CONCEPTUAL MODEL DESIGN

WORK REENGINEERING

In the Work Reengineering task, you perform the following basic steps

- Reengineer the Current User Task Organization Model and the Task Scenarios and Use Cases.

- Validate and refine the Reengineered Task Organization Model and the Reengineered Task Sequence Models.

- Document the Reengineered Task Organization Model and the Reengineered Task Sequence Models.

CONCEPTUAL MODEL DESIGN INTRODUCTION

- A conceptual model design suggests high level presentation rules.

- A conceptual model design facilitates the overall design process.

- A conceptual model is **the mental model that people carry of how something should be done**.

- The conceptual model is the foundation, framework, or overall architecture of the interface, as opposed to the details, such as screen content, layout and design, and feedback.

- This foundation needs to be simple, natural, intuitive, consistent, and a good match to the tasks being supported by the application.

- Conceptual Model Design assumes a particular visual and interaction style and specifies how its components are used to present the underlying task organization model.

INTRODUCTION

- Conceptual model design can be designed for very different technology platforms with different capabilities. E.g web applications, factory equipment's(knobs, dials, led displays)

- A set of presentation rules that consistently present categories of functional components.

PURPOSE

- To provide a coherent, rule based framework that will provide a unifying foundation for all the detailed interface design decisions to come.

- unifying foundation: a set of rules that allows designers to reduce the full complexity of a product to a smaller set of information to learn and remember.

- Helps to learn the product more quickly, efficiently, effectively and to predict behavior of product.

- Taps into the ways in which most naturally think, reason and learn.

PURPOSE

- A conceptual model design is a high level UI design(formal documentation not recommended)
- In general you will be generating several sketchy alternative conceptual model designs
- You can iteratively improve and develop a conceptual design model
- Conceptual model design will be based on small, representative pieces of the total product functionality.
- Only high level, presentation and navigation issues are considered and designed.

PRODUCT ORIENTED APPLICATIONS

- A product oriented model best fits in which there are clear, identifiable work products that users individually create, name and save.

- Examples of product oriented applications: MS EXCEL, word and power point.

- Allow users to maintain, modify, create and maintain documents

CONCEPTUAL MODEL DESIGN DECISIONS

- Whether to design a product/ process based conceptual model

- How to define product or process

- How to present product or processes

- What rules to follow for the use of window types

- How to define major displays and the navigational pathways between them

ROLES AND RESOURCES

- Task leader: UI designer

- Other resources. All project team members who participated in user profile, contextual task analysis and task organization phases should participate and provide input and feedback

EXAMPLES

- Parts of the menu bar would be designed to represent navigational pathways

- Major type of windows would be identified and named(not filled with content)

- The conceptual model design would be continuously refined and updated in response to all later usability evaluation tasks

EXAMPLES

- In word processing application, Primary Product is document and predefined one tool is styles and formats.

- In Spreadsheet software primary product is spreadsheet and tool would be standard equations

- In software and hardware inventory system primary product will be procurement form and tool may be vendor contact list

- Tools are used to build primary products

PRESENTATION IN PRODUCT ORIENTED APPLICATIONS

In a product-oriented GUI interface, almost everything presented to users can be classified as one of three things: *products*, which are entities that users would think of as things or products that they can create, name, and save; *tools*, which are objects supplied by the system that users might use in building products; and *actions*, which users can perform in order to create and modify products. There are in turn two types of products (things users create, name, and save): *primary products*, which are the main point of the application, and *secondary products*, which users may create to use as a tool in building or modifying the primary products.

PROCESS ORIENTED APPLICATIONS

- A Process oriented model best fits in which there are no clear, identifiable primary work products.

- Support some work process

- Information may be stored or retrieved but all users will have access to same information in general

- Not creating individual work products like documents

- E.g financial management system, floor monitoring, inventory tracking etc

PROCESS ORIENTED APPLICATIONS

- Tabs

- Menu bar

- Pull down from menu bar

resented. For example, perhaps a tab metaphor is being considered, and first-level processes in the hierarchy are represented as tabs, second-level processes as menu bar picks within tabs, and third-level processes as pull-downs from menu bar picks

STEP BY STEP PROCEDURE

- Define the conceptual Model as either product or process oriented

- Clearly identify products or processes

- Design presentation rules for products or processes

- Design rules for windows(movable, resizable, scrollable, can be represented as icons, share application window menu bar, message box, modal dialog boxes or not, primary and secondary windows, distinguish among applications, document, dialog box and message box with the help of colors etc..)

