Software Engineering I

Dr. Elham Mahmoudzadeh

Isfahan University of Technology

mahmoudzadeh@iut.ac.ir

2021

Chapter 10
Human–Computer Interaction Layer Design(II)

Steps(I)

1. Preparing proposal
2. Requirements determination
   - User story
3. Abstract Business Process Modelling
4. Analysis
   - Functional Modelling
   - Structural Modelling
   - Behavioral Modelling

Steps(II)

5. Design
   - Optimization
   - Database Management
   - User Interface
   - Physical Architecture

Navigation Design

- The navigation component of the interface enables the user to enter commands to navigate through the system and perform actions to enter and review information it contains.
- It also presents messages to the user about the success or failure of his or her actions.
- The goal of the navigation system is to make the system as simple as possible to use.
- A good navigation component is one the user never really notices. It simply functions the way the user expects, and thus the user gives it little thought.

Basic Principles of navigation design

- Assume that users have not read the manual, have not attended training, and do not have external help readily at hand.
- All controls should be clear and understandable and placed in an intuitive location on the screen.
- Ideally, the controls should anticipate what the user will do and simplify his or her efforts.
  - For example, many setup programs are designed so that for a typical installation, the user can simply keep pressing the Next button.

1- Prevent Mistakes

- Prevent the user from making mistakes. A mistake costs time and causes frustration. Worse still, a series of mistakes can cause the user to discard the system.
- Mistakes can be reduced by labeling commands and actions appropriately and by limiting choices.
  - Too many choices can confuse the user, particularly when the choices are similar and hard to describe in the short space available on the screen.
- Never display a command that cannot be used. Many Windows applications gray out commands that cannot be used; they are displayed on pull-down menus in a very light colored font, but they cannot be selected. This shows that they are available but cannot be used in the current context. It also keeps all menu items in the same place.
- When the user is about to perform a critical function that is difficult or impossible to undo, it is important to confirm the action with the user (and make sure the selection was not made by mistake).

2- **Simplify Recovery from Mistakes**

- No matter what the system designer does, users will make mistakes.
- The system should make it as easy as possible to correct these errors.

3- **Use Consistent Grammar Order**

- One of the most fundamental decisions is the *grammar order.*
- Most commands require the user to specify an object, and the action to be performed on that object.
- The interface can require the user to first choose the object and then the action (an *object–action order*) or first choose the action and then the object (an *action–object order*).
- The grammar order should be consistent throughout the system, both at the data element level and at the overall menu level.

Types of Navigation Controls

- There are two traditional hardware devices that can be used to control the user interface.
  - Keyboard
  - Pointing device such as a mouse, trackball, or touch screen.
- Today, depending on the hardware being used, voice recognition systems can also be used to control the user interface.
- There are three basic software approaches for defining user commands.
  - Languages,
  - Menus,
  - Direct manipulation.

Languages

- With a *command language*, the user enters commands using a special language developed for the computer system (e.g., UNIX and SQL both use command languages).
- Command languages sometimes provide greater flexibility than other approaches because the user can combine language elements in ways not predetermined by developers.
- However, they put a greater burden on users because users must learn syntax and type commands rather than select from a well-defined, limited number of choices.

Menus(I)

- The most common type of navigation system today is the *menu.* A menu presents a user with a list of choices, each of which can be selected.
- Menus are easier to learn than languages because a limited number of available commands are presented to the user in an organized fashion.
- Menus need to be designed with care because the submenus behind a main menu are hidden from users until they click on the menu item.
- It is better to make menus broad and shallow (i.e., each menu containing many items with only one or two layers of menus) rather than narrow and deep (i.e., each menu containing only a few items, but each leading to three or more layers of menus).
- A broad and shallow menu presents the user with the most information initially so that he or she can see many options and requires only a few mouse clicks or keystrokes to perform an action.
- A narrow and deep menu makes users hunt for items hidden behind menu items and requires many more clicks or keystrokes to perform an action.

