BINUS UNIVERSITY

People
Innovation
Excellence

Course : COMP6100/Software Engineering

Effective Period : Desember 2017

Design Concepts and Engineering

Session 06

BINUS UNIVERSITY

Acknowledgement

These slides have been adapted from Pressman, R.S. (2015). *Software Engineering : A Practioner's Approach. 8th ed*. McGraw-Hill Companies.Inc, Americas, New York. ISBN : 978 1 259 253157. Chapter 12,13, 14, 15, 16, 17 and 18

Learning Objectives

LO 2 : Explain the software engineering practices and business environment

People
Innovation
Excellence

BINUS UNIVERSITY

Contents

- Design Concept
- Architecture Design
- Component-Level Design
- User Interface Design
- Pattern-Based Design
- Web Apps Design
- MobileApps Design

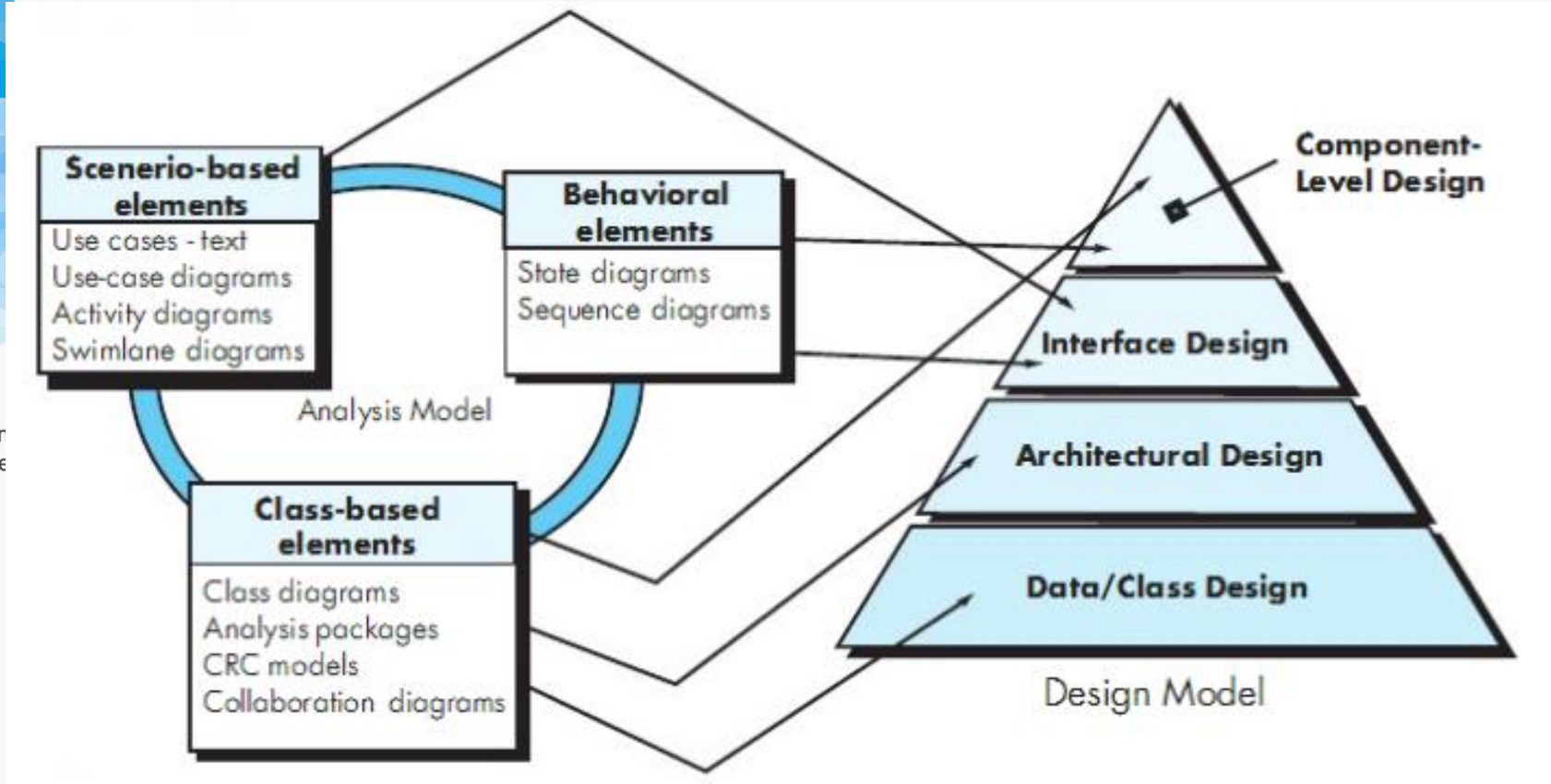
People
Innovation
Excellence

Design Concept

- Mitch Kapor, the creator of Lotus 1-2-3, presented a "software design manifesto" in *Dr. Dobbs Journal.* He said, *Good software design should exhibit:*
  - ***Firmness:*** A program should not have any bugs that inhibit its function.
  - ***Commodity:*** A program should be suitable for the purposes for which it was intended.
  - ***Delight:*** The experience of using the program should be pleasurable one.

Design Concept

Translating the requirements model into the design model



Design Concept

Design and Quality

- The design must implement all of the explicit requirements contained in the analysis model, and it must accommodate all of the implicit requirements desired by the customer.
- The design must be a readable, understandable guide for those who generate code and for those who test and subsequently support the software.
- The design should provide a complete picture of the software, addressing the data, functional, and behavioral domains from an implementation perspective.

