Ch6 Interaction Design and Styles

Human Computer Interaction Class

1. Introduction

- To create a physical UI design, you need to understand how users interact with computer system.
- We consider a model of the interaction process known as the **human action cycle**.

2. The Human Action Cycle

- The **human action cycle** (Norman, 1988) is a *psychological model* that describes the steps users take when they interact with computer systems.
- The cycle shows the way users perform actions and tasks to achieve their goals.

2.1 The details of the Human Action Cycle

- The flow of the activities in the human action cycle is illustrated in Figure 1
  - From a goal
  - Creates and executes action that move toward that goal
  - Perceives and interprets the outcome of executing the actions to see whether the goal will be achieved as anticipated
  - Recognized that if the goal cannot be achieved, it may have to be reformulated and the cycle repeated
- Thus, the human action cycle involve both cognitive and physical activities.

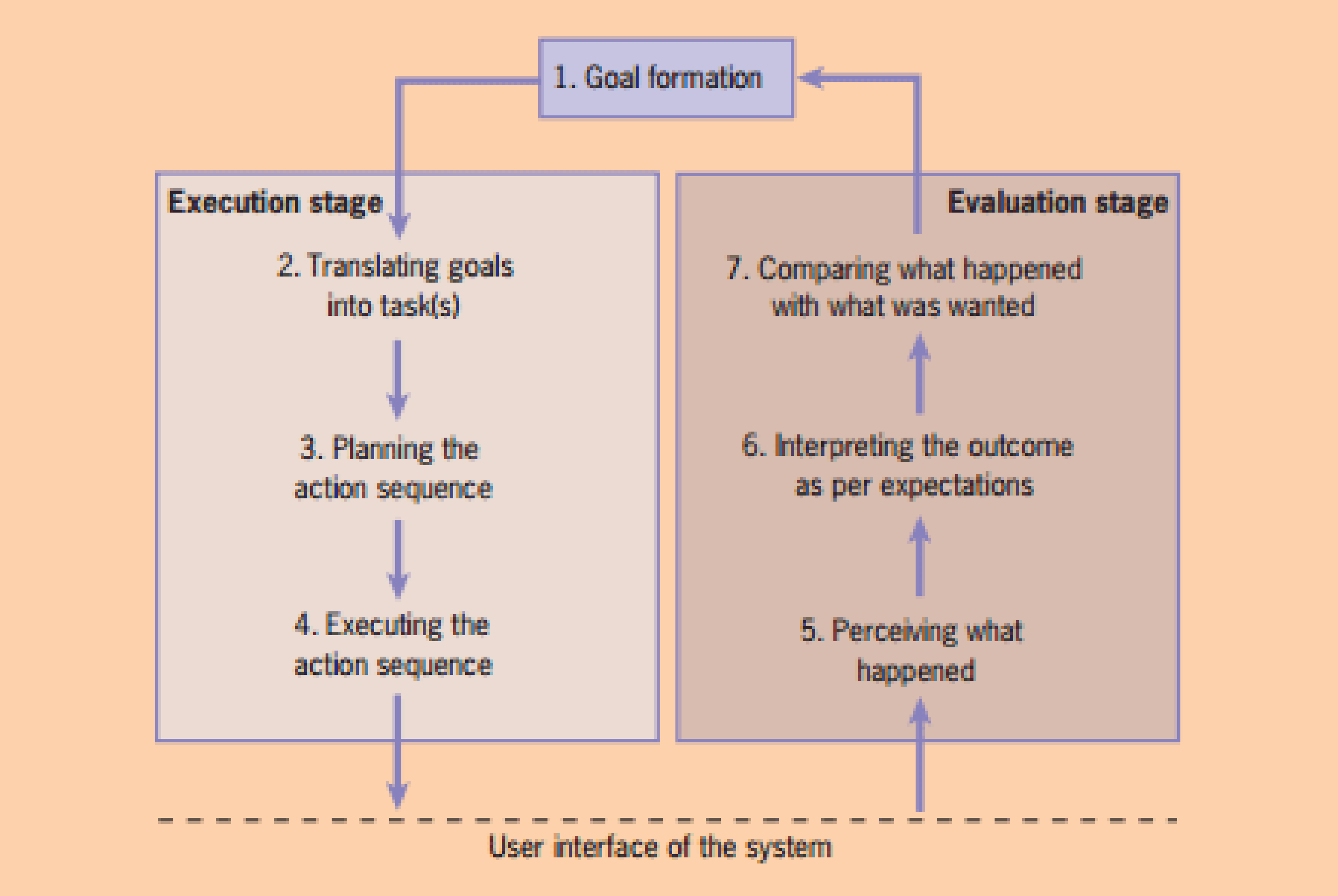
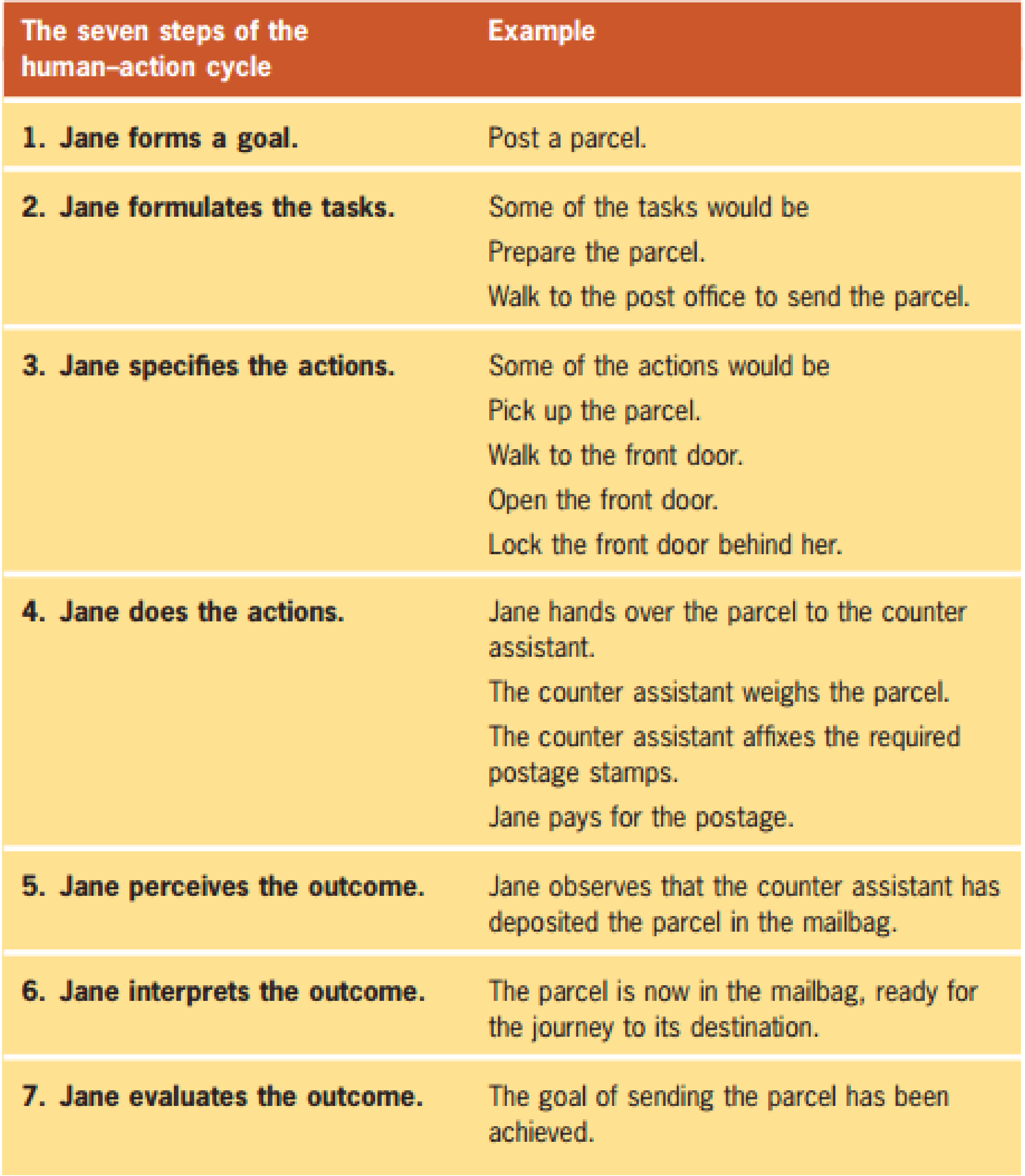

