

CHAPTER-12

View Layer: Designing Interface Objects

Introduction:-

- Main goal of UI is to display and obtain needed information in an accessible, efficient manner.
- ⇒ It's important for a design to provide users the information they need & to clearly tell them how to complete a task.
- ⇒ Designing view layer by mapping UI objects to the view layer objects.

Designing View layer classes:-

- View layer object (or) Interface objects are the only exposed object of an application with which user can interact.
 - View layer represent set of operations in the business, the user must perform to complete their task.
 - Any object that have direct contact with outside world are visible in Interface objects.
 - View layer objects responsible for 2 things
 - 1.) Input ⇒ Responding to user interaction.
 - 2.) Output ⇒ displaying or printing business objects
 - The Process of designing View layer classes
- 1.) Macro level UI design process : Identify view layer objects :-
- Mostly takes place during analysis phase.
 - Goal : Identify classes that interact with human actors by analyzing the use-case developed in analysis phase.

→ This use case should capture a complete, unambiguous, consistent picture of interface requirements of the system.

→ Use-cases describes what the system does rather than how it does.

→ Sequence/collaboration allows to find actor-system interaction & identify interface classes that interact with human actors.

2.) Micro level UI Design activities:-

1.) Designing the view layer objects by applying design axioms & guidelines:-

→ decide the use of components so they best support application specific functions.

2.) Prototyping the view layer interface:-

3.) Testing usability and user satisfaction.

4.) Refining and Iterating the design.

\Rightarrow Macro level process : identifying view classes by analyzing use cases:-

→ Interface objects acts as an buffer b/w user & rest of business objects.

→ Business objects lie inside the business layer & involve no interaction with actors.

Ex: Calculation of Employee salary for overtime duty

It is the work of business layer → but data need to be entered through interface objects

→ Interface objects have the coordinating responsibility for those tasks that come into direct contact with the user.

View layer macro process consists of 2 steps:-

- 1.) For every class identified determine if the class interacts with a human actor, if so do
 - a.) Identify interface objects of the class.
→ use sequence / collaboration diagrams.
 - b.) Define the relationship among view (interface) objects.

2) Iterate & refine.

Micro level process:-

→ Design of View layered objects must be user driven or user centered.

→ User-centered interface replicates user's view of doing things.

Ex: To automate a paper work.

↓
→ So design the application where users can apply their previous real-world knowledge (what they did in paper) to the application interface.

→ The main goal of View layer is to address user's needs.

Process of designing View layer (Interface objects):-

- 1.) For every object identified in macro UI process, apply the micro-level UI design rules & guidelines to develop the UI.

design rules & GUI guidelines

3 UI Design rules:

1) Rule:- Making the Interface simple (Corollary 2)

→ Since users are unaware of the tools and mechanism of application so it becomes complicated. So user's must have simple interface so they can learn new application more easily.

Ex: Driver need not want to know about car engines, just they want to know driving.

→ Corollary 2 says that each class must have single, clearly defined purpose.

2) KISS (Keep it Simple and Straightforward)

→ If a user cannot use the screen developed, without asking questions, then the design is bad.

→ Sometimes additional features & short cuts can affect the product. So to evaluate, consider

1.) Every additional feature potentially affects the performance.

2.) Harder to fix a design problem after the release of product.

3.) Making something simple to use often requires a good work and code.

4.) Features implemented by small extension do not have proportional effect Btw.

2) Rule 2:- Making The Interface transparent and Natural (Corollary 4)

→ Users can identify what to be done next by applying their previous knowledge of doing task.
→ so an application should reflect user goals & task needed to achieve that goal.

Corollary 4.1.1 → There should be a strong mapping b/w user's view of doing things and vi clauses.

Metaphor :- Two unrelated things, using one to denote another.

→ Metaphors are a way to develop user's conceptual model of an application. (makes work simple by using symbols, which are known to the user)

→ Metaphor - need to choose in a way that it should meet the expectations user's have bcz of their real world experience.

① → often an application design is based on single metaphor.

→ The goal of user interface design is to make the user interaction with the computer as simple & natural as possible.

3) Rule 3:- Allowing users to be in control of the S/w
(Corollary 1):-

→ User always should feel in control of the S/w, rather than feeling controlled by S/w.

3. Implications of this concept

1.) Implication 1:-

→ Actions are started by the user rather than the computer.

→ So the user plays an active rather than reactive role.

2.) Implication 2:-

→ Users must be able to customize the aspects of the interface (ex: color, fonts etc.,).

3.) Implication 3:-

→ S/w should be as interactive & responsive as possible.

→ Avoid modes whenever possible.

Mode ⇒ state that excludes general instruction or otherwise limits the user to specific interactions).

→ But we cannot omit modes always, there are situations in which modes are useful.

→ Some the ways to put users in control are

- 1.) Make the interface forgiving
- 2.) Make the interface visual
- 3.) Provide immediate feedback
- 4.) Avoid modes

5.) Make the interface consistent.

