three main interface categories. Each category can be further subdivided into various interaction styles.

2 Key-Modal

Key-Modal Interaction

- typically through keyboard or function keys
- typically user-interface exhibits multiple modes
  - same response from user may produce different effect, depending on mode (state) of the system
- styles
  - menu-based interaction The user interface presents the user with a display of options, and the selection of an option may generate a further set of options.
- **question-and-answer** The system presents a series of questions in text form, and the user enters the answers via a keyboard.

- **function-key interaction** The user makes a series of inputs by pushing function keys or using other special-purpose hardware, prompted along the way with displayed information.

- **(structured) voice-based interaction** The user is presented with options by recorded or synthesized voice, makes choices say with telephone keypad, may record voice response.
  
  * e.g., voice-mail retrieval

  - common for walk-up-and-use systems

  - system knows current state so can guide user to available options

---

Notes

One important category of interface is *key-modal*. The name comes from the two main features of such interfaces: the user interacts by pressing keys of some sort; and the system exhibits different *modes*, or states, so that the same key press will have different effects depending on the current mode (state) of the system.

An automatic teller machine (ATM) is a good example of a (mostly) key-modal interface. See Figure 1. The user’s input is by pressing various keys. The same key that on one screen (mode) might select “Account Balance” on another screen might say “Yes” to the question of whether to print a transaction record.

Within the category of key-modal interfaces it is possible to distinguish a number of different interaction styles, as listed above. Keep in mind, however, that any given interface may exhibit a number of different interaction styles, in different parts of the interface.

The remote control (or front panel) of a DVD player is an example of function-key interaction. There are special-purpose keys (buttons) for functions like “play” and “stop”. Several of these buttons have multiple functions, depending on the state of the system—for example, it’s common to have both “play” and “pause” on the same button. However, some parts of the interface are more menu-based.

Key-modal interfaces are particularly appropriate for inexperienced users performing simple tasks, which is why that are often the choice for “walk up and use” interfaces, like information kiosks in public places.

Because key-modal interfaces generally have simple inputs (key presses) and have states (modes), they are often modelled formally as finite-state automata.
Menu-Based Interaction

- Selection by:
  - Typing option number or letter
  - Cursor or tab keys
  - Pointing and selecting with mouse or trackball
  - Touch screen
  - Function key alongside displayed item

- Can be embedded in other systems: e.g. hypertext, GUI

- Strengths:
  - Guides users, lets them know what’s available

- Weaknesses:
  - Doesn’t well support complex actions, like commands with arguments
– Can be slow, especially for users who know what they want

Notes
A particularly common and important style of key-modal interface is menu-based interaction. Here the system presents the user with a number of options to choose from. The user chooses one, by various means of selection. The user’s choice may cause something to happen, but will often lead to another set of options.

Menus can appear in different forms, and the selection can be done in different ways.

Menus can be embedded in many other kinds of interactive systems. For example, you can create a network of webpages, so that each page presents a number of options as hyperlinks, and following link might take you to another page of options, and so on. This would be a menu-based system, even if the word “menu” is never mentioned. And while direct manipulation dominates most of today’s graphical user interfaces, menus are ubiquitous.

3 Direct Manipulation

Direct Manipulation

• manipulate individual objects

• apply actions directly to objects

• after each user action, display responds immediately to show any resultant change

• styles

  – graphical direct manipulation Information is displayed in the form of graphical objects, which the user can query and manipulate graphically with a pointing device.

  – forms fill-in The system displays a set of text fields on the screen, and the user can select individual fields and enter or modify their contents.

Notes
It would be fair to say that direct manipulation is the dominant form of interaction in most user interfaces today. Why?
And even though graphical direct manipulation, as seen in the ubiquitous computer desktop, is the most obvious example, filling in a form, like a web-form, has much of the same characteristics of direct manipulation—the user can freely act upon any of the form elements—and so belongs in this category.

4 Linguistic

Linguistic Interaction

- essentially word and language based
- styles
  - command-line interaction The user types on a keyboard a command in some command language, and the results are displayed back to the user, usually in the form of text.
  - spoken or text-based natural language User types or speaks queries and provides information in some natural language, and the system responds in the same.

Notes

The linguistic category covers interfaces that make significant use of language in their interaction, whether it’s a computer language in command-line interaction, or human language.

Just making use of language does not qualify an interface as being linguistic. For example, a question-and-answer interface asks its questions in natural language, but the user’s responses are just simple words and numbers, maybe data, like addresses, to go into a database. Only trivial parsing and interpretation of input needs to be done. Or in voice-based interaction, like voice mail, while the output from the system is in speech (usually pre-recorded, sometimes synthesized, or at least assembled, on the fly), the user’s responses are just key presses. Even when the user speaks, say to record an answering message, the user’s speech is not interpreted by the system (except perhaps trivially to decide whether it’s silence or not), just stored.

Some voice-interaction systems allow the user to respond by speaking simple isolated words, like “yes” or “no” or the numeric digits. Even though the user’s speech is being interpreted, and the user isn’t pressing keys, the nature of such a interface is much more like a key-modal interface than a linguistic interface, with the simple spoken words being a slot-in replacement for the key presses.
To qualify as a linguistic interface, there has to be some linguistic richness to the user’s input, and some non-trivial parsing and interpretation of the input.

Why natural language processing is hard

- Fully general speech recognition, natural-language understanding and generation is a very hard AI problem.
- *Time flies like an arrow.*
- *Morty shot Ferdie. He died.*
- *Morty shot Ferdie. He didn’t die.*
- *Morty shot Ferdie. He felt sorry about it.*
- Can be feasible in limited domains, e.g., airline reservations.

