Ch.15 User Interface Design

- **Interface Design**

  - Easy to learn?
    - Easy to use?
      - Easy to understand?

- **Interface Design**

  - *Typical Design Errors*
    - lack of consistency
    - too much memorization
    - no guidance / help
    - no context sensitivity
    - poor response
    - Arcane/unfriendly

- **Golden Rules**

  - *Place the user in control*
  - *Reduce the user's memory load*
  - *Make the interface consistent*

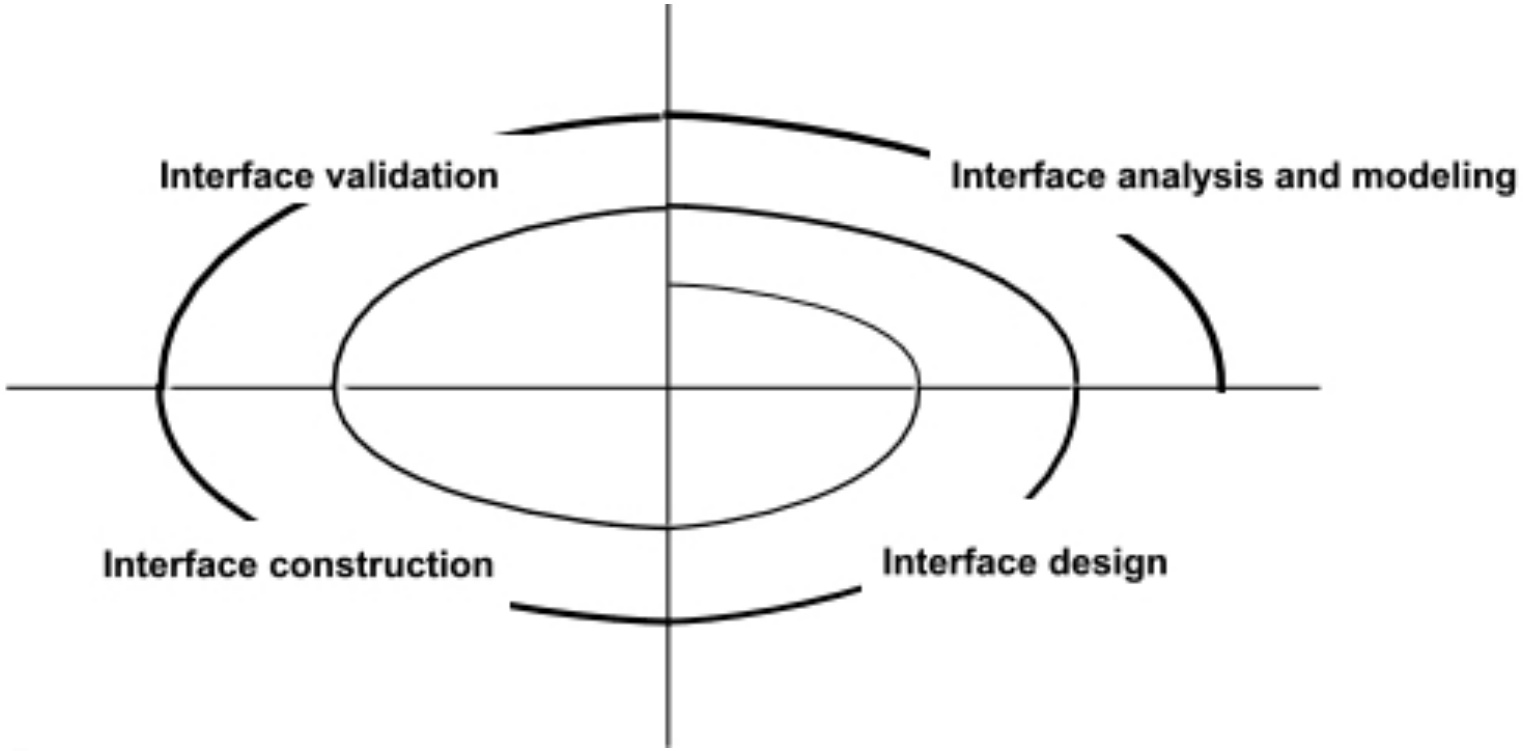
- **Place the User in Control**
  - Define interaction modes in a way that does not force a user into unnecessary or undesired actions.
  - Provide for flexible interaction.
  - Allow user interaction to be interruptible and undoable.
  - Streamline interaction as skill levels advance and allow the interaction to be customized.
  - Hide technical internals from the casual user.
  - Design for direct interaction with objects that appear on the screen.

- **Reduce the User's Memory Load**
  - Reduce demand on short-term memory.
  - Establish meaningful defaults.
  - Define shortcuts that are intuitive.
  - The visual layout of the interface should be based on a real world metaphor.
  - Disclose information in a progressive fashion.