1. Goal formation
Execution stage
2. Translating goals into task(s)
3. Planning the action sequence
4. Executing the action sequence
Evaluation stage
7. Comparing what happened with what was wanted
6. Interpreting the outcome as per expectations
5. Perceiving what happened
User interface of the system

Figure 1 The human action cycle

Three main stages in the cycle

- Goal formation.
  - This step **one**, which is a cognitive activity
  - Users need to be able to form appropriate goal(s) to use the UI effectively
- Execution stage
  - Step two, there, four
  - During the execution stage, the users perform both cognitive and physical activities.
  - Cognitive activities include translating the goals into tasks and planning the action sequences whereas physical activities involves executing the sequence of actions
- Evaluation stage
  - Steps **five, six, seven**. These are cognitive activities involving checking what has happened and comparing it with the desired outcome (the goal(s) that were formed in step one)

Table 1 The human action cycle – an example



The seven steps of the human–action cycle	Example
1. Jane forms a goal.	Post a parcel.
2. Jane formulates the tasks.	Some of the tasks would be Prepare the parcel. Walk to the post office to send the parcel.
3. Jane specifies the actions.	Some of the actions would be Pick up the parcel. Walk to the front door. Open the front door. Lock the front door behind her.
4. Jane does the actions.	Jane hands over the parcel to the counter assistant. The counter assistant weighs the parcel. The counter assistant affixes the required postage stamps. Jane pays for the postage.
5. Jane perceives the outcome.	Jane observes that the counter assistant has deposited the parcel in the mailbag.
6. Jane interprets the outcome.	The parcel is now in the mailbag, ready for the journey to its destination.
7. Jane evaluates the outcome.	The goal of sending the parcel has been achieved.

- Example of each of the seven stages of the human action cycle for sending a parcel at a post office

2.2 Using the Human Action Cycle to influence the Design Process

- One way to critically evaluate this aspect of a UI prototype is to walk through the prototype, checking to see if it satisfies the requirements of a use scenario and asking questions based on the human action cycle.
- Answering these questions can provide you with various types of information:
  - You may be able to predict difficulties that the users may face with the design and suggest modifications.
  - You may be able to suggest suitable changes in the users' environment and the system's technology.
  - You may be able to suggest necessary skills for the users when they work with the UI, or identify training needs.

