User Interface Design

Ashish Kumar Dwivedi

Introduction

- User interface design creates an effective communication medium between a human and a computer.
- Following a set of interface design principles, design identifies interface objects and actions and then creates a screen layout that forms the basis for a user interface prototype.
- A software engineer designs the user interface by applying an iterative process that draws on predefined design principles.

Why UI?

- If software is difficult to use, if it forces you into mistakes, or if it frustrates your efforts to accomplish your goals, you won't like it, regardless of the computational power it exhibits, the content it delivers, or the functionality it offers.
- The interface has to be right because it molds a user's perception of the software.
- User interface design begins with the identification of user, task, and environmental requirements.
- Once user tasks have been identified, user scenarios are created and analyzed to define a set of interface objects and actions.
- Tools are used to prototype and ultimately implement the design model, and the result is evaluated for quality.

Golden Rules

1. 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**
   - **Hide technical internals from the casual user**
   - **Design for direct interaction with objects that appear on the screen**
2. Reduce the user's memory load.
   - **Reduce demand on short-term memory**
   - **Establish meaningful defaults**
   - **Define shortcuts that are intuitive**
3. 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.**

UI Design Principles

Structure Simplicity Visibility Feedback Tolerance

Usability

In an insightful paper on usability, Larry Constantine [Con95] asks a question that has significant bearing on the subject: "What do users want, anyway?" He answers this way:

> What users really want are good tools. All software systems, from operating systems and languages to data entry and decision support applications, are just tools. End users want from the tools we engineer for them much the same as we expect from the tools we use. They want systems that are easy to learn and that help them do their work. They want software that doesn't slow them down, that doesn't trick or confuse them, that doesn't make it easier to make mistakes or harder to finish the job.

Constantine argues that usability is not derived from aesthetics, state-of-the-art interaction mechanisms, or built-in interface intelligence. Rather, it occurs when the architecture of the interface fits the needs of the people who will be using it.

A formal definition of usability is somewhat illusive. Donahue and his colleagues [Don99] define it in the following manner: "Usability is a measure of how well a computer system . . . facilitates learning; helps learners remember what they've learned; reduces the likelihood of errors; enables them to be efficient, and makes them satisfied with the system."

The only way to determine whether "usability" exists within a system you are building is to conduct usability assessment or testing. Watch users interact with the system and answer the following questions [Con95]:

