Mensch-Maschine-Interaktion 1

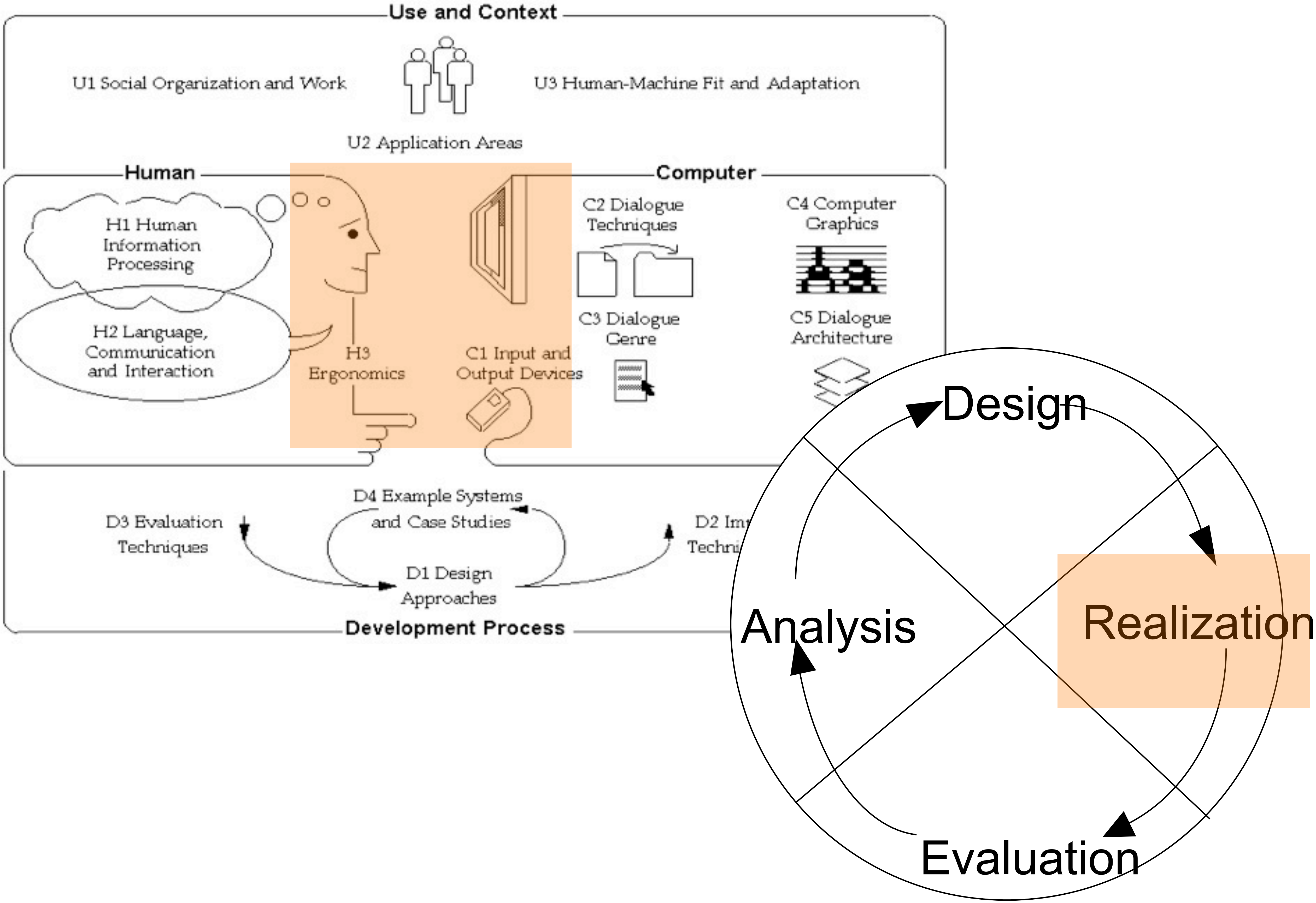
Chapter 8:

Implementing Interactive Systems

Slides partially based on material by Albrecht Schmidt + Paul Holleis

Implementing Interactive Systems

- Designing Look-And-Feel
- Implementation Technologies for Interactive Systems
- Standards and Guidelines
- Software Development Process Models
- User-Centered Development
- Integrating Usability into the Development Process

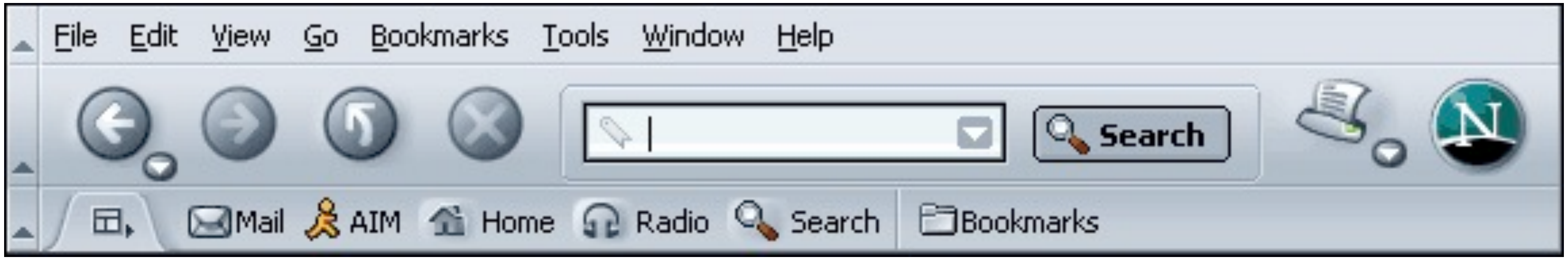

Use and Context
U1 Social Organization and Work
U3 Human-Machine Fit and Adaptation
U2 Application Areas
Human
Computer
H1 Human Information Processing
H2 Language, Communication and Interaction
H3 Ergonomics
C1 Input and Output Devices
C2 Dialogue Techniques
C3 Dialogue Genre
C4 Computer Graphics
C5 Dialogue Architecture
D4 Example Systems and Case Studies
D3 Evaluation Techniques
D1 Design Approaches
D2 Im
Techni
Development Process
Design
Analysis
Realization
Evaluation

Visual Design

- Visual Arts versus Visual Design
  - Goal of the artist: to create an observable artefact that provokes an aesthetic response (kind of self-expression)
  - Goal of the designer: to find the representation that is best suited to the communication of some specific information (oriented towards the goals of other people)
- Graphic Design and Visual Interface Design
  - Aesthetic concerns placed within the constraints of a functional framework
  - Designers working on interfaces needs to understand
    - » colour, typography, form, composition, …
    - **» and** interaction, behaviour, human capabilities, ...
- Industrial Design and Interface Design
  - New relationship coming up as more physical artefacts become software-enabled
  - have to work increasingly hand-in-hand

Source: A. Cooper

Some Principles of Visual Design

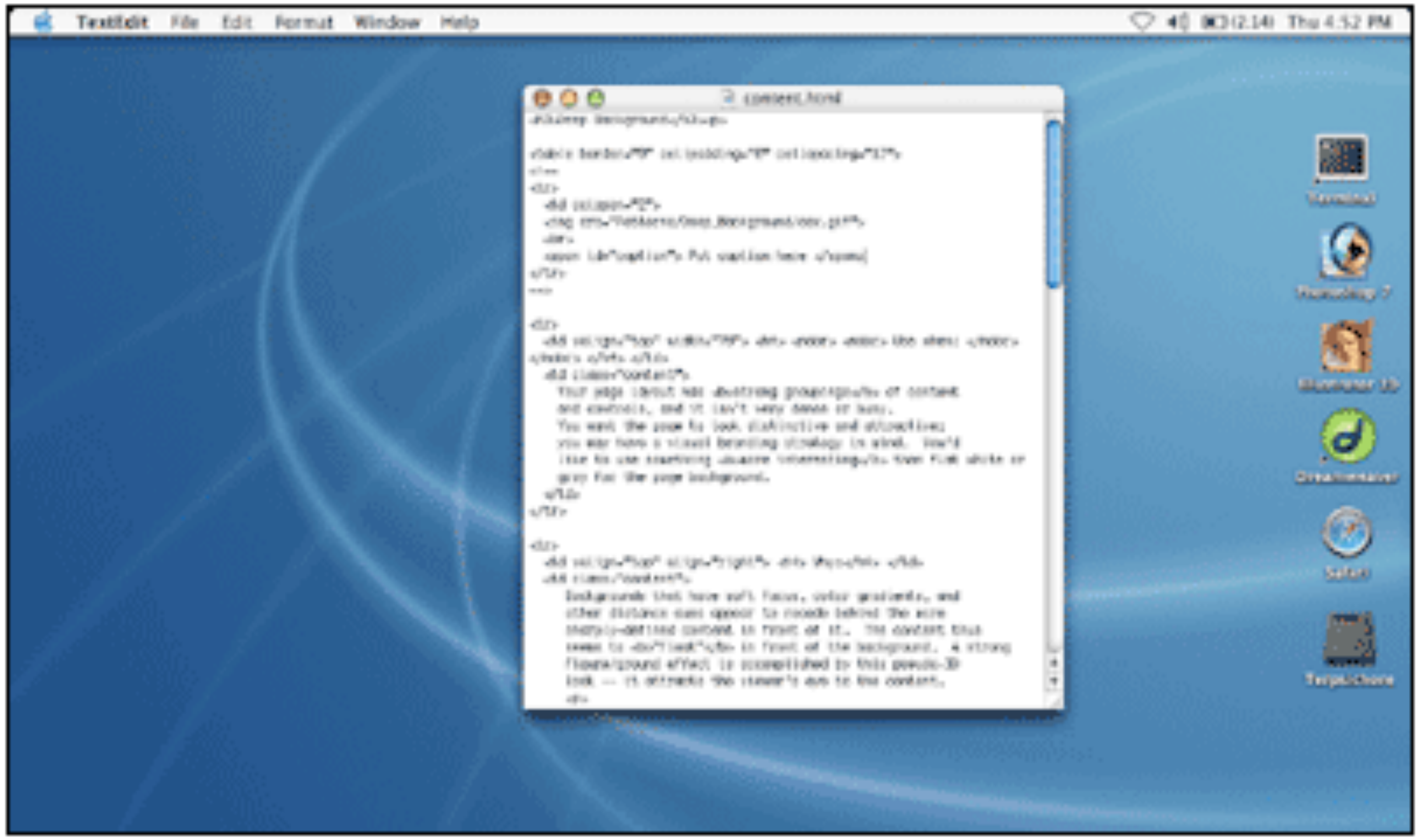


- Avoid visual noise and clutter
  - No superfluous elements that distract the user
- Use contrast, similarity and layering to distinguish and organize elements *(visual patterns)*
  - Dimensional contrast (depth)
  - Layering
  - Figure and ground
- Provide visual structure and flow at each level of organization
- Use cohesive, consistent and contextually appropriate imagery
- Integrate style and function comprehensively and purposefully
  - Form and function, branding

Based on Mullet/Sano 1995

Pattern: Deep Background (Tidwell)

Deep Background



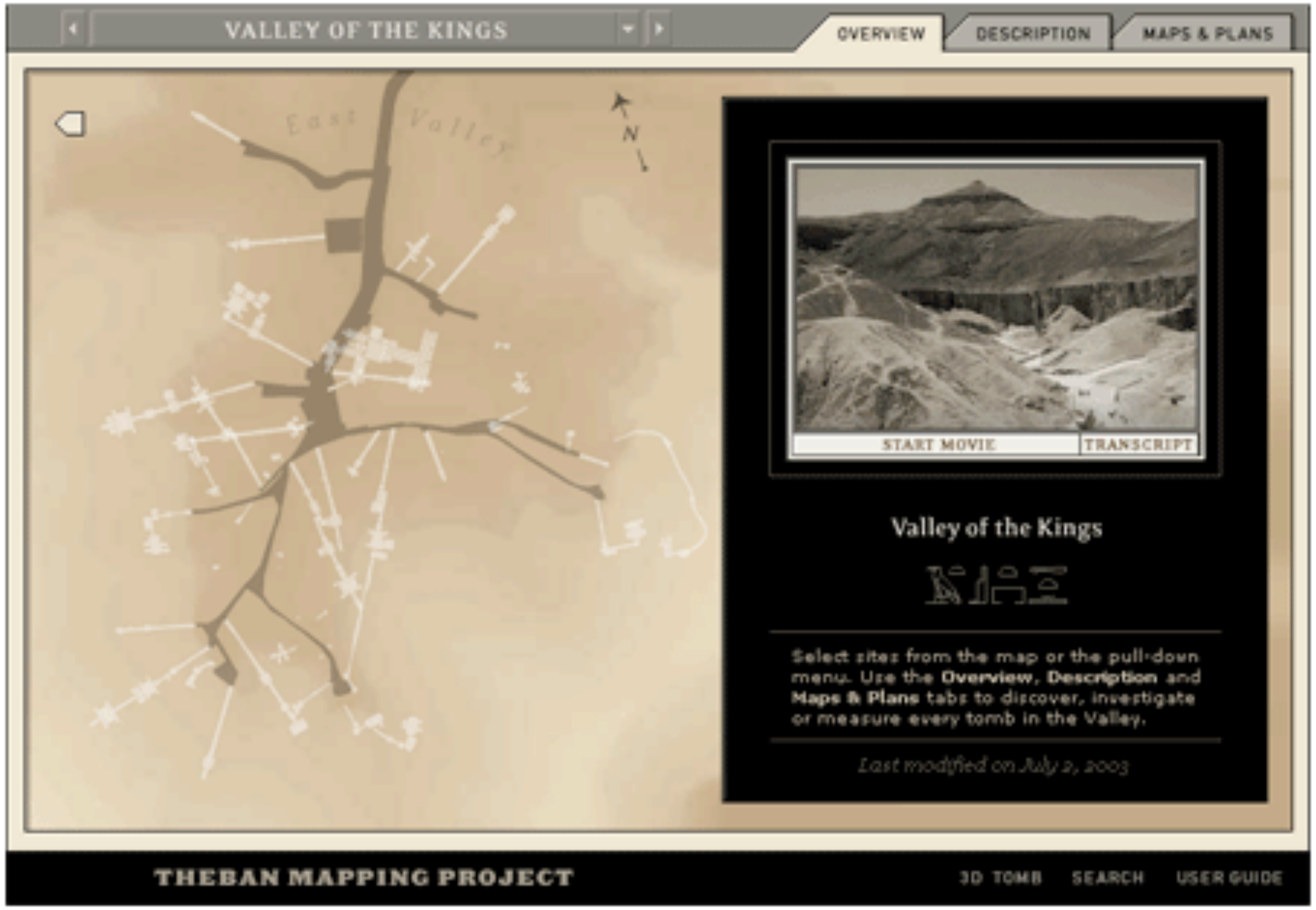
```
<h1>Deep Background</h1><p>

<table border="0" cellpadding="0" cellspacing="12">
<!--
<tr>
  <td colspan="2">
  <img src="Patterns/Deep_Background/osx.gif">
  <br>
  <span id="caption"> Put caption here </span>
</tr>
-->

<tr>
  <td valign="top" width="79"> <h4> <nobr> <nobr> Use when: </nobr> </nobr> </h4> </td>
  <td class="content">
    Your page layout has <b>strong groupings</b> of content
    and controls, and it isn't very dense or busy.
    You want the page to look distinctive and attractive;
    you may have a visual branding strategy in mind.  You'd
    like to use something <b>more interesting</b> than flat white or
    gray for the page background.
  </td>
</tr>

<tr>
  <td valign="top" align="right"> <h4> Why</h4> </td>
  <td class="content">
     Backgrounds that have soft focus, color gradients, and
     other distance cues appear to recede behind the more
     sharply-defined content in front of it.  The content thus
     seems to <b>"float"</b> in front of the background.  A strong
     figure/ground effect is accomplished by this pseudo-3D
     look -- it attracts the viewer's eye to the content.
     <p>
```

From Mac OS/X

What: Place an image or gradient into the page's background that visually recedes behind the foreground elements.

Pattern: Few Hues, Many Values (Tidwell)

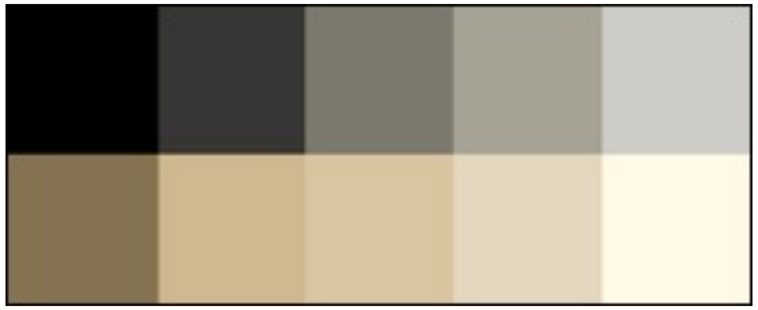
Few Hues, Many Values



Valley of the Kings

Select sites from the map or the pull-down menu. Use the **Overview**, **Description** and **Maps & Plans** tabs to discover, investigate or measure every tomb in the Valley.