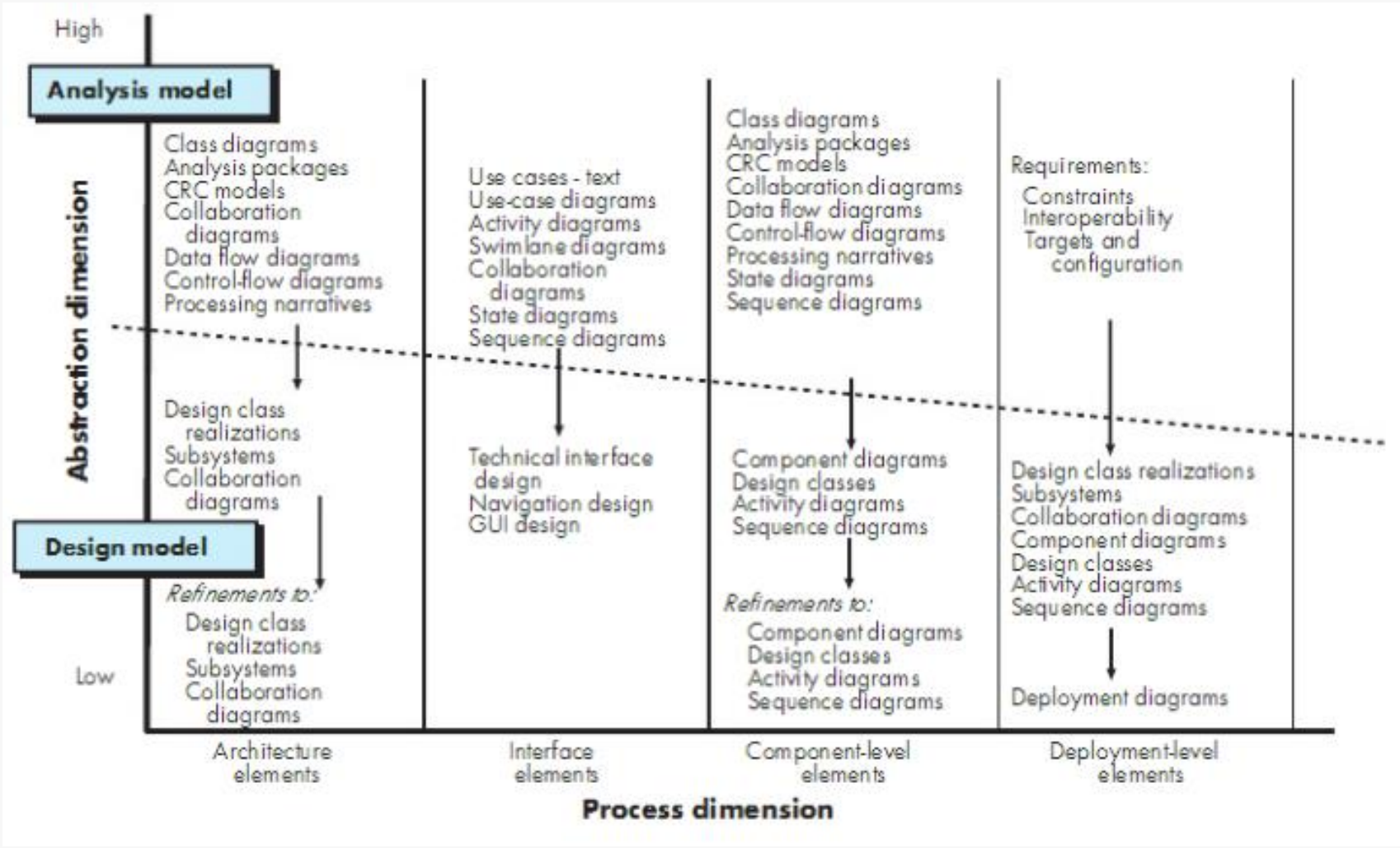
Design Concept

Quality Guidelines

- A design should exhibit an architecture
- A design should be modular
- A design should contain distinct representations.
- A design should lead to data structures that are appropriate.
- A design should lead to components that exhibit independent functional characteristics.
- A design should lead to interfaces that reduce the complexity.
- A design should be derived using a repeatable method.
- A design should be represented using a notation that effectively communicates its meaning.

BINUS UNIVERSITY

Design Model



People
Innovation
Excellence

Architectural Design

The architecture is not the operational software. Rather, it is a representation that enables a software engineer to:

(1) analyze the effectiveness of the design in meeting its stated requirements,

(2) consider architectural alternatives at a stage when making design changes is still relatively easy, and

(3) reduce the risks associated with the construction of the software.

Architectural Design

- The software must be placed into context
  - the design should define the external entities (other systems, devices, people) that the software interacts with and the nature of the interaction
- A set of architectural archetypes should be identified
  - An *archetype* is an abstraction (similar to a class) that represents one element of system behavior
- The designer specifies the structure of the system by defining and refining software components that implement each archetype

Architecture Design

Analyzing Architectural Design

1. Collect scenarios.
2. Elicit requirements, constraints, and environment description.
3. Describe the architectural styles/patterns that have been chosen to address the scenarios and requirements:
    - module view
    - process view
    - data flow view
4. Evaluate quality attributes by considered each attribute in isolation.
5. Identify the sensitivity of quality attributes to various architectural attributes for a specific architectural style.
6. Critique candidate architectures (developed in step 3) using the sensitivity analysis conducted in step 5.

Component- Level Design

Component-Level Design Guidelines

- **Components**
  - Naming conventions should be established for components that are specified as part of the architectural model and then refined and elaborated as part of the component-level model
- **Interfaces**
  - Interfaces provide important information about communication and collaboration (as well as helping us to achieve the OPC)
- **Dependencies and Inheritance**
  - it is a good idea to model dependencies from left to right and inheritance from bottom (derived classes) to top (base classes).

Component- Level Design

Component Design for WebApps

- WebApp component is
  - (1) a well-defined cohesive function that manipulates content or provides computational or data processing for an end-user, or
  - (2) a cohesive package of content and functionality that provides end-user with some required capability.
- Therefore, component-level design for WebApps often incorporates elements of content design and functional design.

Interface Design

Easy to learn?

Easy to use?

Easy to understand?

Typical Design Errors

lack of consistency
too much memorization
no guidance / help
no context sensitivity
poor response
Arcane/unfriendly

People
Innovation
Excellence

Interface Design

Golden Rules

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

Interface Design

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.

