中山大學
SUN YAT-SEN UNIVERSITY

Lecture 26
Web Engineering Basics

SE-805 Web 2.0 Programming (supported by Google)

http://my.ss.sysu.edu.cn/courses/web2.0/

School of Software, Sun Yat-sen University

Outline

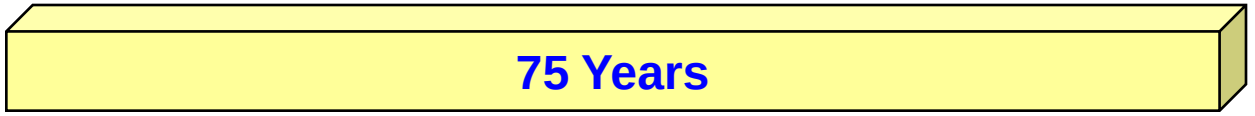
- **Web Engineering & Web applications**
- Web engineering process & Management
- Requirements for WebApps analysis
- Design Modeling for WebApps

Web Engineering is important

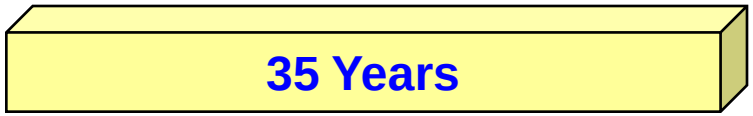
- The World Wide Web and the Internet that empowers it are arguably the most important developments is the history of computing.
  - Billions more person share information any where
  - Create information together and collaborate
- Web Engineering (WebE) is the process that is used to create high-quality *ubiquitously usable* Web App.
  - Process
  - Methods
  - Tools
- Based on your practice, what's new you have known that contrast with definition of Software Engineering?
  - Techniques, method, tools, process
  - Requirements, design, implementing, testing,……