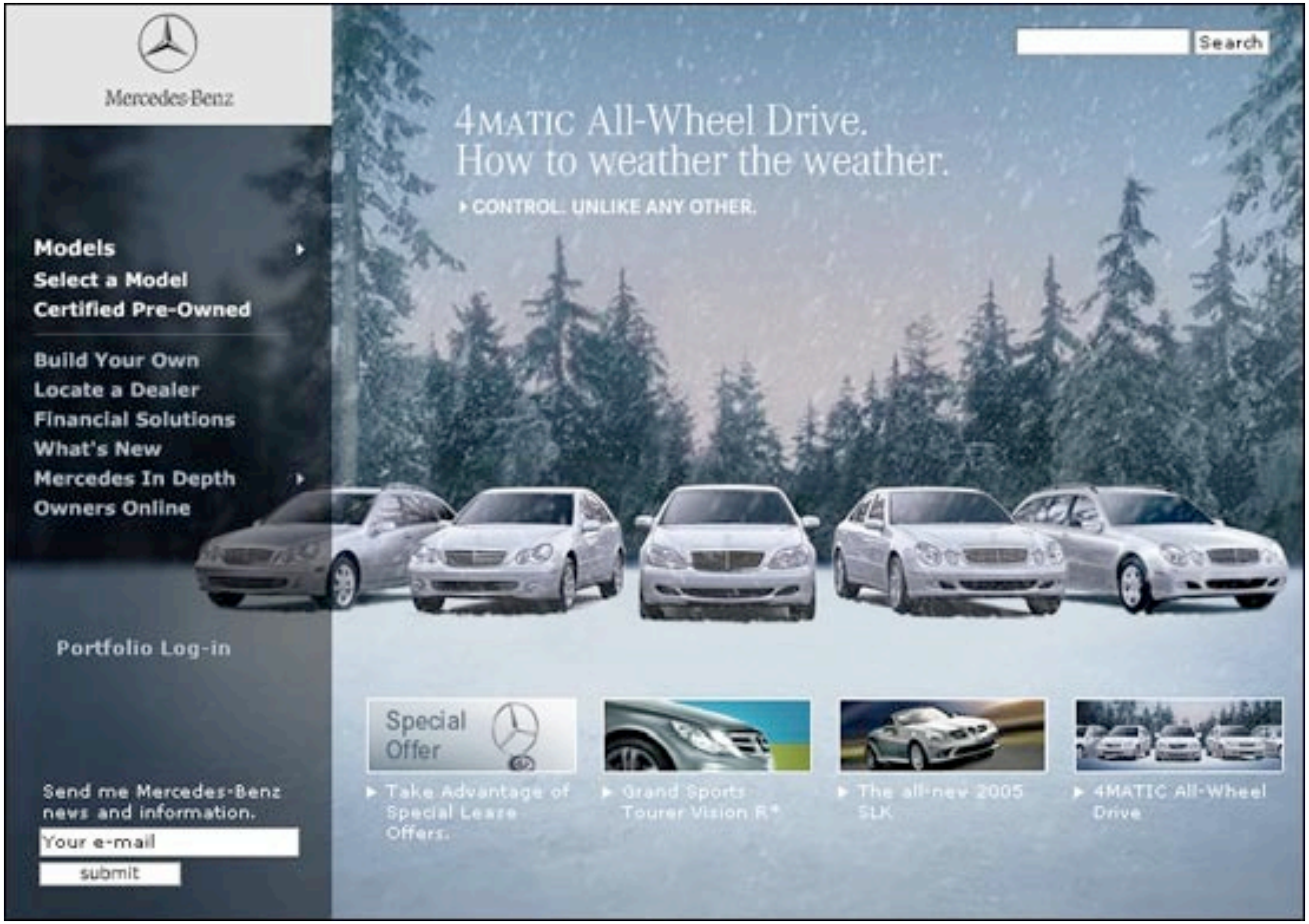
Last modified on July 2, 2003



From http://thebanmappingproject.org

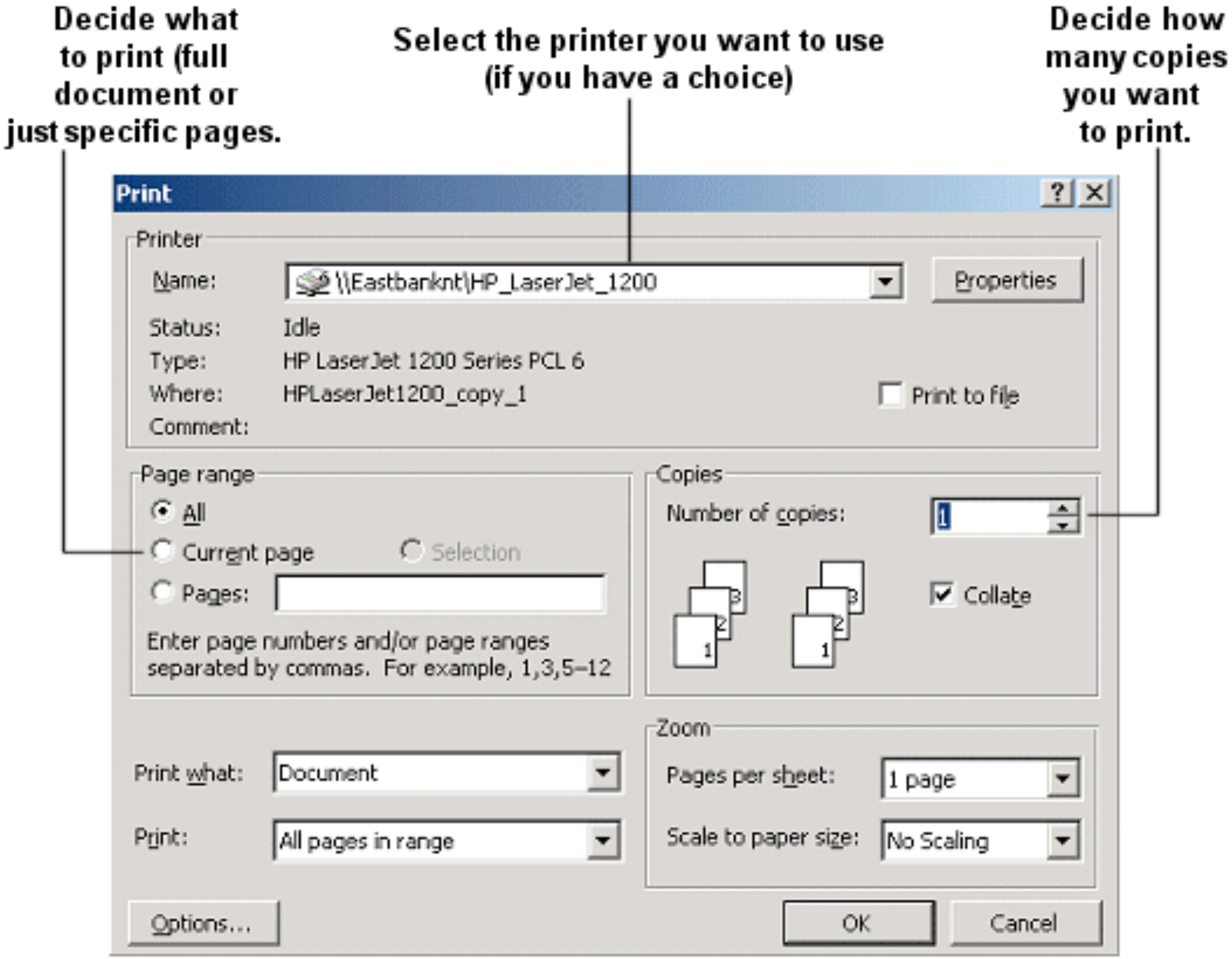
What: Choose one, two, or at most three major color hues to use in the interface. Create a color palette by selecting assorted values (brightnesses) from within those few hues.

Example: Layering



Source: Tidwell

Example: Visual Flow



Decide what to print (full document or just specific pages.

Select the printer you want to use (if you have a choice)

Decide how many copies you want to print.

Printer

Name: \\Eastbanknt\HP_LaserJet_1200

Status: Idle

Type: HP LaserJet 1200 Series PCL 6

Where: HPLaserJet1200_copy_1

Comment:

□ Print to file

Page range

● All

○ Current page

○ Selection

○ Pages:

Enter page numbers and/or page ranges separated by commas. For example, 1,3,5–12

Copies

Number of copies: 1

☑ Collate

Print what: Document

Print: All pages in range

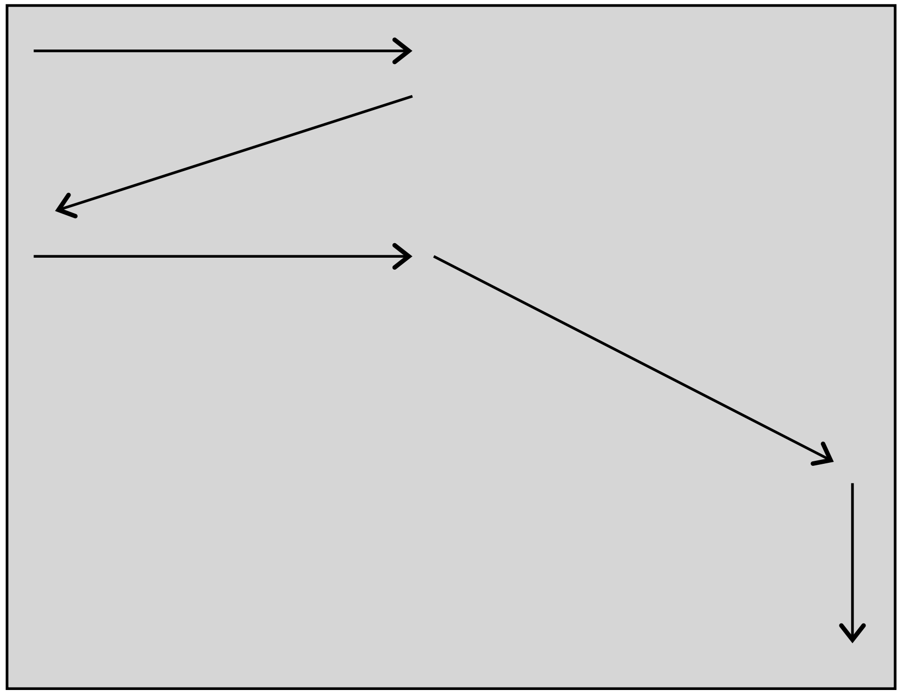
Zoom

Pages per sheet: 1 page

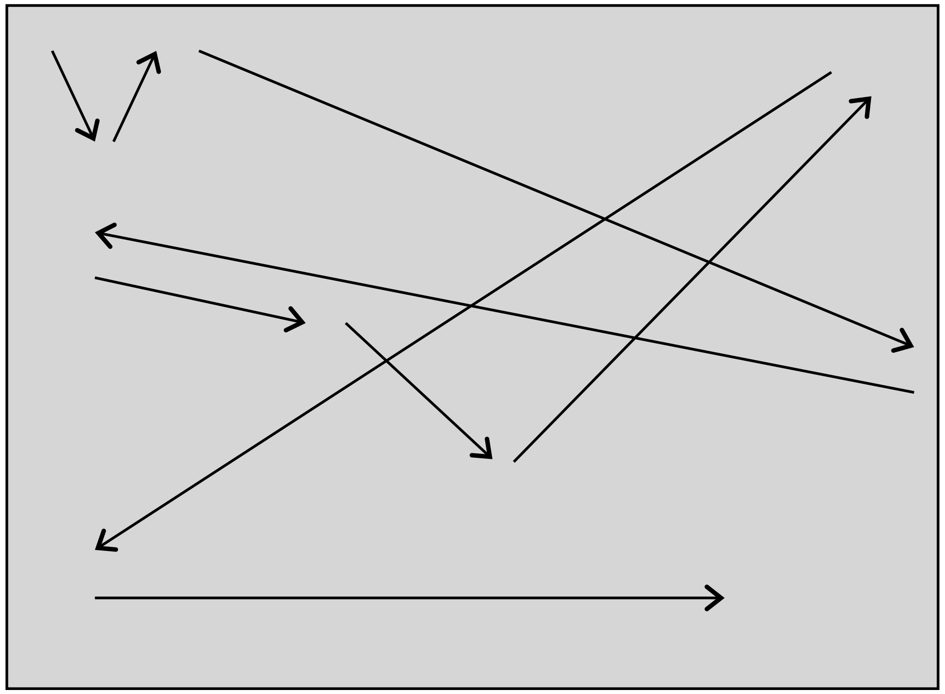
Scale to paper size: No Scaling

Grid
Group boxes

Good and Bad Logical Flow



Eye movements match the logical path through The interface

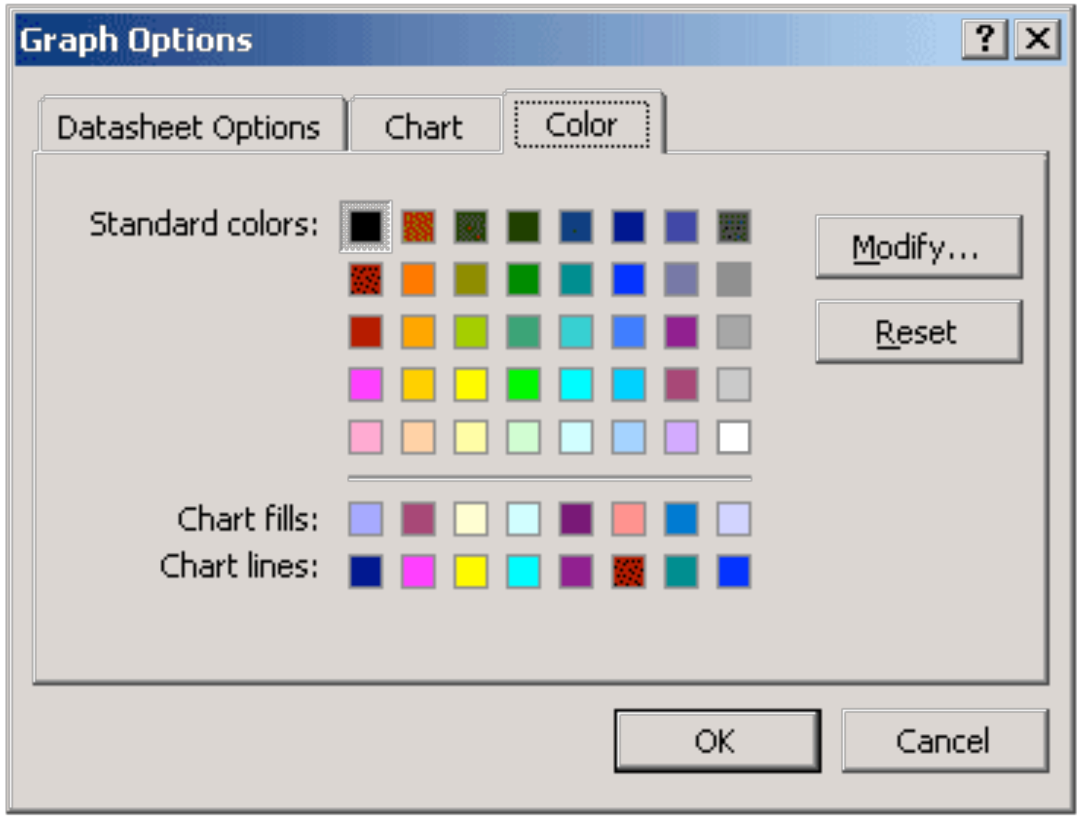


Everything is all over the place

Symmetry and Balance

- Symmetry gives interfaces a solid, stable look
- Balance of visual weights in asymmetric design

Diagonal Balance



Word's Graph Options dialog box

Tidwell:

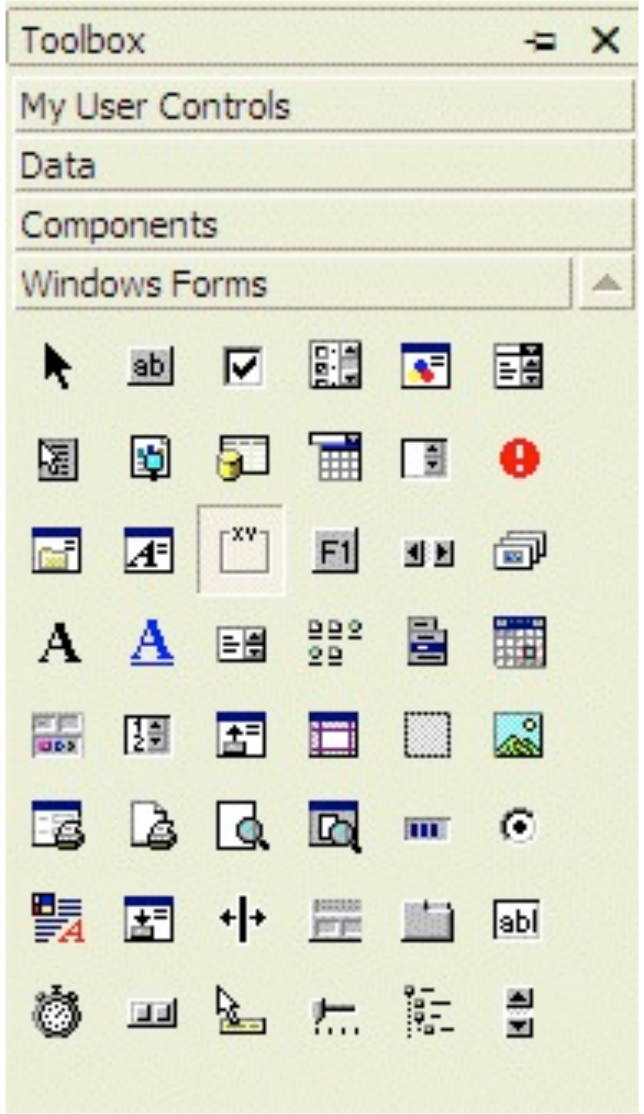
What: Arrange page elements in an asymmetric fashion, but balance it by putting visual weight into both the upper-left and lower-right corners.

Implementing Interactive Systems

- Designing Look-And-Feel
- Implementation Technologies for Interactive Systems
- Standards and Guidelines
- Software Development Process Models
- User-Centered Development
- Integrating Usability into the Development Process

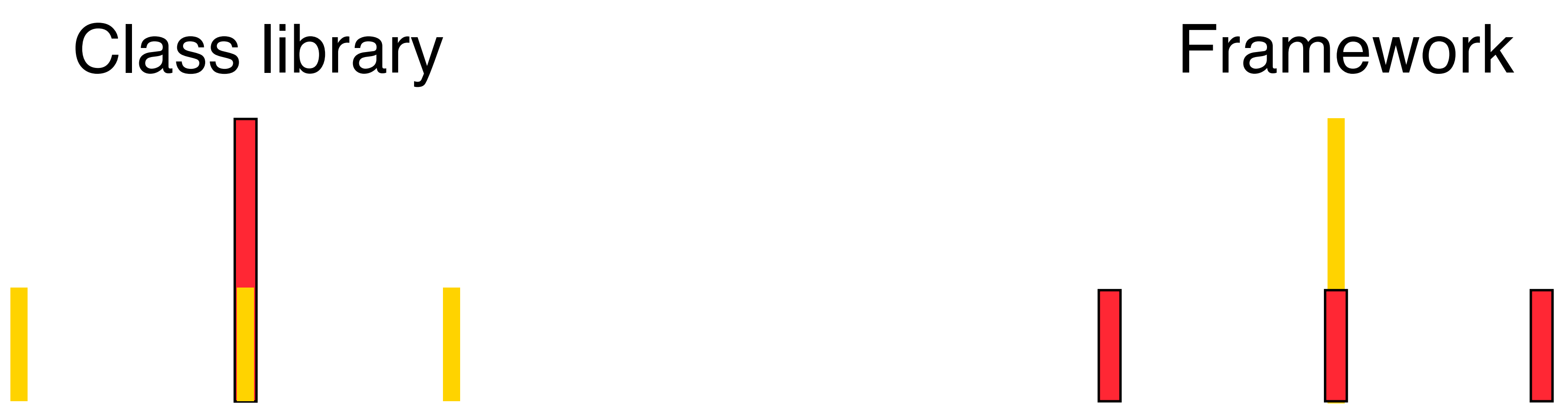
User Interface Toolkits

- Various forms:
  - Libraries
  - Frameworks
  - (Visual) components *(widgets)*
- Dependencies on
  - Programming language
  - Development tool (in particular for visual components)
  - Operating system
- Examples:
  - Java AWT & Swing
  - Microsoft MFC (C++, Windows)
  - Windows Forms (C#, Windows)
  - Qt (C++, Unix)
  - Cocoa (MacOS)



Visual C++

Class Library vs. Framework



Class library

Framework

🟥 Application-specific parts

🟨 Pre-fabricated parts

"Don't call us, we call you"

- A framework defines a stand-alone, executable basis for a class of applications.

- Framework:
  Application-specific code ***is called from pre-fabricated code***.
- Class library:
  Application-specific code ***calls pre-fabricated code***.