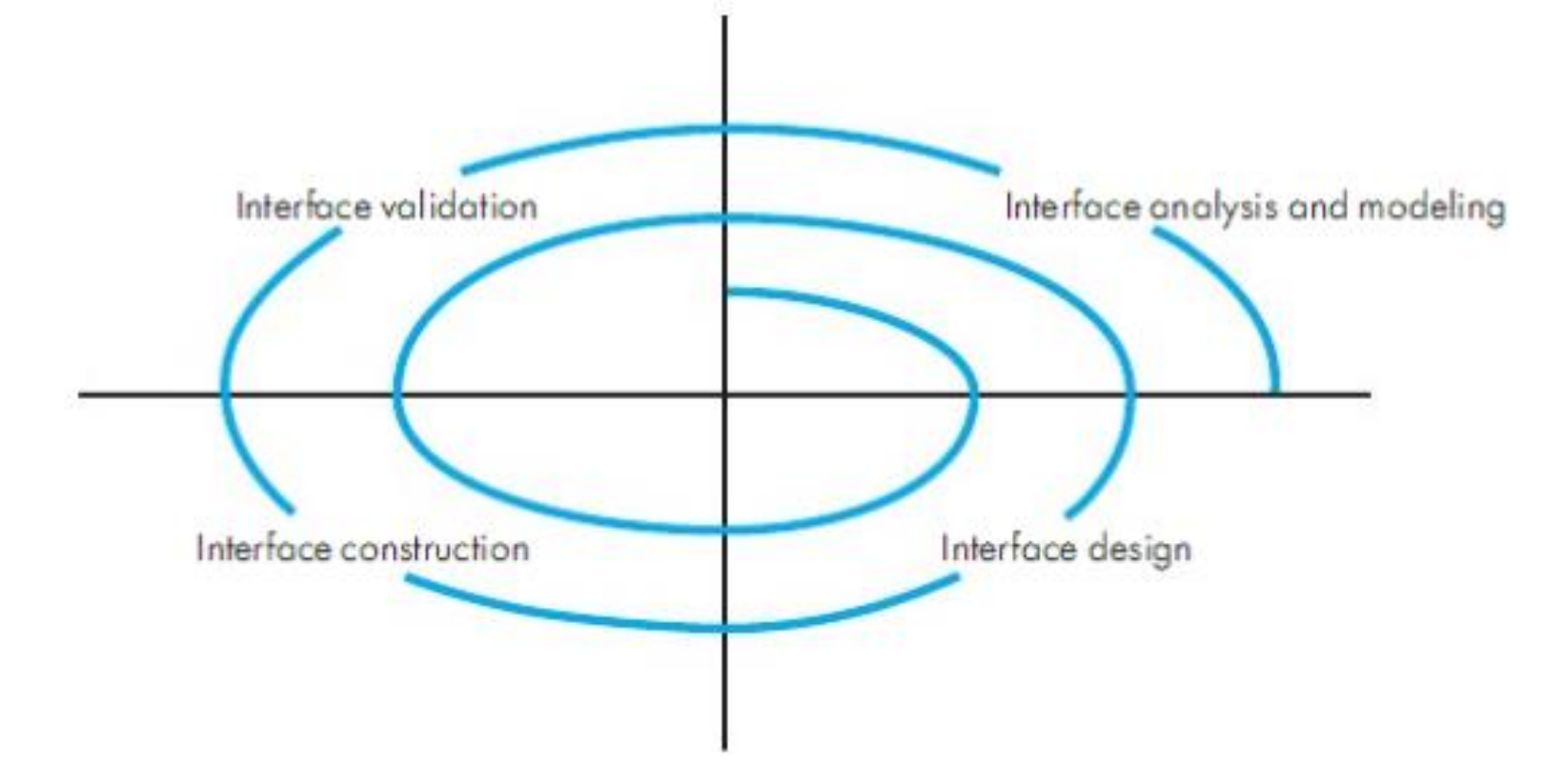
Interface Design

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.

Interface Design

The user interface design process



People
Innovation
Excellence

Interface Design

Design Issues

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

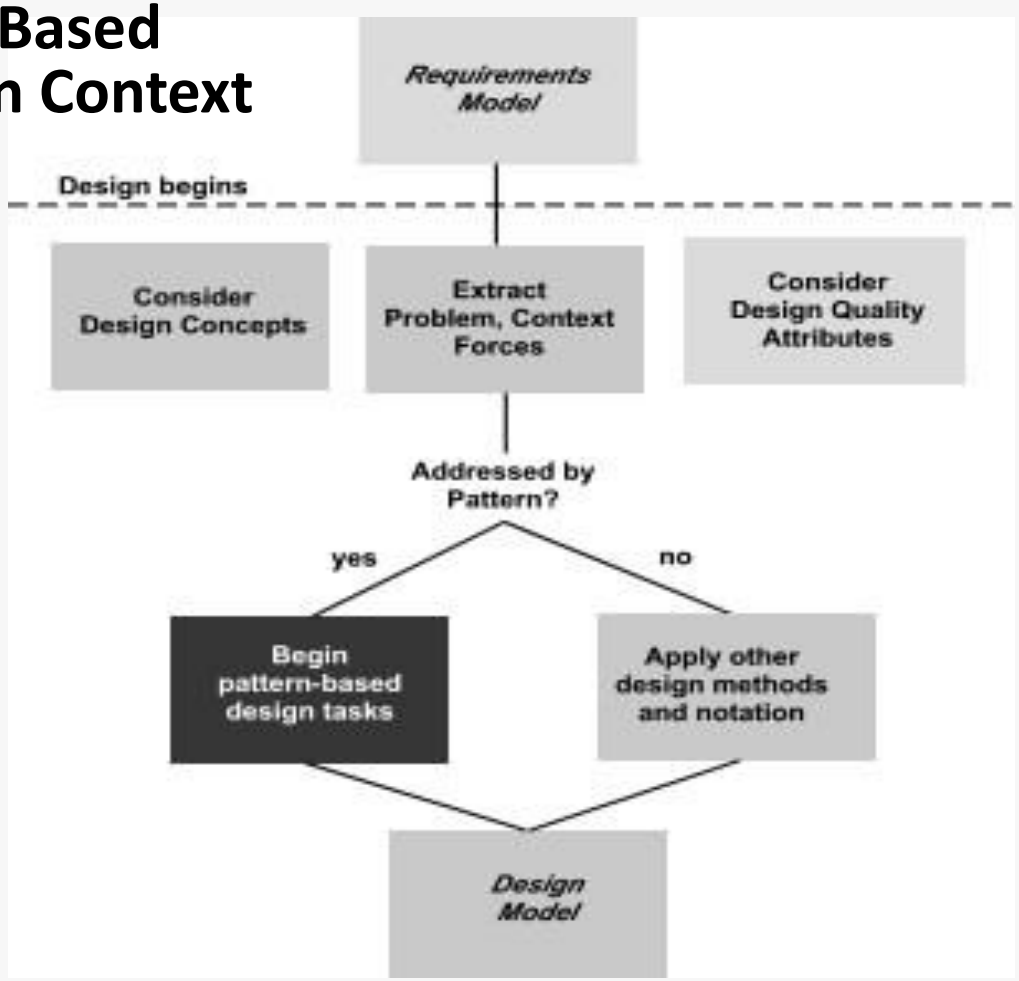
Pattern-Based Design

Effective Patterns

- Coplien [Cop05] characterizes an effective design pattern in the following way:
  - ***It solves a problem***: Patterns capture solutions, not just abstract principles or strategies.
  - ***It is a proven concept***: Patterns capture solutions with a track record, not theories or speculation.
  - ***The solution isn't obvious***: Many problem-solving techniques (such as software design paradigms or methods) try to derive solutions from first principles. The best patterns *generate* a solution to a problem indirectly--a necessary approach for the most difficult problems of design.
  - ***It describes a relationship***: Patterns don't just describe modules, but describe deeper system structures and mechanisms.
  - ***The pattern has a significant human component (minimize human intervention)***. All software serves human comfort or quality of life; the best patterns explicitly appeal to aesthetics and utility.