Menus(II)

- Research suggests that in an ideal world, any one menu should contain no more than eight items, and it should take no more than two mouse clicks or keystrokes from any menu to perform an action (or three from the main menu that starts a system). However, analysts sometimes must break this guideline in the design of complex systems by grouping menu items separated by a horizontal line.
- Often menu items have *hot keys* that enable experienced users to quickly invoke a command with keystrokes in lieu of a menu choice (e.g., on a Windows machine, across many applications, Ctrl-F tends to invoke the Find command; on a Mac, you use Command-F instead).
- Menus should put together like items so that the user can intuitively guess what each menu contains. Most designers recommend grouping menu items by interface objects (e.g., customers, purchase orders, inventory) rather than by interface actions (e.g., new, update, format), so that all actions pertaining to one object are in one menu, all actions for another object are in a different menu, and so on. However, this is highly dependent on the specific interface.

Direct Manipulation

- User enters commands by working directly with interface objects.
  - For example, users can change the size of objects in Microsoft PowerPoint by clicking on them and moving their sides,
  - Users can move files in Windows Explorer by dragging the filenames from one folder to another.
- Direct manipulation can be simple, but it suffers from two problems. First, users familiar with language- or menu-based interfaces don’t always expect it. Second, not all commands are intuitive.

Messages(I)

- Messages are the way the system responds to a user and informs him or her of the status of the interaction.
- Messages should be clear, concise, and complete.
- All messages should be grammatically correct and free of jargon and abbreviations.
- Avoid negatives because they can be confusing (e.g., replace Are you sure you do not want to continue? with Do you want to quit?).
- Avoid humor.
- Messages should require the user to acknowledge them, rather than being displayed for a few seconds and then disappearing. The exceptions are messages that inform the user of delays in processing, which should disappear once the delay has passed.
- In general, messages are text, but sometimes, standard icons are used.

Messages(II)

- Messages (and especially error messages) should always explain the problem in polite, succinct terms (e.g., what the user did incorrectly) and explain corrective action as clearly and as explicitly as possible so that the user knows exactly what needs to be done.
- In the case of complicated errors, the error message should display what the user entered, suggest probable causes for the error, and propose possible user responses. When in doubt, provide either more information than the user needs or the ability to get additional information.
- Error messages should provide a message number. Message numbers are not intended for users, but their presence makes it simpler for help desks and customer support lines to identify problems and help users because many messages use similar wording.

Types of messages

Type of Messages	When to Use	Notes
Error message Informs the user that he or she has attempted to do something to which the system cannot respond	When the user does something that is not permitted or not possible	Always explain the reason and suggest corrective action. Traditionally, error messages have been accompanied by a beep, but many applications now omit it or permit users to remove it.
Confirmation message Asks users to confirm that they really want to perform the action they have selected	When user selects a potentially dangerous choice, such as deleting a file	Always explain the cause and suggest possible action. Often include several choices other than OK and cancel.
Acknowledgment message Informs the user that the system has accomplished what it was asked to do	Seldom or never. Users quickly become annoyed with all the unnecessary mouse clicks	Acknowledgment messages are typically included because novice users often like to be reassured that an action has taken place. The best approach is to provide acknowledgment information without a separate message on which the user must click. For example, if the user is viewing items in a list and adds one, then the updated list on the screen showing the added item is sufficient acknowledgment.
Delay message Informs the user that the computer system is working properly	When an activity takes more than seven seconds	Should permit the user to cancel the operation in case he or she does not want to wait for its completion. Should provide some indication of how long the delay will last.
Help message Provides additional information about the system and its components	In all systems	Help information is organized by table of contents and/or keyword search. Context-sensitive help provides information that depends on what the user was doing when help was requested. Help messages and online documentation are discussed in Chapter 12.

Input Design

- Inputs facilitate the entry of data into the computer system, whether highly structured data, such as order information (e.g., item numbers, quantities, costs) or unstructured information (e.g., comments).
- Input design means designing the screens used to enter the information as well as any forms on which users write or type information (e.g., timecards, expense claims).