- Identify major displays(e.g primary window(document) or dialog box(secondary windows))

- Define and design major navigational pathways(define interaction, alternative controls for navigation e.g push button)

- Document alternative conceptual model design in sketches and explanatory(representation of application and its associated documents, menu bar and pull down choices(e.g file, open), pathways)

SAMPLE

The Standard GUI Conceptual Model

1. **Define the Conceptual Model as either Product- or Process-Oriented.**

The conceptual model for Microsoft Word is product-oriented.

2. **Clearly Identify Products or Processes.**
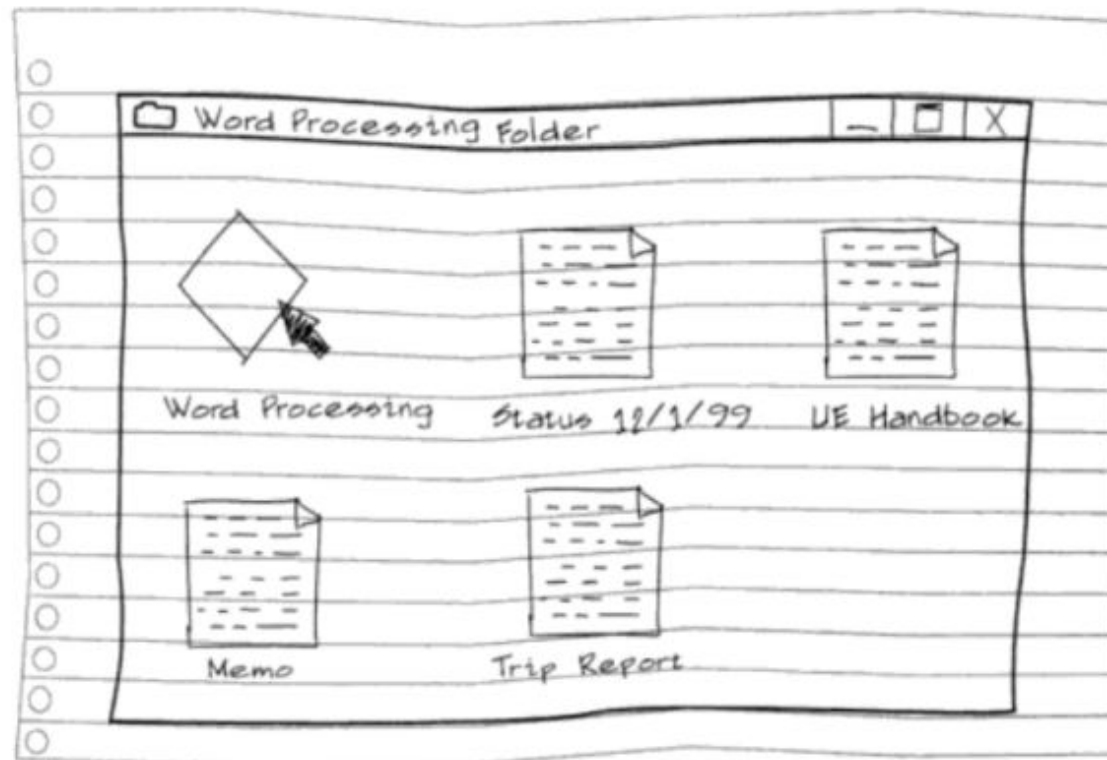
There is one type of primary product: a document. There is at least one type of secondary product—styles—and there likely will be others.

3. **Design Presentation Rules for Products or Processes.**

Primary products (documents) are represented at the desktop or operating system level as large icons with labels. They are also represented within the application through the Open dialog box as labeled small icons in a list box.