Pattern-Based Design

Pattern-Based Design in Context



People
Innovation
Excellence

BINUS UNIVERSITY

People
Innovation
Excellence

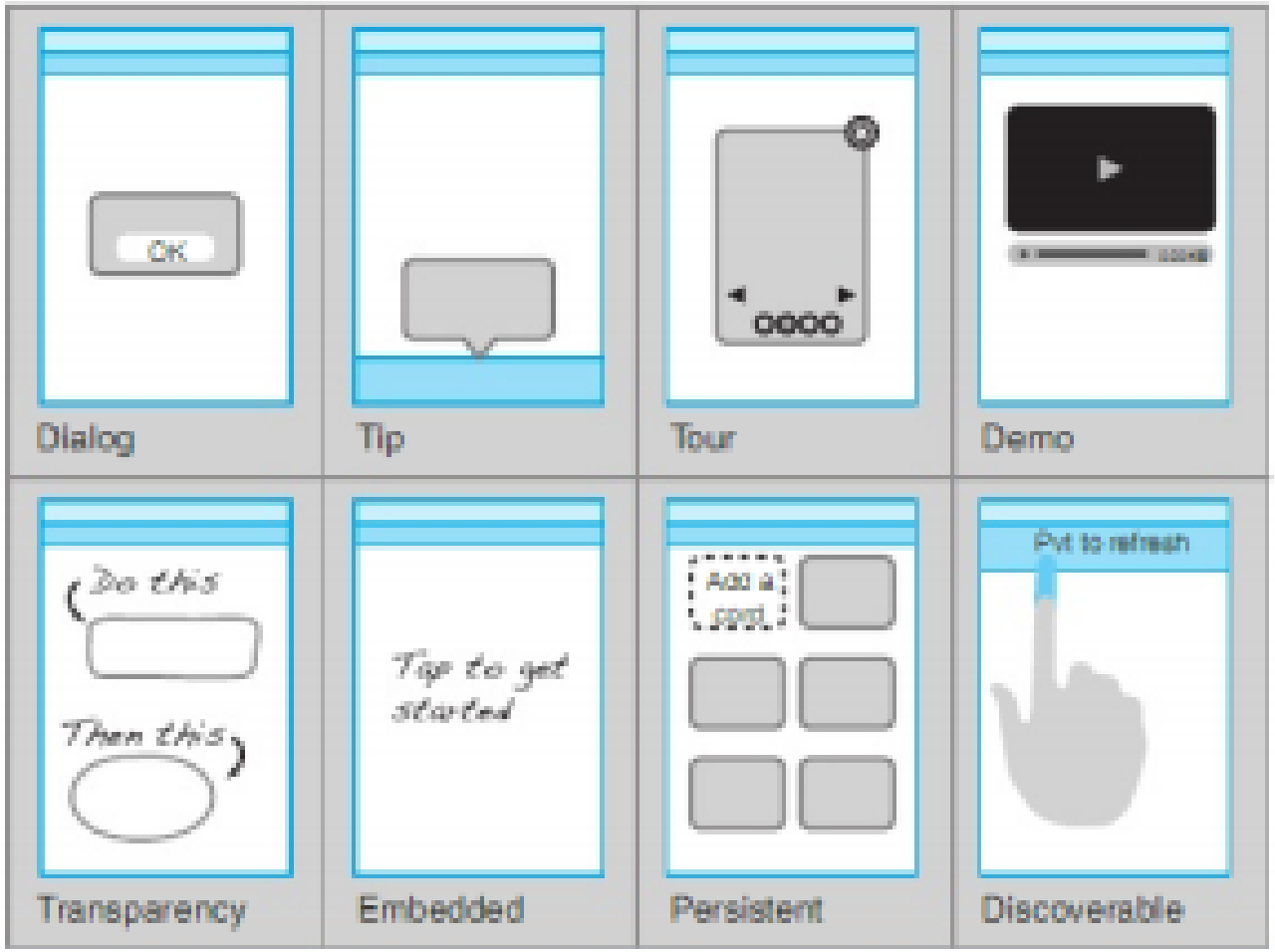
Pattern-Based Design

Pattern Organizing Table

	Database	Application	Implementation	Infrastructure
Data/Content				
Problem statement ...	PatternName(s)		PatternName(s)	
Problem statement ...		PatternName(s)		PatternName(s)
Problem statement ...	PatternName(s)			PatternName(s)
Architecture				
Problem statement ...		PatternName(s)		
Problem statement ...		PatternName(s)		PatternName(s)
Problem statement ...				
Component-level				
Problem statement ...		PatternName(s)	PatternName(s)	
Problem statement ...				PatternName(s)
Problem statement ...		PatternName(s)	PatternName(s)	
User Interface				
Problem statement ...		PatternName(s)	PatternName(s)	
Problem statement ...		PatternName(s)	PatternName(s)	
Problem statement ...		PatternName(s)	PatternName(s)	