3. Communicating the Designer’s Understanding of the System

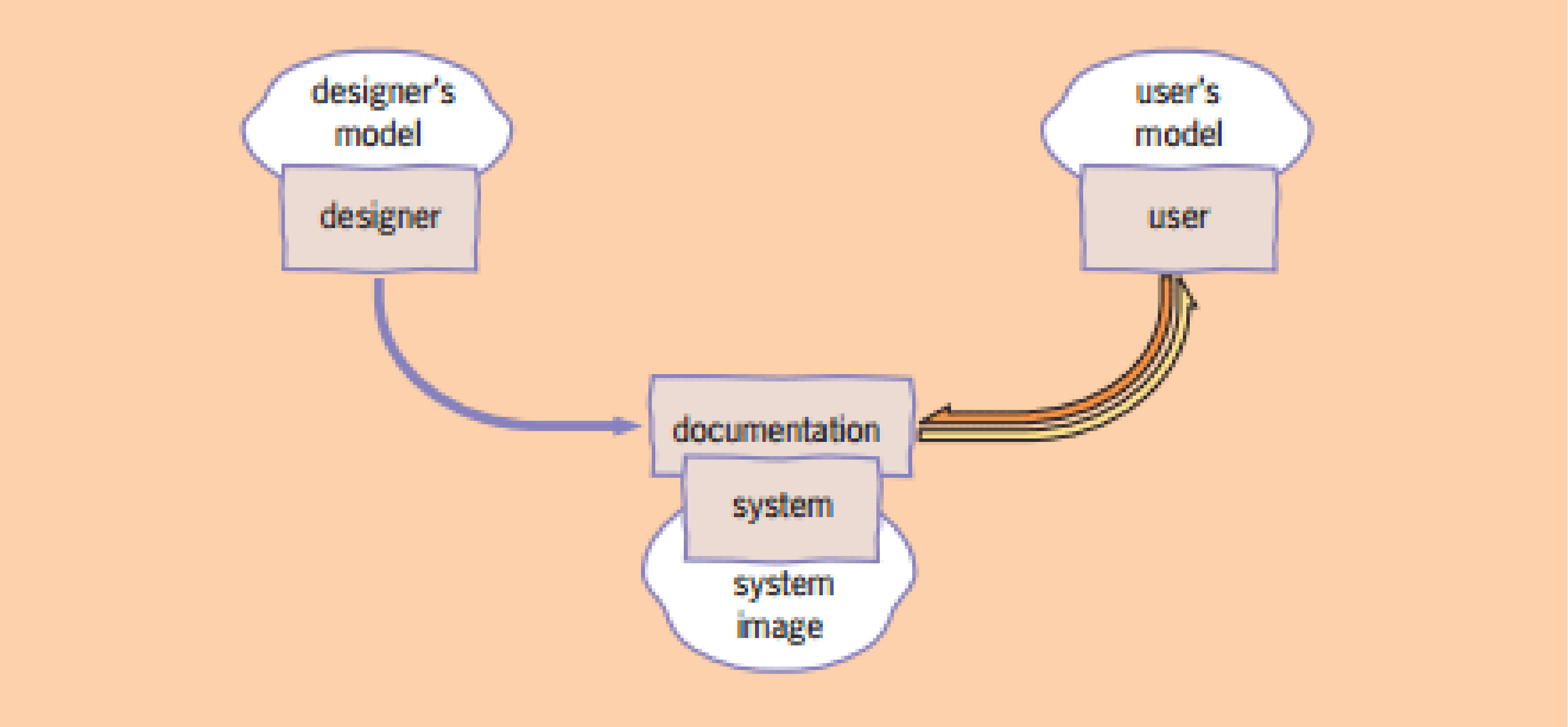


Figure 2 The designer’s model, the user’s model, and the system image (From Norman and Drapr, 1986)

- *The designer’s model*.
  - This is the designer’s understanding of a system.
  - It includes both the structural and functional aspects of the system — both how it works and how to operate it.
  - The designer’s model is usually complete and correct.

- *The user’s model*.
  - This is the user’s understanding of the system.
  - Typically it contains mainly functional information, possibly incomplete and incorrect.

- *The system image*.
  - This includes the UI, supporting documentation, training, and so on.
  - The purpose of the system image is to communicate enough of the designer's model to the users so that they can create an accurate mental model of the system.
  - The design of the system image should also take into account the users' existing knowledge of the area.

3.1 Designer’s Model

The **designer’s model** is an explicit and consciously developed model, derived from:

- The structure and organization of the UI, as represented by the content diagram
- An understanding of the domain and the system’s purpose and functionality
- An understanding of users’ requirements, their characteristics, tasks, and expectations of the new system
- The system’s technology and the environment
- Any hardware platform or implementation constraints or any trade-offs

3.2 System image

- The **system image** is the means by which the functionality and state of a system is presented to the user, and this can be through a variety of channels: the UI, training, documentation including instruction manuals, and online help.
- The users acquire their knowledge of the system from the system image. The most important part of the system image is the UI since, through interaction with it, users build up their own mental model of the system and how it works.
- Users do not always read instruction documents, use the help function, or receive training. Therefore, the maximum burden of influencing the user’s mental model is on the UI.

3.3 How the User Interface Enables the User to Develop an Accurate Mental Model

- As the figure 2 illustrates, the UI needs to take the following points into account:

1. The *existing user model*. The users will have expectations of the UI, and if these are not met they may experience difficulties.