User Interface Management System (UIMS)

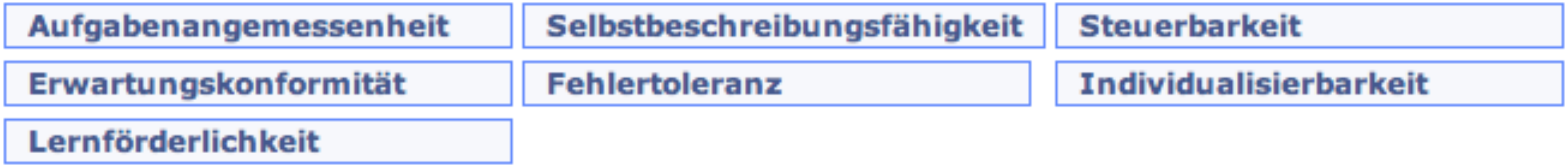
- UIMS is a term used with a wide range of meanings:
  - Conceptual architecture for the structure of an interactive system
    - » separating application logic and interface
  - Techniques for implementing application and presentation parts
    - » providing the separation but preserving the intended connection
  - Support techniques for managing a run-time interactive environment
- In the following:
  - Focus on software architecture
- Advantages of separation between presentation and application:
  - Portability
  - Reusability
  - Multiple interfaces
  - Customization of interface

Implementing Interactive Systems

- Designing Look-And-Feel
- Implementation Technologies for Interactive Systems
- Standards and Guidelines
- Software Development Process Models
- User-Centered Development
- Integrating Usability into the Development Process

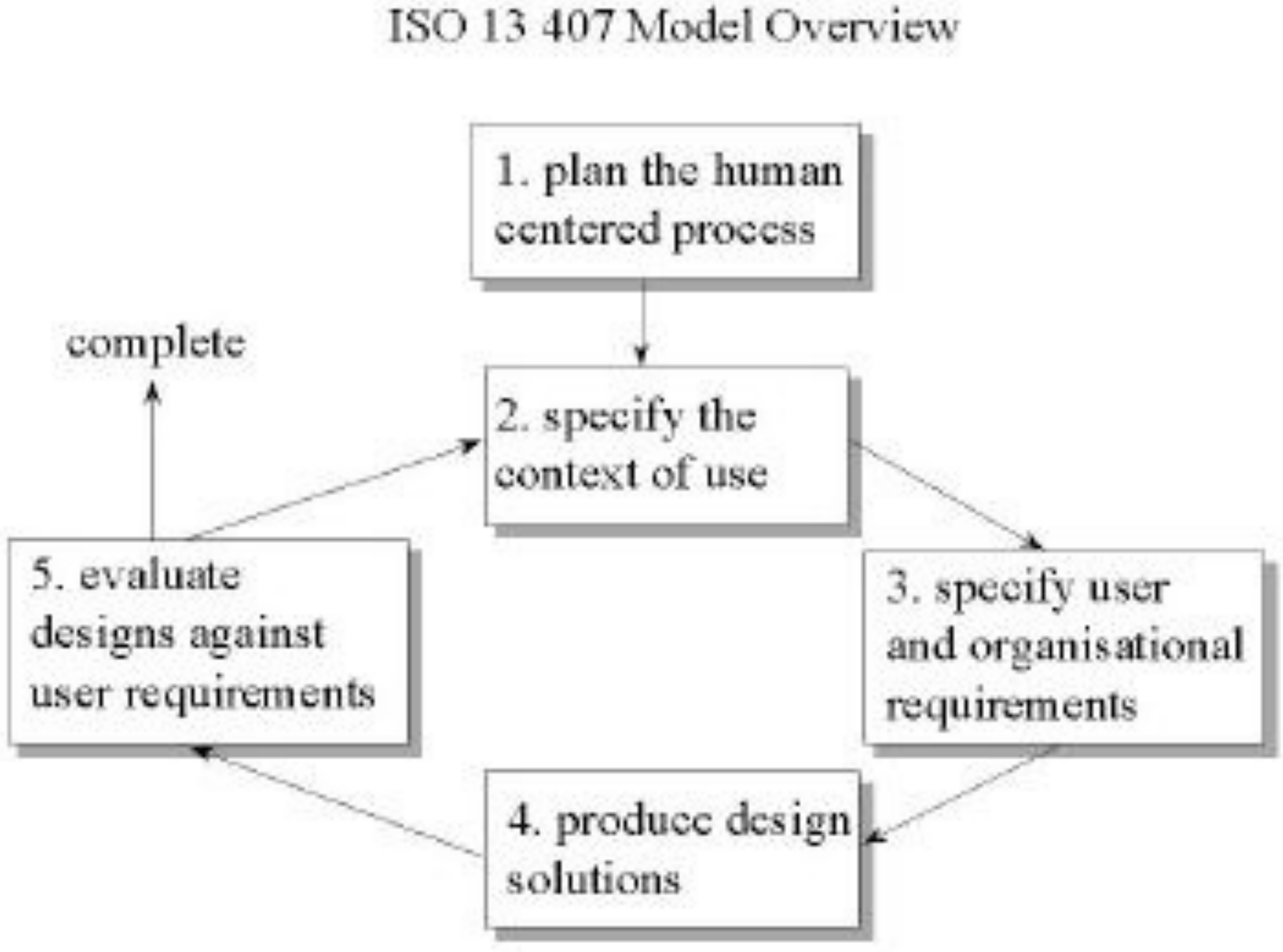
Standards (1)

- ISO 9241
  - Original title: *Ergonomic requirements for office work with visual display terminals (VDTs)*
  - New title: *Ergonomics of Human System Interaction*
  - Example: ISO 9241 Part 110 “Dialogue Principles”
    - » Suitability for the task
    - » Self-descriptiveness
    - » Controllability
    - » Conformity with user expectations
    - » Error tolerance
    - » Suitability for individualisation
    - » Suitability for learning



Standards (2)

- ISO 13407
  - Human-centered development process
  - See later in this lecture
- ISO 14915
  - Design principles for multimedia user interfaces
- ISO 16071
  - Acessibility of human-computer interfaces
- BITV
  - Barrierefreie Informationstechnik-Verordnung
- BildscharbV
  - Bildschirmarbeitsverordnung


ISO 13 407 Model Overview
1. plan the human centered process
complete
2. specify the context of use
5. evaluate designs against user requirements
3. specify user and organisational requirements
4. produce design solutions

Hix and Hartson's Guidelines (1)

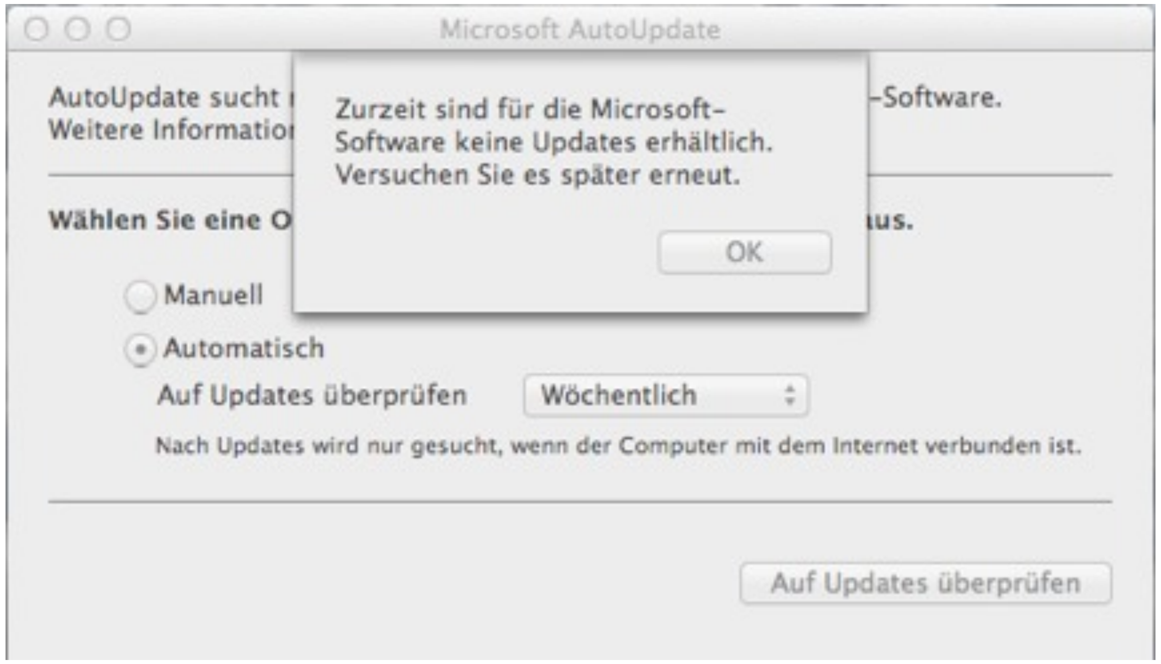
- User centered design
- Know the user
- Involve the user
- Prevent user errors
- Optimize user operation
- Keep control with the user
- Help the user to get started

- Give a task-based mental model
- Be consistent
- Keep it simple
- Design for memory limitations
- Use recognition rather than recall
- Use cognitive directness
- Draw on real world analogies

(Hix and Hartson, Developing User Interfaces, Wiley, 1993)

Hix and Hartson’s Guidelines (2)

- Use informative feedback
- Give status indicators
- Use user-centred wording
- Use non-threatening wording
- Use specific constructive advice
- Make the system take the blame
- Do not anthropomorphise



AutoUpdate sucht … -Software.
Weitere Informatio…

Zurzeit sind für die Microsoft-Software keine Updates erhältlich. Versuchen Sie es später erneut.

Wählen Sie eine O… …us.

○ Manuell
● Automatisch

Auf Updates überprüfen: Wöchentlich

Nach Updates wird nur gesucht, wenn der Computer mit dem Internet verbunden ist.

- Use modes cautiously
- Make user action reversible
- Get attention judiciously
- Maintain display inertia
- Organize screen to manage complexity
- Accommodate individual difference

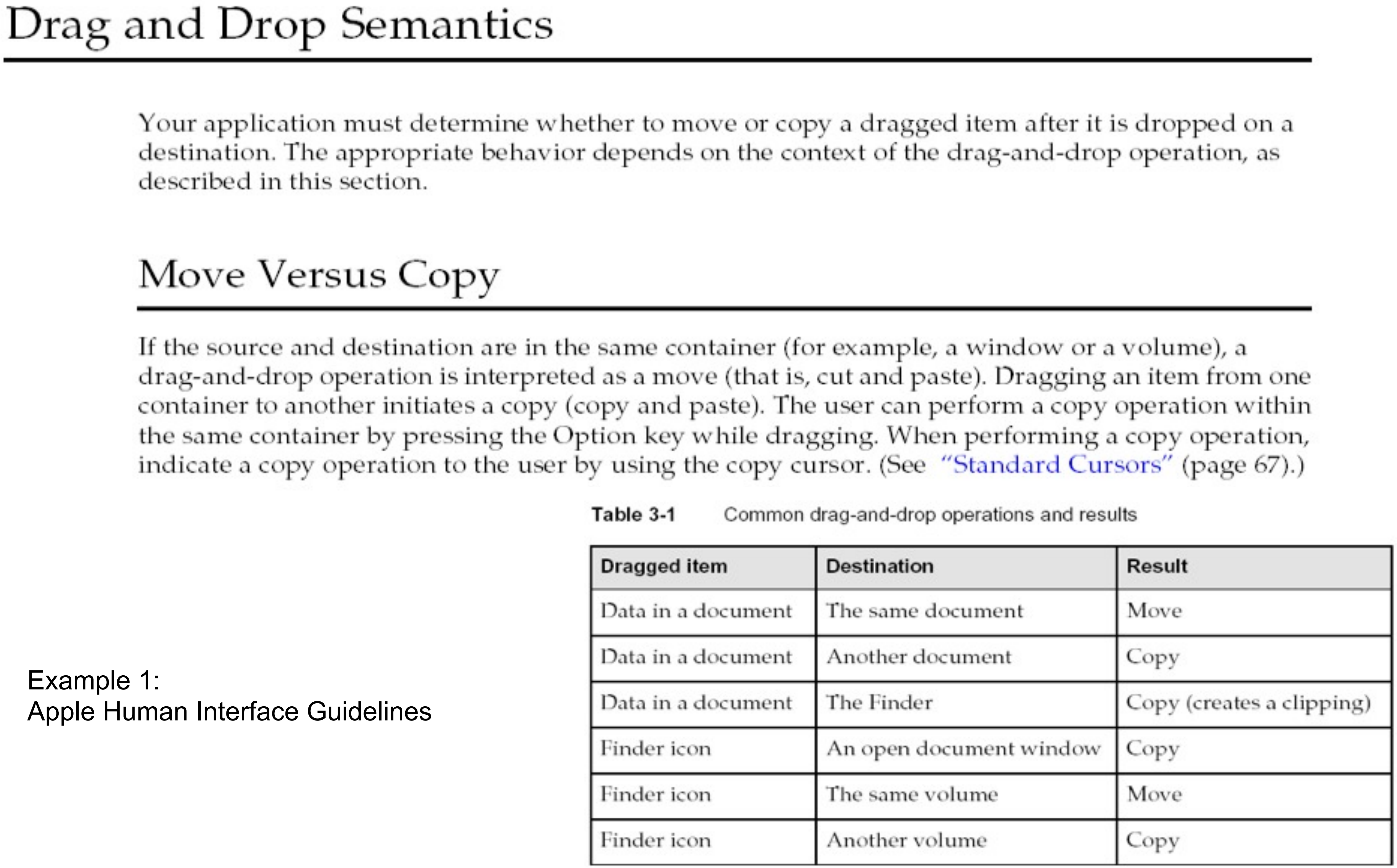
(Hix and Hartson, Developing User Interfaces, Wiley, 1993)

GNOME Guideline

- 1. Usability Principles
  - Design for People
  - Don't Limit Your User Base
  - Accessibility
  - Internationalization and Localization
  - Create a Match Between Your Application and the Real World
  - Make Your Application Consistent
  - Keep the User Informed
  - Keep It Simple and Pretty
  - Put the User in Control
  - Forgive the User
  - Provide Direct Manipulation
- 2. Desktop Integration
  - Placing Entries in the Applications Menu
  - Menu Item Names
  - …

https://developer.gnome.org/hig-book/

- 3. Windows
  - Titles
  - …
  - Layout
  - Common Dialogs
- 4. Menus
  - The Menubar
  - Types of Menu
  - Drop-down Menus
  - …
  - Help
- 5. Toolbars
  - Appearance and Content
  - …
- 6. Controls
  - …
  - Sliders
  - Buttons
  - Check Boxes
  - …



Drag and Drop Semantics

Your application must determine whether to move or copy a dragged item after it is dropped on a destination. The appropriate behavior depends on the context of the drag-and-drop operation, as described in this section.

Move Versus Copy

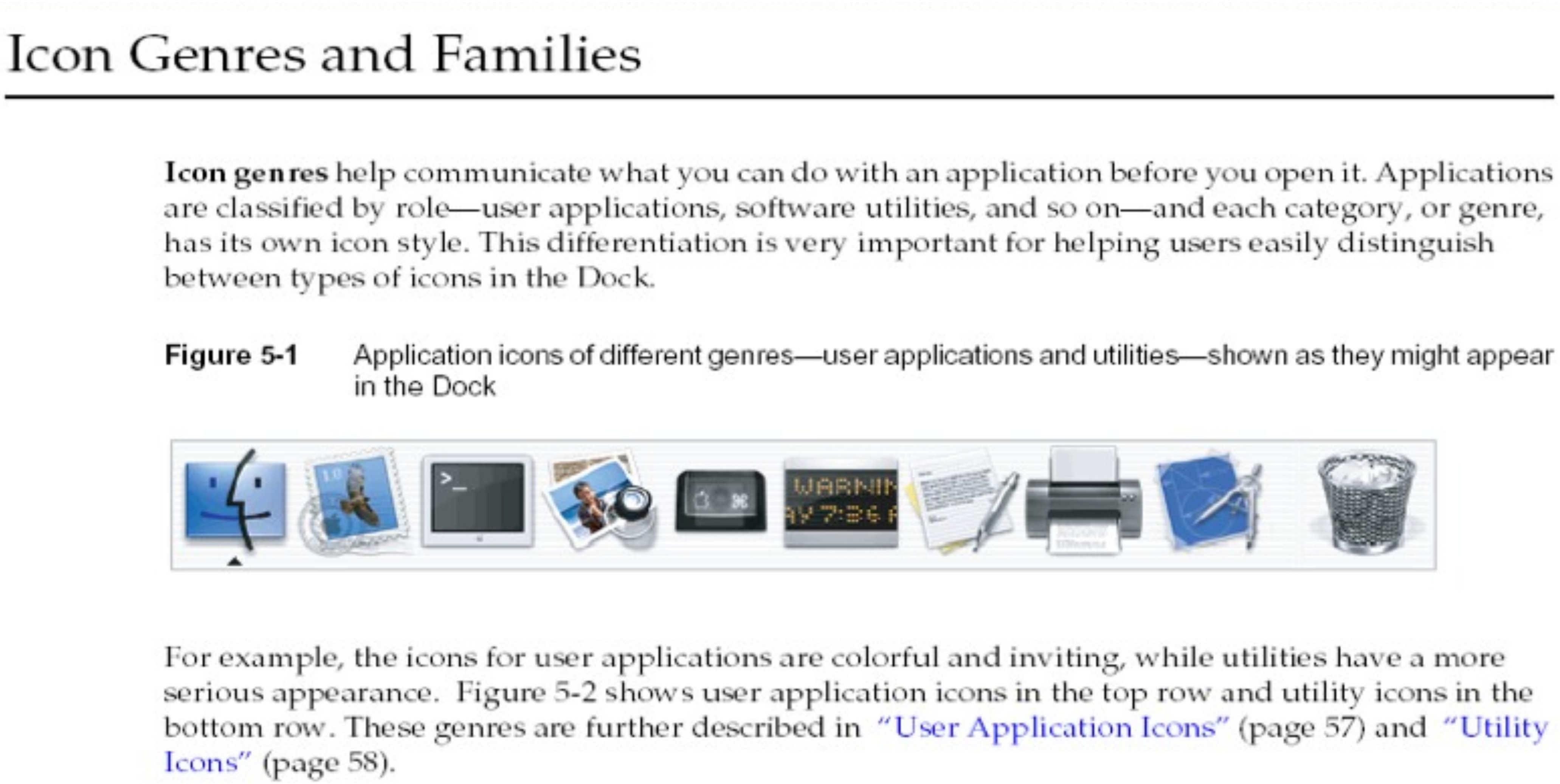
If the source and destination are in the same container (for example, a window or a volume), a drag-and-drop operation is interpreted as a move (that is, cut and paste). Dragging an item from one container to another initiates a copy (copy and paste). The user can perform a copy operation within the same container by pressing the Option key while dragging. When performing a copy operation, indicate a copy operation to the user by using the copy cursor. (See "Standard Cursors" (page 67).)

Table 3-1 Common drag-and-drop operations and results

Dragged item	Destination	Result
Data in a document	The same document	Move
Data in a document	Another document	Copy
Data in a document	The Finder	Copy (creates a clipping)
Finder icon	An open document window	Copy
Finder icon	The same volume	Move
Finder icon	Another volume	Copy

Example 1:
Apple Human Interface Guidelines

http://developer.apple.com/library/mac/#documentation/UserExperience/Conceptual/AppleHIGuidelines/Intro/Intro.html



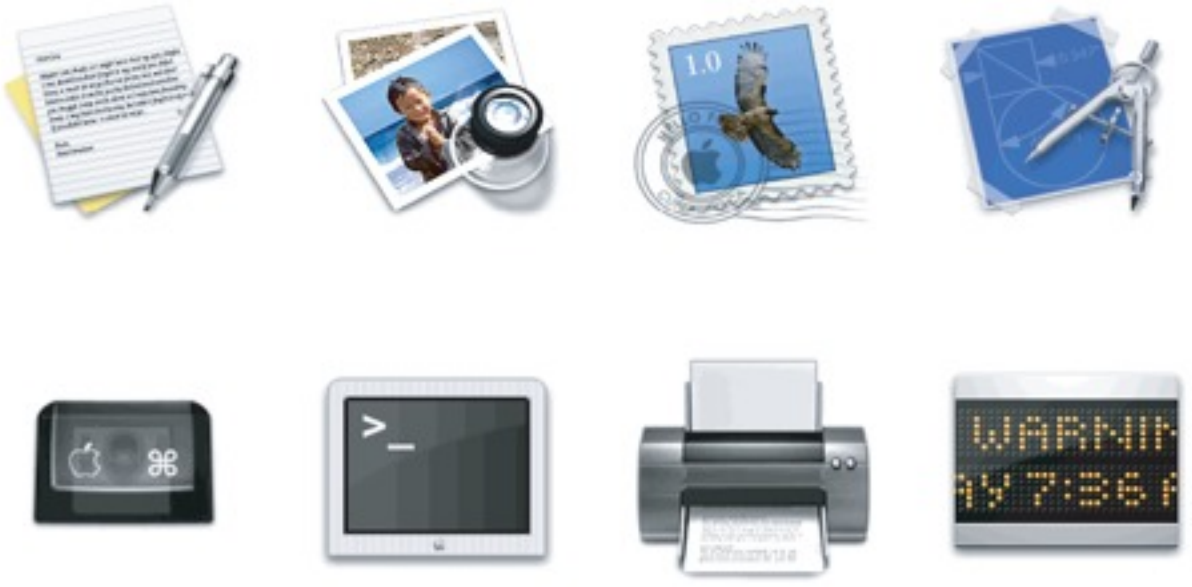
Icon Genres and Families

Icon genres help communicate what you can do with an application before you open it. Applications are classified by role—user applications, software utilities, and so on—and each category, or genre, has its own icon style. This differentiation is very important for helping users easily distinguish between types of icons in the Dock.