Some facts: Rate of Growth

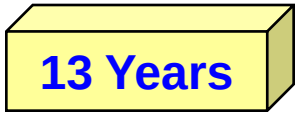
- Time to reach 50 million people



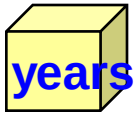
Telephone



Radio



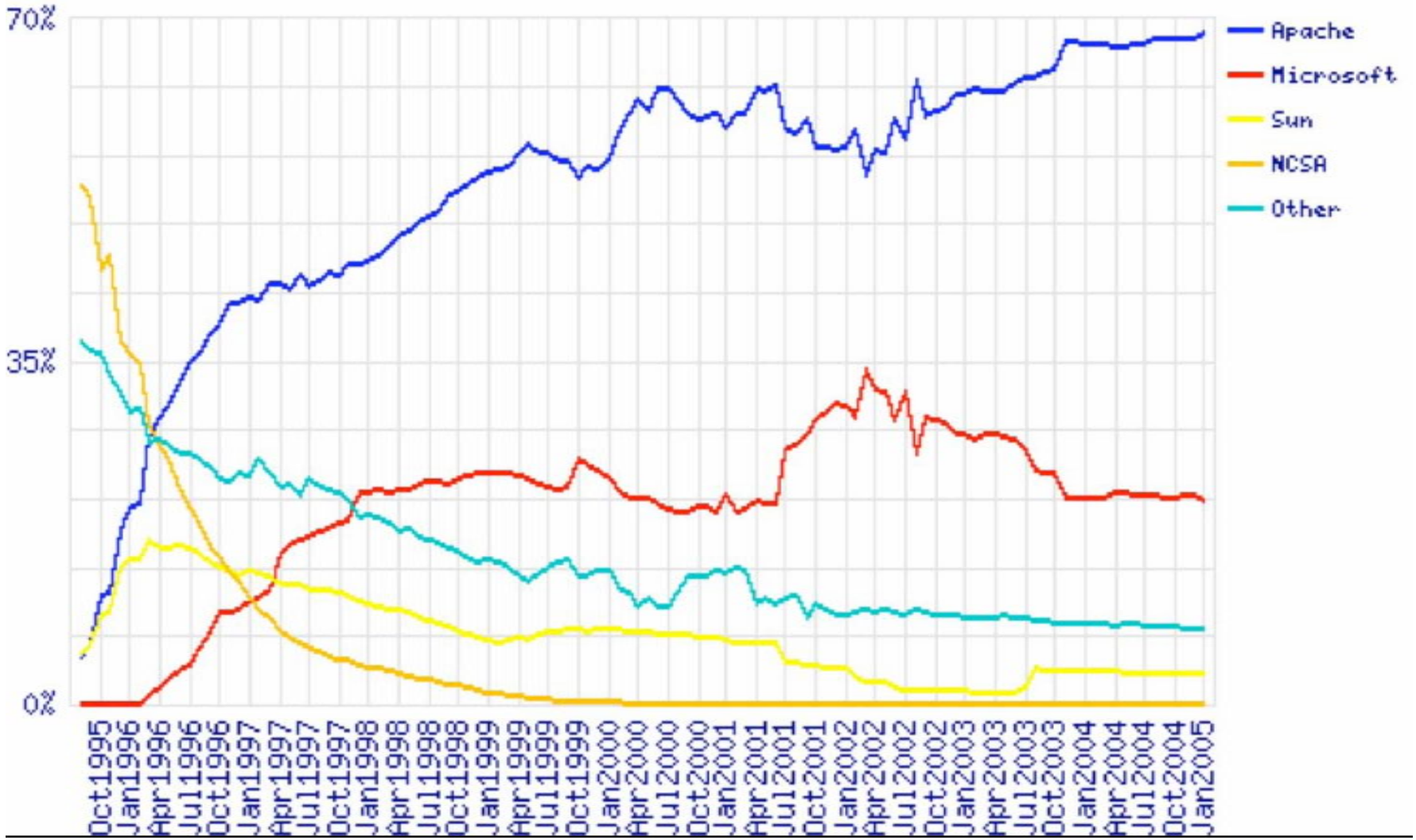
Television



The Web

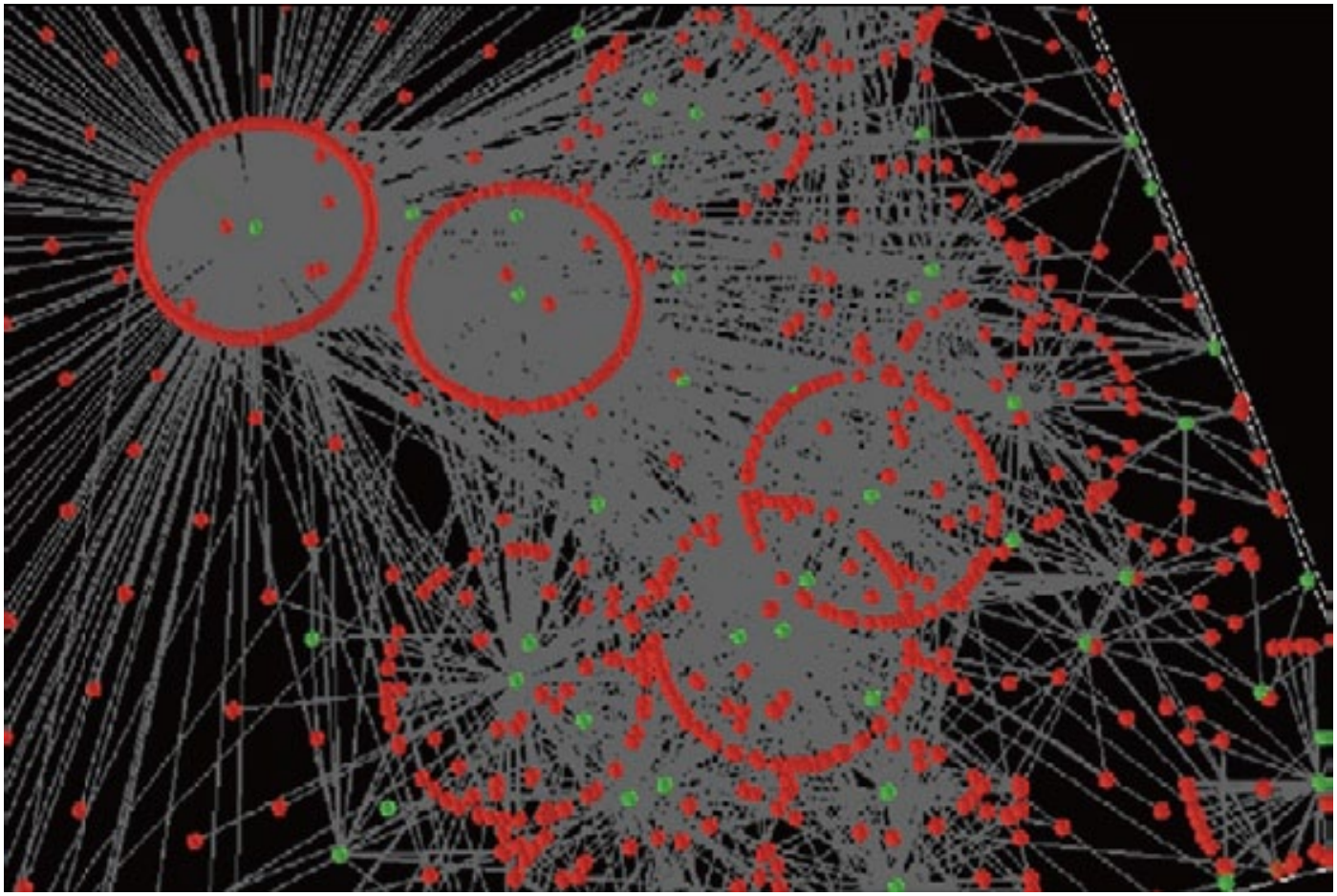
Some facts: Growth of Web Servers

Market Share for Top Servers Across All Domains August 1995 - January 2005



Posted by wss at 11:24 PM UTC on Jan 1, 2005 in Web Server Survey

Some facts: Complexity



Mush up Web Service with two dimension diagram

(Green node is Web Service, Red node is mush up application)

True or False?

- Web Application is always βversion online.
- Web Application can satisfy all customs.
- Web Application as Service
- Web Engineering is now era of Software Engineering
- Ad-hoc is keyword in Web Application
- Software can not be built by users
- Service on demand promoted by IBM
- Cloud Computing is now Technique in Web Engineering

Web VS. Software Engineering

	Software Engineering	*Web engineering*
User range	Small	Large
Number of simultaneous users	Small	Very Large
User Requirements	Specific	Changes Fast
Growth and Change	Slow	Fast
Design and Development expertise	Few	Wide Range
Budgets	Varies in a wide range according to the size of the company	Comparatively Small and varies in a wide range
Timing Constrains	Longer time (3 months-2 years)	Comparatively less time (1 week – 4 months)
Hardware and Software Environments constrains	Specific	Unknown (should cater for all possible combinations)
Adherence to Standards and Protocols	Not much important	Very Much important
Security and Legal issues	Not much important	Very Much important
Look and Feel of the Final Product	Not much important	Varies according cultures, access devices etc.

We Have Done this before...

Evolution of Engineering Disciplines

- Civil Engineering
  - Leads back to 3000 BC
- Mechanical Engineering
  - Leads back to 100 BC
- Electrical Engineering
  - Leads back to 1900
- Telecommunication Engineering
  - Leads back to 1840
- Software Engineering
  - Early stages - 1955, As an Engineering Practice 1970's
- Web Engineering
  - In the Process of **Making the history**

Outline

- Web Engineering & Web applications
- **Web engineering process & Management**
- Requirements for WebApps analysis
- Design Modeling for WebApps