Pattern-Based Design

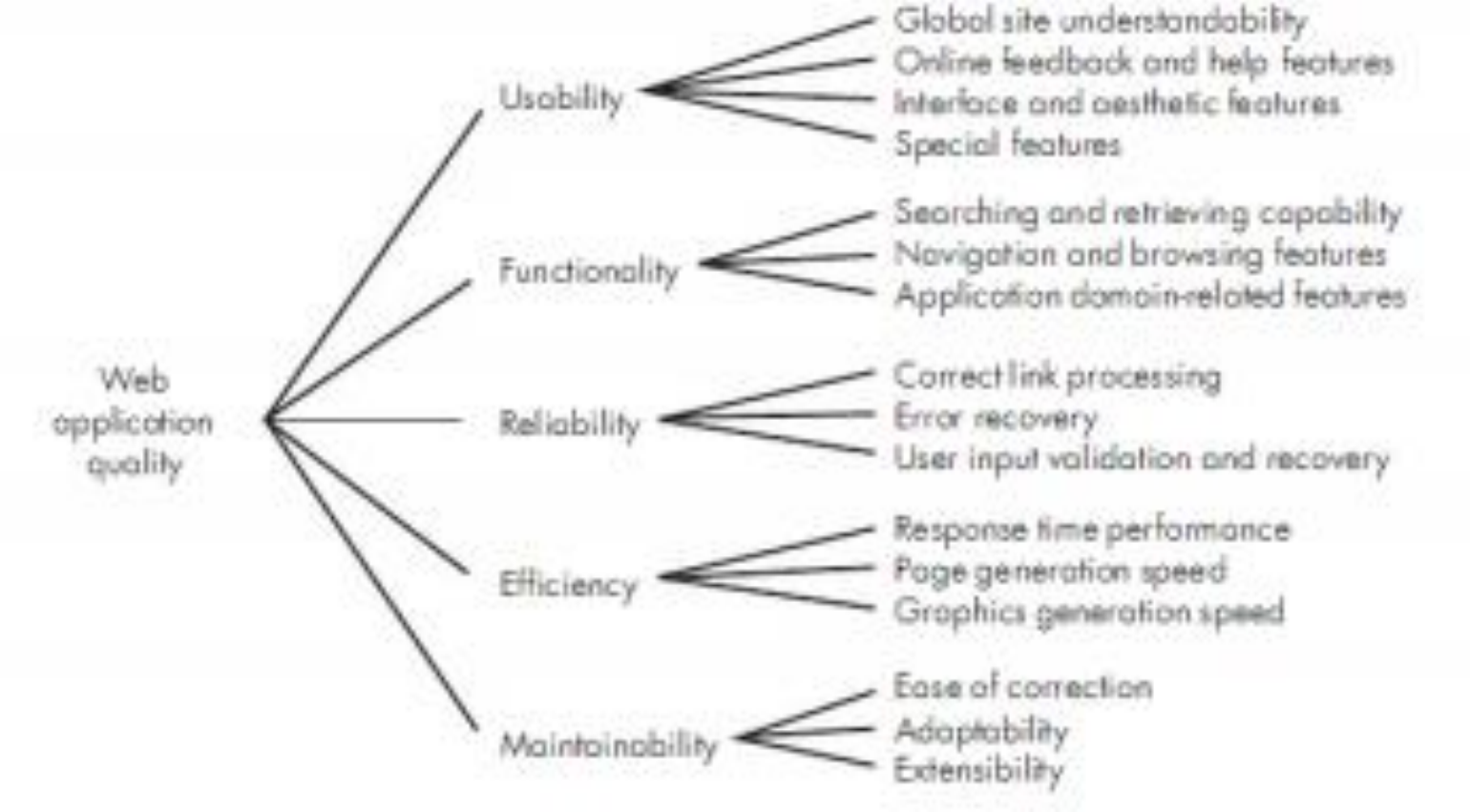
Example of mobile invitation patterns



People
Innovation
Excellence

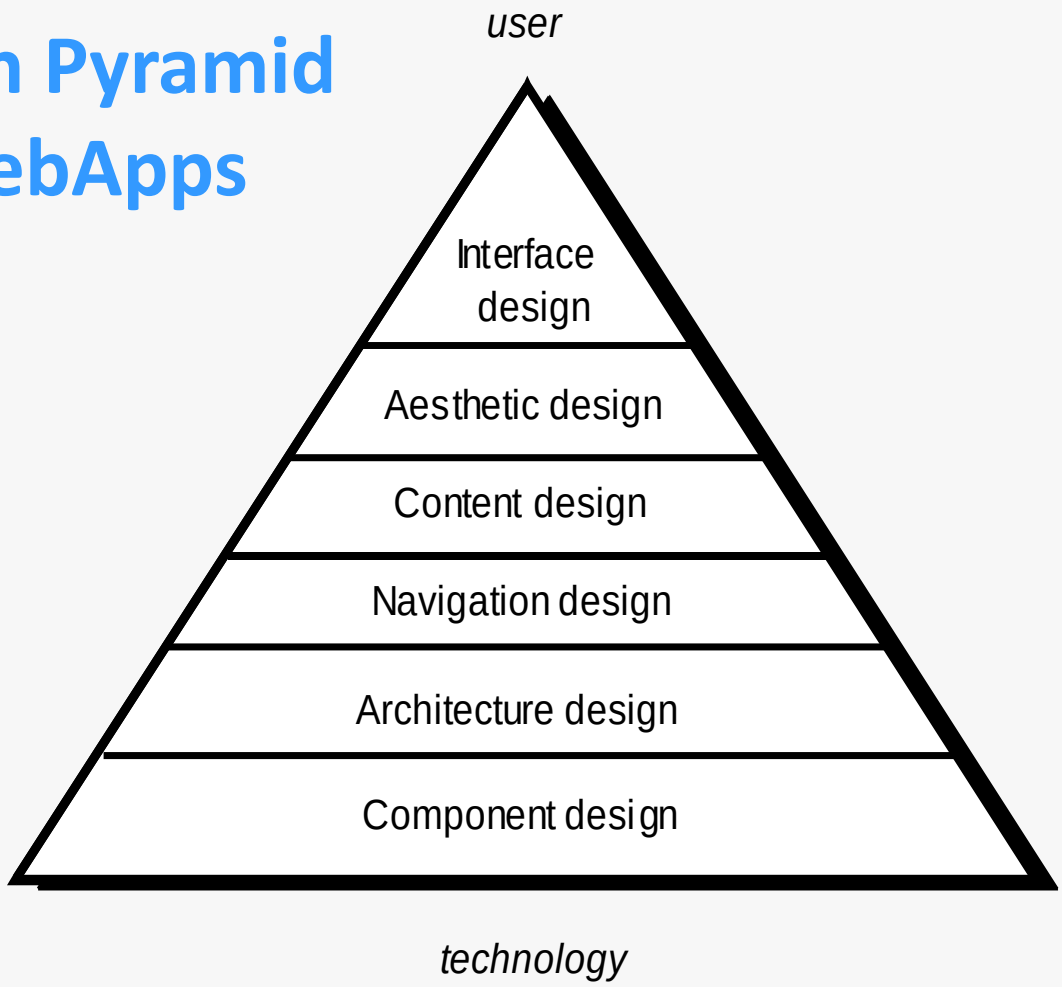
WebApp Design

Quality requirements tree



WebApp Design

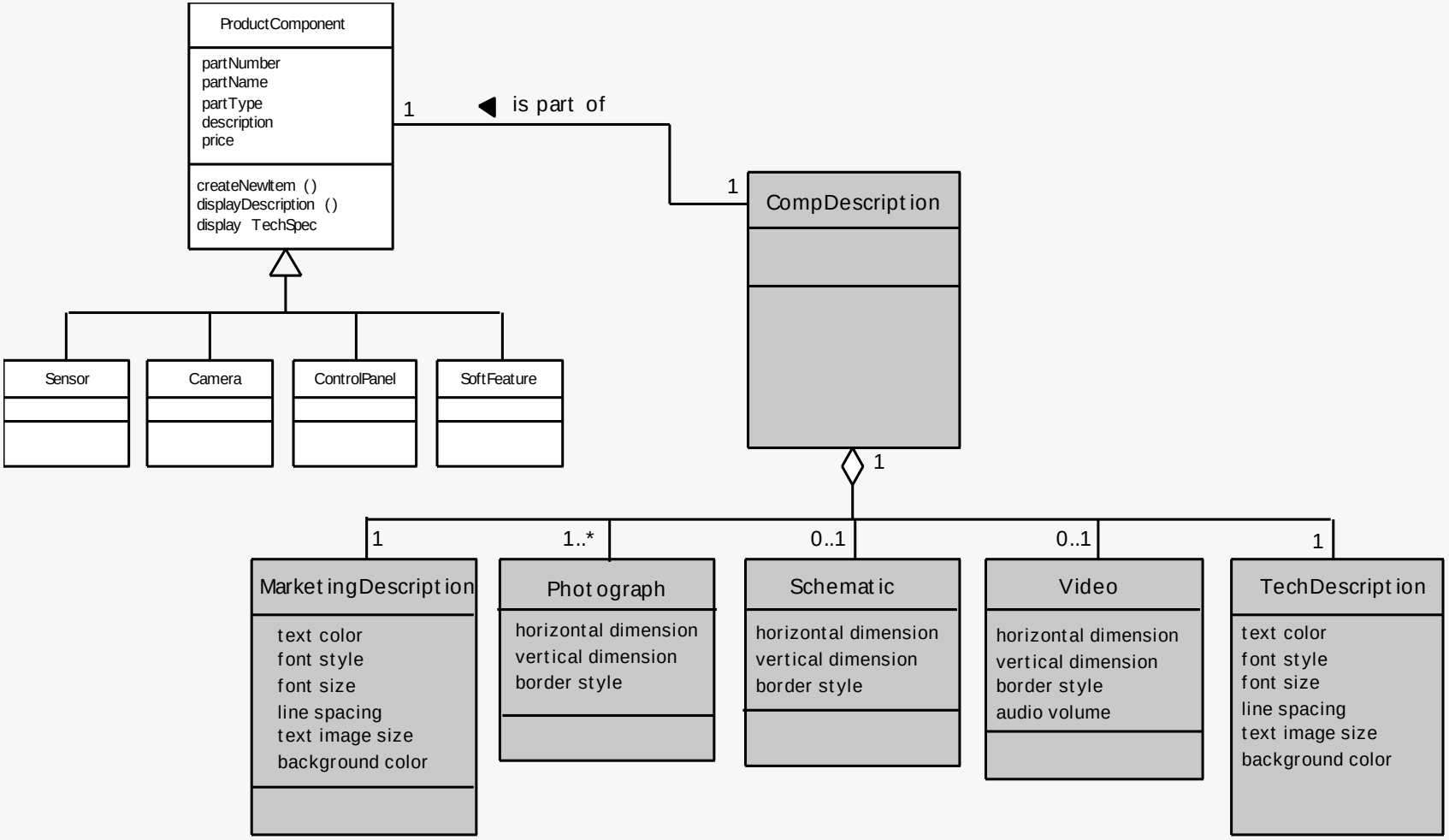
Design Pyramid for WebApps



People
Innovation
Excellence

WebApp Design

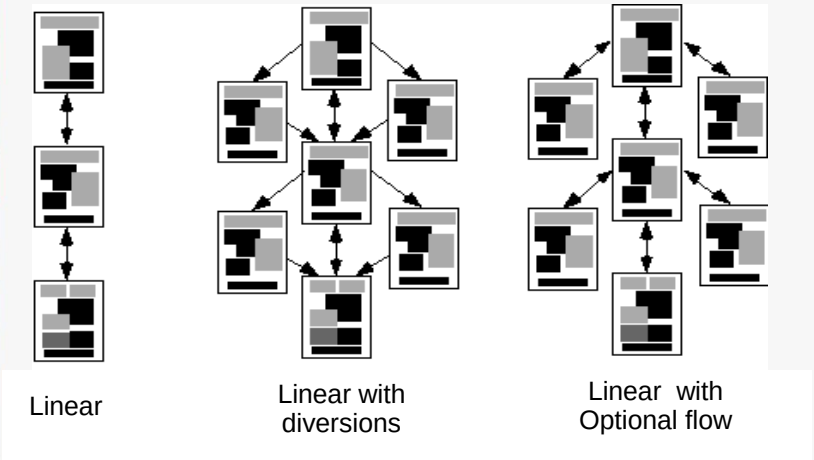
Design of Content Objects

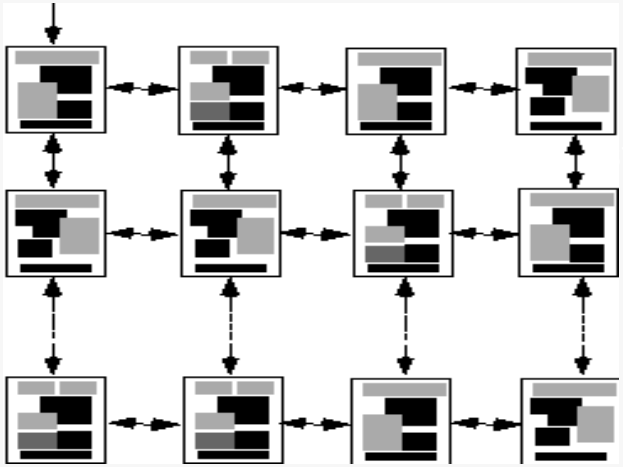