The only other entities represented as icons at the desktop level are folders and applications.

The user may have created and saved any number of documents. The user might also have created a folder at the desktop level and in it stored the application and all associated documents. If not, then the application would be stored somewhere in the Start button hierarchy, and the documents would be stored on directories in the Windows Explorer.

Double-clicking on the application icon at the desktop or operating system level opens the application in a primary window, and an empty, untitled product (document) in a primary window within it.
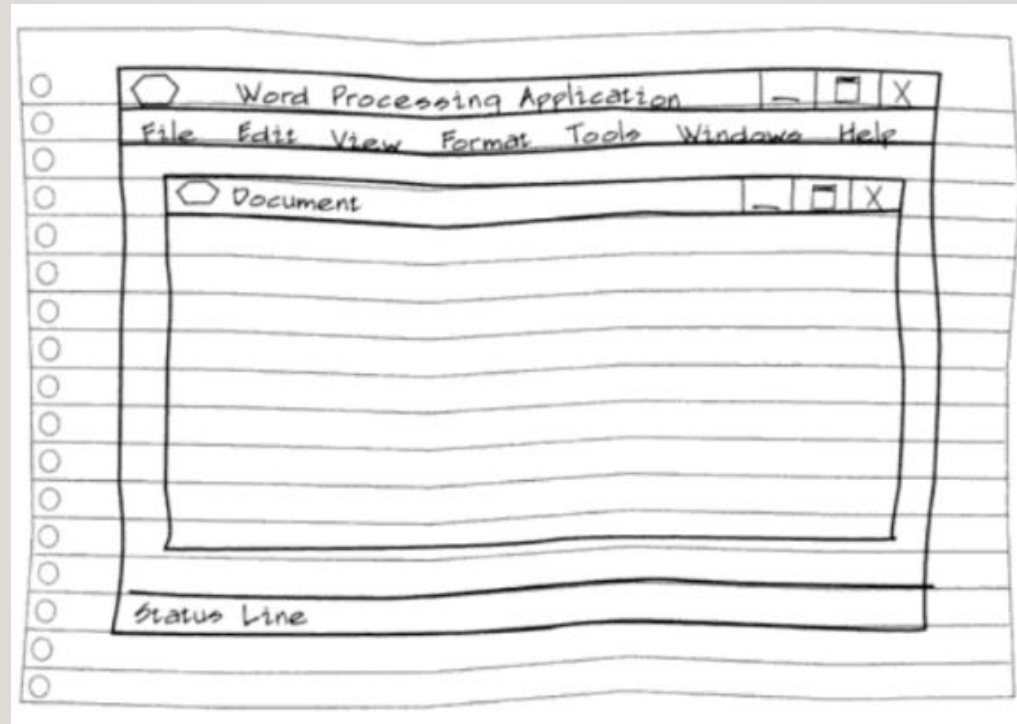
Double-clicking on a product (document) icon from the operating system or desktop level opens the application window and a product (document) window displaying that product (document).

Secondary products, such as styles, are presented as labels in a list box within a dialog box accessed through the menu bar.

APPLICATION WINDOW

MAJOR PATHWAYS

SUMMARY OF CONCEPTUAL MODEL DESIGN DECISIONS

- Conceptual model design decisions include:

- 1) Choice of a high level set of user objects, and decisions about how they will be presented and how the user will navigate among them,

- 2) Design of a set of simple rules for the consistent use of primary windows, dialog boxes and message boxes, and the assignment of behavioral properties (e.g., modal vs. modeless, resizable vs. non-resizable, movable vs. non movable, scrollable vs. non-scrollable) to those windows,

- 3) Decisions on when and when not to use action bars in windows, and the design of the organization and labeling of picks in the action bar(s) and pull downs

- 4) Design of a simple set of rules for when to use the different available GUI "widgets" (e.g. radio 'buttons, list boxes, drop downs etc.)

- 5) Organizing functionality and designing navigation through the interface to best support the users' task,

- 6) Using color effectively.

REFERENCE

- Chapters 8, "The Usability Engineering Lifecycle" by Deborah J. Mayhew