→ mode in a distinct setting that produces different results from the same user input than other settings.

1) Make the Interface forgiving:-

→ User's action should be easily reversed.

→ So user can use the application without fear of causing irreversible mistake.

→ So user's can backup or undo previous actions.

→ Unexpected loss of data should require a confirmation.

→ User's feel more comfortable with a system when their mistakes do not cause serious or irreversible results.

2) Make the Interface visual:-

Design Interfaces where user's can see.

3.) Provide Immediate feedback:-

Users should never press a key or select an action without receiving immediate visual feedback.

4.) Avoiding modes:-

→ Mode forces users to focus on the way an application works, instead on the task they want to complete.

→ There are some modes that can be used in user interface.

a) Modal dialogue:-

→ Sometimes an application needs information to continue, (like name of a file into which users want to save something).

→ When an error occurs, users may be required to perform an action before they continue their task.

b.) Spring-loaded modes:-

Users want to continually take some actions to remain in that mode. (dragging mouse etc.)

c.) Tool-Driven modes:-

→ Like pen, pencil tool used for drawing.
After selecting the tool the mouse pointer changes to match the selected tool.

2.) Make the Interface consistent:-

→ Gives the user the feel that he/she is in control.

→ User interfaces should be consistent throughout the applications.

Purpose of a view layer interface:-
we can view the application using this view layer.

↓ connecting one module to another

Interface contains one or more windows, where each window serves a specific purpose.

1.) Forms and data entry window:-

→ This window provides access to data that users can retrieve, display and change.

2.) Dialog boxes:-

Display status information or ask user to supply information.

2) Application window (Main window):-

It contains entire application with which user can interact.

10) Guidelines for designing forms and Data entry windows:-

1) Identify the information you want to display or change. The issues are:

- a.) what kind of information will users work with and why?
- b.) Do users need access to all the information in the table or just some information?
- c.) In what order do users want rows to appear?

2) Identify the tasks that users need to work with data on the form. It includes

- a.) Navigating through rows
- b.) Adding or deleting rows
- c.) Changing data in rows
- d.) Saving and abandoning changes.

3) Guidelines for choosing layout:-

- a.) Use existing paper form, as a starting point of your design.
- b.) If it contains too much info. to fit on a screen, use main window with optional smaller windows.
- c.) Put required / frequently entered information on top & left side of the form. (bcz user behaviour is reading from left-to-right).

- d.) Align fields at their left edges, as users expect to type info. from left-to-right and top-to-bottom.
- e.) Put similar information together.

a.)

Guidelines for Designing Dialog Boxes and Error Messages

Dialog box provides exchange of Information b/w user and application.

If a dialog box is for an error message, use the following guidelines.

a.) Your Error message should be positive,-
Ex: "Enter date as dd/mm/yyyy".

b.) Error message should be constructive:-
→ In this, the user should feel as they are controlling the system rather than the software is controlling them.
Ex: "Press undo button & try again"

c.) Error message should be brief and meaningful
→ It depends on place where it emerges.

Ex: "ERROR: off command" → This is useful for programmer but not for user.

d.) Arrange the controls in dialog box in people reading direction behaviour.
→ Left-to-right.
→ Top-to-bottom.

3) Guidelines for Command Button layouts:-

- Usually on upper-right border dialog box.
- Sometimes on bottom of dialog box.
- On web interfaces → Placing on left border is very popular
 - If command buttons are placed with in a window but not on tabbed page, they apply to entire window.

4) Guidelines for Designing Application windows:-

Application window usually contains common drop-down menus.

1.) File menu:-

- Interface for primary operation.
- Open, Save, Save As, Print, Exit, close.

2.) Edit menu:-

- Cut, Copy, Paste commands, Undo, Find, delete

3.) View menu and other command menus:-

- It changes the user's view of data in the window.
- Zoom, Show Rulers etc.,

4.) Window menu:-

Managing the windows in main workspace.

5.) Help menu:-

→ Contains command that provide access to Help Information.

6.)

Guidelines for using colors:-

→ To add visual appeal to the form.

→ Guidelines

- a) Use identical or similar colors to indicate related information.
- b) For object background, use contrasting but complementary color.
- c) Can use bright colors to call attention to certain elements.
- d) Don't use too many colors.
- e) Allow user's to modify color configuration of the application.

6.)

Guidelines for using fonts:-

→ Use commonly installed fonts & not specialized fonts.

→ Use bold for control labels

→ Don't use too many font styles.

→ Emphasize text by increasing font size on the form.

Prototyping the user interface:-

It is highly needed to prototype the user interface during the analysis to better understand the system requirement.

Creating user interface consist of 3 steps

- 1.) Create user interface object (like buttons)
- 2.) Link or assign appropriate behaviours or actions to these UI objects.
- 3.) Test, debug.