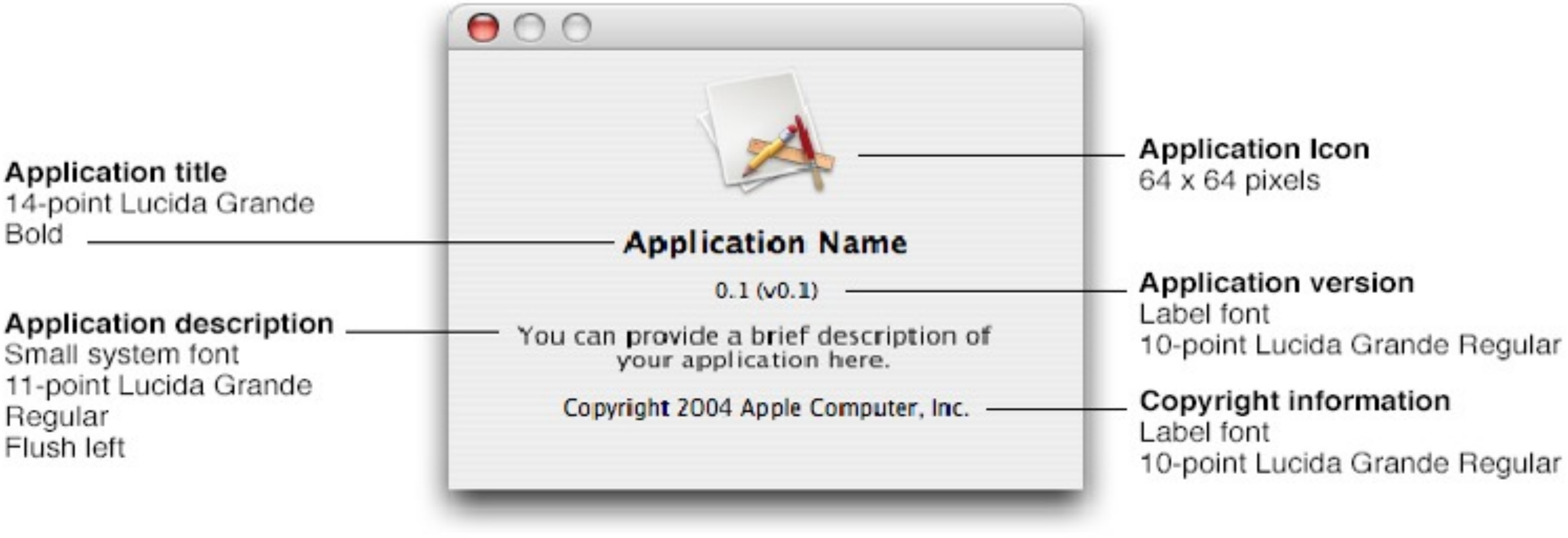
Figure 5-1 Application icons of different genres—user applications and utilities—shown as they might appear in the Dock

For example, the icons for user applications are colorful and inviting, while utilities have a more serious appearance. Figure 5-2 shows user application icons in the top row and utility icons in the bottom row. These genres are further described in "User Application Icons" (page 57) and "Utility Icons" (page 58).

Figure 5-2 Two icon genres: User application icons in top row; utility icons in bottom row



Example 2:
Apple Human Interface Guidelines
(page 55)



Application Name

0.1 (v0.1)

You can provide a brief description of your application here.

Copyright 2004 Apple Computer, Inc.

Application title
14-point Lucida Grande
Bold

Application description
Small system font
11-point Lucida Grande
Regular
Flush left

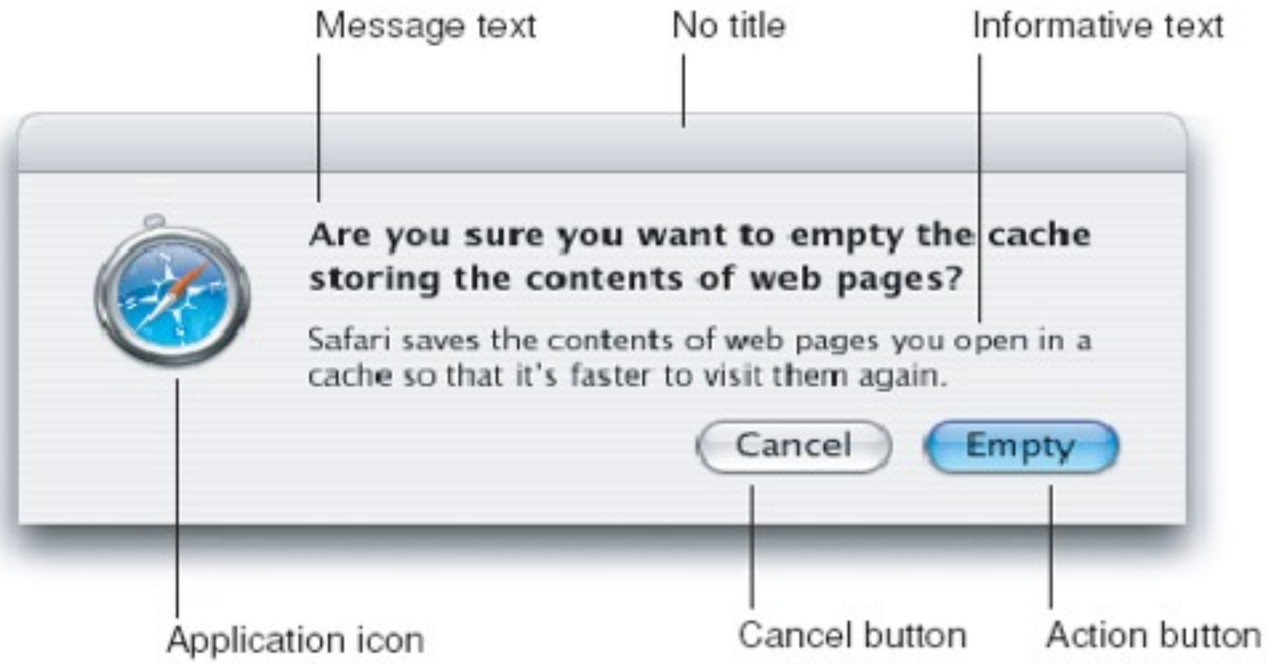
Application Icon
64 x 64 pixels

Application version
Label font
10-point Lucida Grande Regular

Copyright information
Label font
10-point Lucida Grande Regular

Example 3:
Apple Human Interface Guidelines
(page 126 & 134)

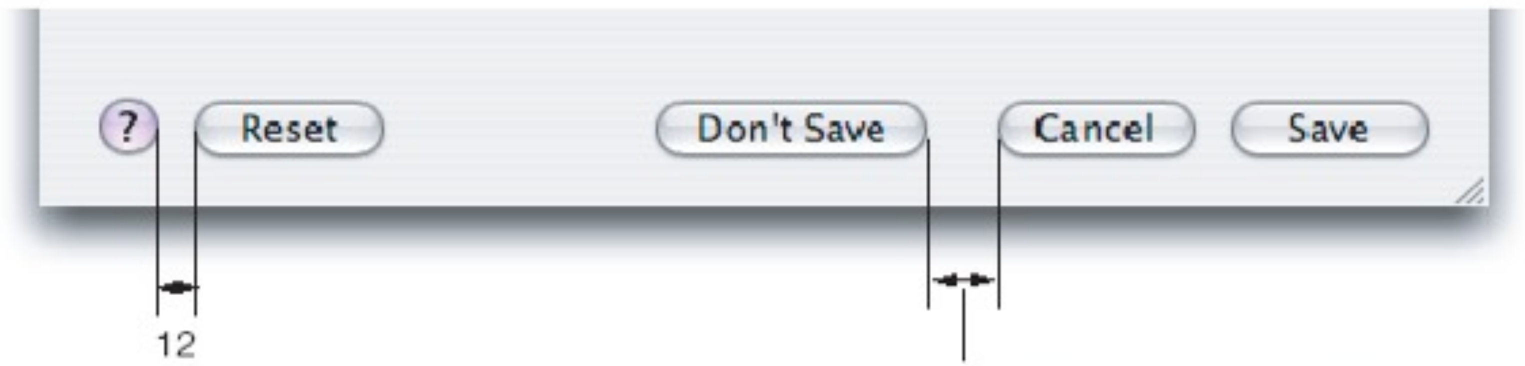
Figure 9-2 A standard alert



Are you sure you want to empty the cache storing the contents of web pages?

Safari saves the contents of web pages you open in a cache so that it's faster to visit them again.

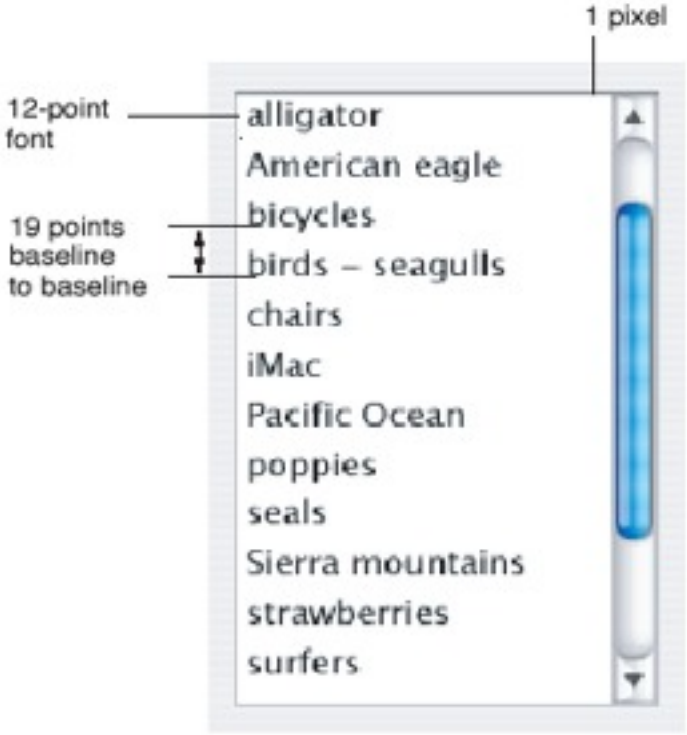
Figure 9-7 Position of buttons at the bottom of a dialog



A button that could result in lost data should be at 24 pixels away from "safe" buttons.

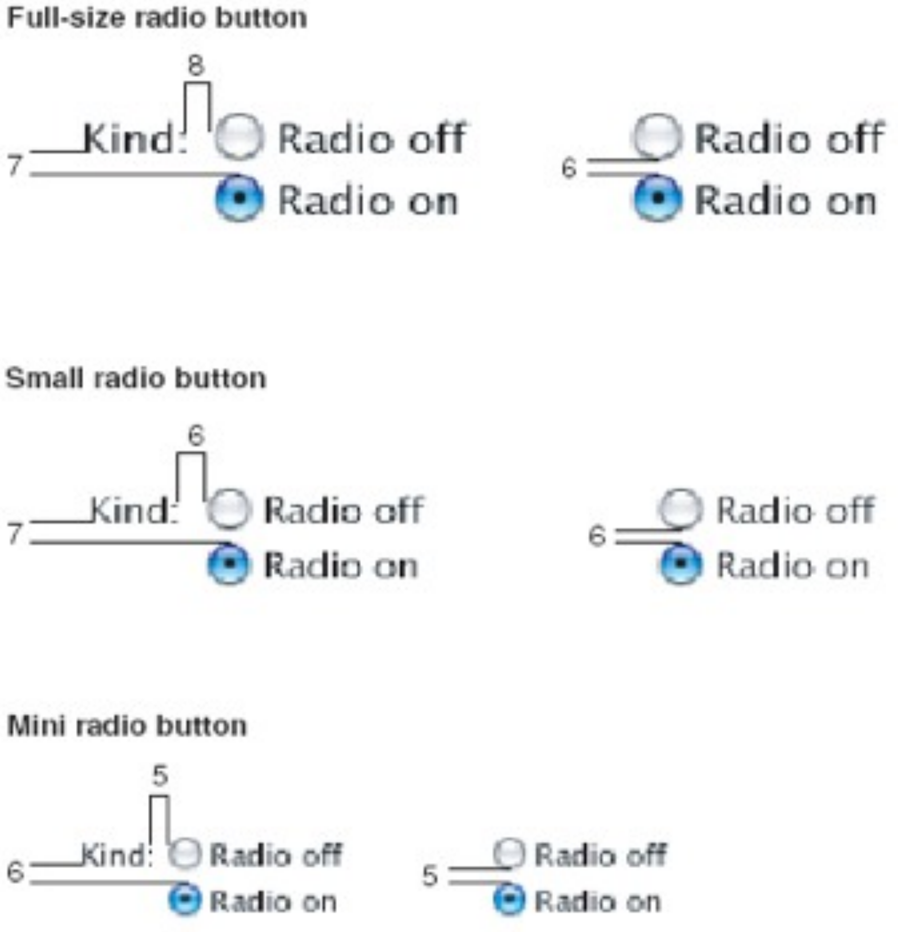
Scrolling List Specifications

Figure 10-44 Scrolling list dimensions



Radio Button Specifications

Figure 10-14 Radio button spacing


Full-size radio button

Small radio button

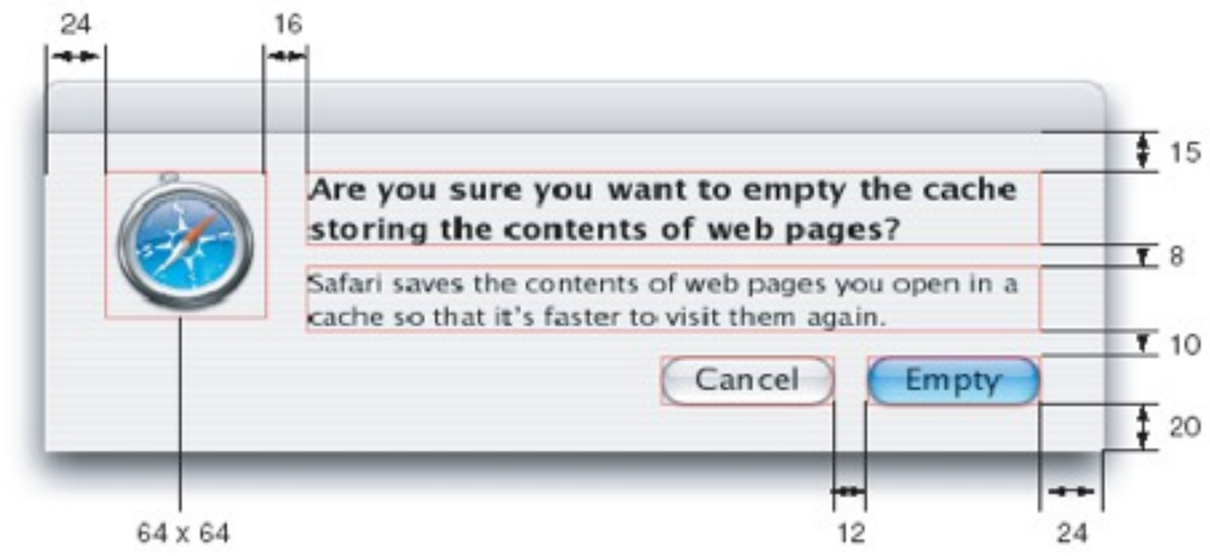
Mini radio button

Align the baselines of the label and the first button's text.

Example 4:
Apple Human Interface Guidelines
(page 138, 163 & 190)

Example 5:
Apple Human Interface Guidelines
(page 207, 209 & 210)

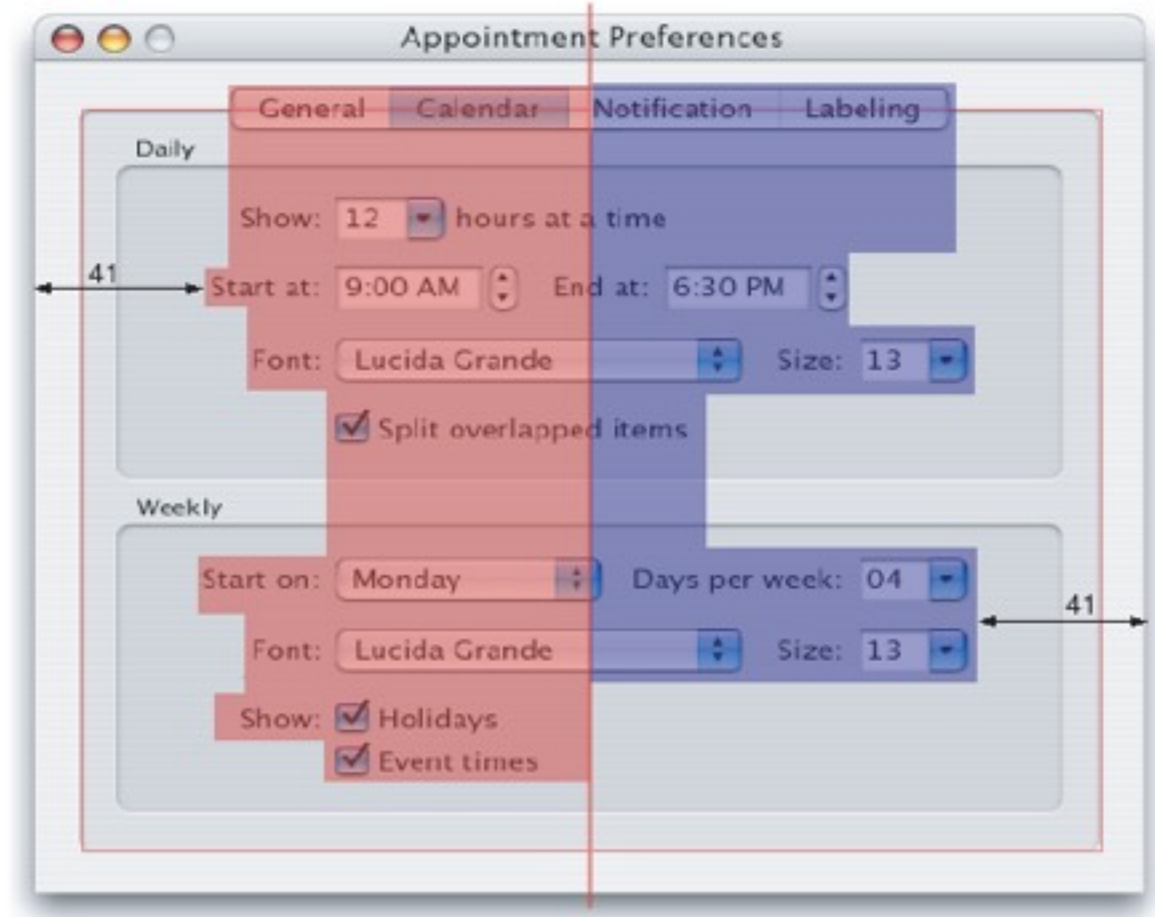
Figure 11-10 Layout dimensions for a standard alert



Are you sure you want to empty the cache storing the contents of web pages?