2. The *designer's model*. The UI needs to *communicate effectively* the relevant parts of the designer's model, in particular *the functional aspects* — how the users need to operate the UI in order to achieve their goals.

To take into account the **user's model**, you need to consider a number of design issues:

- *Matching the users' expectations*. As a designer, if you are aware of the users' expectations, then you will be able to create a UI design that is more immediately understandable and easier to learn. It may not always be possible for you to exactly match the UI to the users' expectations, either because of technological constraints or because the system requirements are different, but it is always worth trying.

- *Shaping the user's model*. A user may find a system daunting if it includes a lot of functions. One way to overcome this is to organize the UI so that the familiar functions are easy to pick out. This means the user can gain confidence in using these functions before moving on to the unfamiliar functions. For example, on a cell phone it should be immediately apparent how to dial a number, so users can try this before experimenting with more advanced features such as changing the ringing tone.

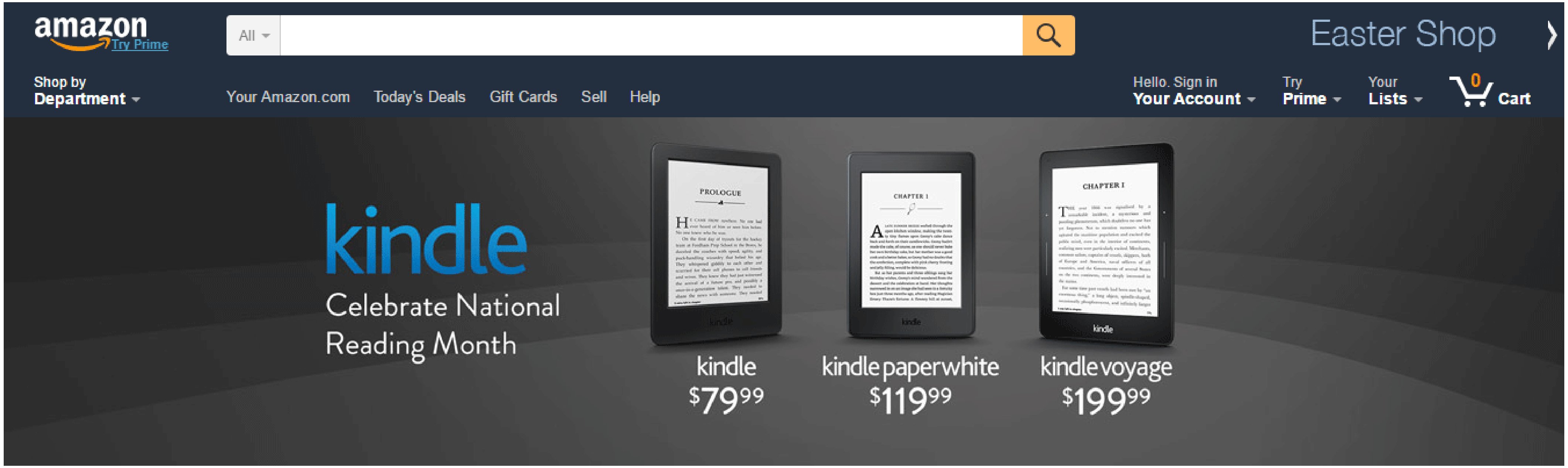
- *Flexibility*. Typically, the user's model is not static but grows through interaction with the UI. A user slowly develops a picture of what is in the system, what it is capable of doing, and how it responds to various actions. Since user's models change with time, the designer should create a flexible UI — for example, providing different ways of accomplishing the same tasks to suit both novices and experts.

In conclusion, the designer's aim is to create a UI that effectively communicates his or her model while reflecting an awareness of the user's existing mental model.

4. Using Metaphor to Develop Accurate Mental Models

- In the previous section, we emphasized the importance of designing a UI that matches the users’ mental models or supports the adaptation of the users’ mental models toward the one desired.
- In this section, we show you how metaphors can help users develop accurate mental models.

4.1 The Benefits of Metaphor



Metaphor is a figure of speech in which a word or phrase denoting one kind of object or action is used in place of another to suggest a likeness or analogy between them.

- Metaphors are often used to unify UIs. Consider the example of e-shopping.
  - The experience of shopping online — selecting goods, proceeding to the checkout, and entering details of your credit card and delivery address — is very different from the shopping experience in a physical store.
  - To help bridge this gap, many e-shopping (or e-commerce) sites present a shopping metaphor by using terminology, icons, and graphics that are analogous to shopping in the real world.

4.2 Problem with Metaphor

- As you can see, metaphors can be extremely powerful, and it would be easy to conclude that they are the solution to all UI problems. Unfortunately, it is not that simple. The following sections describe two important classes of problem that can arise when using metaphors.
  - ➢Metaphors that do not match the users' experience of the world.
  - ➢Metaphors that relate to objects or concepts that are outside the user's experience

4.3 Choosing a Suitable Metaphor or Set of Metaphor

- As with other activities in the design of UIs, the approach to the choice of metaphors should be iterative and user centered.
- You may come up with several to start with, then retain some and reject others.
- During requirements gathering, the analysis of users, tasks, and environment may provide ideas for metaphors that could be used in the UI design.
- In particular, the task scenarios and use cases are often a good source of metaphors.
- Conversations during requirements gathering may also give you an insight into the metaphors that users tend to employ.