The Web Engineering Scenario

System Conception & Stakeholders

用户

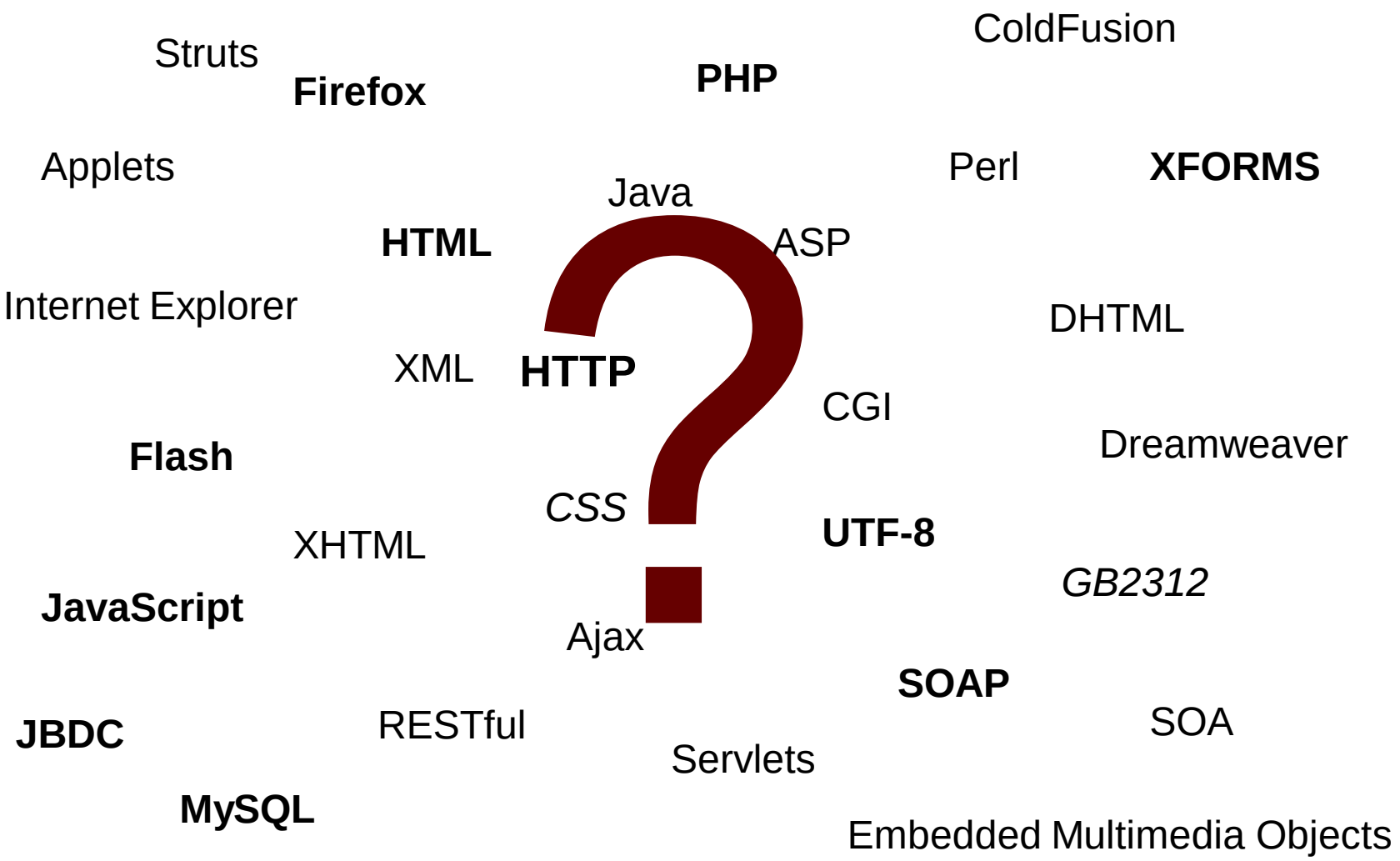
投资人

运营商

业务专家

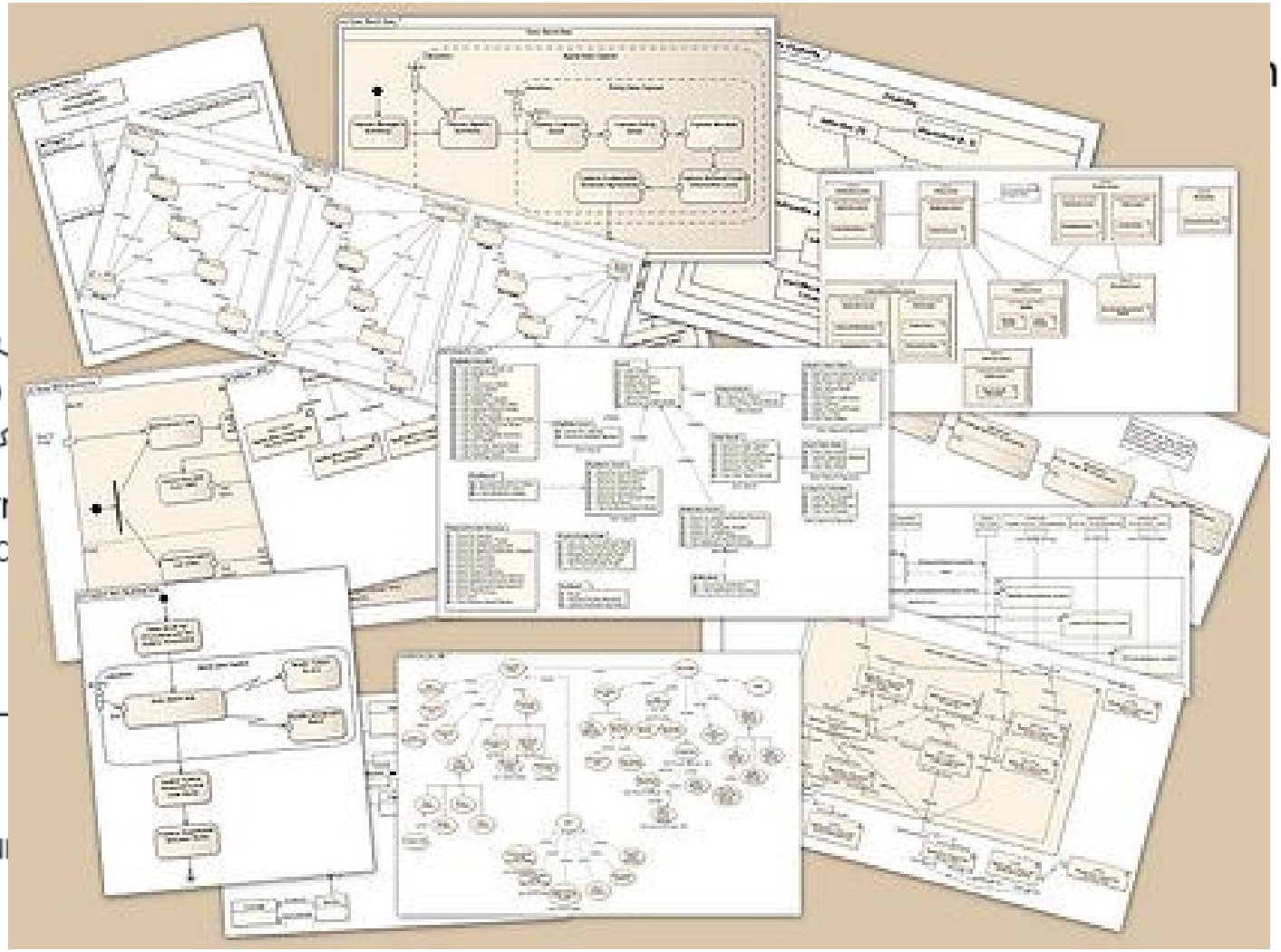
The Web Engineering Scenario

Resource collection & Technique selections



The Web Engineering Scenario

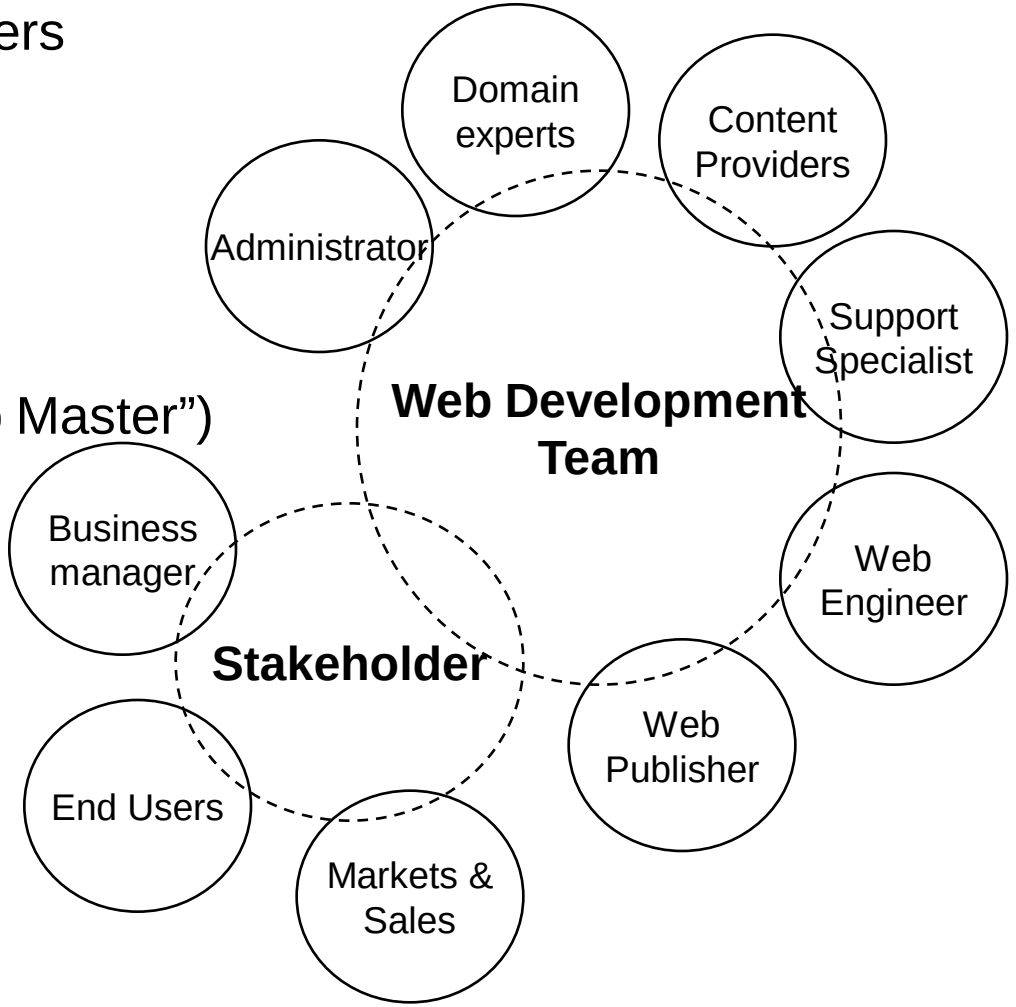
System Analysis Design & Modeling



The Web Engineering Scenario

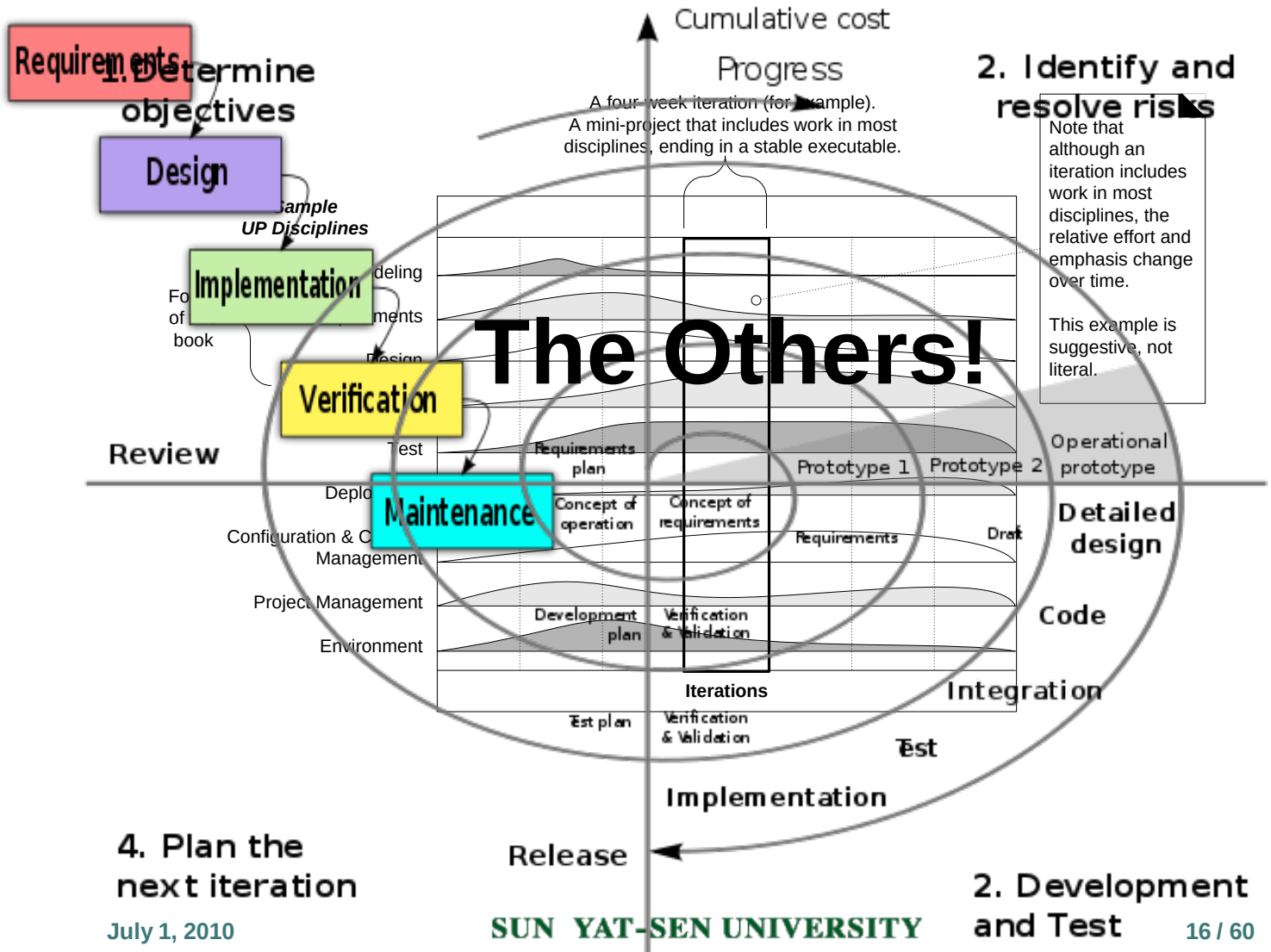
Team building

- WebE team roles
  - Content Developer/Providers
  - Web Publisher
  - Web Engineer.
  - Business domain experts
  - Support Specialist
  - Administrator (a.k.a. “Web Master”)