- **Make the Interface Consistent**
  - Allow the user to put the current task into a meaningful context.
  - Maintain consistency across a family of applications.
  - If past interactive models have created user expectations, do not make changes unless there is a compelling reason to do so.

- **User Interface Design Models**
  - *User model* — a profile of all end users of the system
  - *Design model* — a design realization of the user model
  - *Mental model (system perception)* — the user's mental image of what the interface is
  - *Implementation model* — the interface "look and feel" coupled with supporting information that describe interface syntax and semantics

- **User Interface Design Process**



- **Interface Analysis**

  - Interface analysis means understanding
    - (1) the people (end-users) who will interact with the system through the interface;
    - (2) the tasks that end-users must perform to do their work,
    - (3) the content that is presented as part of the interface
    - (4) the environment in which these tasks will be conducted.

- **User Analysis**
  - Are users trained professionals, technician, clerical, or manufacturing workers?
  - What level of formal education does the average user have?
  - Are the users capable of learning from written materials or have they expressed a desire for classroom training?
  - Are users expert typists or keyboard phobic?
  - What is the age range of the user community?
  - Will the users be represented predominately by one gender?
  - How are users compensated for the work they perform?
  - Do users work normal office hours or do they work until the job is done?
  - Is the software to be an integral part of the work users do or will it be used only occasionally?
  - What is the primary spoken language among users?
  - What are the consequences if a user makes a mistake using the system?
  - Are users experts in the subject matter that is addressed by the system?
  - Do users want to know about the technology the sits behind the interface?

- **Task Analysis and Modeling**
  - Answers the following questions …
    - What work will the user perform in specific circumstances?
    - What tasks and subtasks will be performed as the user does the work?
    - What specific problem domain objects will the user manipulate as work is performed?
    - What is the sequence of work tasks—the workflow?
    - What is the hierarchy of tasks?
  - *Use-cases* define basic interaction
  - *Task elaboration* refines interactive tasks
  - *Object elaboration* identifies interface objects (classes)
  - *Workflow analysis* defines how a work process is completed when several people (and roles) are involved