Interaction Styles

1. Introduction

- To create a physical UI design, you need to understand how users interact with computer systems.

2. Interaction Styles

- There are different ways a user can communicate with a computer system and a computer system can communicate with a user. These are called **interaction styles**.
- An **interaction style** is a *collection of user interface controls and their associated behavior.*
- The interaction style provides both the *look (appearance)* and *feel (behavior) of the user interface components*, indicating the way a user will communicate with the system.

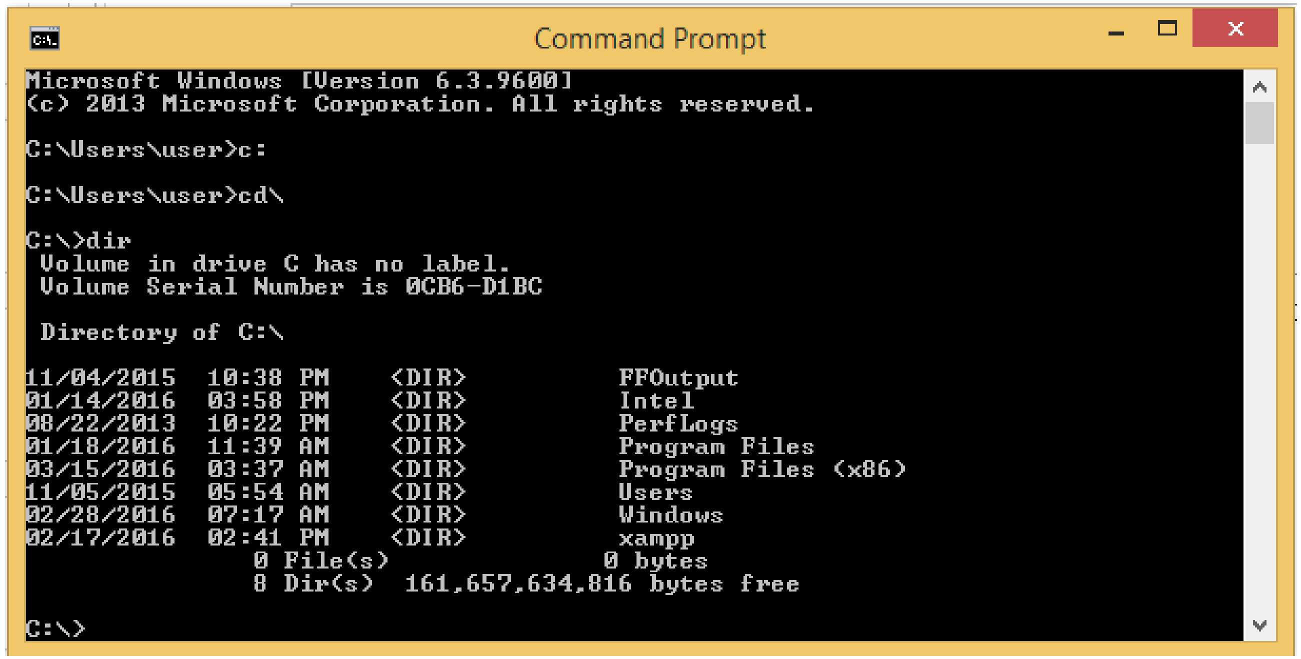
Five interaction styles

- Command line
- Menu selection
- Form-fill
- Direct manipulation
- Anthropomorphic

The choice of interaction style will also depend on the choice of interaction devices (both input and output devices, and vice versa

2.1 Command line

- The command line interface was the first interactive dialog style to be commonly used.
- It provides a means of directly instructing the system, using function keys on a keyboard, single characters, abbreviations, or whole-word commands.
- As the user types characters onto the screen, they appear as a line across the screen, hence the term **“command line”**



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\user>c:

C:\Users\user>cd\

C:\>dir
 Volume in drive C has no label.
 Volume Serial Number is 0CB6-D1BC

 Directory of C:\

11/04/2015  10:38 PM    <DIR>          FFOutput
01/14/2016  03:58 PM    <DIR>          Intel
08/22/2013  10:22 PM    <DIR>          PerfLogs
01/18/2016  11:39 AM    <DIR>          Program Files
03/15/2016  03:37 AM    <DIR>          Program Files (x86)
11/05/2015  05:54 AM    <DIR>          Users
02/28/2016  07:17 AM    <DIR>          Windows
02/17/2016  02:41 PM    <DIR>          xampp
               0 File(s)              0 bytes
               8 Dir(s)  161,657,634,816 bytes free