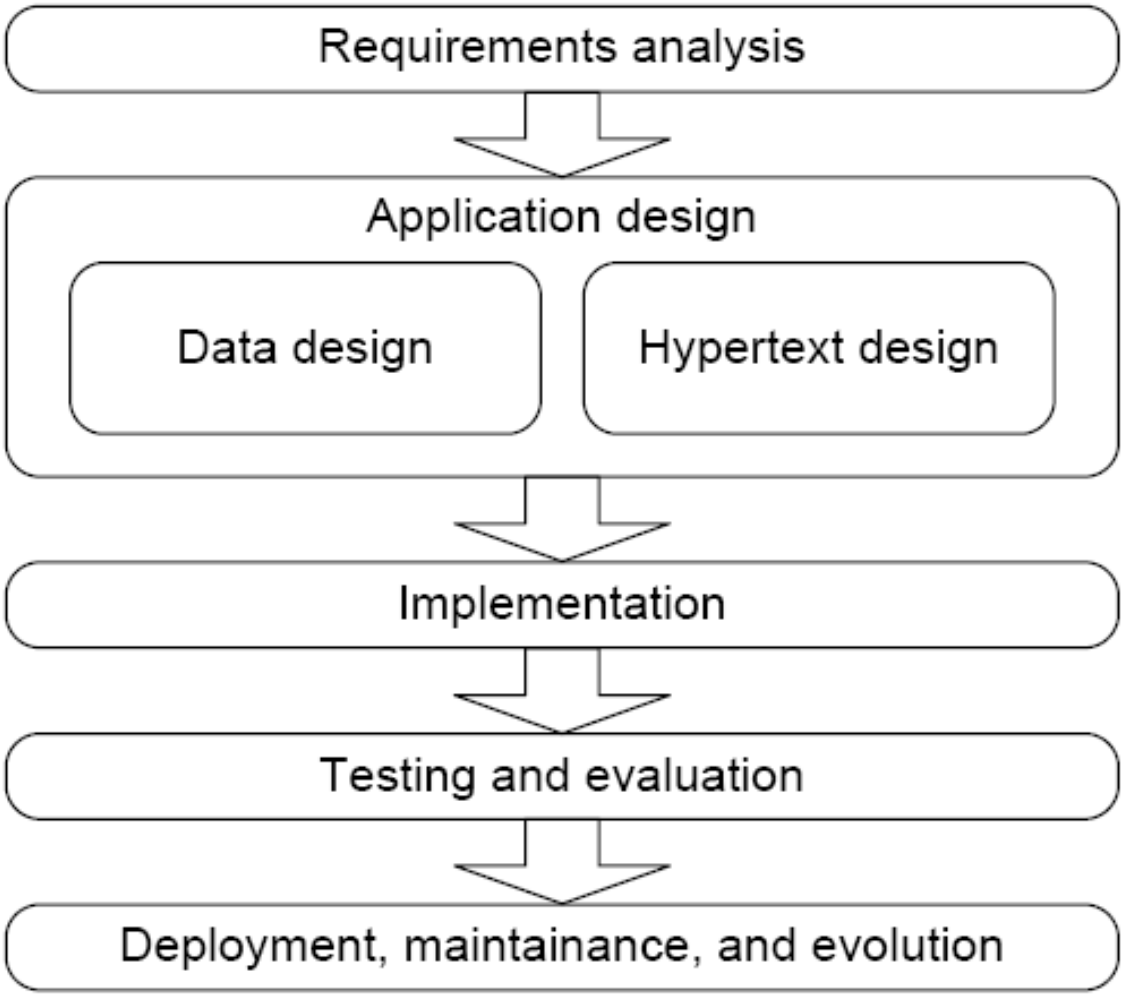
Web Engineering Processing

software life cycle: Waterfall model, Spiral Model, Unified Model,which one?



The WebML Model?

Web Modeling Language (WebML) proposed in 2000, a visual language and development method for specifying the content structure of a Web application



Use cases and business process models

Data model and hypertext model

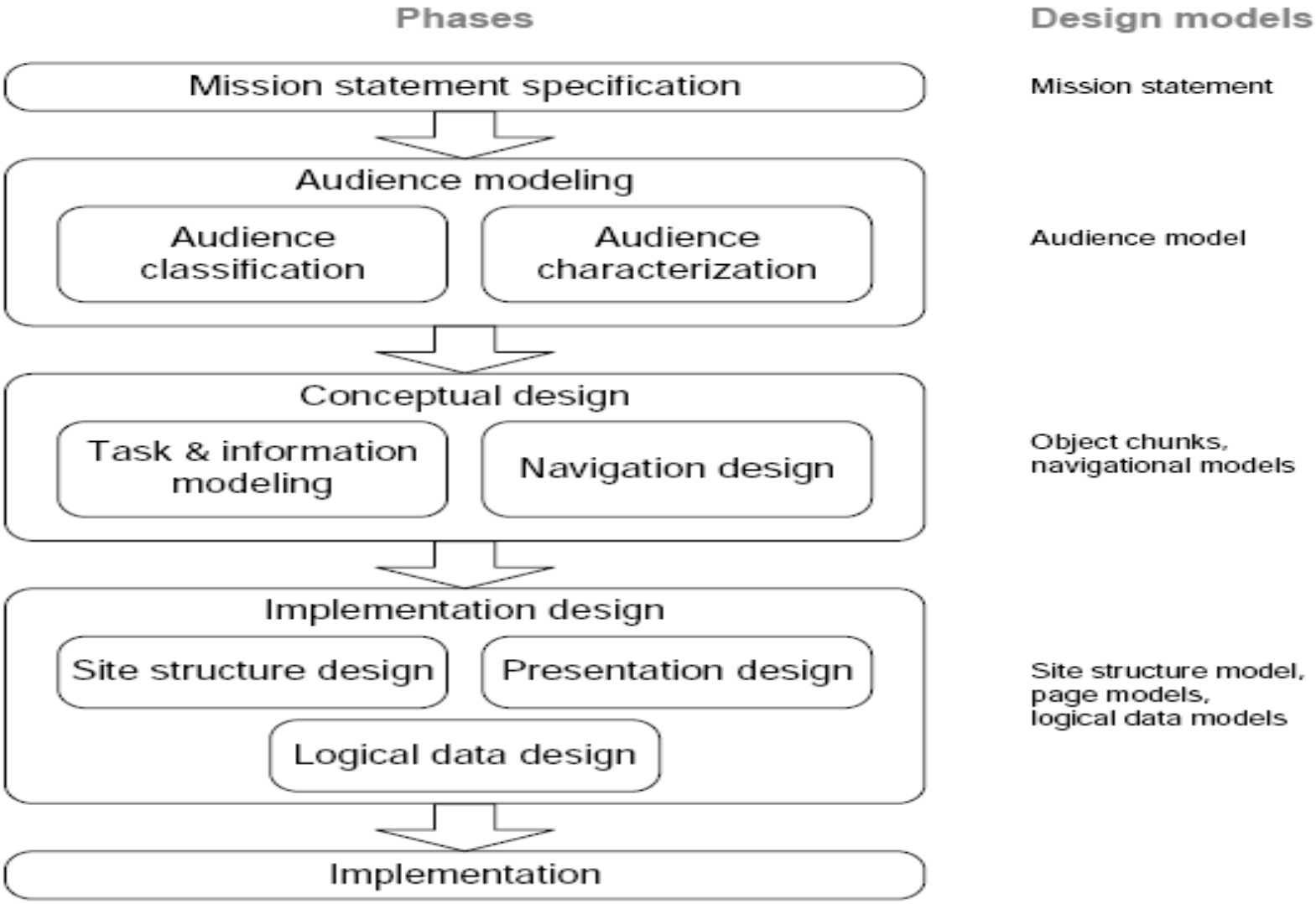
Database and Web application

Assessment of Web application and WebRatio code generator

Conceptual model changes

WSDM*?