Safari saves the contents of web pages you open in a cache so that it's faster to visit them again.

Figure 11-6 Center-equalization in a changeable pane dialog



Appointment Preferences

Daily

Show: 12 hours at a time

Start at: 9:00 AM End at: 6:30 PM

Font: Lucida Grande Size: 13

☑ Split overlapped items

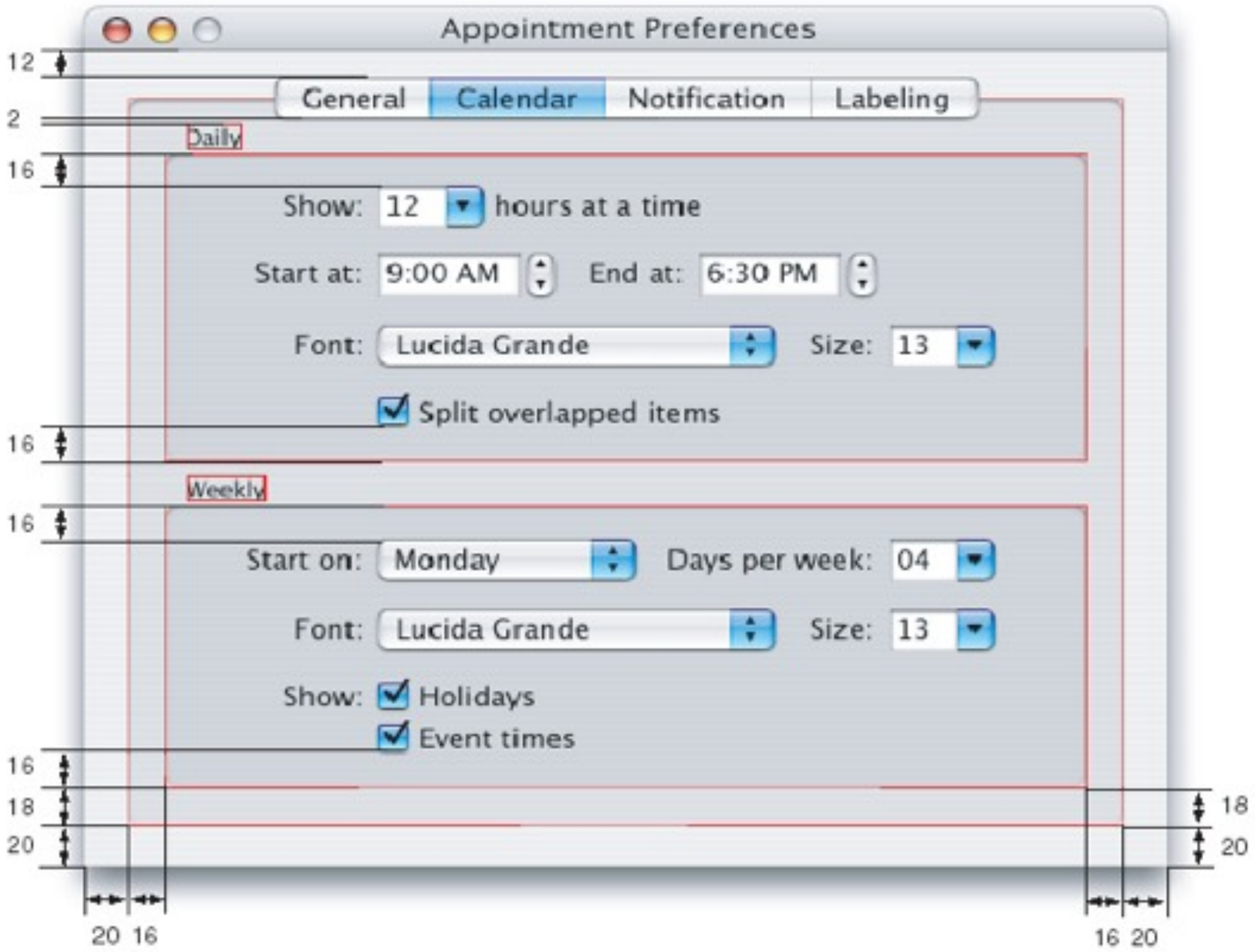
Weekly

Start on: Monday Days per week: 04

Font: Lucida Grande Size: 13

Show: ☑ Holidays
☑ Event times

Figure 11-8 Layout dimensions for a changeable pane dialog



Appointment Preferences

Daily

Show: 12 hours at a time

Start at: 9:00 AM End at: 6:30 PM

Font: Lucida Grande Size: 13

☑ Split overlapped items

Weekly

Start on: Monday Days per week: 04

Font: Lucida Grande Size: 13

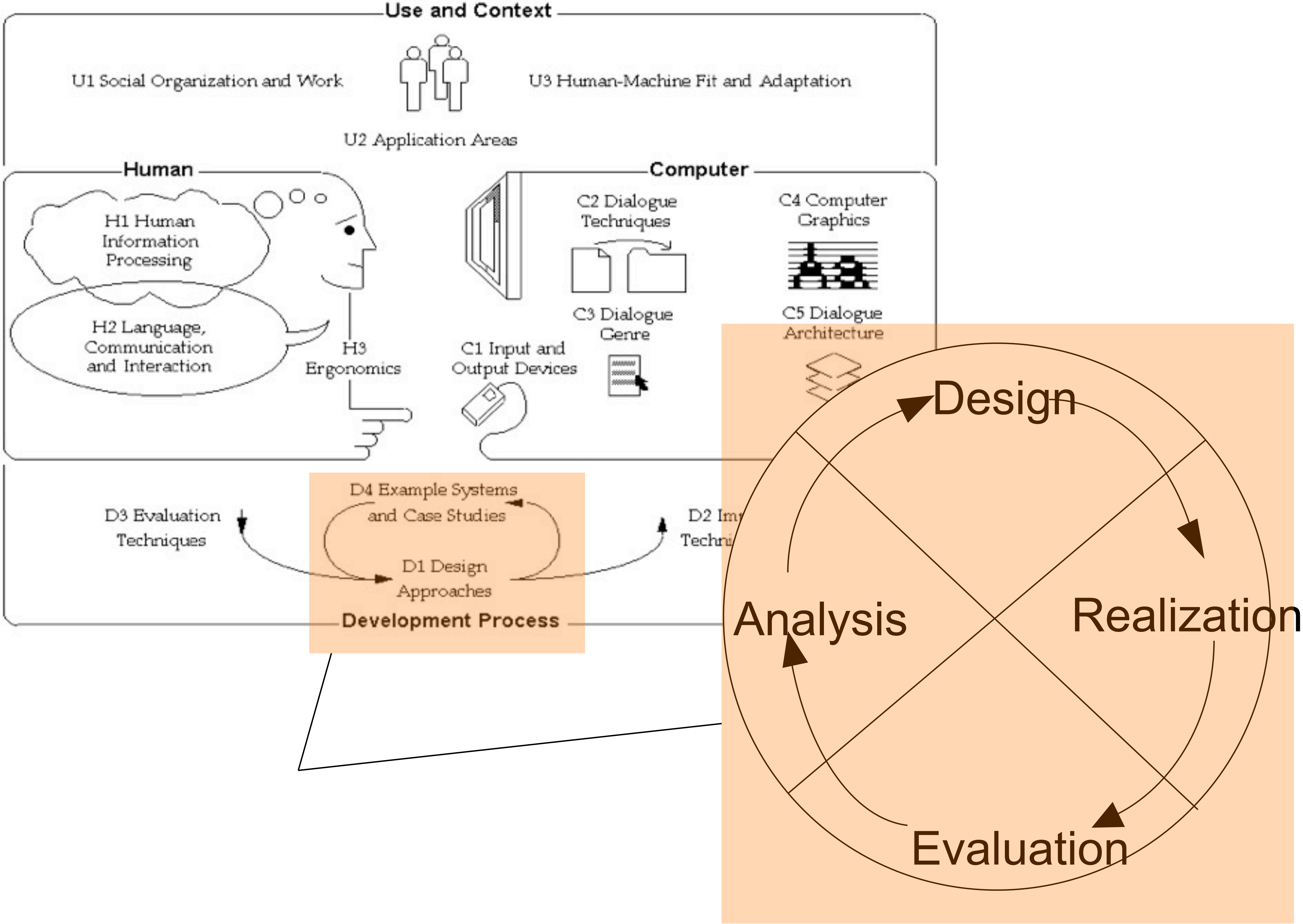
Show: ☑ Holidays
☑ Event times

Specific Guidelines for
Operating Systems, Window Managers, and the WWW

- Introduction to the Apple Human Interface Guidelines
  http://developer.apple.com/documentation/UserExperience/index.html
- KDE User Interface Guidelines
  http://developer.kde.org/documentation/design/ui/
  http://developer.kde.org/documentation/standards/kde/style/basics/
- Palm OS® User Interface Guidelines
  http://www.accessdevnet.com/docs/ui/UIGuidelinesTOC.html
- MSDN - User Interface Design and Development
  http://msdn.microsoft.com
- GNOME Human Interface Guidelines (V2.3)
  http://developer.gnome.org/projects/gup/hig/draft_hig_new/
- Web Guidelines
  http://www.webstyleguide.com/wsg3/index.html
- … and many others!

Implementing Interactive Systems

- Designing Look-And-Feel
- Implementation Technologies for Interactive Systems
- Standards and Guidelines
- Software Development Process Models
- User-Centered Development
- Integrating Usability into the Development Process

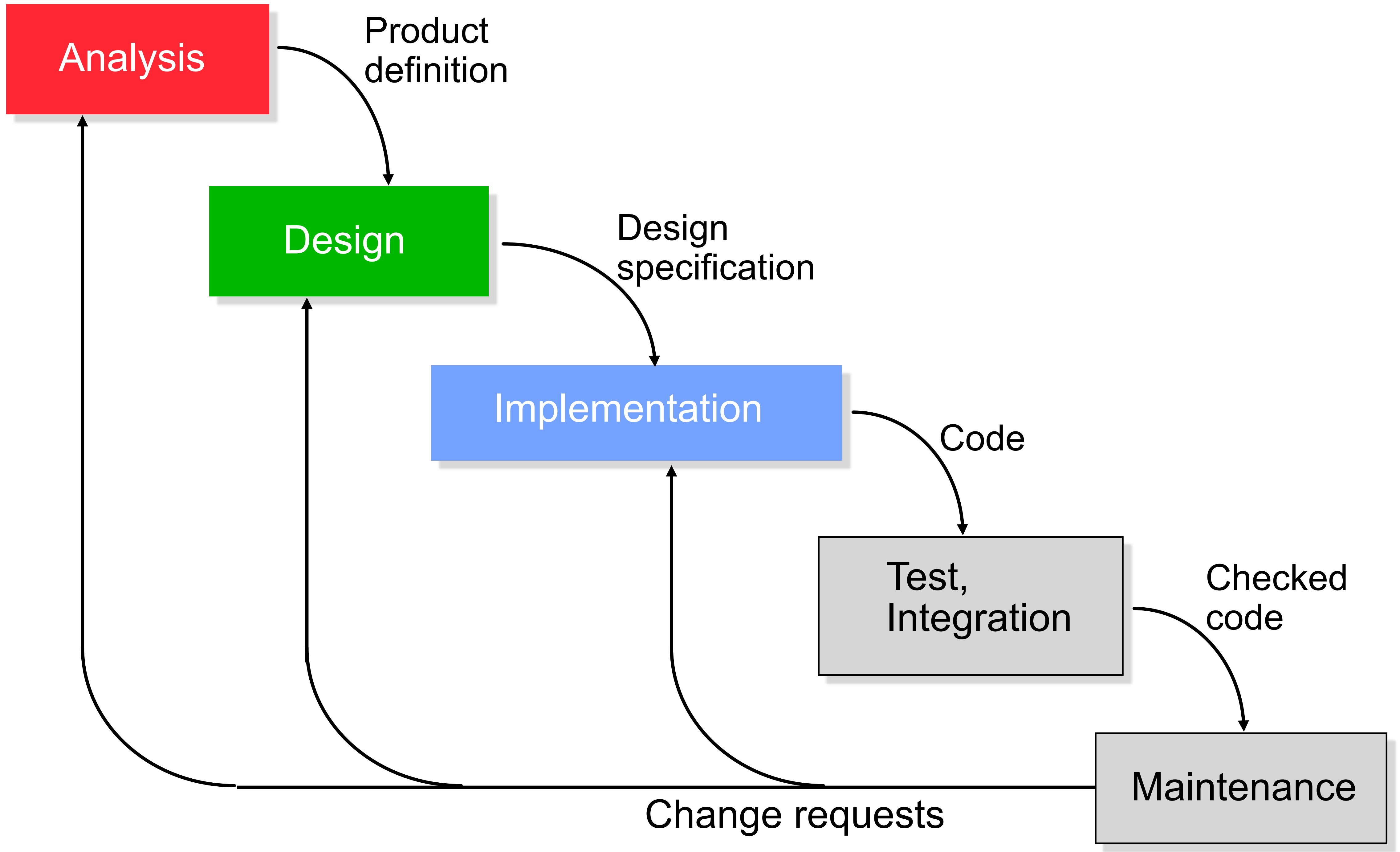


Use and Context
U1 Social Organization and Work
U2 Application Areas
U3 Human-Machine Fit and Adaptation
Human
H1 Human Information Processing
H2 Language, Communication and Interaction
H3 Ergonomics
Computer
C1 Input and Output Devices
C2 Dialogue Techniques
C3 Dialogue Genre
C4 Computer Graphics
C5 Dialogue Architecture
D1 Design Approaches
D2 Im
Techni
D3 Evaluation Techniques
D4 Example Systems and Case Studies
Development Process
Design
Realization
Evaluation
Analysis

Software Development Process Model

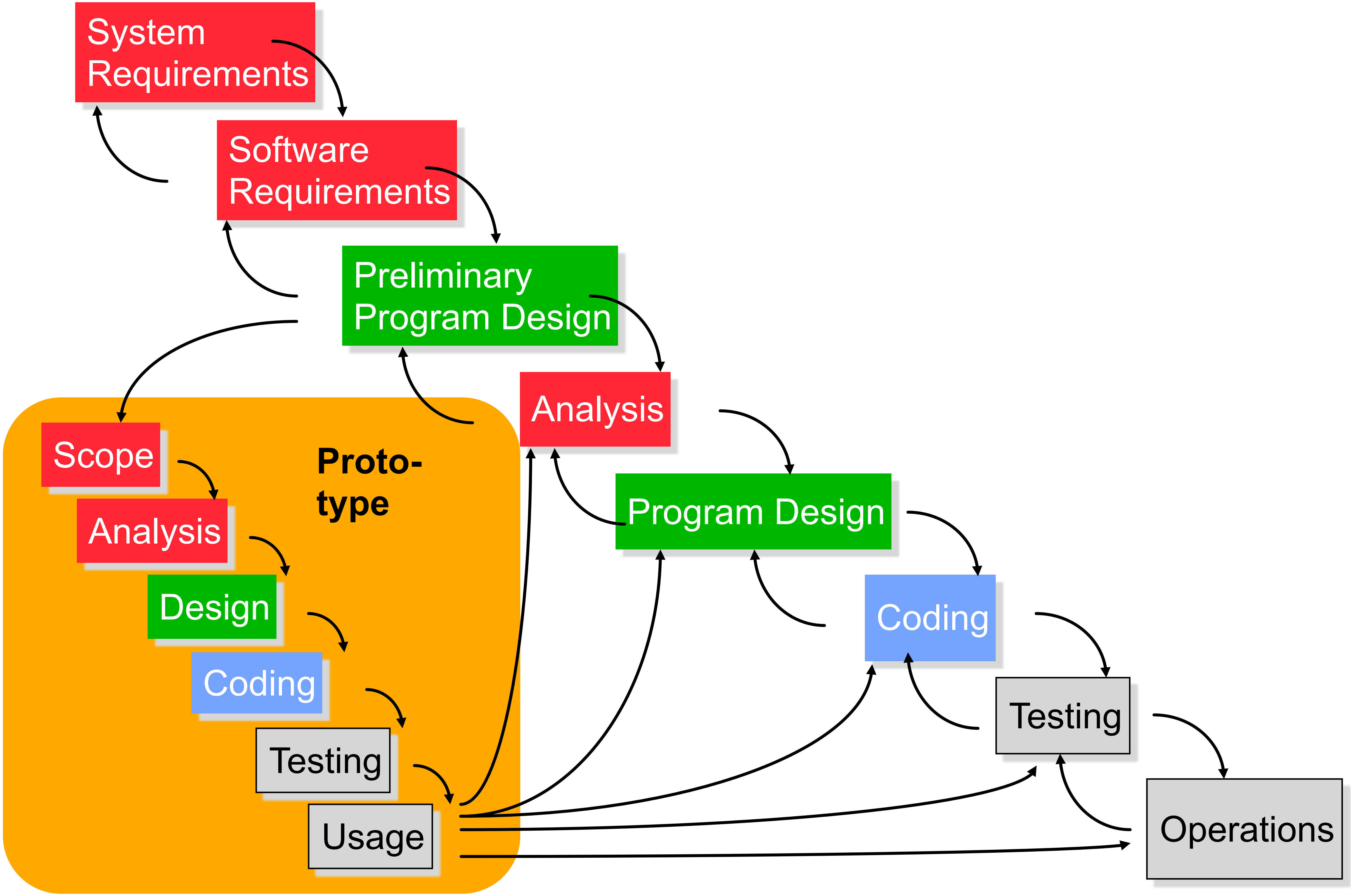
- ***Process model***
  - – Segmentation of the overall (team) activity of software development into smaller portions of work
    - high-level structure: phases
    - low-level structure: steps, activities
  - – Definition of an order for carrying out work units
  - – Guideline for the production of intermediate results
- Basic activities covered in all models:
  - – Analysis
  - – Design
  - – Implementation
  - – Validation (in particular Test, Integration)
  - – Deployment (in particular Maintenance)