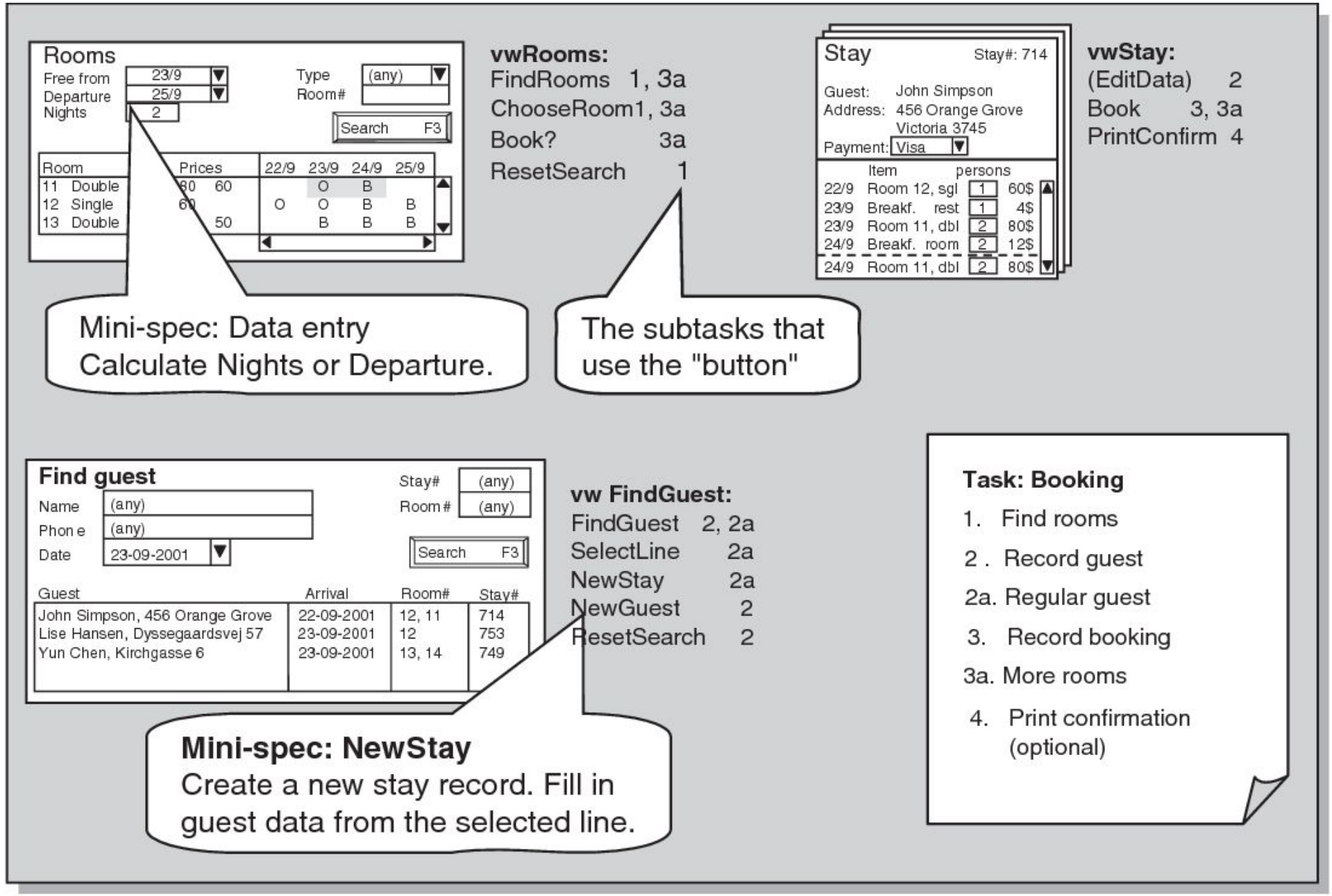
- Is the system usable without continual help or instruction?
- Do the rules of interaction help a knowledgeable user to work efficiently?
- Do interaction mechanisms become more flexible as users become more knowledgeable?
- Has the system been tuned to the physical and social environment in which it will be used?
- Is the user aware of the state of the system? Does the user know where she is at all times?
- Is the interface structured in a logical and consistent manner?
- Are interaction mechanisms, icons, and procedures consistent across the interface?
- Does the interaction anticipate errors and help the user correct them?
- Is the interface tolerant of errors that are made?
- Is the interaction simple?

If each of these questions is answered "yes," it is likely that usability has been achieved.

Among the many measurable benefits derived from a usable system are [Don99]: increased sales and customer satisfaction, competitive advantage, better reviews in the media, better word of mouth, reduced support costs, improved end-user productivity, reduced training costs, reduced documentation costs, reduced likelihood of litigation from unhappy customers.