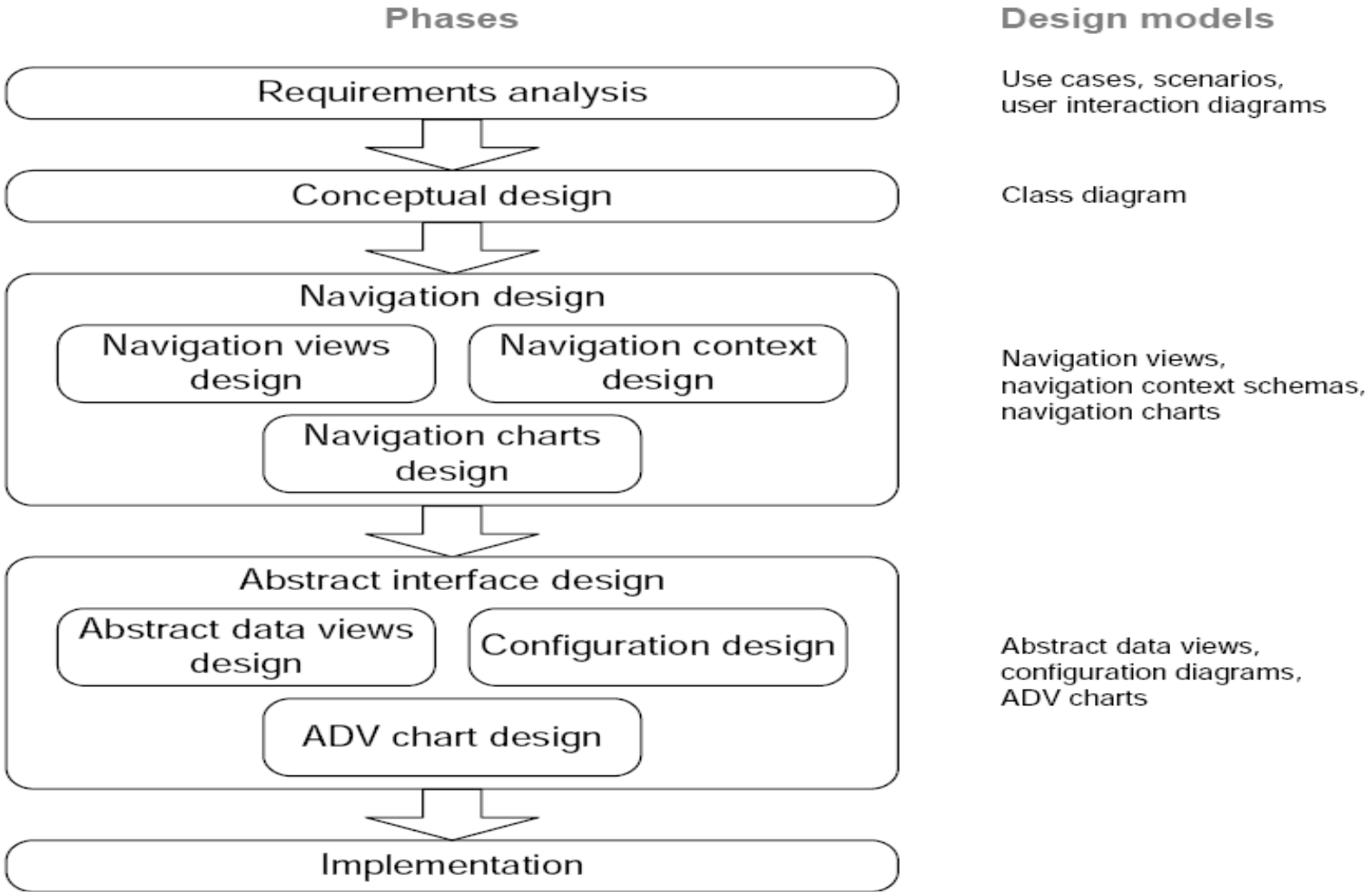
Web Site Design Method was initiated in 1998, The first Web design methods



* Later re-baptized as Web Semantics Design Methods.

The OOHDM Model?

Object-Oriented Hypermedia Design Method (OOHDM) is one of the first methods adopted for Web application development projects.(2000)



Metrics for Web Engineering

- Function Point ?
- Use Case Point ?

- For Static Web Site
  - Pages
  - Links
  - Medias

- For Dynamic Web Site
  - Client functions
  - Mush up services
  - Server actions
  - External Services
  - External Data

WebE Process Summary

- Re-arrange your development processes following general characteristics
  - ***Continuous*** and ***fast*** development and release times are paramount.
  - Web development processes are less documentation-based and, rather, put high emphasis on ***prototypes*** (prototypes are much more expressive than technical documents, especially to unskilled customers).
  - High ***user involvement*** and early feedback is desirable.
  - A new actor enters the development process: the ***graphic designer***.

- Review "The Agile Manifesto"

Outline

- Web Engineering & Web applications
- Web engineering process & Management
- **Requirements for WebApps analysis**
- Design Modeling for WebApps

Case study

- mini-project “Context Driven Call Center On-line System” (CCOS)
- The Scenarios
  - The Customer visit “taobao”[http://www.taobao.com/], and press tag “contact with sale” when he need help
  - The Content Developer want to embed it easy.

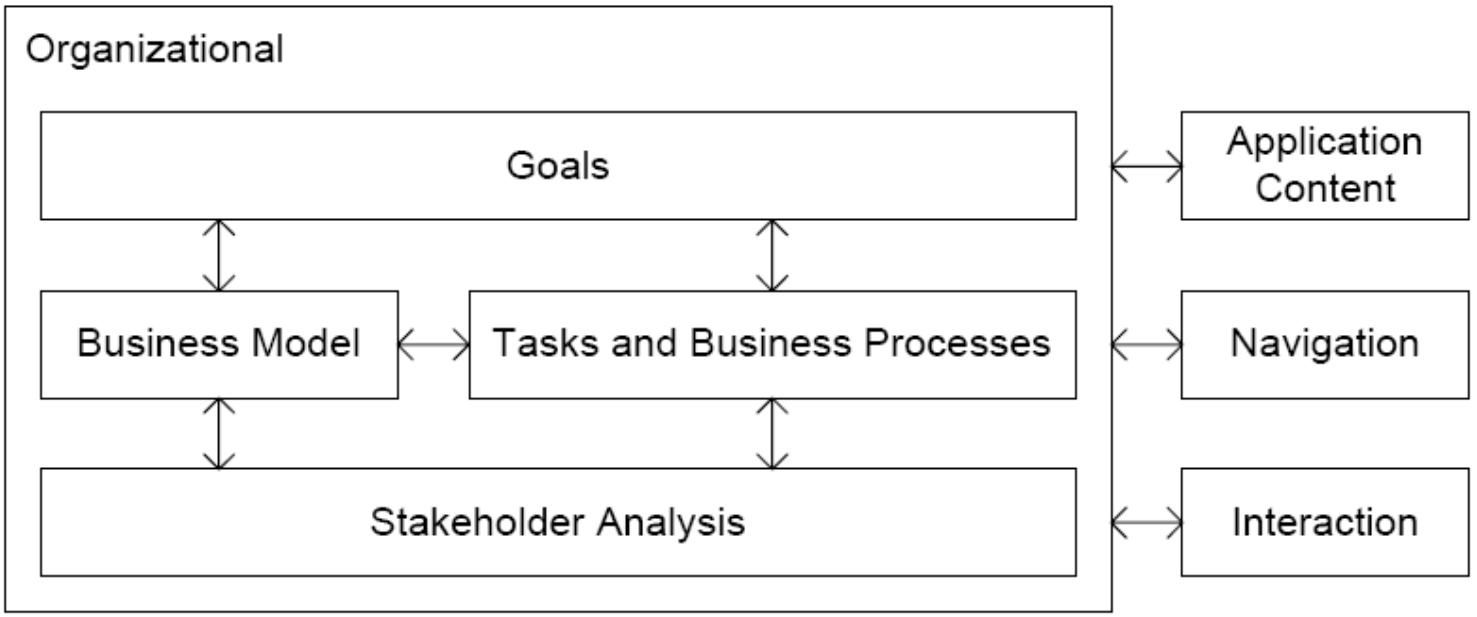
Nokia/诺基亚N97 包安装导航 另有正港行带发票送证书软件光盘

卖家： 国际联讯 和我联系

  - The shop Manager want that the team will give Customers instant services by text or voice on-line. And common Q & A will be listed before service
- As a service provider, you must develop your CCOS just like Taobao
  - Steps and activities
  - Analysis and design method & documentation it
  - Code will not needed