- **Swimlane Diagram**

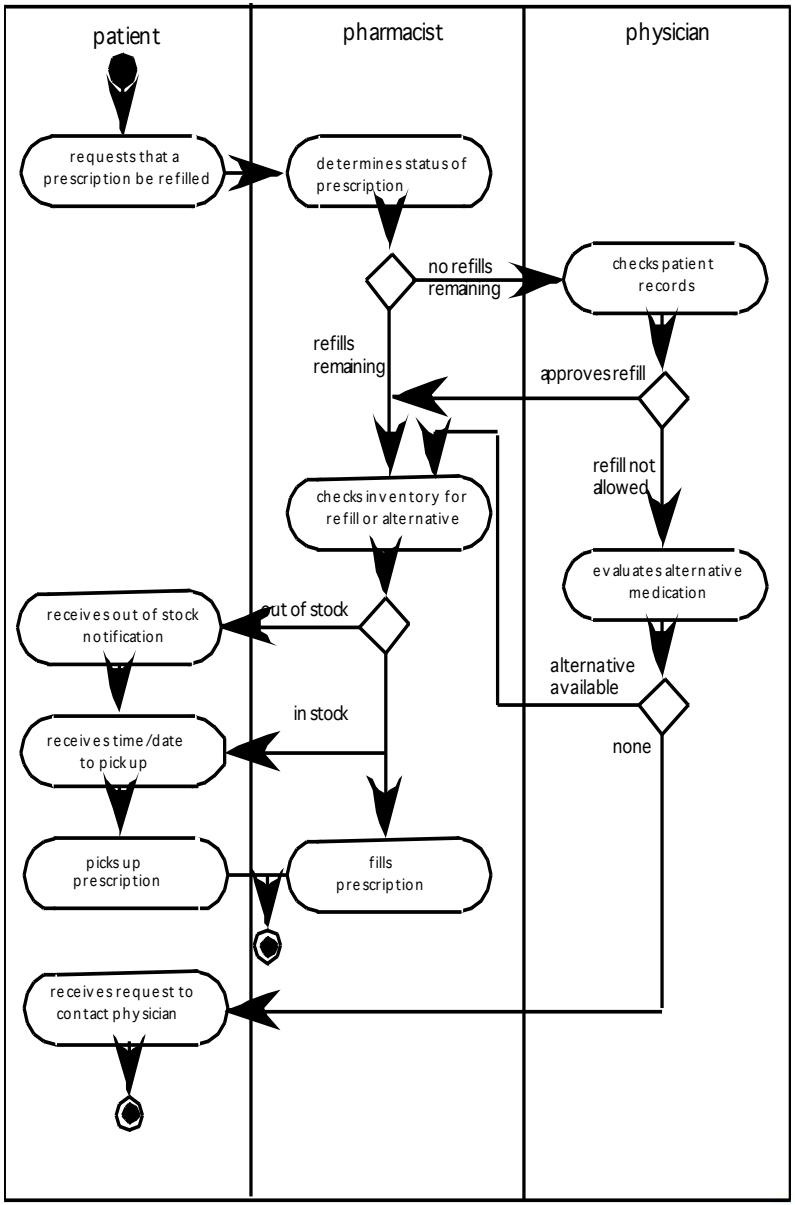


Figure 12.2 Swimlane diagram for prescription refill function

- **Analysis of Display Content**
  - Are different types of data assigned to consistent geographic locations on the screen (e.g., photos always appear in the upper right hand corner)?
  - Can the user customize the screen location for content?
  - Is proper on-screen identification assigned to all content?
  - If a large report is to be presented, how should it be partitioned for ease of understanding?
  - Will mechanisms be available for moving directly to summary information for large collections of data.
  - Will graphical output be scaled to fit within the bounds of the display device that is used?
  - How will color to be used to enhance understanding?
  - How will error messages and warning be presented to the user?

- **Interface Design Steps**
  - Using information developed during interface analysis, *define interface objects and actions (operations).*
  - *Define events (user actions)* that will cause the state of the user interface to change. Model this behavior.
  - *Depict each interface state* as it will actually look to the end-user.
  - *Indicate how the user interprets the state of the system* from information provided through the interface.

- **Design Issues**
  - Response time
  - Help facilities
  - Error handling
  - Menu and command labeling
  - Application accessibility
  - Internationalization

- **Web and Mobile App Interface Design**
  - *Where am I?* The interface should
    - provide an indication of the WebApp that has been accessed
    - inform the user of her location in the content hierarchy.
  - *What can I do now?* The interface should always help the user understand his current options
    - what functions are available?
    - what links are live?
    - what content is relevant?
  - *Where have I been, where am I going?* The interface must facilitate navigation.
    - Provide a “map” (implemented in a way that is easy to understand) of where the user has been and what paths may be taken to move elsewhere within the WebApp.

- **Effective Web and Mobile App Interfaces**
  - Bruce Tognozzi [TOG01] suggests…
    - *Effective interfaces are visually apparent and forgiving*, instilling in their users a sense of control. Users quickly see the breadth of their options, grasp how to achieve their goals, and do their work.
    - *Effective interfaces do not concern the user with the inner workings of the system.* Work is carefully and continuously saved, with full option for the user to undo any activity at any time.
    - *Effective applications and services perform a maximum of work*, while requiring a minimum of information from users.

- **Interface Design Principles - I**
  - *Anticipation*—A WebApp should be designed so that it anticipates the use's next move.
  - *Communication*—The interface should communicate the status of any activity initiated by the user
  - *Consistency*—The use of navigation controls, menus, icons, and aesthetics (e.g., color, shape, layout)
  - *Controlled autonomy*—The interface should facilitate user movement throughout the WebApp, but it should do so in a manner that enforces navigation conventions that have been established for the application.
  - *Efficiency*—The design of the WebApp and its interface should optimize the user's work efficiency, not the efficiency of the Web engineer who designs and builds it or the client-server environment that executes it.