C:\>
```

Figure 1 Command line

2.1 Command line

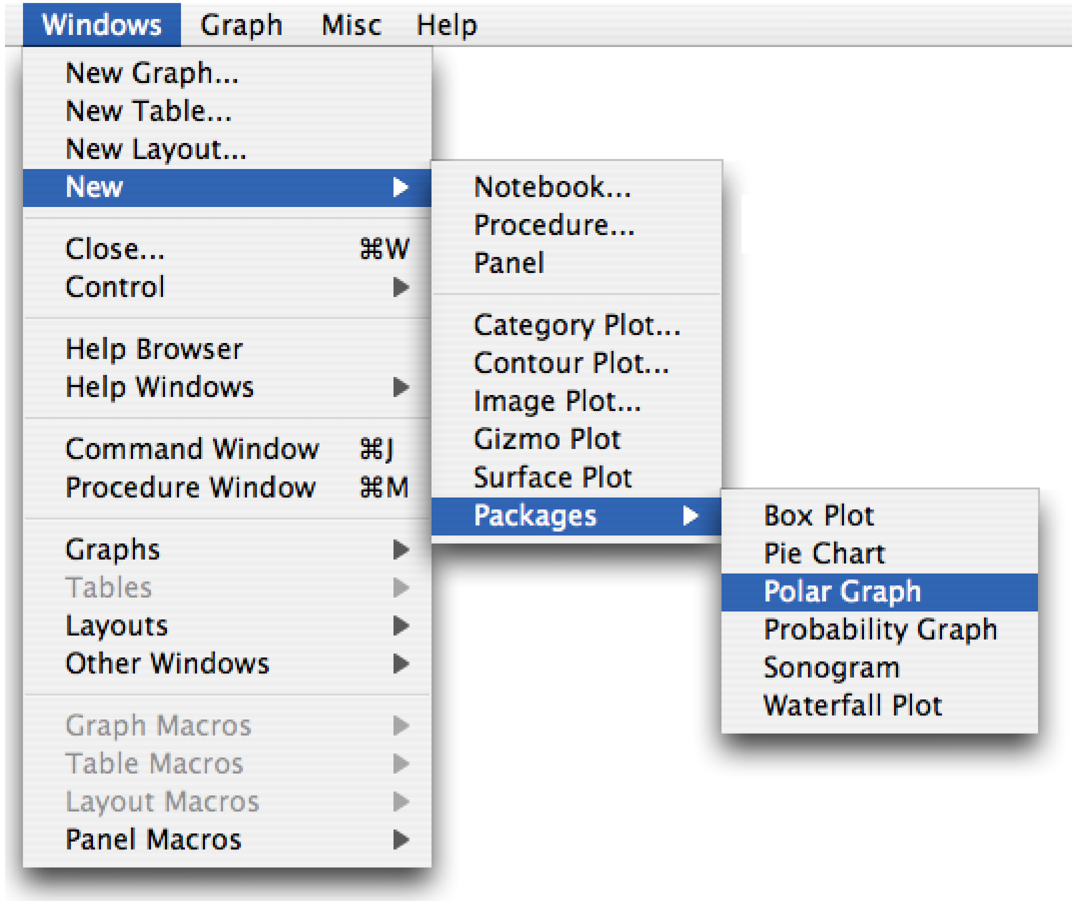
- **Command line** interfaces are **powerful** because they offer access to system functionality.
- They are also **flexible**:
  - The command line has a number of options or parameters that will vary its behavior in some way
  - It can be applied to many objects at once
  - Making it useful for repetitive tasks

- One limitation of command line UI is the commands need to remembered.

Command line interfaces are better for **expert users** than for novices.
For expert users, command languages provide a sense of being in control.

2.2 Menu Selection

- **Menu selection** avoids many of the problem associated with command line interface.
- A **menu** is a set of options form which the user must choose.
- The interface displays the options as menu items or icons and the user indicates a choice with a pointing device or keystroke, receiving feedback that indicates which option he or she has choosen, and the outcome of the command being executed.

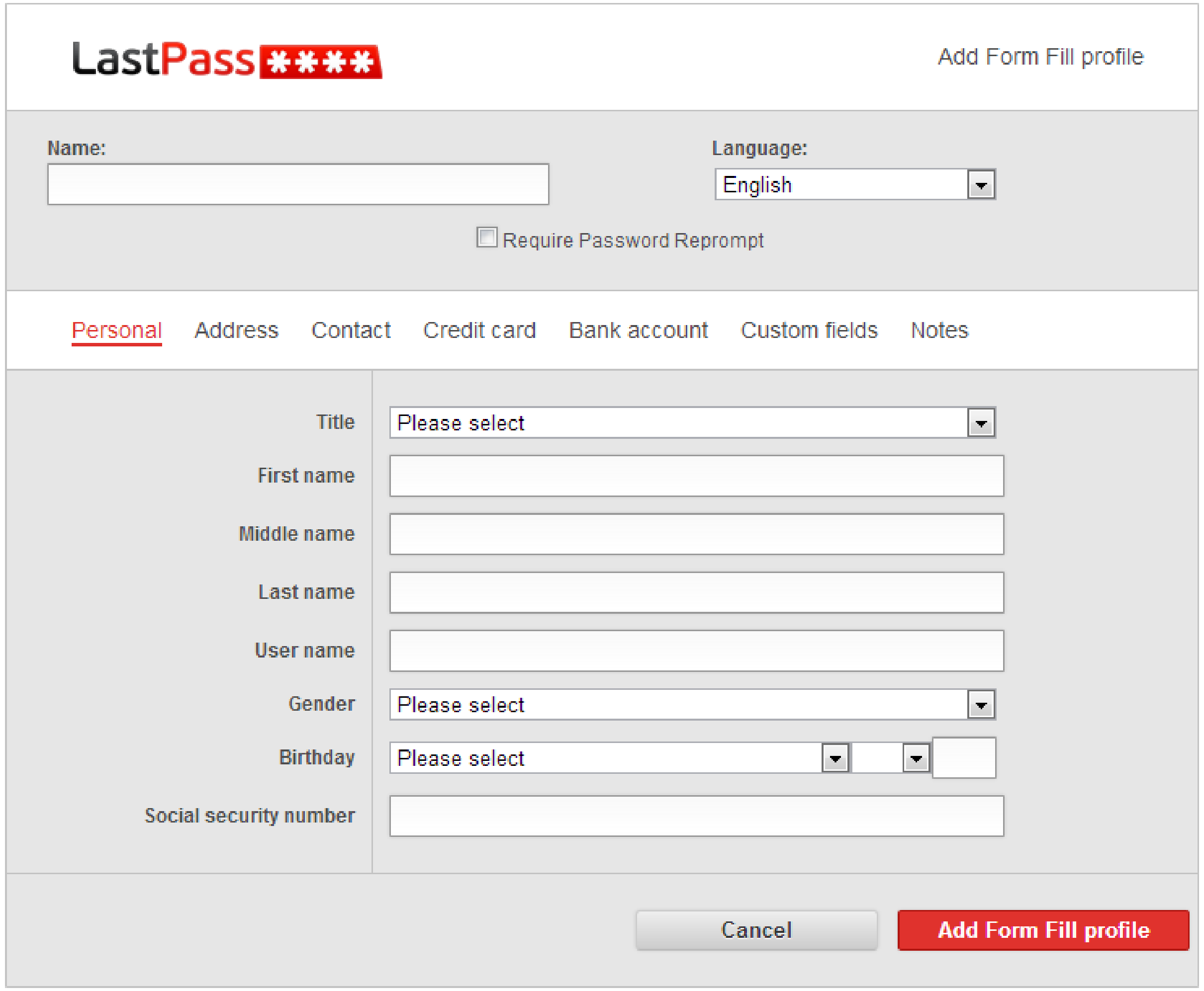


Guidelines for designing menu selection interface

- Use task semantics (flow of tasks and interaction) to organize menu
- Give menu items titles that reflect their functions
- Group items meaningfully
- Avoid lengthy menus
- Order menu items meaningfully
- Use short names for menu items
- Use consistent grammar, layout, and terminology
- Provide online context-sensitive help
- Consider the screen’s size when deciding the number of menu items

2.3 Form-Fill

- If your interface has to *gather a lot of information from the user*, then it often helps if you provide a form to fill in.



Name:

Language: English

☐ Require Password Reprompt

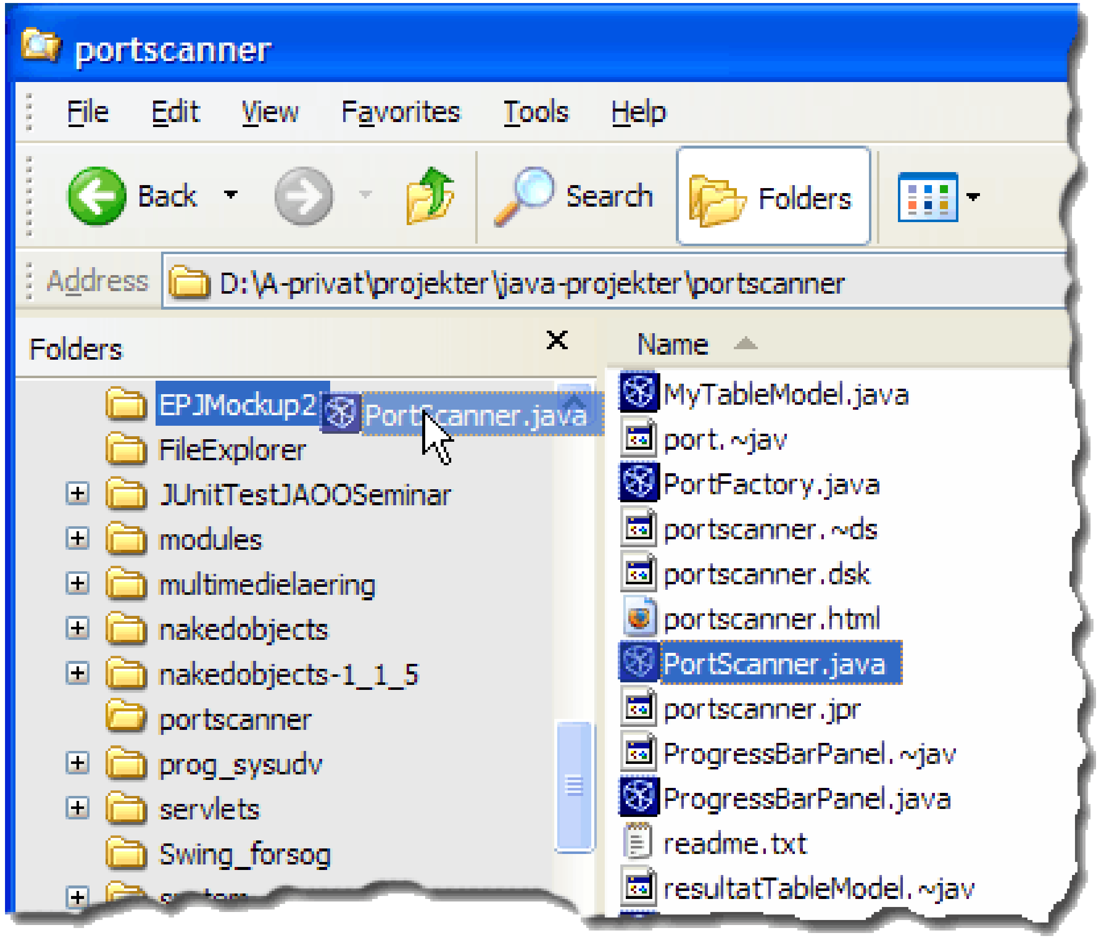
Title	Please select
First name	
Middle name	
Last name	
User name	
Gender	Please select
Birthday	Please select
Social security number	

Guidelines for designing form-fill interface

- Give meaningful titles or labels to the fields
- Give familiar field labels (use the user’s language)
- Provide comprehensible instructions
- Incorporate a logical grouping and sequencing of the fields
- Present a visually appealing layout for the form
- Use consistent terminology and abbreviations
- Provide white space and boundaries
- Restrict the characters that can be entered or provide default value
- Allow for convenient cursor movement
- Provide error correction for individual characters and entire fields
- Provide error message for unacceptable values and error indicators as soon as possible
- Indicate required fields
- Provide explanatory message for the fields

2.4 Direct Manipulation

- **Direct manipulation** (DM) interface allow users to interact directly with the UI objects.