“Pure” Waterfall Model

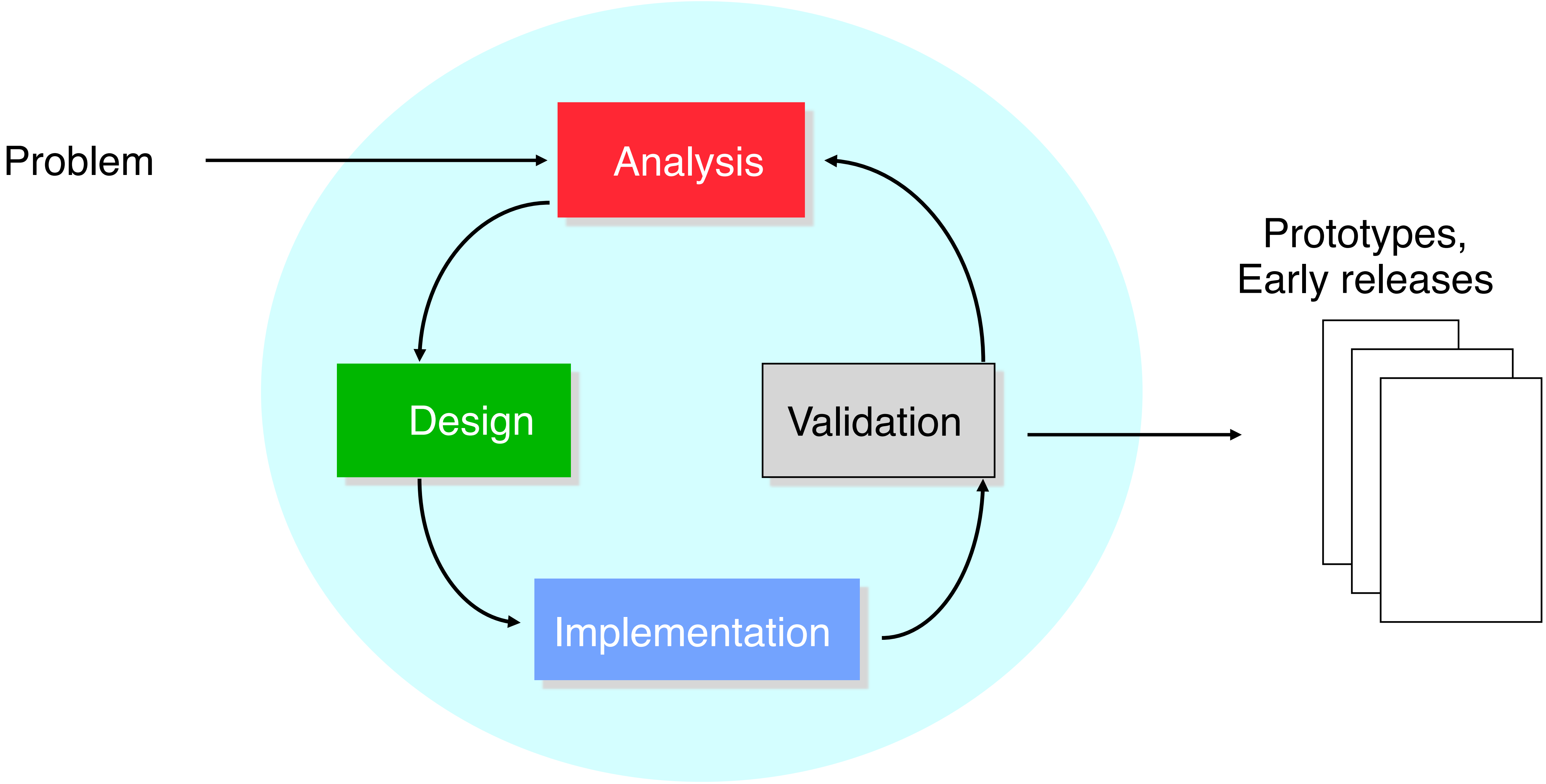


W. Royce (1970)

“Waterfall” with Feedback Loops and Prototyping

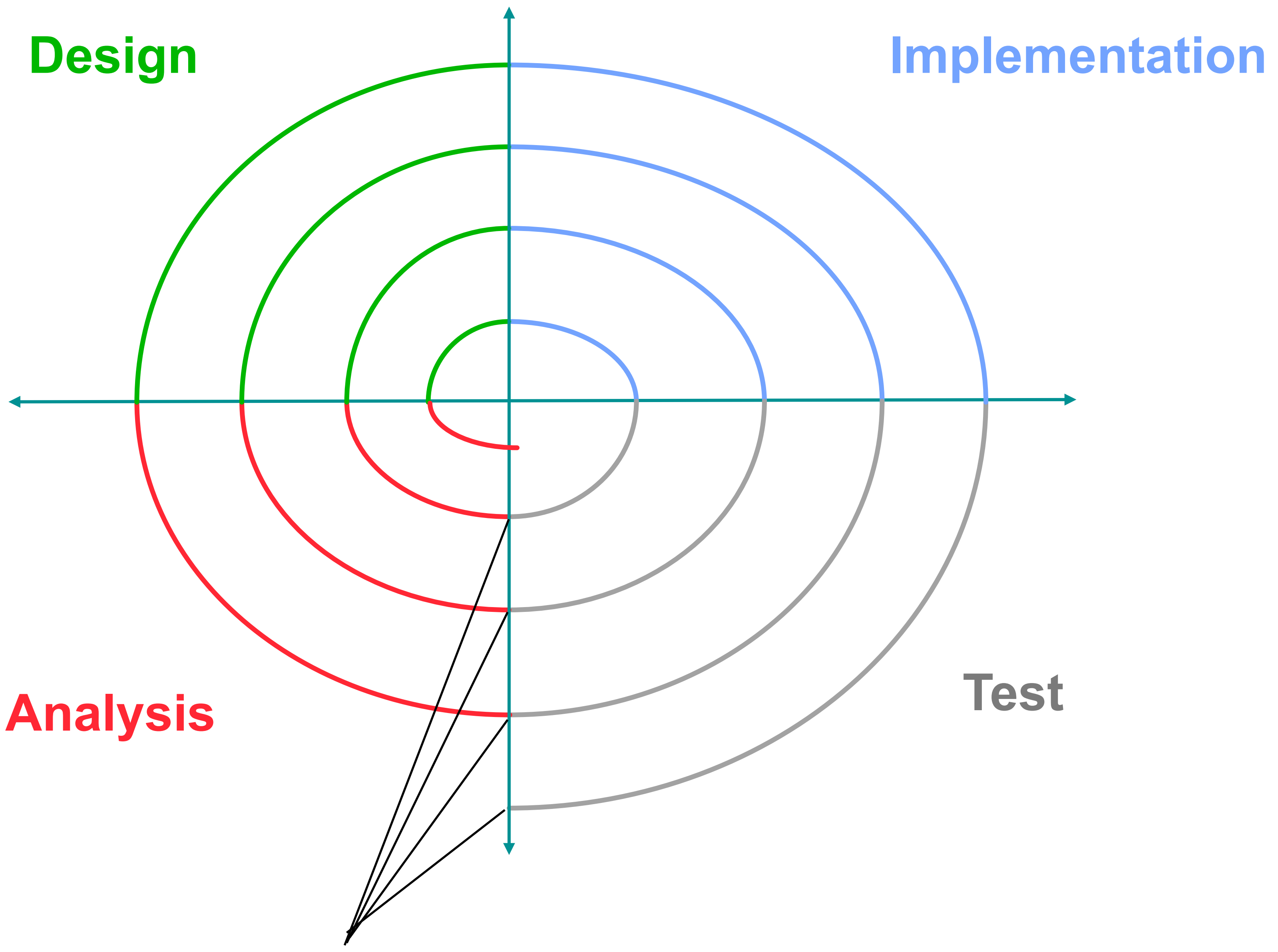


Evolutionary Development



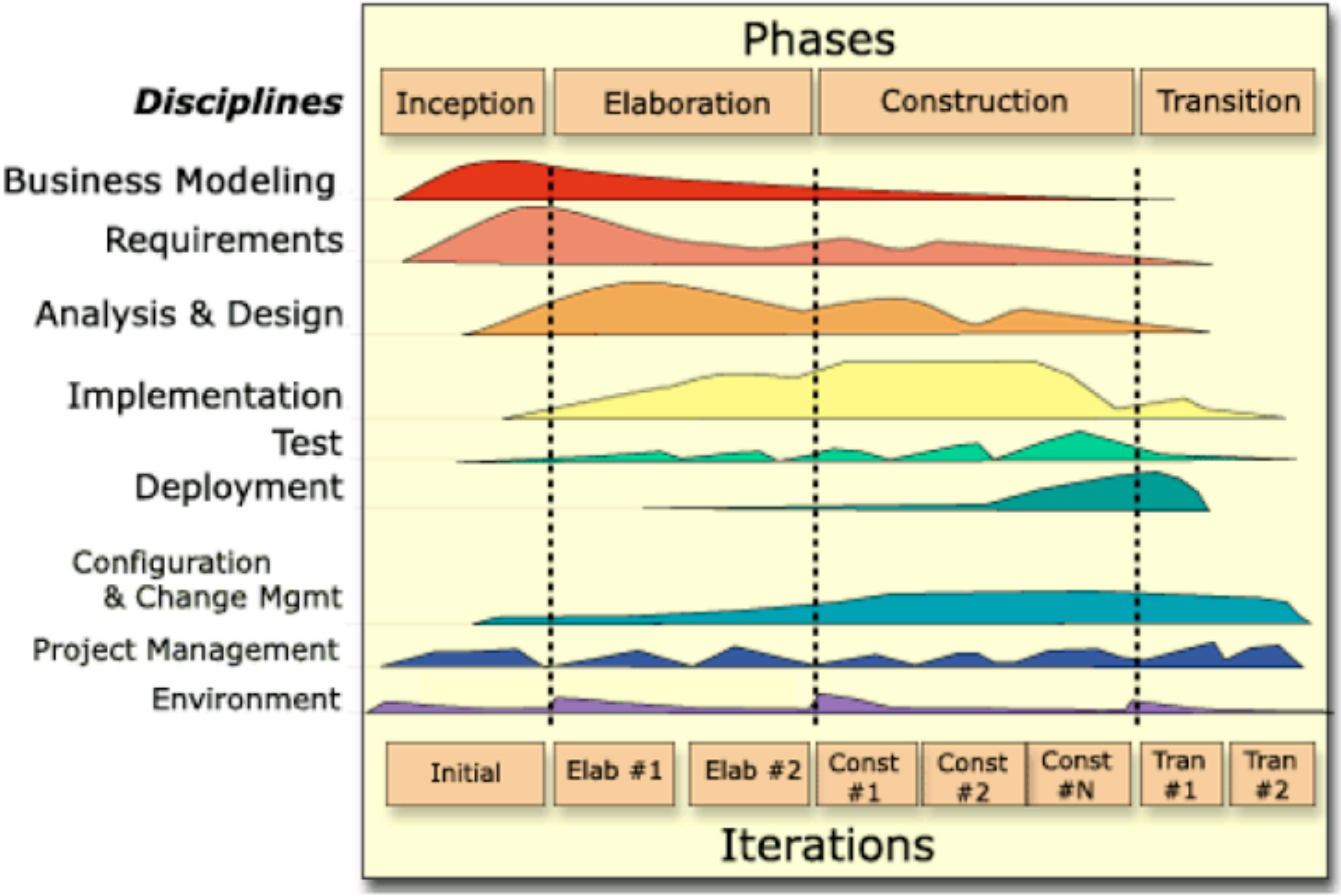
- Typical for small projects and experimental systems
- Technological progress (e.g. object orientation) may have improved scalability to large systems

Modern Adaptation – “Spiral Model”



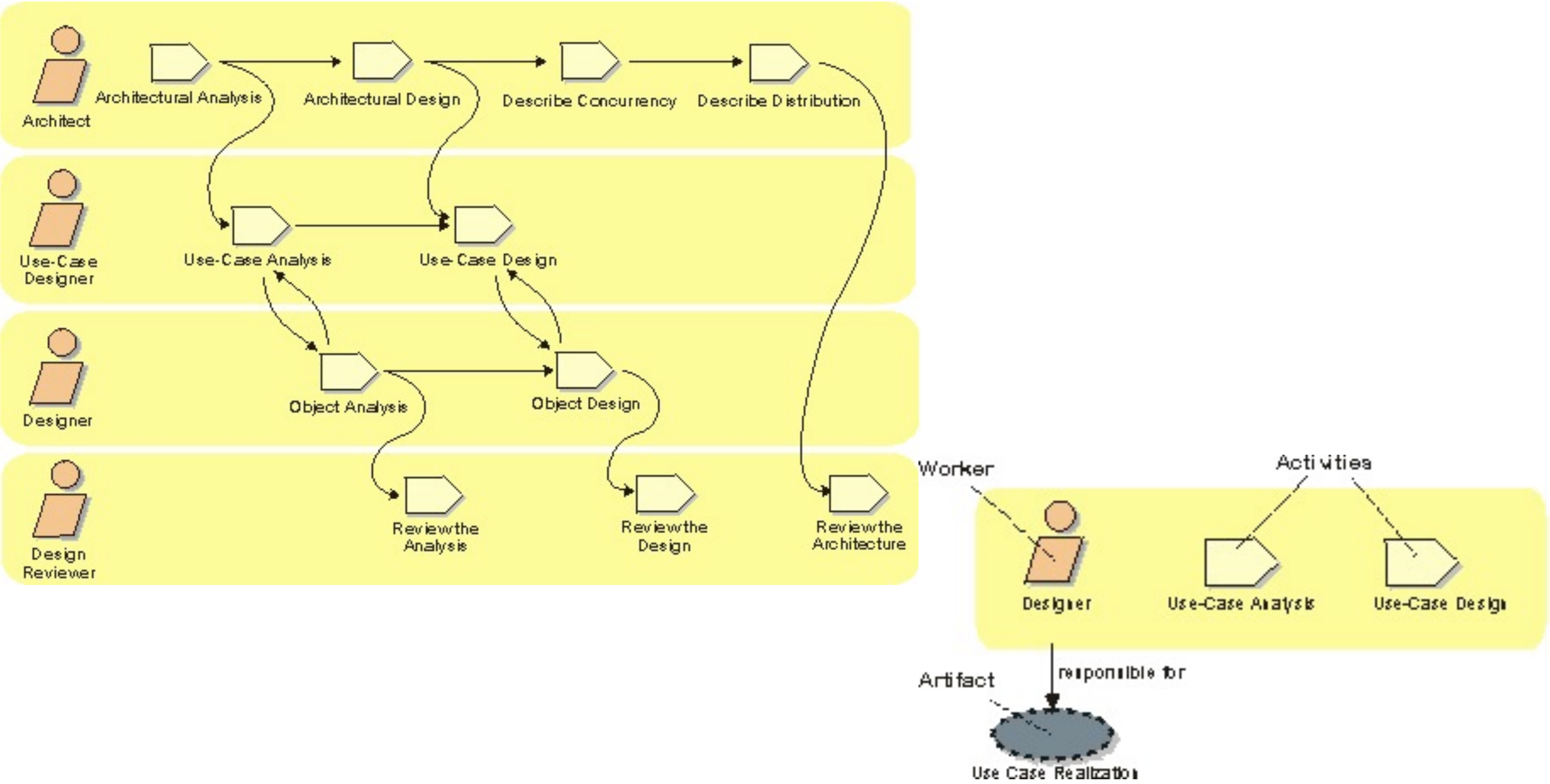
Products (Releases)
including *Prototypes*

Rational Unified Process (RUP)



Software Process framework is a commercial product of Rational, now IBM.
http://www-01.ibm.com/software/rational/rup/

Detailed Prescriptions in RUP



- Developers often consider this as not flexible enough for creative work.

Think different ... be creative ...

Agile Development

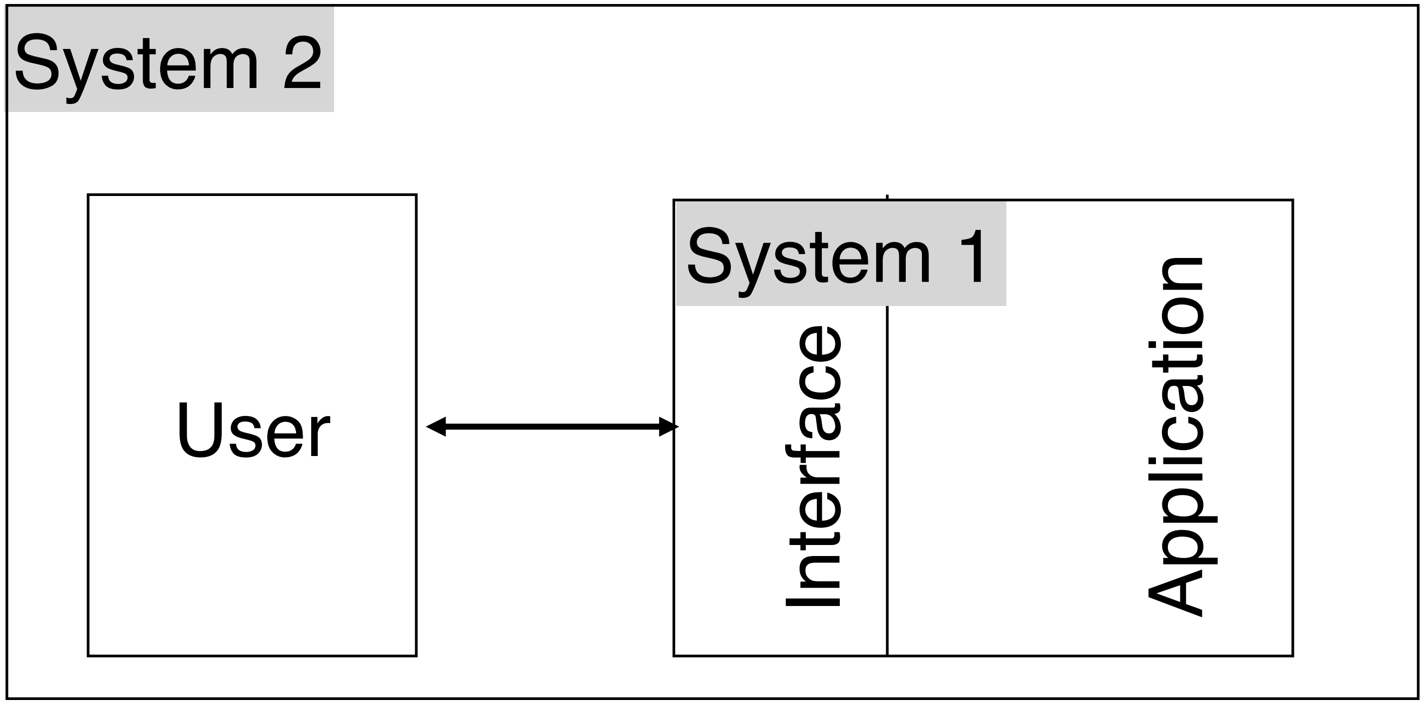
- "Agile" Software development (www.agilemanifesto.org):
  - E.g. Extreme Programming (XP), Crystal, Scrum
  - see lecture MMP by H. Hussmann...
- Recent trend in software development processes
  - Radical evolutionary development
- Key characteristics of agile development:
  - Individuals and interactions (rather than processes and tools)
  - Working software (code rather than extensive documentation)
  - Customer collaboration (instead of contract negotiations)
  - Responding to change (instead of following a plan)
- Agile development is not just “hacking along”!
  - Clear and strict rules
- Mixed information about success in practice
  - Good experiences in small and innovative projects
  - Large-scale projects tend to stay “conservative”, mainly due to transparency for project management

Implementing Interactive Systems

- Designing Look-And-Feel
- Implementation Technologies for Interactive Systems
- Standards and Guidelines
- Software Development Process Models
- User-Centered Development
- Integrating Usability into the Development Process

Usability Aspects are Mostly Ignored by Software Engineers

- Example:
  - IEEE “SWEBOK” body of knowledge definition for SE mentions HCI as “related discipline” under the name “software ergonomics”
- System perspectives
  - SW Engineers take the “System 1” perspective
  - Usability Engineers take the “System 2” perspective (following examples)