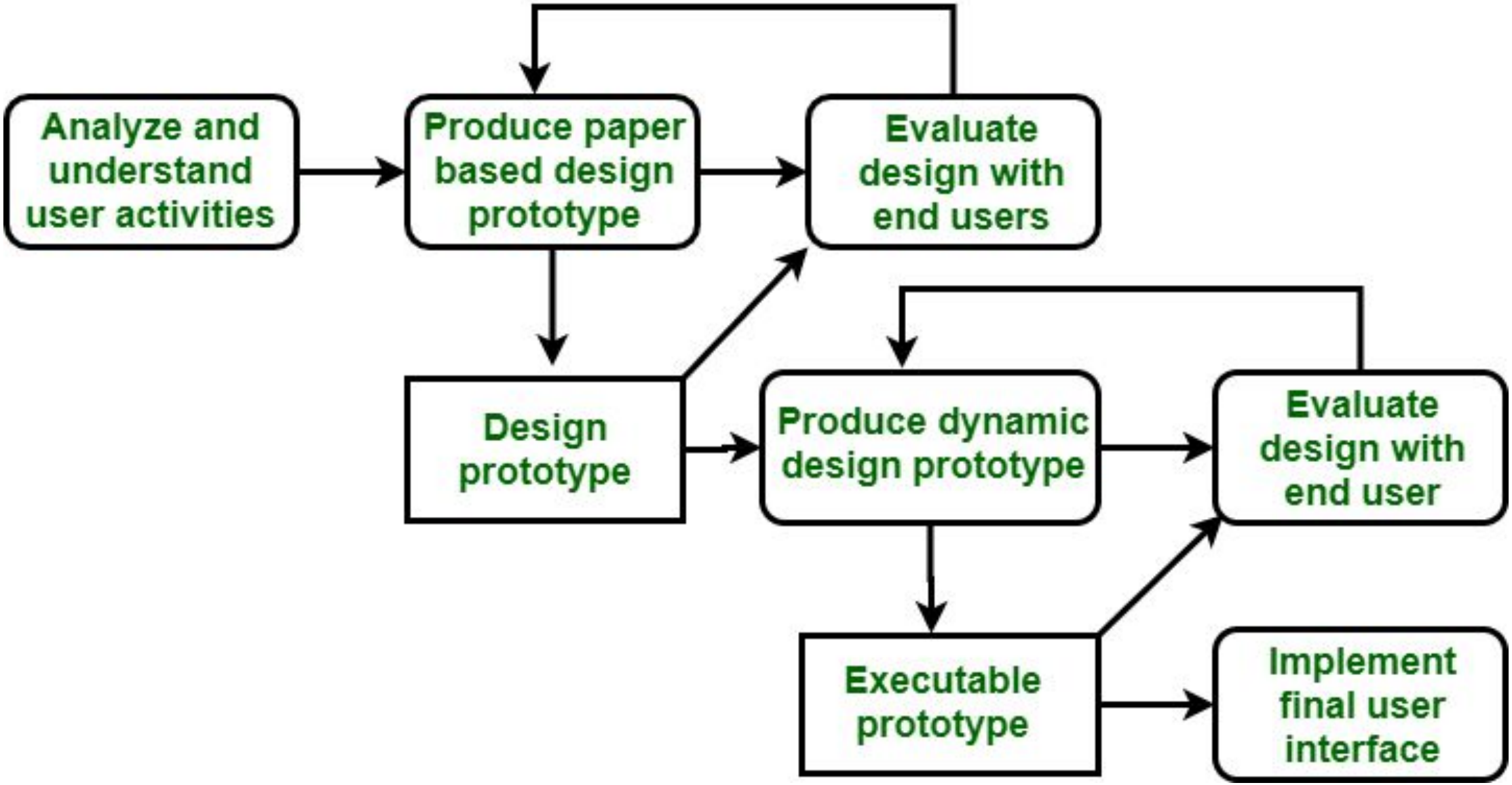
UI Analysis and Design

- Four different models come into play when a user interface is to be analyzed and designed.
- A human engineer (or the software engineer) establishes a *user model,* the software engineer creates a *design model,* the end user develops a mental image that is often called the user's *mental model* or the *system perception,* and the implementers of the system create an *implementation model*.
- Unfortunately, each of these models may differ significantly. Your role, as an interface designer, is to reconcile these differences and derive a consistent representation of the interface.
- users can be categorized as: *Novices*, *Knowledgeable, intermittent users*, *Knowledgeable, frequent users*