Basic Principles

- The goal of the input mechanism is to simply and easily capture accurate information for the system.
- The fundamental principles for input design reflect the nature of the inputs (whether batch or online) and ways to simplify their collection.

1- **Online versus Batch Processing(I)**

- There are two general formats for entering inputs into a computer system: online processing and batch processing.
- With *online processing* (sometimes called *transaction processing*), each input item is entered into the system individually, usually at the same time as the event or transaction prompting the input.
- Online processing is most commonly used when it is important to have *real-time information* about the business process.
  - For example, when you reserve an airline seat, the seat is no longer available for someone else to use.

1- **Online versus Batch Processing(II)**

- With *batch processing*, all the inputs collected over some time period are gathered together and entered into the system at one time in a batch.
- Some business processes naturally generate information in batches. Batch processing is also used for transaction processing systems that do not require real-time information.
- This batching simplifies the data communications process and often saves in communications costs, but it does mean that inventories are not accurate in real time but rather are accurate only at the end of the day after the batch has been processed.

2- Capture Data at the Source(I)

- Perhaps the most important principle of input design is to capture the data in an electronic format at its original source or as close to the original source as possible.
- *Source data automation* refers to using special hardware devices to automatically capture data without requiring anyone to type it.
  - *Bar-code readers, Smart cards, …*
- As well as reducing the time and cost of data entry, these systems reduce errors because they are far less likely to capture data incorrectly.

3- **Minimize Keystrokes**

- Another important principle is to minimize keystrokes.
- Keystrokes cost time and money, whether they are performed by a customer, user, or trained data-entry operator.
- A system should not require a user to type information that can be selected from a list; selecting reduces errors and speeds entry.
- In many cases, some fields have values that often recur. These frequent values should be used as the *default value* for the field so that the user can simply accept the value and not have to retype it time and time again.

Types of Inputs(I)

- Each data item that has to be input is linked to a *field* on the form into which its value is typed. Each field also has a field *label*, which is the text beside, above, or below the field that tells the user what type of information belongs in the field. Often the field label is similar to the name of the data element. In some cases, a field displays a template over the entry box to show the user exactly how data should be typed.

Types of Inputs(II)

- **Text** As the name suggests, a *text box* is used to enter text. Text boxes can be defined to have a fixed length or can be scrollable and can accept a virtually unlimited amount of text. We never use a text box if we can use a selection box.
  - Text boxes should permit standard GUI functions, such as cut, copy, and paste.
- **Numbers** A *number box* is used to enter numbers. Some software can automatically format numbers as they are entered, so that 3452478 becomes $34,524.78. Dates are a special form of numbers that sometimes have their own type of number box. Never use a number box if you can use a selection box.
- **Selection Box** A *selection box* enables the user to select a value from a predefined list. The items in the list should be arranged in some meaningful order. The default selection value should be chosen with care. A selection box can be initialized as unselected. However, it is better to start with the most commonly used item already selected.

Input Validation

- All data entered into the system need to be validated to ensure their accuracy.
- Ideally, computer systems should not accept data that fail any important validation check to prevent invalid information from entering the system.