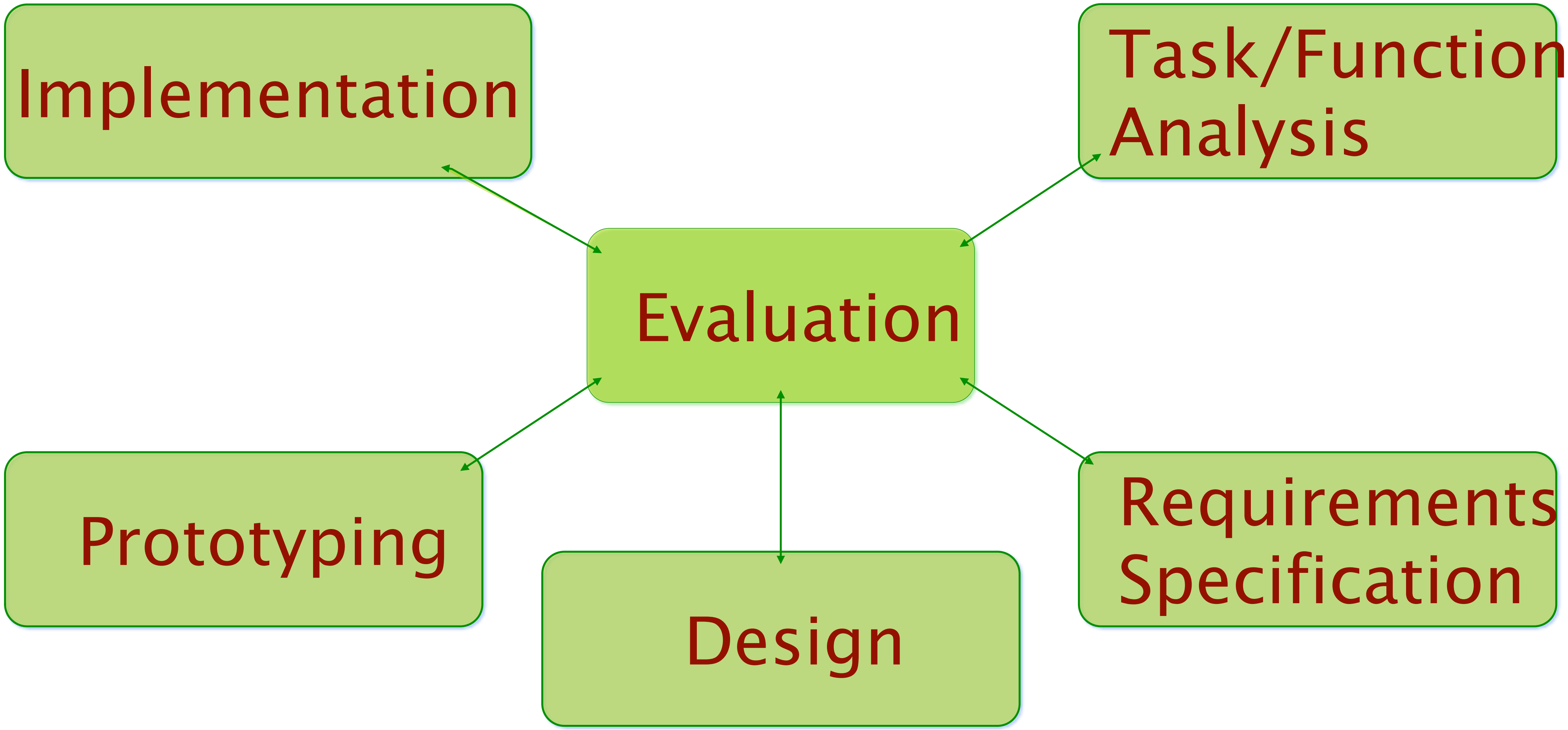


Seffah/Desmarais/Metzker

Separation between Interaction Design and Technical Design

- For interactive applications a separation into a two stage process is often advisable
- 1st – Interaction design (iterative)
  - concept
  - Interaction analysis
  - Prototypes
  - Evaluation
  - Stable and tested design
- 2nd – Technical realization
  - Technical analysis
  - Technical specification (e.g. architecture, platform)
  - Implementation
  - Evaluation and Quality management

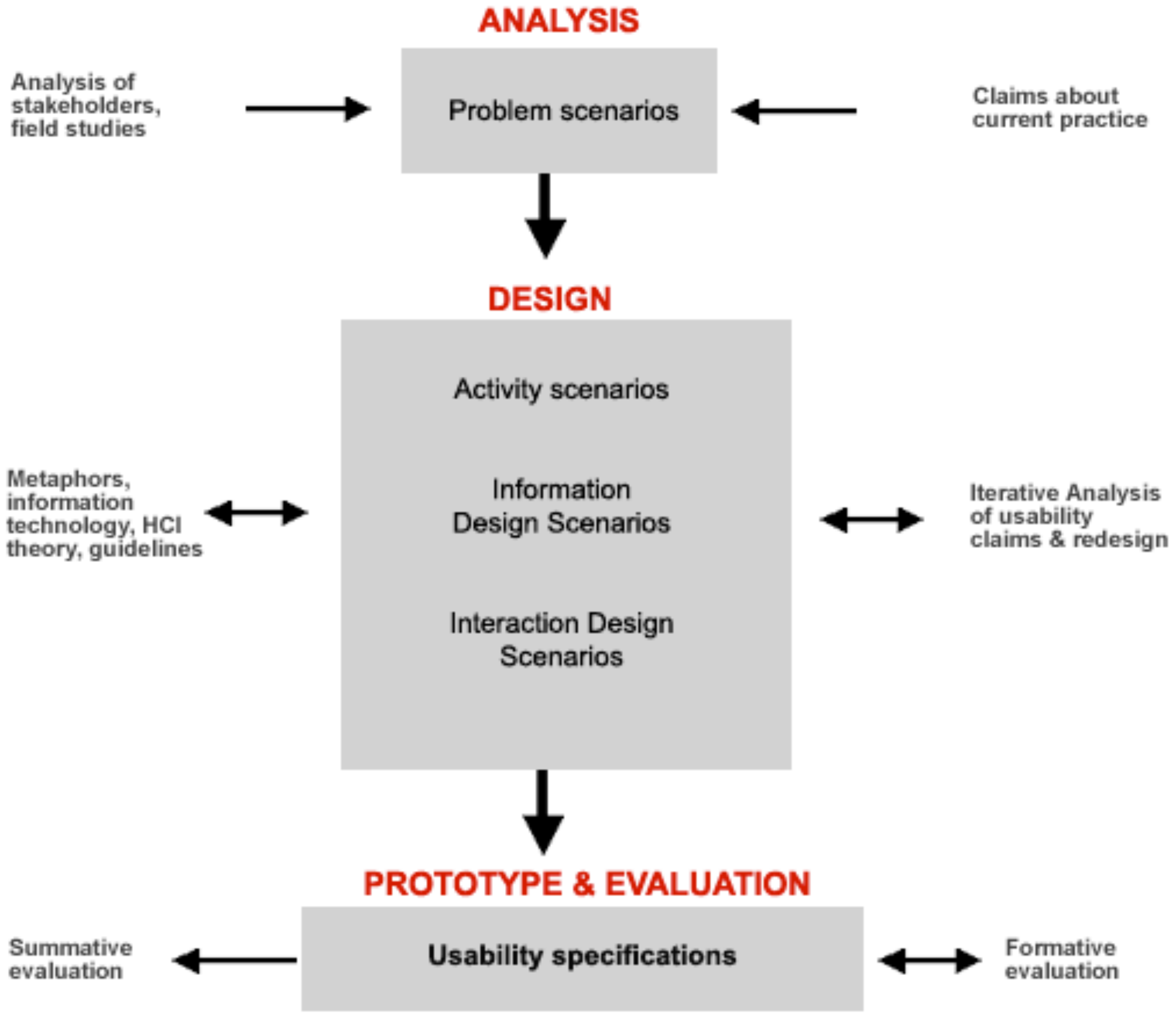
Star Lifecycle



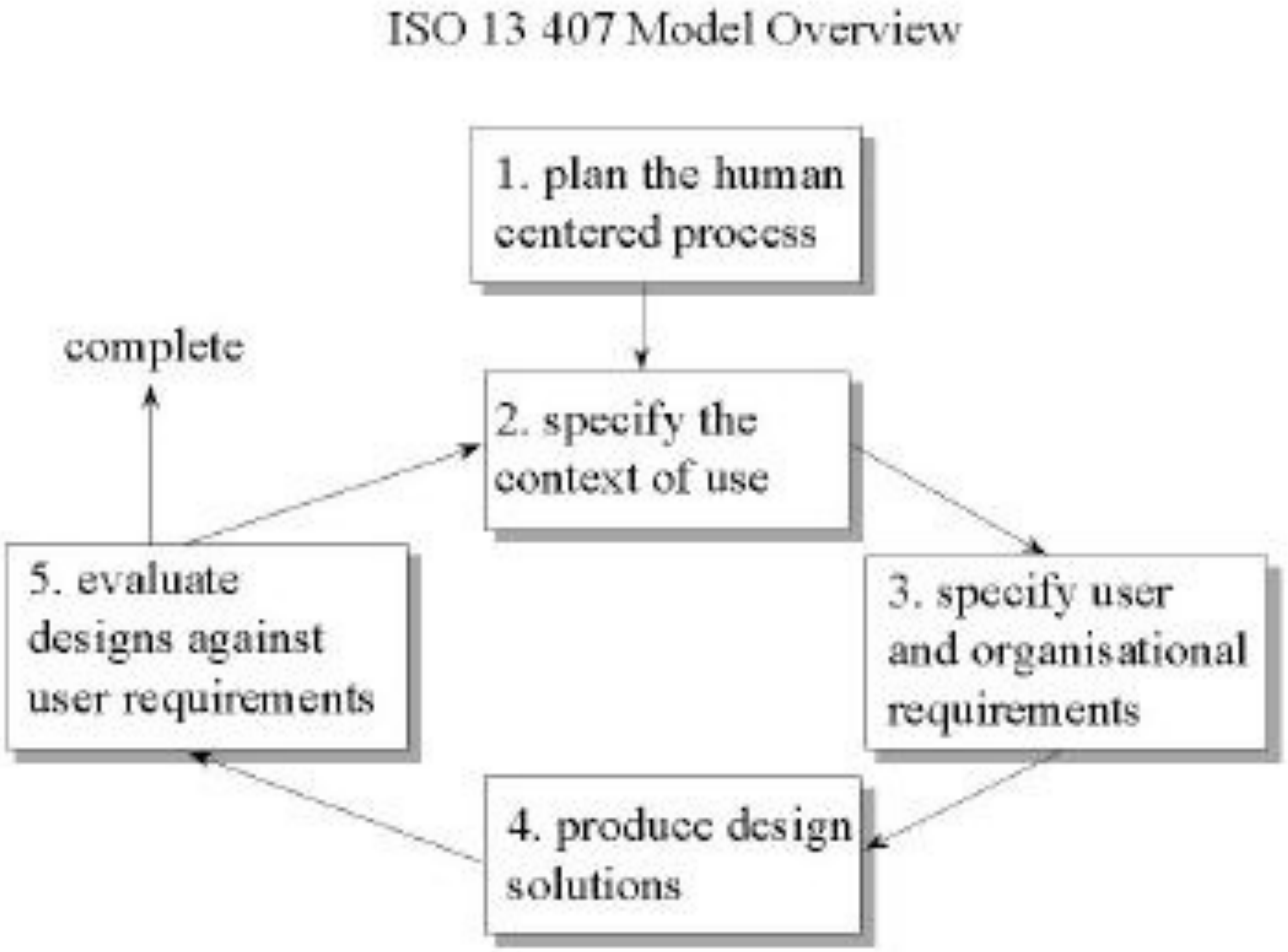
- Hix, Hartson 1993
  - Non-sequential: any order of activities
  - Evaluation-centric: every activity is evaluated
  - Interconnected: evaluation connects everything

Scenario-Based Design

- Mary Beth Rosson, John M. Carroll: Usability Engineering - Scenario-Based Development of Human-Computer Interaction, Academic Press 2002


ANALYSIS
Analysis of stakeholders, field studies
Problem scenarios
Claims about current practice
DESIGN
Activity scenarios
Metaphors, information technology, HCI theory, guidelines
Information Design Scenarios
Iterative Analysis of usability claims & redesign
Interaction Design Scenarios
PROTOTYPE & EVALUATION
Summative evaluation
Usability specifications
Formative evaluation

ISO 13407


ISO 13 407 Model Overview
1. plan the human centered process
2. specify the context of use
3. specify user and organisational requirements
4. produce design solutions
5. evaluate designs against user requirements
complete

- Guidelines for integrating usability aspects into the development process
  - Proposes iterative process
  - Stresses evaluation
  - Design solutions cover also lightweight prototypes, mock-ups etc.
- See e.g. http://www.ucc.ie/hfrg/emmus/methods/iso.html

Problems of User Centered Design

- Users may be wrong
- Users may be resistant to change
- Users may expect disadvantages (e.g. being replaced by software)

- Be aware – you are expected to create an optimal system with regard to ***the goals specified***
  - this is unfortunately NOT necessarily the system users would like to have (e.g. trade-off between employers and employees)

Implementing Interactive Systems

- Designing Look-And-Feel
- Implementation Technologies for Interactive Systems
- Standards and Guidelines
- Software Development Process Models
- User-Centered Development
- Integrating Usability into the Development Process

Concurrent Workflows, Competing Cultures

- User Interaction Design and Software/System Design are concurrent activities
  - One depends on the other, one influences the other
- Separate cultures led to competing models of the development process
  - Software Engineering: Artefact-centric (e.g. design documents), disciplined order of steps, quantitative evaluation (metrics, tests), highly compatible to project management needs
  - User-Centred Development: Interdisciplinary, loose (e.g. rough guidelines), flexible in order of steps, open to late changes, continuous qualitative evaluation (e.g. user tests with prototypes), difficult to “sell” to project managers
- Ambiguous overlaps in terminology
  - The same terms are used in many methods with differently defined or weakly defined semantics
    - E.g. “scenario”, “use case”, “test”, “user experience“
- Integration of process models
  - “Interface development is transitioning from an artistic exercise into an engineering discipline.” (Curtis/Hefley)

Concurrency of UI and SW Engineering

User Interface Engineering		Development Phase		Software Engineering
User & task analysis	⇐	**Requirements analysis**	⇒	Application design
Human vs. Machine	⇐	**Requirements allocation**	⇒	Hardware vs. Software
Dialogue design	⇐	**Preliminary design**	⇒	Architectural design
Screen design	⇐	**Detailed design**	⇒	Logical design
Coding	⇐	**Implementation**	⇒	Coding
Usability lab	⇐	**Implementation testing**	⇒	Unit & Integration testing
Contextual observation	⇐	**System testing**	⇒	System testing
Human performance	⇐	**Optimization**	⇒	Machine performance

(Curtis/Hefley)

User Experience “Plugin” for the RUP

- Extensions of roles, activities and (UML) artifacts
  – Use cases extended by “use case storyboards”
  – UI Prototyping as a specific activity
  – Screens as special cases of classes (derived from Conallen’s UML-based Web Design Method)

- Steps to create User Experience Storyboards:
  1. Add actor characteristics to the use case.
  2. Add usability guidance and usability requirements to the use case.
  3. Identify UX elements.
  4. Model the use-case flows with the UX elements.
  5. Model screen navigation for the use case.

http://www-128.ibm.com/developerworks/rational/library/content/RationalEdge/nov03/f_usability_jh.pdf