- For example, dragging a file from one folder and dropping it into another.
- This actually achieved by using a continuous input device, such as a mouse, pen or joystick.



Address D:\A-privat\projekter\java-projekter\portscanner

Well-designed DM interface typically have a number of characteristics:

- There is a visible and continuous representation of the task objects and their actions, Consequently, there is little syntax to remember.
- The task objects are manipulated by physical actions, such as clicking or dragging, rather than by entering complex syntax.
- Operations are rapid, incremental, and reversible; their effects on task objects are immediately visible. Thus, users can instantly see if their action are furthering their goal and if they are not, can simply change the direction of their activity.
- While interacting with DM interfaces, users fell as if they are interacting with the domain rather than with the interface.
- Novice can learn the basic functionality quickly.

Guideline for designing direct manipulation interface

- Choose any metaphor carefully to ensure that they promote the rapid development of an accurate mental model.
- Create visual representations of the user's tasks.
- Provide rapid, incremental, and reversible actions.
- Replace typing with pointing/selecting.
- Present a visually appealing layout.
- Make the results of actions immediately visible – provide quickly visual or auditory feedback.

Anthropomorphic

- **Anthropomorphic interfaces** *aim to interact with users in same way that human interact with each other*. Natural language interfaces and interfaces that recognize *gestures*, *facial expressions*, or *eye movements* all belong to this category.
- The design and development of anthropomorphic interfaces requires an understanding not only of hardware and software, but also of how humans communicate with each other through language, gestures facial expression, and eye contact.

2.6 Blending Interaction Styles

- Most modern UIs blend more than one interaction style.
- Blending can occur in a variety of ways. For example,
  - commands can lead the user to form-fill where data entry is required, or menu can be used to control an otherwise DM environment when a suitable visualization of actions cannot be found.
  - List boxes can supplement form-fill for less knowledgeable user.

Table 6.1 The advantages and disadvantage of the five primary interaction styles

Interaction Style	Advantages	Disadvantages