Types of Validation check

Type of Validation	When to Use	Notes
Completeness check Ensures all required data have been entered	When several fields must be entered before the form can be processed	If required information is missing, the form is returned to the user unprocessed.
Format check Ensures data are of the right type (e.g., numeric) and in the right format (e.g., month, day, year)	When fields are numeric or contain coded data	Ideally, numeric fields should not permit users to type text data, but if this is not possible, the entered data must be checked to ensure it is numeric. Some fields use special codes or formats (e.g., license plates with three letters and three numbers) that must be checked.
Range check Ensures numeric data are within correct minimum and maximum values	With all numeric data, if possible	A range check permits only numbers between correct values. Such a system can also be used to screen data for "reasonableness"—e.g., rejecting birthdates prior to 1880 because people do not live to be a great deal over 100 years old (most likely, 1980 was intended).
Check digit check Check digits are added to numeric codes	When numeric codes are used	Check digits are numbers added to a code as a way of enabling the system to quickly validate correctness. For example, U.S. Social Security numbers and Canadian Social Insurance numbers assign only eight of the nine digits in the number. The ninth number—the check digit—is calculated using a mathematical formula from the first eight numbers. When the identification number is typed into a computer system, the system uses the formula and compares the result with the check digit. If the numbers don't match, then an error has occurred.
Consistency checks Ensure combinations of data are valid	When data are related	Data fields are often related. For example, someone's birth year should precede the year in which he or she was married. Although it is impossible for the system to know which data are incorrect, it can report the error to the user for correction.
Database checks Compare data against a database (or file) to ensure they are correct	When data are available to be checked	Data are compared against information in a database (or file) to ensure they are correct. For example, before an identification number is accepted, the database is queried to ensure that the number is valid. Because database checks are more expensive than the other types of checks (they require the system to do more work), most systems perform the other checks first and perform database checks only after the data have passed the previous checks.

Output Design

- Outputs are what the system produces, whether on the screen, on paper, or in other media, such as the Web.
- Outputs are perhaps the most visible part of any system because a primary reason for using an information system is to access the information that it produces.

Basic Principles

- The goal of the output mechanism is to present information to users so that they can accurately understand it with the least effort.
- The fundamental principles for output design reflect how the outputs are used and ways to make it simpler for users to understand them.

1- Understand Report Usage

- The first principle in designing reports is to understand how they are used.
- Web reports that are intended to be read from start to finish should be presented in one long scrollable page, whereas reports that are used primarily to find specific information should be broken into multiple pages, each with a separate link.
- The frequency of the report can also play an important role in its design and distribution.
- *Real-time reports* provide data that are accurate to the second or minute at which they were produced. *Batch reports* are those that report historical information that may be months, days, or hours old, and they often provide additional information beyond the reported information.
    - If the information in a report is time critical, then real-time reports have value. This is particularly important because real-time reports are often expensive to produce; unless they offer some clear business value, they might not be worth the extra cost.

2- Manage Information Load

- The goal of a well-designed report is to provide all the information needed to support the task for which it was designed.
- This does not mean that the report needs to provide all the information available on the subject—just what the users decide they need in order to perform their jobs.
- In some cases, this can result in the production of several different reports on the same topics for the same users because they are used in different ways.
- Information should be provided in a format that is usable without modification. The user should not need to re-sort the report's information; instead critical information should be highlighted so that users can find it more easily amid a mass of data, or perform additional mathematical calculations.

3- **Minimize Bias**

- No analyst sets out to design a biased report. The problem with bias is that it can be very subtle; analysts can introduce it unintentionally.
- *Bias* can be introduced by the way lists of data are sorted because entries that appear first in a list can receive more attention than those later in the list. Data are often sorted in alphabetical order, making those entries starting with the letter *A* more prominent. Data can be sorted in chronological order (or reverse chronological order), placing more emphasis on older (or most recent) entries. Data may be sorted by numeric value, placing more emphasis on higher or lower values.

International And Cultural Issues And User Interface Design

- With the World Wide Web, virtually any firm can have a global presence.
- International and cultural issues
  - Multilingual requirements,
  - Color,
  - Cultural differences.

Multilingual Requirements(I)

- The first and most obvious difference between applications used in one region and those designed for global use is language.
- Global applications often have *multilingual requirements*, which means that they have to support users who speak different languages and write using non-English letters.
- One of the most challenging aspects in designing global systems is getting a good translation of the original language messages into a new language.
- Rules
  - Keep the writing short and simple. It is much easier to avoid mistranslations.
  - Avoid humor, jargon, slang, clichés, puns, analogies, and metaphors. These tend to be too culturally specific.
  - Use good grammar. Be sure to punctuate everything correctly.
- Another challenge is often screen space. In general, English-language messages usually take 20 percent to 30 percent fewer letters than their French or Spanish counterparts. Designing global systems requires allocating more screen space to messages than might be used in the English-language version.