Notes

Why aren’t there more natural-language linguistic interfaces in use? Simply because interpreting (and generating) natural language (whether written or spoken) is a very hard AI problem, which researchers have been working on since the 1960s, with only modest success (including the Language Technology Group in this Department).

To give you some sense of why natural-language processing (NLP) is so hard, consider the sentences above. Take the first one. Probably the interpretation that first comes to most peoples mind is “Time [subject] flies [verb] like an arrow [adverbial phrase]”. To paraphrase, “Time seems to pass as fast as an arrow flies.” But that is not the only interpretation possible. It could be “Time [noun phrase] flies [subject] like [verb] an arrow [object]”. That is, there are a special kind of flies, “time flies”, who have a liking for an arrow, just as we might say “Fruit flies like a banana”. Or perhaps we’re at a race, where both flies and arrows are competing, and the sentence is a command to “Time [imperative verb] flies [object] like an arrow [adverbial phrase], just as we might say “Time racing horses like a racing car”. Of course, while the other interpretations are grammatically correct and even (with a bit of imagination) semantically possible, they are practically speaking much less plausible.

In English (as in most other languages) we have **pronouns**, like “he”, “she”, “it”, which (like variables in a programming language) stand for other things. In linguistics this phenomenon is called **anaphora**. In the first
sentence of the sentence pairs above, two individuals, Morty and Ferdie, are referred to by name. The second sentence of each pair uses the pronoun “he”. But to which individual does “he” refer? Most people would have no difficulty in inferring that in the first two Morty and Ferdie utterances, “he” refers to Ferdie, while in the third it refers to Morty. But what are the (unconscious) reasoning processes that lead to that conclusion and how could we program machines to do the same? Clearly it can’t be simply the rule that the pronoun refers to the nearest noun, since in the sentences above, it isn’t always the nearest. Maybe the interpretation is the one that gives a sentence that is mostly likely true. That is, in the first Morty and Ferdie utterance, Ferdie has been shot, and probably killed, so the second sentence of the pair is mostly likely true with “he” referring to Ferdie. But that rule also fails in the next utterance, in which the second sentence says the opposite. What’s going on is some more subtle reasoning. Having heard that Ferdie has been shot, we the listeners (and the speaker) are aware that whether he died or not is at issue. So that in a following sentence that answers that question (whether in the positive or the negative), “he” probably refers to Ferdie. There’s a similar thing going on in the last sentence. Almost certainly, Ferdie (if still alive) was sorry about being shot, but we take “he” here as referring to Morty. Morty has just performed some significant action, so how he feels about its consequences is a topic of interest.

Sorting all this out requires some subtle reasoning about the world and how people behave and feel, and about the intentions and capabilities of the speaker and listener.

The point of all this is not the particular difficulties of NLP, for example parsing and anaphora, but just that NLP is difficult and that we won’t have useful, full natural-language capabilities in user interfaces for quite a few years to come.

Notwithstanding that, it is, with current technology, reasonably feasible to have a natural language interface in restricted domains, like airline flight reservations. Here there’s a quite limited vocabulary of words (and the concepts behind them) and only a limited range of question and sentence constructions used.

Comments

- The classification into different styles provides a useful framework and insights, but don’t take it too rigidly.

- Hybrid systems are possible, and done well can take advantage of the strengths of multiple styles
- e.g. ddd, bash

- A particular system may partake of a number of styles.
- Be aware of each style’s strengths and weaknesses in the context of the users and their situation

Notes

This classification scheme provides a useful framework for thinking and talking about interfaces. However, there will be existing interfaces that don’t quite fit into this classification, and doubtless as interfaces evolve, new forms of interface will be developed.

The graphical debugger, ddd (which you probably used in 252), is one example of a hybrid interface. It has a graphical direct-manipulation component: for example, you can click on a source-code line to set a break-point. It also has a command-line interface to gdb. This gives you the best of both worlds: the immediacy of a direct-manipulation interface, with the power of a command-line interface. Notice in this case the two interfaces overlap in their functionality. What is provided through the direct-manipulation interface is a useful subset of the command-line gdb functionality. In general, providing a graphical user-interface wrapper around a command-line tool is a very useful technique, much used in the Unix world (MacOSX, GNU/Linux, Solaris, the various BSD derivatives, for example).

The bash shell you might think of as mainly a linguistic, command-line interface. But some aspects of it, like immediate editing of command lines, and moving through the history with arrow keys, have the flavor of direct manipulation. Some aspects of the bash interface, like the various kinds of completion (filenames, commands, hostnames), are harder to categorize.

Many other user-interfaces make use of different interaction categories and styles in different parts of their interfaces. Such interfaces are sometimes called “multimodal”—although that’s using “modal” in a slightly different sense than in “key-modal”. Here it’s being used to say that the interface has multiple modes (ways) of operation rather than just different states.

The main idea to take away from all this is that there are many kinds of interfaces possible (more than mentioned here). Your job is to select the kinds of interface that are going to work best for your users in their situation for the tasks they have to perform.

Basic Categories of Interaction Revisited

- key-modal
• direct manipulation
• linguistic

5 Example Interfaces

Peep
• “Network Auralizer”
• event sounds
• state sounds

Bumptop
• 3D desktop with physics

Haptics
• touch/force feedback

Looking Glass
• desktop extended with 3D
• Sun, Java, Java3D, GPL
• Hideya Kawahara

Croquet
• 3D multiuser collaborative interactive environments
• Squeak Smalltalk, opensource
• Alan Kay
• educational motivations
WiiMote Fun

- head tracking
- multi-touch screen
- ...
- Johnny Chung Lee, CMU