User Experience “Plugin” for the RUP

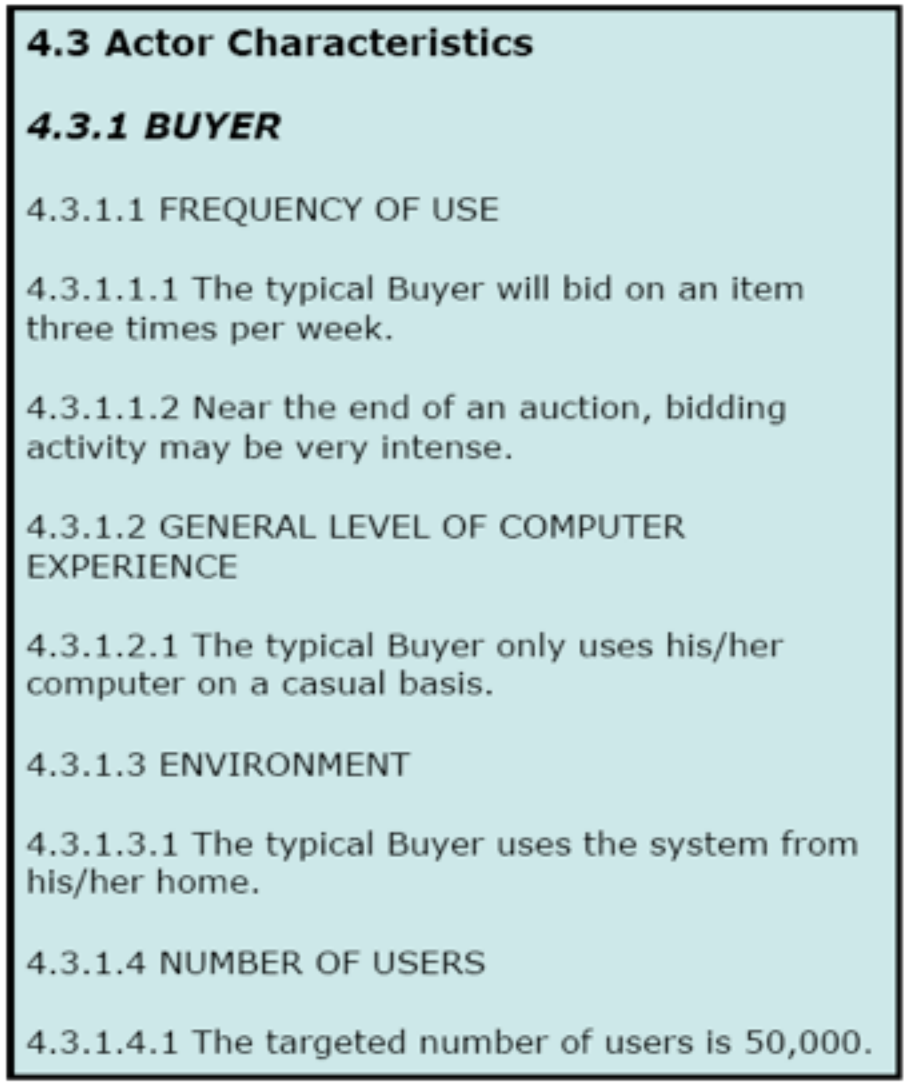
Actor characteristics

Usability guidance / requirements

UX elements

Use-case flows

Screen navigation



4.3 Actor Characteristics

4.3.1 BUYER

4.3.1.1 FREQUENCY OF USE

4.3.1.1.1 The typical Buyer will bid on an item three times per week.

4.3.1.1.2 Near the end of an auction, bidding activity may be very intense.

4.3.1.2 GENERAL LEVEL OF COMPUTER EXPERIENCE

4.3.1.2.1 The typical Buyer only uses his/her computer on a casual basis.

4.3.1.3 ENVIRONMENT

4.3.1.3.1 The typical Buyer uses the system from his/her home.

4.3.1.4 NUMBER OF USERS

4.3.1.4.1 The targeted number of users is 50,000.

Figure 5: Actor characteristics for the Bid on Item use case

User Experience “Plugin” for the RUP

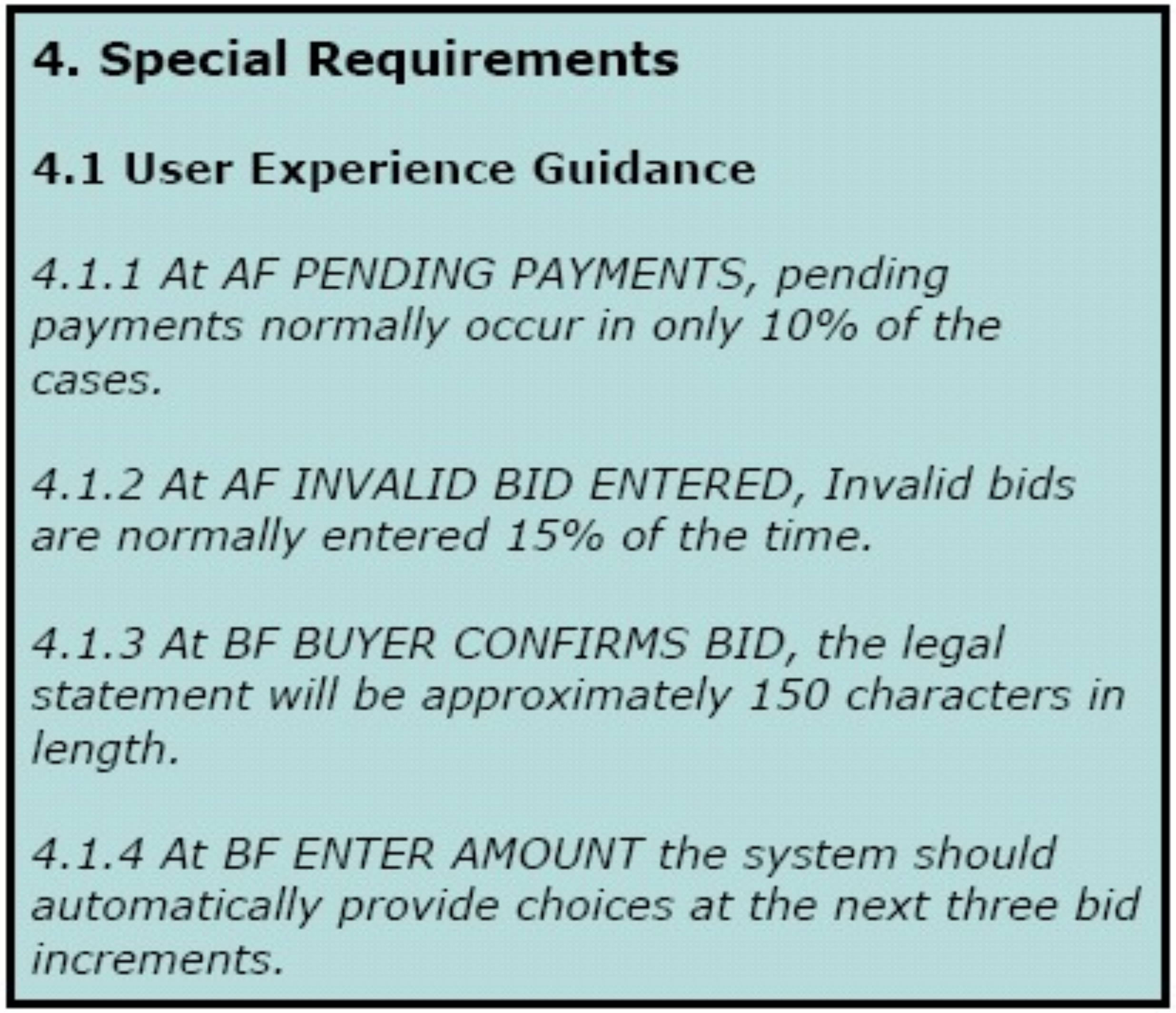
Actor characteristics

Usability guidance / requirements

UX elements

Use-case flows

Screen navigation



4. Special Requirements

4.1 User Experience Guidance

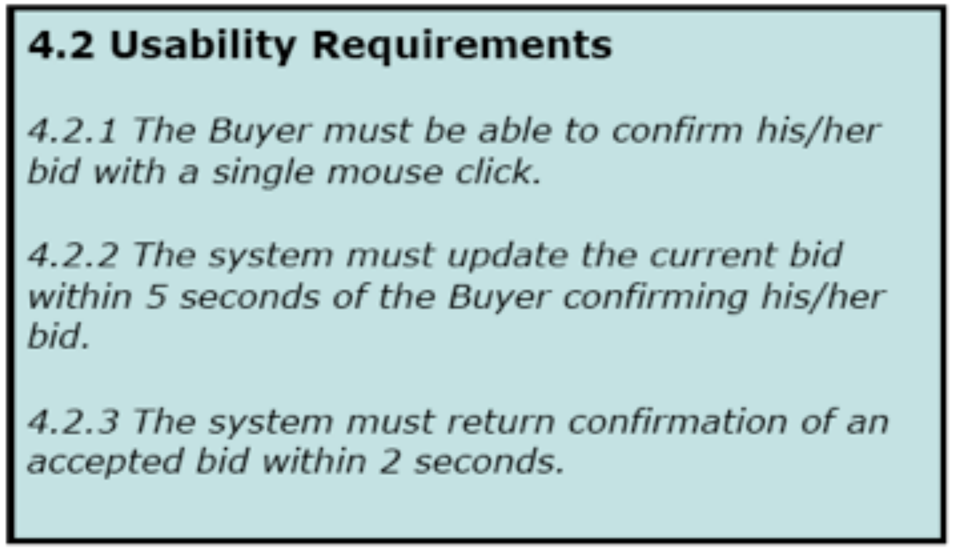
4.1.1 At AF PENDING PAYMENTS, pending payments normally occur in only 10% of the cases.

4.1.2 At AF INVALID BID ENTERED, Invalid bids are normally entered 15% of the time.

4.1.3 At BF BUYER CONFIRMS BID, the legal statement will be approximately 150 characters in length.

4.1.4 At BF ENTER AMOUNT the system should automatically provide choices at the next three bid increments.

Figure 6: Usability guidance for the Bid on Item use case



4.2 Usability Requirements

4.2.1 The Buyer must be able to confirm his/her bid with a single mouse click.

4.2.2 The system must update the current bid within 5 seconds of the Buyer confirming his/her bid.

4.2.3 The system must return confirmation of an accepted bid within 2 seconds.

Figure 7: Usability requirements for the Bid on Item use case

User Experience “Plugin” for the RUP

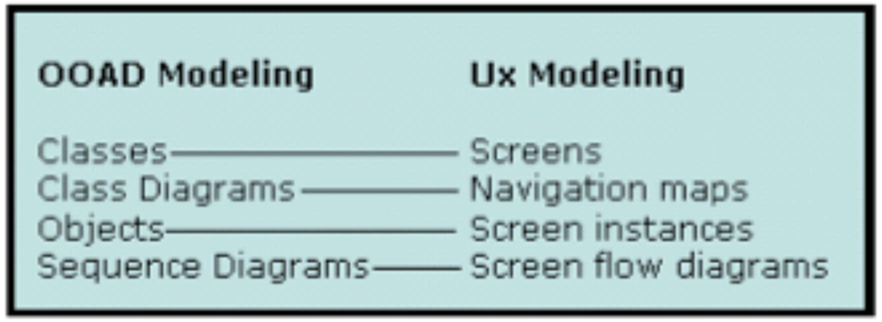
Actor characteristics

Usability guidance / requirements

UX elements

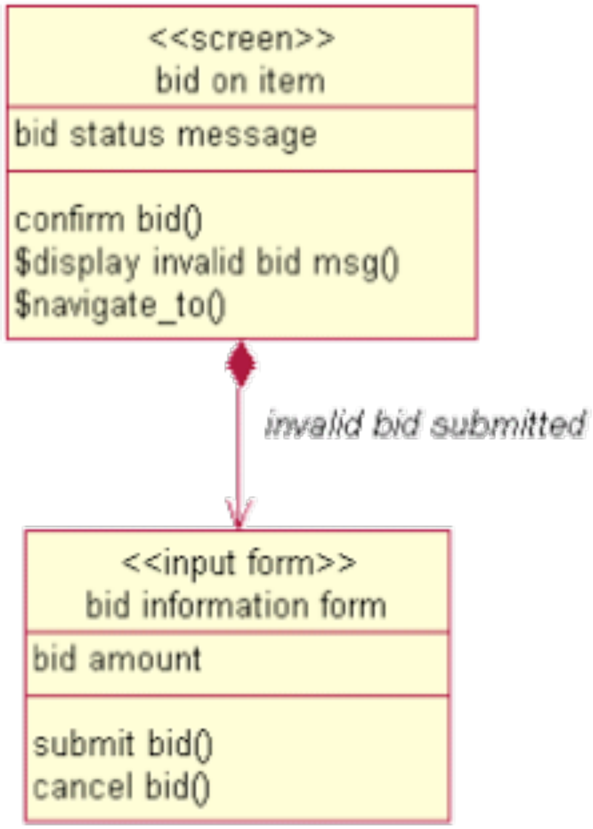
Use-case flows

Screen navigation



OOAD Modeling	Ux Modeling
Classes	Screens
Class Diagrams	Navigation maps
Objects	Screen instances
Sequence Diagrams	Screen flow diagrams

Figure 8: Mapping between OOAD and UX modeling elements



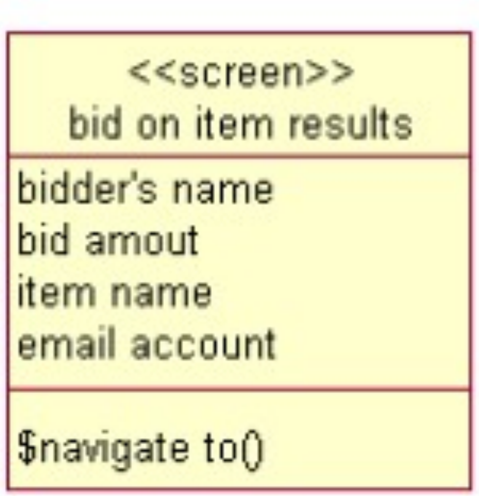


Figure 12: UML representation of a screen for Bid on Item use case (basic flow)

Figure 18: UML representation of a screen with an input form

User Experience “Plugin” for the RUP

Actor characteristics

Usability guidance / requirements

UX elements

Use-case flows

Screen navigation

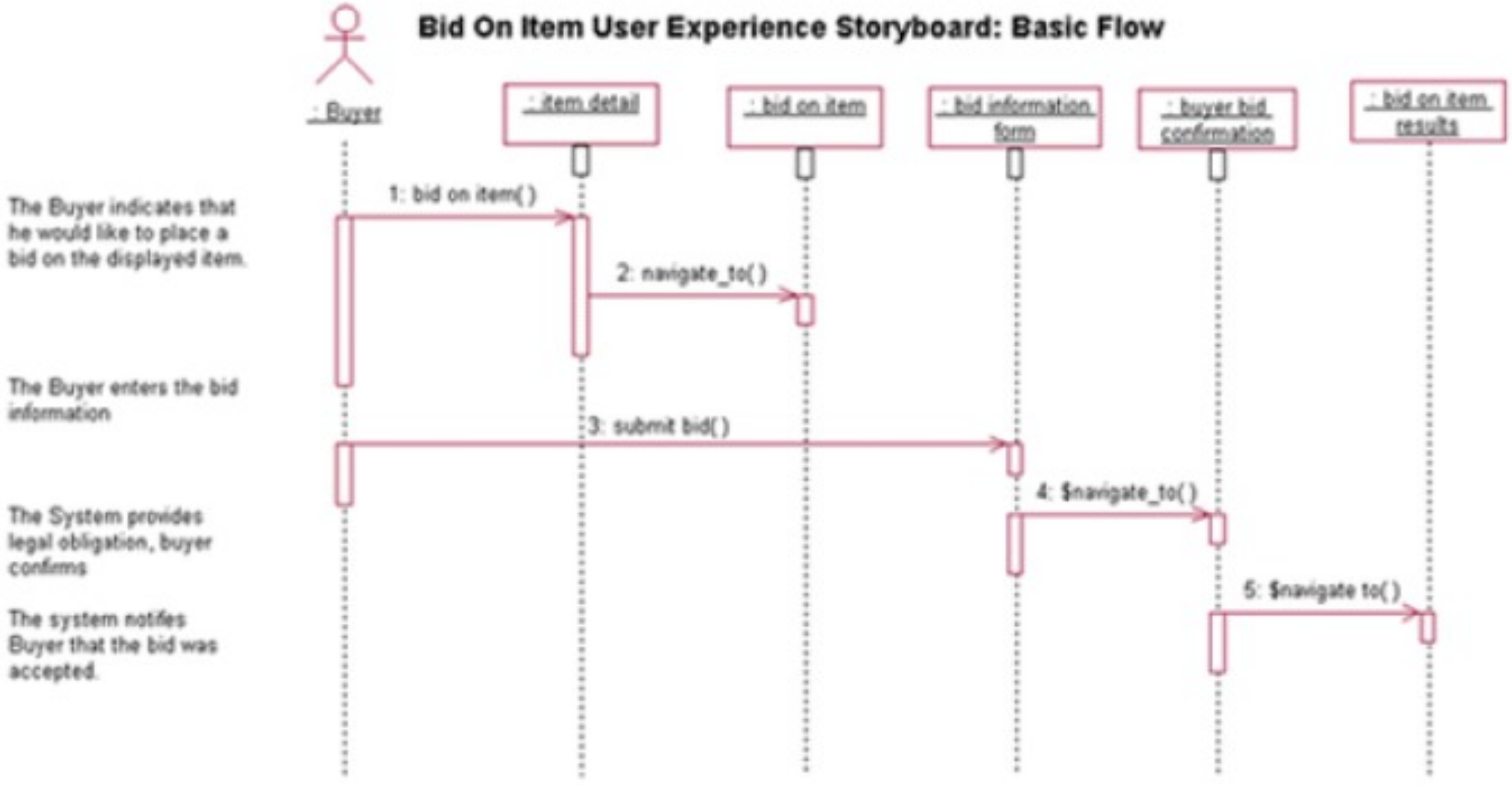

Bid On Item User Experience Storyboard: Basic Flow
: Buyer
: item detail
: bid on item
: bid information form
: buyer bid confirmation
: bid on item results
1: bid on item( )
2: navigate_to( )
3: submit bid( )
4: $navigate_to( )
5: $navigate to( )
The Buyer indicates that he would like to place a bid on the displayed item.
The Buyer enters the bid information
The System provides legal obligation, buyer confirms
The system notifies Buyer that the bid was accepted.

Figure 19: Sequence diagram showing the basic flow of events for the Bid on Item use case

User Experience “Plugin” for the RUP

- Actor characteristics
- Usability guidance / requirements
- UX elements
- Use-case flows
- **Screen navigation**

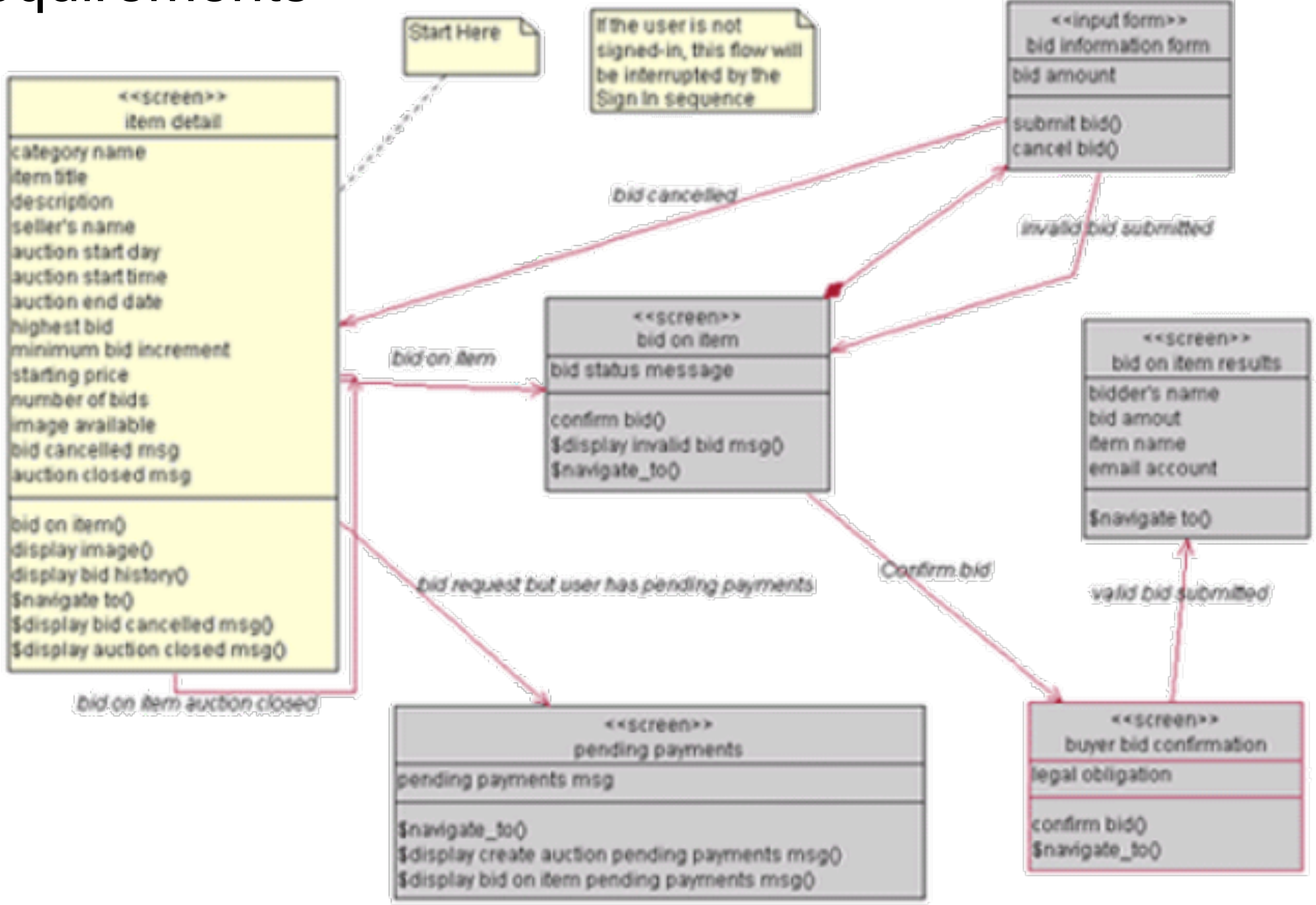


Figure 20: Navigation diagram for the Bid on Item use case

Wrap-Up & Outlook

- Intro, what is HCI?
- User Requirements
- Principles in HCI
- Designing Interactive Systems
- User Study design and Statistics
- Models in HCI
- Capabilities of Humans and machines
- Implementing Interactive Systems

- Mensch-Maschine-Interaktion 2:
  - HCI and the Web
  - Interactive Surfaces
  - Mobile and Ubiquitous User Interfaces
- CG3 = Information Visualization

References

- B. Shneiderman: Designing the User Interface: Strategies for Effective Human-Computer Interaction, Third Edition. 1997.
- A. Cooper: About Face 2.0. Chapter 1 and 19 ff.
- Alan Dix, Janet Finlay, Gregory Abowd and Russell Beale: Human-Computer Interaction (third edition), Prentice Hall 2003
- D. A. Norman. The Design of Everyday Things. Basic Books 2002. Chapter 4.
- Jennifer Tidwell: Designing Interfaces - Patterns for Effective Interaction Design, O'Reilly 2005 http://designinginterfaces.com/
- Ahmed Seffah, Jan Gulliksen, Michel C. Desmarais (eds.): Human-Centered Software Engineering - Integrating Usability in the Development Process, Springer 2005
- Mary Beth Rosson, John M. Carroll: Usability Engineering - Scenario-Based Development of Human-Computer Interaction, Academic Press 2002
- Deborah Hix and H. Rex Hartson: Developing User Interfaces Ensuring Usability Through Product & Process, John Wiley 1993
- Bill Curtis, Bill Hefley: A WIMP no more: the Maturing of User Interface Engineering, ACM *interactions* 1(1), January 1994, 22-34
- John M. Carroll, Mary Beth Rosson: Getting Around the Task-Artifact How to Make Claims and Design Cycle: by Scenario, ACM Transactions on Information Systems, Vol. 10, No. 2, April 1992, 181-212