Multilingual Requirements(II)

- Some systems supports several different languages simultaneously (a concurrent multilingual system). Other systems contain separate parts that are written in each language and must be reinstalled before a specific language can be used; that is, each language is provided by a different version of the system so that any one installation will use only one language.
  - Either approach can be effective, but this functionality must be designed into the system well in advance of implementation.
- Finally, one other consideration that must be considered is reading direction. In most Western societies, readers read from left to right and top to bottom. This is not true for many cultures.

Color

- To begin with, color is not black and white. The meaning associated with a color is totally culturally dependent.
- In most Western cultures, black is associated with death, mourning, and grief or with respect and formality. In many Eastern cultures, white is associated with death or the color of robes worn by religious leaders.
- In the United States, red implies excitement, spice passion, sex, and even anger; in Mexico, it indicates religion; in the United Kingdom, it indicates authority, power, and government; in Scandinavian countries, it indicates strength; and in China, it means communism, joy, and good luck. Blue is associated with cleanliness in Scandinavia; love and truth in India; loyalty in Germany; and trust, justice, and "official" business in the United States. In Ireland, green signifies nationalism and Catholicism, and in the United States it denotes health, environmentalism, safety, greed, and envy. In the Arab Middle East green is a sign of holiness, in France it represents criminality, and in Malaysia it signifies danger and disease. Yellow also has many culturally dependent meanings. In the United States, it is associated with caution and cowardice; in Scandinavia, warmth; in Germany, envy; and in India, commerce. Purple signifies death, nobility, or the Church in Latin America, the United States, and Italy, respectively.
- Obviously, when building a website for a global audience, colors must be chosen carefully; otherwise, unintentional messages will be sent.

Cultural Differences

- When developing a website for an international audience, you need to think about what message needs to be sent to a local culture from your global organization to achieve the business goals of the firm.
- Consequently, you need to be able to understand the different local cultures.

Three dimensions that are directly relevant to user interface design(I)

- Speed of messages, context, and time.
- The *speed of messages* dimension deals with how fast a member of a culture is expected to understand a message and how "deep" the content of a typical message will be in a culture. The deeper the message content, the longer it will take for a member of a culture to understand the message. For example, two different approaches to describe a historical event would be a news headline (fast and shallow) and a documentary (slow and deep).
- Different cultures have different expectations of the content and response to a message. For a Western audience, minimizing the number of words contained in a user interface makes sense. Westerners prefer to get to the point as fast as possible. However, this is not true for Eastern cultures. Consequently, for a firm like Amazon.com, providing detailed reviews and short excerpts from a book provides support for a slow and deep culture, while providing bullet point types of comments supports the fast and shallow culture. By providing both, Amazon.com addresses both needs.

Three dimensions that are directly relevant to user interface design(II)

- The second dimension, *context*, deals with the level of implicit information that is used in the culture versus the information needing to be made explicit. In high-context cultures, most information is known intrinsically and does not have to be made explicit. Therefore, the actual content of the message is fairly limited. However, in low-context cultures, everything must be spelled out explicitly to avoid any ambiguity, and therefore the message needs to be very detailed. You will find this dimension causing problems when attempting to close a business deal.
- From a website design perspective, it is pointed out that in a high-context culture, focusing the design on aesthetics, politeness, and humility produces an effective website, but in a low-context culture, things such as the terms and conditions of a purchase, the “rank” of the product and firm, and the use of superlatives in describing the product and firm are critical attributes of a successful website.

Three dimensions that are directly relevant to user interface design(III)

- *Time*, addresses how a culture deals with many different things going on simultaneously.
- In a *polychronic time* culture, members of the culture tend to do many things at the same time but are easily distracted and view time commitments as very flexible. With *monochronic time* cultures, members of the culture solve many things by focusing on one thing at a time, are single-minded, and consider time commitments as something that is set in stone. When designing for a polychronic culture, the liberal use of “pop-up” messages might be fun and engaging, while in a monochronic culture, pop-up messages simply annoy the user.
- However, with the use of e-mail interruptions and text messaging, this could change over time.

