Page No.
Date : / /

SE Assignment 1

Name : Ayush Jain
Sap Id : 60004200132
Div : B

1 Elaborate the task set for creating Component Level Design in OO Projects

Ans :

- Component level design is elaborative in nature
- It transforms information from requirements and architectural models into a design representation that provides sufficient detail to guide the construction (coding and testing) activity
- The following steps represent a typical task set for component level design when applied to object oriented projects:-

STEP 1: Identify all design classes that correspond to the problem domain.
-> using the requirements and architectural model each analysis class and architectural component is
STEP 2: Identify all design classes that correspond to the infrastructure domain
->These classes are not described in the requirements model and are often missing from the architecture model but they must be described at this point.
->classes and components in this category include :
GUI components
operating system components
object and data management components.

Page No.
Date : / /

STEP 3: Elaborate all design classes that aren't acquired as reusable components
-> Elaboration requires that all interfaces attributes and operations necessary to implement the class be described in detail
Design heuristics (eg component cohesion and coupling]

STEP 3(a): Specific message details when classes or components collaborate
->The requirements model makes use of a collaboration diagram to show how analysis classes collaborate with one another
STEP 3(b): Identify appropriate interfaces for each
->within the context of component-level design a UML Interface group of Public [externally visible] operations
->the interface contains no internal structure has no attributes or associations

STEP 3(c): Elaborate attributes and define data types and data structures required to implement them
-> In general data structures and types used to define attributes are defined within the content Programming language that is used for the implementation
->UML defines an attribute's data type using the following syntax:
name: type-expression initial -value {Property string }
->Here name is the attribute name type expression is the data type initial value is the value that the attribute takes when an object is created and

Page No.
Date : / /

Property string defines a property or characteristic of the attribute

STEP 3 (D): Describe processing flow within each operation in detail
->This may be accomplished using a programming language-based pseudocode or with a UML activity
-> Each software component is elaborated through number of iterations that apply the refinement
STEP 4: Describe persistent data sources [databases and files] and identify the classes required to them
->Databases and files normally transcend the design description of manage them an individual component In most cases these persistent data stores are initially specified as part of architectural design

STEP 5: Develop and elaborate behaviorial representations for a class or component
-> UML State diagrams were used as a part of requirements model to represent the externally observable behaviour of the system.
STEP 6: Elaborate deployment diagrams to provide additional implementational detail Deployment diagrams are used as part of architecture
STEP 7: Refractor every component-level design representation and always consider alternatives.

2. Explain the golden rules of user Interface Design
Ans: User Interface Design creates an effective communication medium between a human and a

Scanned with CamScanner

Page No.
Date : / /

computer following a set of interface design Principles design identifies interface objects and actions and then creates a screen layout that forms the basics for a user interface Prototype

The Golden Rules of User - Interface Design are :-
(i) Place the user in control
a) Design interaction modes in a way that does not force a user into unecessary or undesired actions
->An interaction mode is the current state of the interface For example if spell check is selected in a word Processor menu the software moves to a spell-checking mode The user should be able to enter and exit the mode with little or more effort

b)Provides flexible interaction
->Because different users have different interaction Preferences choices should be Provided
->For example software might allow a user to interact via keyboard commands mouse movements digitizer Pen a multitouch screen recognition commands However not every action is amenable to every interaction mechanism
For example There will be difficulty in using keyboard Commands or voice inputs to draw a complex shape.
c) allow user interaction to be interruptible and undoableEven when involved in a sequence of actions the user should be able to interrupt the sequence to do something else.
->The user should also be able to "undo" any action.

Scanned with CamScanner

d)Streamline interaction as skill levels advance and allow the interaction to be customised
->users often find that they perform the same sequence of interactions repeatedly
->It is worth while to design a "macro" mechanism that enables an advanced user to customise the Interface to facilitate interaction

(ii) Reduce the user's memory load
->The more a user has to remember the more error prone the interaction with the system will be. It is for the reason that a well-designed user interface doesn't tax the user memory
a) Reduce demand on short-term memory
-> when users are involved in complex tasks memory can be significant. The interface should be designed to reduce the requirement to remember past actions inputs and results
->This can be accomplished by providing visual cues that enable a user to recognise past actions rather than recalling them
b) Establish meaningful defaults
->The initial set of defaults should make sense to the average user but he should be able to specify his individual preferences as well However a "reset" option should be available enabling the redefination of original default values

c) Define shortcuts that are intuitive
-> when mnemonics are used to accomplish a system function (eg alt + P to invoke the print function)

Page No.
Date : / /

the mneumonic should be tied to the action in a way that is easy to remember.
d) The visual layout of the interface should be based on a real-world metaphor
->For example a bill Payment system should use a checkout and check register metaphor guide the user through the bill Paying Process
->This enables the user to rely on well understood visual cues rather than memorizing an arcane interaction sequence.
d) Disclose information in a Progressive fashion.
->The interface should be organized hierarchically
->For example in word Processing applications underlining Function is available in the style menu when the user Picks it all underlying options such as single underline double underline dashed underline are Presented.

(iii) Make the interface consistent.
->The interface should Present and acquire information in a consistent fashion.
This implies that:
(1)All visual information is organized according to design screen rules that are maintained throughout all screen displays
(2)Input mechanisms are constrained to a limited set that is used consistently throughout the
(3) Mechanisms for navigating from task to task are consistently defined and implemented.

Design Principles :-

Page No.
Date : / /

a-> Allow the user to put the current task into a ~~meaning~~ meaningful context
->Many interfaces implement complex layers of interactions with dozens of screen images
->It is important to provide indicators [eg window titles graphical icons the user Consistent color coding] that enables to know the content of the work at hand.

b) Maintain consistency across a family of applications
->A set of applications (or Products) should all implement the same design rules so that consistency is maintained for all interactions

c)If Past interactive models have created user expectations do not make changes unless there is a compelling reason to do so.
->For example once a particular interactive sequence has become a defacto standard [eg: use of ctrl + V for pasting] the user expects this in every application he encounters A change [eg using alt +V to invoke pasting] will cause confusion.