Command line	• Flexible. • Appeals to expert users. • Supports creation of user-defined "scripts" or macros. • Is suitable for interacting with networked computers even with low bandwidth.	• Retention of commands is generally very poor. • Learnability of commands is very poor. • Error rates are high. • Error messages and assistance are hard to provide because of the diversity of possibilities plus the complexity of mapping from tasks to interface concepts and syntax. • Not suitable for non-expert users.
Form-fill	• Simplifies data entry. • Shortens learning in that the fields are predefined and need only be 'recognized'. • *Guides* the user via the predefined rules.	• Consumes screen space. • Usually sets the scene for rigid formalization of the business processes.

Interaction style	Advantage	Disadvantage
Menu selection	• Ideal for novice or intermittent users. • Can appeal to expert users if display and selection mechanisms are rapid and if appropriate "shortcuts" are implemented. • Affords exploration (users can "look around" in the menus for the appropriate command, unlike having to remember the name of a command *and* its spelling when using command language.) • Structures decision making. • Allows easy support of error handling as the user's input does not have to be parsed (as with command language).	• Too many menus may lead to information overload or complexity of discouraging proportions. • May be slow for frequent users. • May not be suited for small graphic displays.

Interaction style	Advantage	Disadvantage
Direct manipulation	- Visually presents task concepts. - Easy to learn. - Errors can be avoided more easily. - Encourages exploration. - High subjective satisfaction. - Recognition memory (as opposed to cued or free recall memory)	- May be more difficult to programed. - Not suitable for small graphic displays. - Spatial and visual representation is not always preferable. - Metaphors can be misleading since the "the essence of metaphor is understanding and experiencing one kind of thing in terms of another" (Lakoff and Johnson 1983: p. 5), which, by definition, makes a metaphor different from what it represents or points to. - Compact notations may better suit expert users.
Anthropomorphic Interface	- Can relieve the burden of learning the syntax for the interaction with the system.	- Can be unpredictable. - Difficult to implement.

Table 6.2 The Relation between task characteristics and interaction style

Task characteristics	Interaction style
A large amount of data entry is required	Form-fill or command line
A paper form exists that must be computerized	Form-fill
Familiar notation exists	Command line
A natural visual representation exists, or a modest number of task objects and actions can represent the task domain	Direct manipulation
Multiple decisions or selections from a large range of unfamiliar options are required	Menu selection or direct manipulation
Exploration is anticipated	Direct manipulation