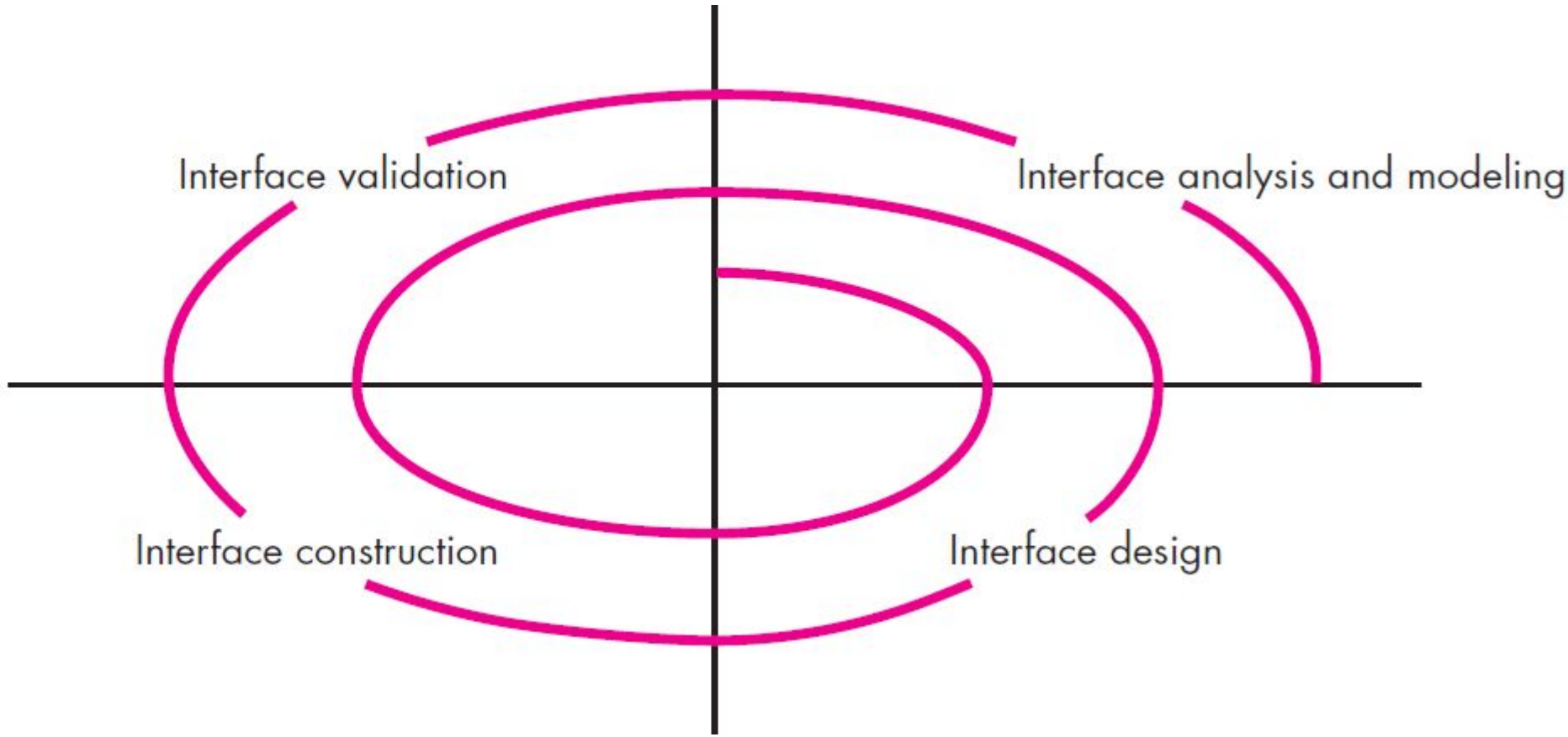

Rooms
Free from 23/9
Departure 25/9
Nights 2
Type (any)
Room#
Search F3
Room
Prices
22/9 23/9 24/9 25/9
11 Double 80 60 O B
12 Single 60 O O B B
13 Double 50 B B B
vwRooms:
FindRooms 1, 3a
ChooseRoom1, 3a
Book? 3a
ResetSearch 1
Stay
Stay#: 714
Guest: John Simpson
Address: 456 Orange Grove
Victoria 3745
Payment: Visa
Item persons
22/9 Room 12, sgl 1 60$
23/9 Breakf. rest 1 4$
23/9 Room 11, dbl 2 80$
24/9 Breakf. room 2 12$
24/9 Room 11, dbl 2 80$
vwStay:
(EditData) 2
Book 3, 3a
PrintConfirm 4
Mini-spec: Data entry
Calculate Nights or Departure.
The subtasks that
use the "button"
Find guest
Name (any)
Phon e (any)
Date 23-09-2001
Stay# (any)
Room # (any)
Search F3
Guest
Arrival
Room#
Stay#
John Simpson, 456 Orange Grove 22-09-2001 12, 11 714
Lise Hansen, Dyssegaardsvej 57 23-09-2001 12 753
Yun Chen, Kirchgasse 6 23-09-2001 13, 14 749
vw FindGuest:
FindGuest 2, 2a
SelectLine 2a
NewStay 2a
NewGuest 2
ResetSearch 2
Mini-spec: NewStay
Create a new stay record. Fill in
guest data from the selected line.
Task: Booking
1. Find rooms
2 . Record guest
2a. Regular guest
3. Record booking
3a. More rooms
4. Print confirmation
(optional)

Model

- The user's *mental model* (system perception) is the image of the system that end users carry in their heads. For example, if the user of a particular word processor were asked to describe its operation, the system perception would guide the response.
- The *implementation model* combines the outward manifestation of the computer-based system (the look and feel of the interface), coupled with all supporting information (books, manuals, videotapes, help files) that describes interface syntax and semantics.


Analyze and understand user activities
Produce paper based design prototype
Evaluate design with end users
Design prototype
Produce dynamic design prototype
Evaluate design with end user
Executable prototype
Implement final user interface

The user interface design process



Process

- The user interface analysis and design process begins at the interior of the spiral and encompasses four distinct framework activities:
- (1) interface analysis and modeling, (2) interface design, (3) interface construction, and (4) interface validation.
- *Interface analysis* focuses on the profile of the users who will interact with the system.
- Skill level, business understanding, and general receptiveness to the new system are recorded; and different user categories are defined. For each user category, requirements are elicited.