Interface Design Principles - II

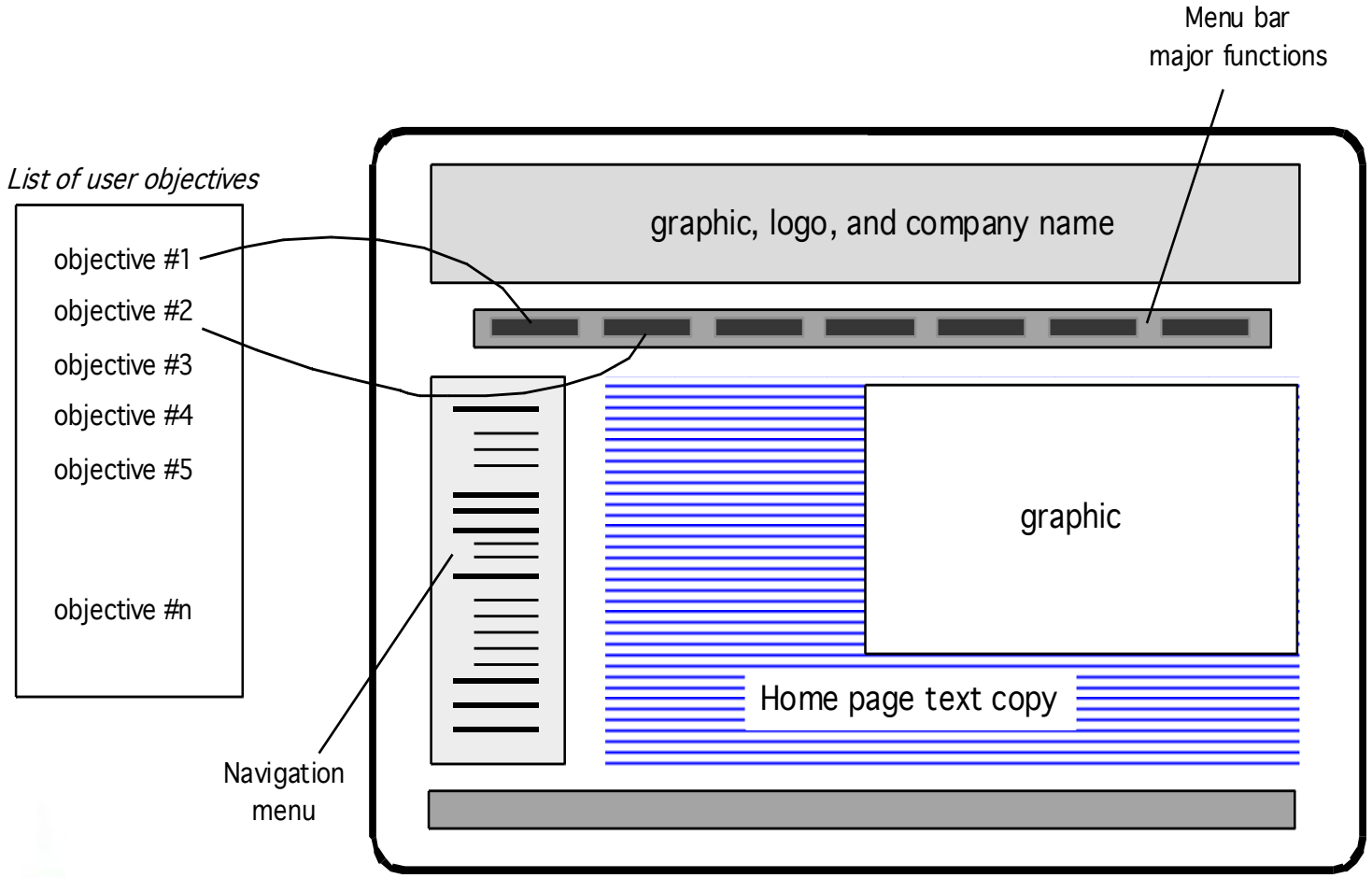
- *Focus*—The WebApp interface (and the content it presents) should stay focused on the user task(s) at hand.
- *Fitt's Law*—"The time to acquire a target is a function of the distance to and size of the target."
- *Human interface objects*—A vast library of reusable human interface objects has been developed for WebApps.
- *Latency reduction*—The WebApp should use multi-tasking in a way that lets the user proceed with work as if the operation has been completed.
- *Learnability*— A WebApp interface should be designed to minimize learning time, and once learned, to minimize relearning required when the WebApp is revisited.

Interface Design Principles - III

- *Maintain work product integrity*—A work product (e.g., a form completed by the user, a user specified list) must be automatically saved so that it will not be lost if an error occurs.
- *Readability*—All information presented through the interface should be readable by young and old.
- *Track state*—When appropriate, the state of the user interaction should be tracked and stored so that a user can logoff and return later to pick up where she left off.
- *Visible navigation*—A well-designed WebApp interface provides "the illusion that users are in the same place, with the work brought to them."

- **Interface Design Workflow - I**
  - Review information contained in the analysis model and refine as required.
  - Develop a rough sketch of the Web or Mobile App interface layout.
  - Map user objectives into specific interface actions.
  - Define a set of user tasks that are associated with each action.
  - Storyboard screen images for each interface action.
  - Refine interface layout and storyboards using input from aesthetic design.

- **Mapping User Objectives**



- **Interface Design Workflow - II**
  - Identify user interface objects that are required to implement the interface.
  - Develop a procedural representation of the user's interaction with the interface.
  - Develop a behavioral representation of the interface.
  - Describe the interface layout for each state.
  - Refine and review the interface design model.

- **Aesthetic Design**
  - Don’t be afraid of white space.
  - Emphasize content.
  - Organize layout elements from top-left to bottom right.
  - Group navigation, content, and function geographically within the page.
  - Don’t extend your real estate with the scrolling bar.
  - Consider resolution and browser window size when designing layout.

- **Design Evaluation Cycle**