Requirements Relevance to the Web

- Functional requirements
  - Organization Requirements
  - Application Domain Requirements
    - Information content, Information flow, Information structure
  - Navigation Requirements
  - Interaction Requirements



Requirements Relevance to the Web

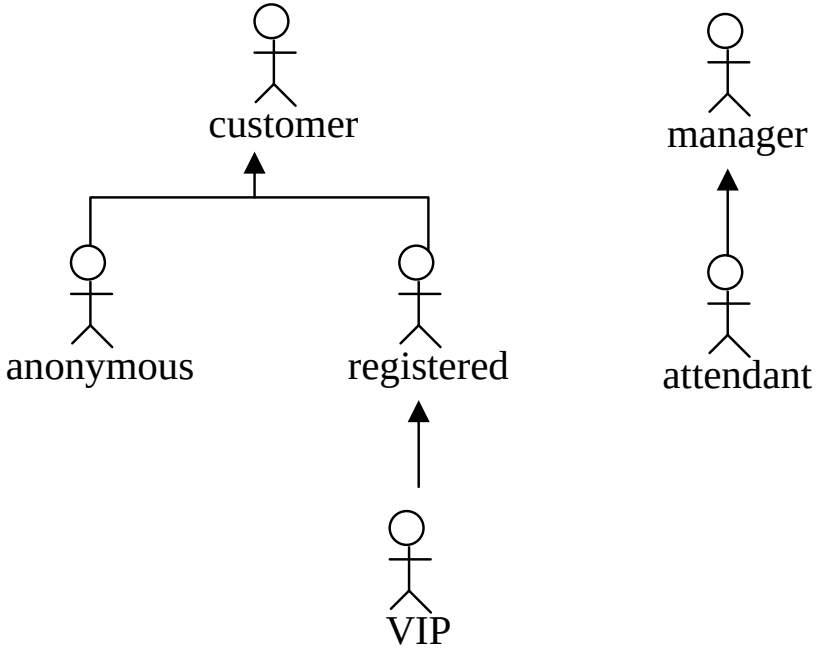
- Nonfunctional requirements
  - Product Requirements: FURPS+ except function
  - Project Organizational Requirements: posed by the customer's and developer's organizations, process standards, programming language, and so on.
  - External requirements: The environment and infrastructure in which the WebApp resides, such as interoperability requirements with external systems

Requirements elaboration and specification

- Organization Requirements Analysis
  - Business Value Model : values will be exchanged between actors that will use the Web application.
  - Business Information Flow Model: information in business.
  - Goals Model : goals of stakeholders in project.
  - Business Process Model : processes of business activities.
  - Task Model : the tasks a user needs to perform.
  - Audience Model : trace matrix of functional requirements
- Application Domain Analysis
- Navigation and Interaction Analysis
  - Navigation Relationships
  - High-Level Interaction and Navigation Units

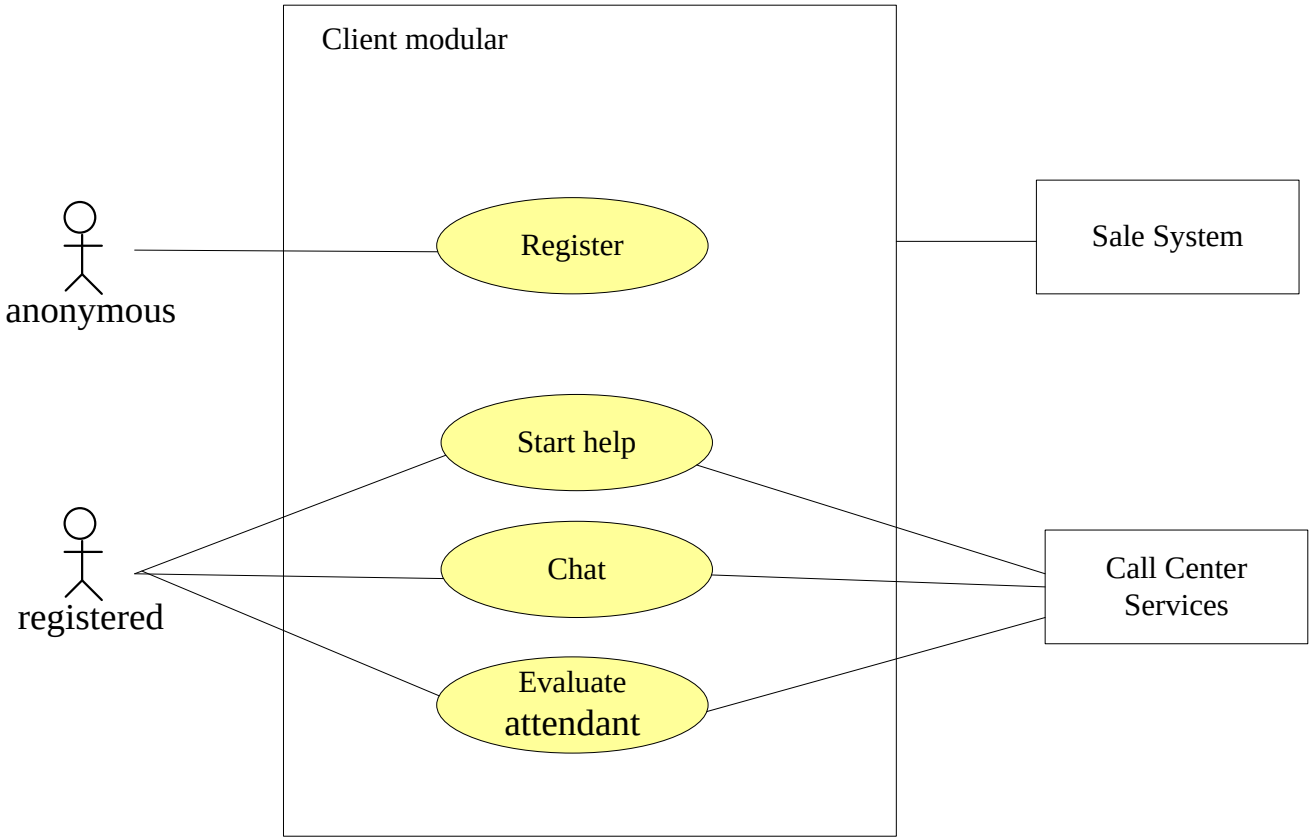
Step1:Finding Actor in Organization

- Goals Model
  - List more business scenarios
  - Find users and modeling with Class Diagram
  - List goals and responsibility briefly