People
Innovation
Excellence

WebApp Design

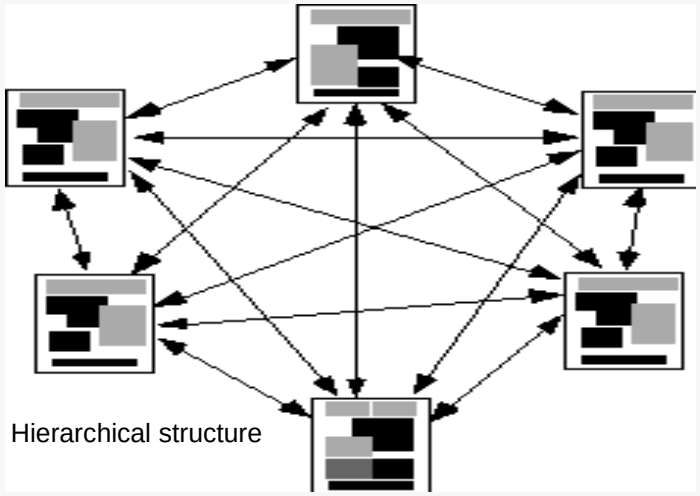
Content Architecture

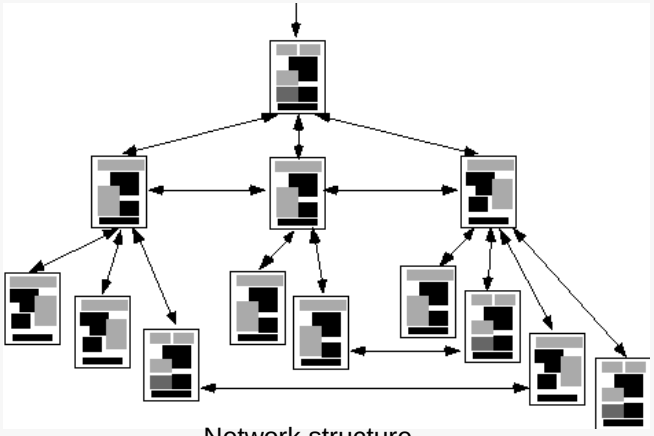




Grid Structure

People
Innovation
Excellence





Network structure

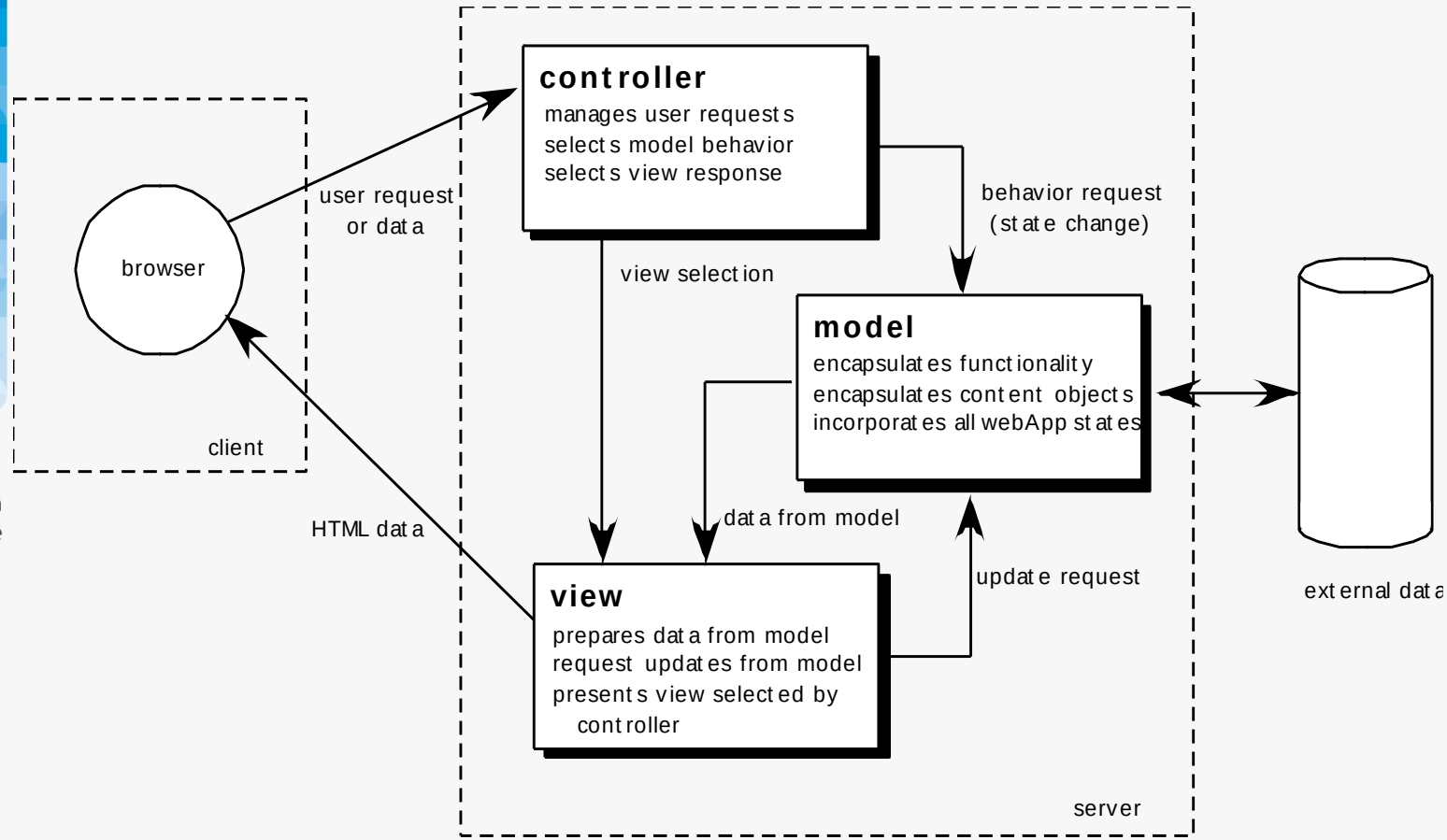
WebApp Design

MVC Architecture

- The ***model*** contains all application specific content and processing logic, including
  - all content objects
  - access to external data/information sources,
  - all processing functionality that are application specific
- The ***view*** contains all interface specific functions and enables
  - the presentation of content and processing logic
  - access to external data/information sources,
  - all processing functionality required by the end-user.
- The ***controller*** manages access to the model and the view and coordinates the flow of data between them.