Other dimensions that are relevant to user interface design(I)

- Power distance, uncertainty avoidance, individualism versus collectivism, and masculinity versus femininity.
- The first dimension, *power distance*, addresses how the distribution of social power is dealt with in the culture. In cultures with a high power distance, members of the culture believe in the authority of the social hierarchy. In cultures with low power distance, members of the culture believe that power should be more equally distributed. Consequently, in cultures with a high power distance, emphasis on the "greatness" of the leaders of the firm, the use of "proper titles" for members of the firm, and the posting of testimonials on behalf of the firm by "prominent" members of society is important. International awards won by the firm, its members, or its products should also be posted prominently on the website.
- The second dimension, *uncertainty avoidance*, addresses to what degree a culture is comfortable with uncertainty. In a culture with a high uncertainty avoidance, members avoid taking risks, value tradition, and are much more comfortable in a rule-driven society. In cultures that score high on uncertainty avoidance, more customer service needs to be provided, more important "local" contacts need to be available, the firm's and product's history and tradition need to be provided on the website, and, in the case of software, the use of free trials and downloads is critical. In other words, you need to build trust and reduce perceived risk between the customer and the firm.

Other dimensions that are relevant to user interface design(II)

- The third dimension, *individualism* versus *collectivism*, is based on the level of emphasis the culture places on the individual or the collective, or group. In North America and Europe, individualism is rewarded. However, in East Asia, it is believed that by focusing on optimizing the group, the individual will be most successful. In other words, it is the group that is the most important. In a collective society, presenting information on how the firm "gives back" to the community; supports "member" clubs, "loyalty" programs, and "chat" facilities; and provides links to "local" sites of interest are very important characteristics for a website. In contrast, in an individualistic society, providing support for personalization of the user's experience with the website, emphasizing the uniqueness of the products that the user is viewing, and emphasizing the privacy policy of the site are critical.
- The fourth dimension, *masculinity* versus *femininity*, does not mean how men and women are treated by the culture. But, instead this dimension addresses how well masculine and feminine characteristics are valued by the culture. For example, in a masculine culture, characteristics such as being assertive, ambitious, aggressive, and competitive are valued, whereas in a feminine culture, characteristics such as being encouraging, compassionate, thoughtful, gentle, and cooperative are valued.

Nonfunctional Requirements And Human–computer Interaction Layer Design(I)

- The human–computer interaction layer is heavily influenced by nonfunctional requirements.
- *Operational requirements*, such as choice of hardware and software platforms, influence the design of the human–computer interaction layer. For example, something as simple as the number of buttons on a mouse (one, two, three, or more) changes the interaction that the user will experience. Other operational nonfunctional requirements that can influence the design of the human–computer interaction layer include system integration and portability.
- *Security requirements* affecting the human–computer interaction layer deal primarily with the access controls implemented to protect the objects from unauthorized access. HCI layer design must include appropriate log-on controls and the possibility of encryption.

Nonfunctional Requirements And Human–computer Interaction Layer Design(II)

- *Performance requirements*, over time, have become less of an issue for this layer. The interaction of the human–computer interaction layer with the other layers must be considered. For example, if the system response is slow, incorporating more-efficient data structures with the problem domain layer, including indexes in the tables with the data management layer, and/or replicating objects across the physical architecture layer could be required.
- In addition to *the international and cultural issues* described previously, unstated norm requirements include having the date displayed in the appropriate format (MM/DD/YYYY versus DD/MM/YYYY). For a system to be truly useful in a global environment, the user interface must be customizable to address local cultural requirements.

Reference

- **Dennis, Wixon, Tegarden**, "**System Analysis and Design, An Object Oriented Approach with UML", 5th Edition, 2015.**