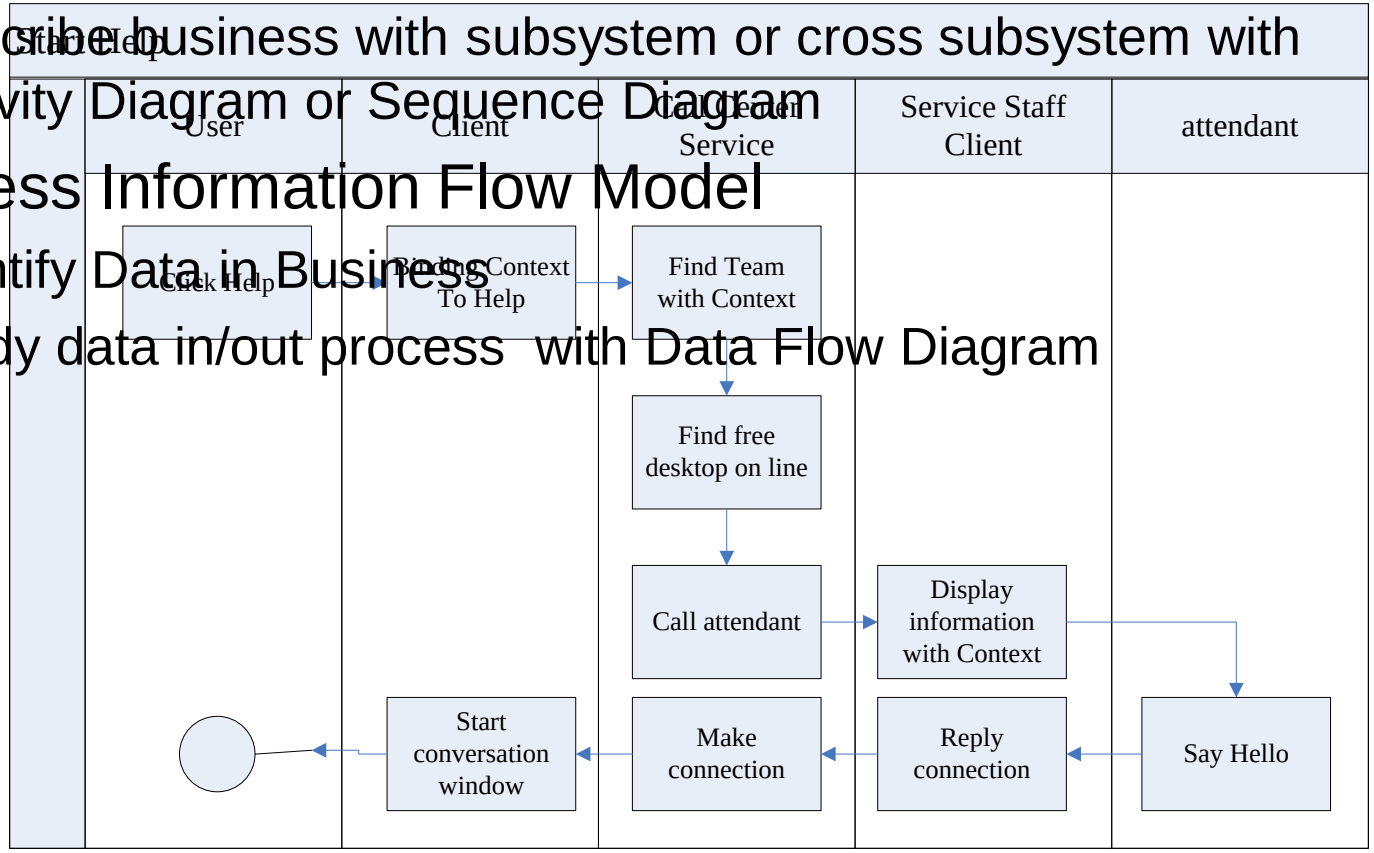
Step2:Developing Use-Case

- Task Model
  - Imitate the user use system
  - Draw use-case Diagram, and then
  - Write use-case business valuable



Step3:Bussness Modeling

- Business Process Model
  - Divide System into subsystem
  - Describe business with subsystem or cross subsystem with Activity Diagram or Sequence Diagram
- Business Information Flow Model
  - Identify Data in Business
  - Study data in/out process with Data Flow Diagram



Step4:The Content Model

- Content objects are extracted from use-cases
- The relationships among content objects and/or the hierarchy of content maintained by a WebApp
  - Relationships—entity-relationship diagram or UML
  - Hierarchy—data tree or UML

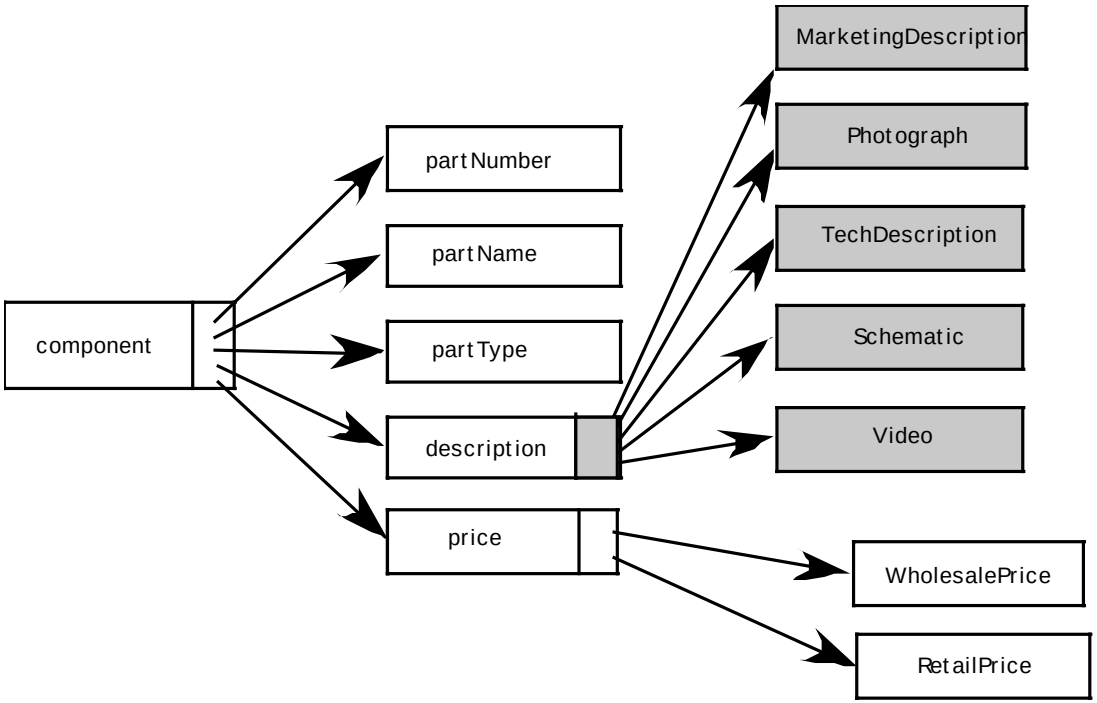


Figure 18.3 Data tree for a *SafeHome* component

Step5:The Domain Model

- grammatical parse is used to identify candidate classes
- Use Class Diagram

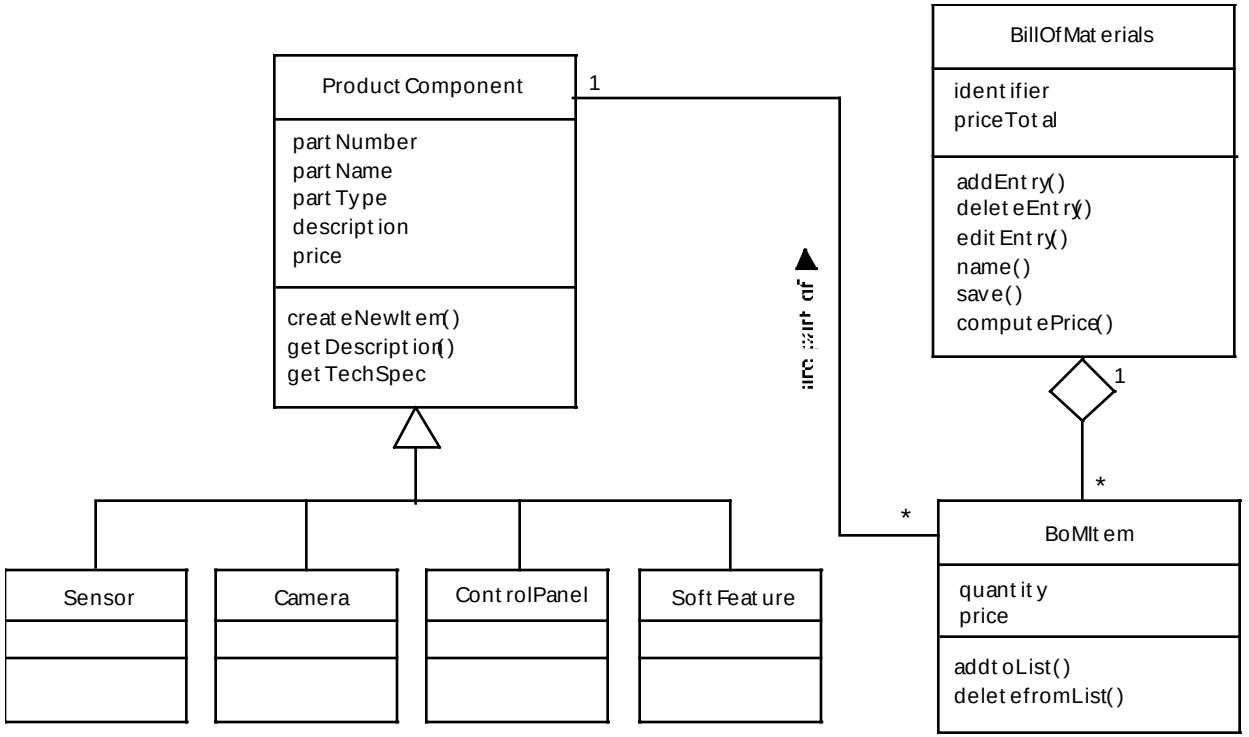


Figure 18.4 Analysis classes for use-case: *select SafeHome components*

Step6: Relationship-Navigation Analysis

- Relationship-navigation analysis (RNA) identifies relationships among the elements uncovered as part of the creation of the analysis model
- Steps:
  - ***Stakeholder analysis***—identifies the various user categories and establishes an appropriate stakeholder hierarchy
  - ***Element analysis***—identifies the content objects and functional elements that are of interest to end users
  - ***Relationship analysis***—describes the relationships that exist among the WebApp elements
  - ***Navigation analysis***—examines how users might access individual elements or groups of elements
  - ***Evaluation analysis***—considers pragmatic issues (e.g., cost/benefit) associated with implementing the relationships defined earlier

Step7:The Configuration Model

Server-side

- Server hardware and operating system environment must be specified
- Interoperability considerations on the server-side must be considered
- Appropriate interfaces, communication protocols and related collaborative information must be specified

- Client-side
  - Browser configuration issues must be identified
  - Testing requirements should be defined

- Deployment Diagram

Tasks for CCOS System

- Find more than three e-commerce site that has call center on-line
- Requirements elaboration and specification.