WebApp Design

MVC Architecture



People
Innovation
Excellence

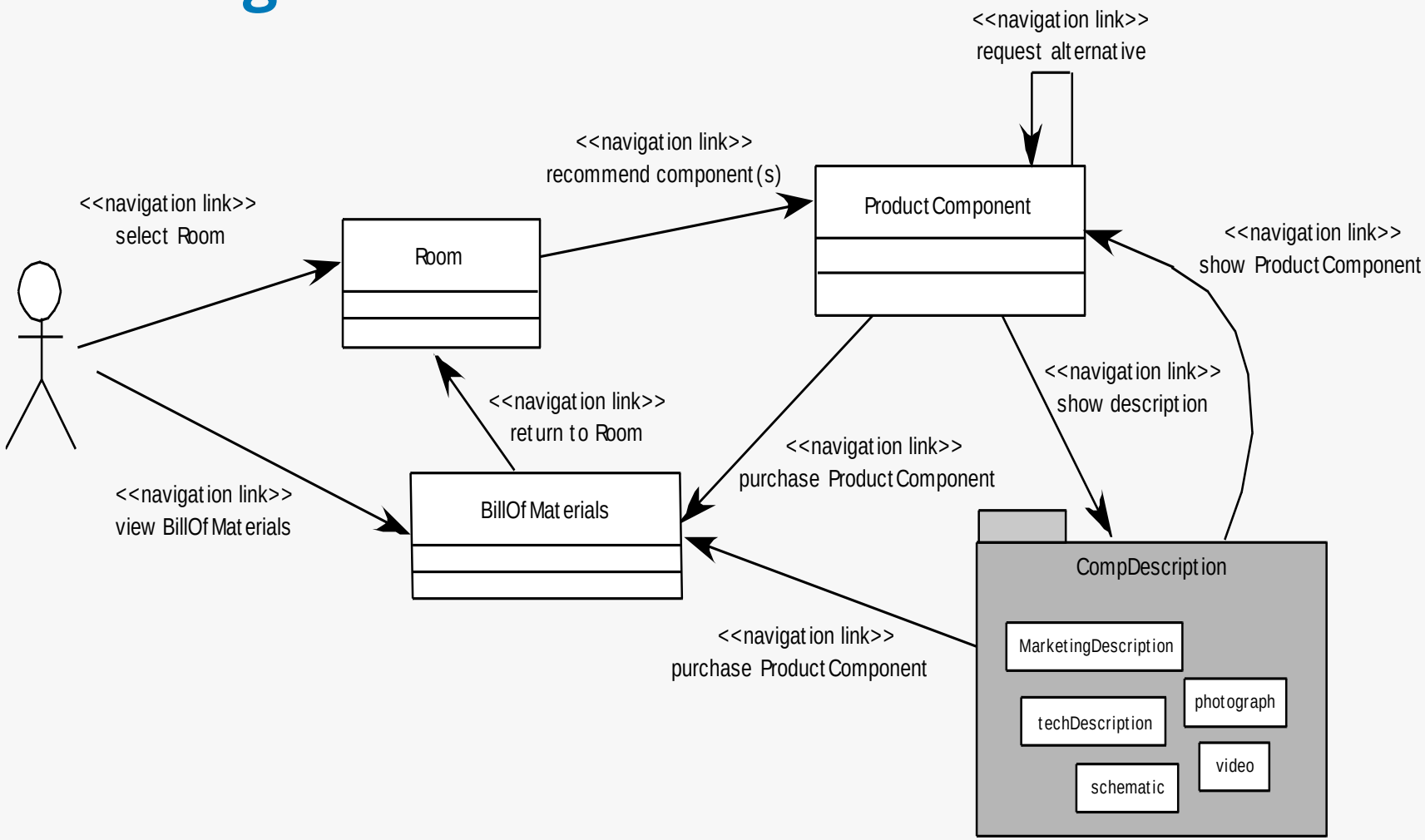
WebApp Design

MVC Architecture - Navigation Design

- Begins with a consideration of the user hierarchy and related use-cases
  - Each actor may use the WebApp somewhat differently and therefore have different navigation requirements
- As each user interacts with the WebApp, she encounters a series of *navigation semantic units* (NSUs)
  - NSU—"a set of information and related navigation structures that collaborate in the fulfillment of a subset of related user requirements"

WebApp Design

Creating an NSU



People
Innovation
Excellence

WebApp Design

Navigation Syntax

- ***Individual navigation link***—text-based links, icons, buttons and switches, and graphical metaphors..
- ***Horizontal navigation bar***—lists major content or functional categories in a bar containing appropriate links. In general, between 4 and 7 categories are listed.
- *Vertical navigation column*
  - lists major content or functional categories
  - lists virtually all major content objects within the WebApp.
- ***Tabs***—a metaphor that is nothing more than a variation of the navigation bar or column, representing content or functional categories as tab sheets that are selected when a link is required.
- ***Site maps***—provide an all-inclusive tab of contents for navigation to all content objects and functionality contained within the WebApp.

MobileApp Design

Developing MobileApps

- Formulation
- Planning
- Analysis
- Engineering
- Implementation and Testing
- User Evaluation

People
Innovation
Excellence

MobileApp Design

User Interface Design

- Is the user interface consistent across applications ?
- Is the device interoperable with different network services ?
- Is the device acceptable in term of stakeholder values in the target market area ?

BINUS UNIVERSITY

MobileApp Design

MobileApp Design – Best Practices

Some important consideration when designing mobile touch screen applications listed by Schumacher include :

- Identify your audience
- Design for context of use
- There is a fine line between simplicity and laziness
- Use the platform as an advantage
- Make scrollbars and selection highlighting more salient
- Increase discoverability of advanced functionality
- Use clear and consistent labels
- Clever icons should never be developed at the expense of user understanding
- Support user expectations for personalization
- Long scrolling forms trump multiple screens on mobile devices

References

- Pressman, R.S. (2015). ***Software Engineering : A Practioner's Approach. 8th ed***. McGraw-Hill Companies.Inc, Americas, New York. ISBN 978 1 259 253157.
- **Introduction to Software Architecture, http://www.youtube.com/watch?v=x30DcBfCJRI**
- **Component-based game engine, http://www.youtube.com/watch?v=_K4Mc3t9Rtc**
- Software design pattern, http://www.youtube.com/watch?v=ehGl_V61WJw

BINUS UNIVERSITY

Q & A

People
Innovation
Excellence