- The construction activity involves prototyping—the only practical way to validate what has been designed.
- Finally, analysis of the user environment focuses on the physical work environment. Among the questions to be asked are:
  - Where will the interface be located physically?
  - Will the user be sitting, standing, or performing other tasks unrelated to the interface?
  - Does the interface hardware accommodate space, light, or noise constraints?
  - Are there special human factors considerations driven by environmental factors?

Process

- The goal of *interface design* is to define a set of interface objects and actions (and their screen representations) that enable a user to perform all defined tasks in a manner that meets every usability goal defined for the system.
- *Interface construction* normally begins with the creation of a prototype that enables usage scenarios to be evaluated.
- *Interface validation* focuses on:
- (1) the ability of the interface to implement every user task correctly, to accommodate all task variations, and to achieve all general user requirements;
- (2) the degree to which the interface is easy to use and easy to learn, and
- (3) the users' acceptance of the interface as a useful tool in their work.

User Analysis

- The phrase “user interface” is probably all the justification needed to spend some time understanding the user before worrying about technical matters.
- **User Interviews.** The most direct approach, members of the software team meet with end users to better understand their needs, motivations, work culture, and a myriad of other issues.
- **Sales input.** Sales people meet with users on a regular basis and can gather information that will help the software team to categorize users and better understand their requirements.
- **Marketing input.** Market analysis can be invaluable in the definition of market segments and an understanding of how each segment might use the software in subtly different ways.
- **Support input.** Support staff talks with users daily. They are the most likely source of information on what works and what doesn’t, what users like and what they dislike, what features generate questions and what features are easy to use.

Task Analysis and Modeling

- The goal of task analysis is to answer 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?

Interface Design Steps

The following 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 behaviour.
- 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.

The interface design evaluation cycle