Outline

- Web Engineering & Web applications
- Web engineering process & Management
- Requirements for WebApps analysis
- **Design Modeling for WebApps**

When should emphasize WebApp design

- when content and function are complex
- when the size of the WebApp encompasses hundreds of content objects, functions, and analysis classes
- when the success of the WebApp will have a direct impact on the success of the business

Design & WebApp Quality

- Security
  - Rebuff external attacks
  - Exclude unauthorized access
  - Ensure the privacy of users/customers
- Availability
  - the measure of the percentage of time that a WebApp is available for use
- Scalability
  - Can the WebApp and the systems with which it is interfaced handle significant variation in user or transaction volume
- Time to Market

Quality Dimensions for End-Users

- *Time*
  - How much has a Web site changed since the last upgrade?
  - How do you highlight the parts that have changed?
- *Structural*
  - How well do all of the parts of the Web site hold together.
  - Are all links inside and outside the Web site working?
  - Do all of the images work?
  - Are there parts of the Web site that are not connected?
- *Content*
  - Does the content of critical pages match what is supposed to be there?
  - Do key phrases exist continually in highly-changeable pages?
  - Do critical pages maintain quality content from version to version?
  - What about dynamically generated HTML pages?

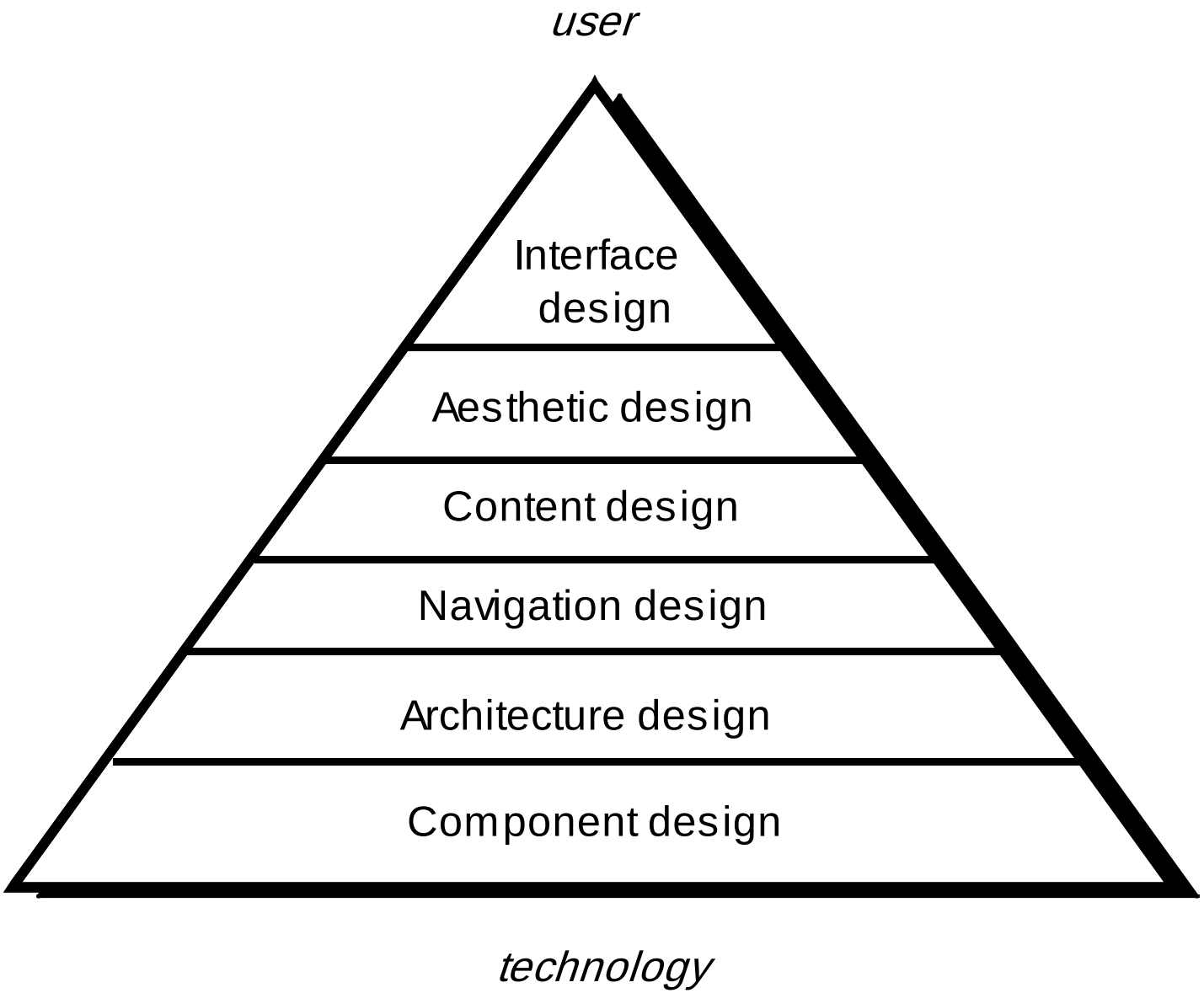
Quality Dimensions for End-Users

- *Accuracy and Consistency*
  - Are today's copies of the pages downloaded the same as yesterday's? Close enough?
  - Is the data presented accurate enough? How do you know?
- *Response Time and Latency*
  - Does the Web site server respond to a browser request within certain parameters?
  - In an E-commerce context, how is the end to end response time after a SUBMIT?
  - Are there parts of a site that are so slow the user declines to continue working on it?
- *Performance*
  - Is the Browser-Web-Web site-Web-Browser connection quick enough?
  - How does the performance vary by time of day, by load and usage?
  - Is performance adequate for E-commerce applications?

WebApp Design Goals

- Consistency
  - ***Content*** should be constructed consistently
  - ***Graphic design (aesthetics)*** should present a consistent look across all parts of the WebApp
  - ***Architectural design*** should establish templates that lead to a consistent hypermedia structure
  - ***Interface design*** should define consistent modes of interaction, navigation and content display
  - ***Navigation mechanisms*** should be used consistently across all WebApp elements
- Robustness
- Navigability
  - designed in a manner that is intuitive and predictable
- Visual appeal
- Compatibility

WebE Design Pyramid



WebApp 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 WebApp 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 Workflow

- Review information contained in the analysis model and refine as required.
- Develop a rough sketch of the WebApp 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.
- **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.

Using Design Patterns

Calendar Picker

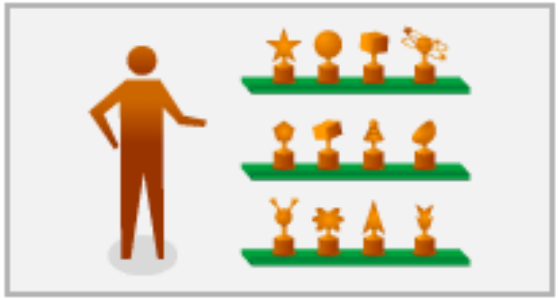
User wants to find or submit a particular piece of information based on a date or between a date range.

Carousel

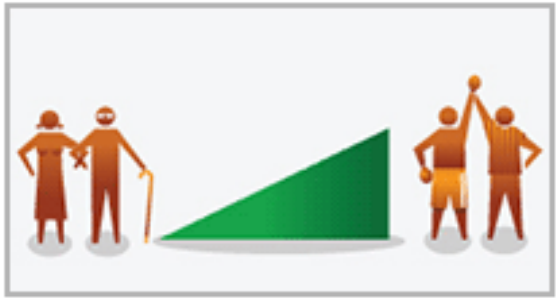
User needs to browse among a set of like objects represented pictorially.

Collapse Transition

An object is no longer of primary importance, but must remain available in a smaller form.



Collectible Achievements



The Competitive Spectrum

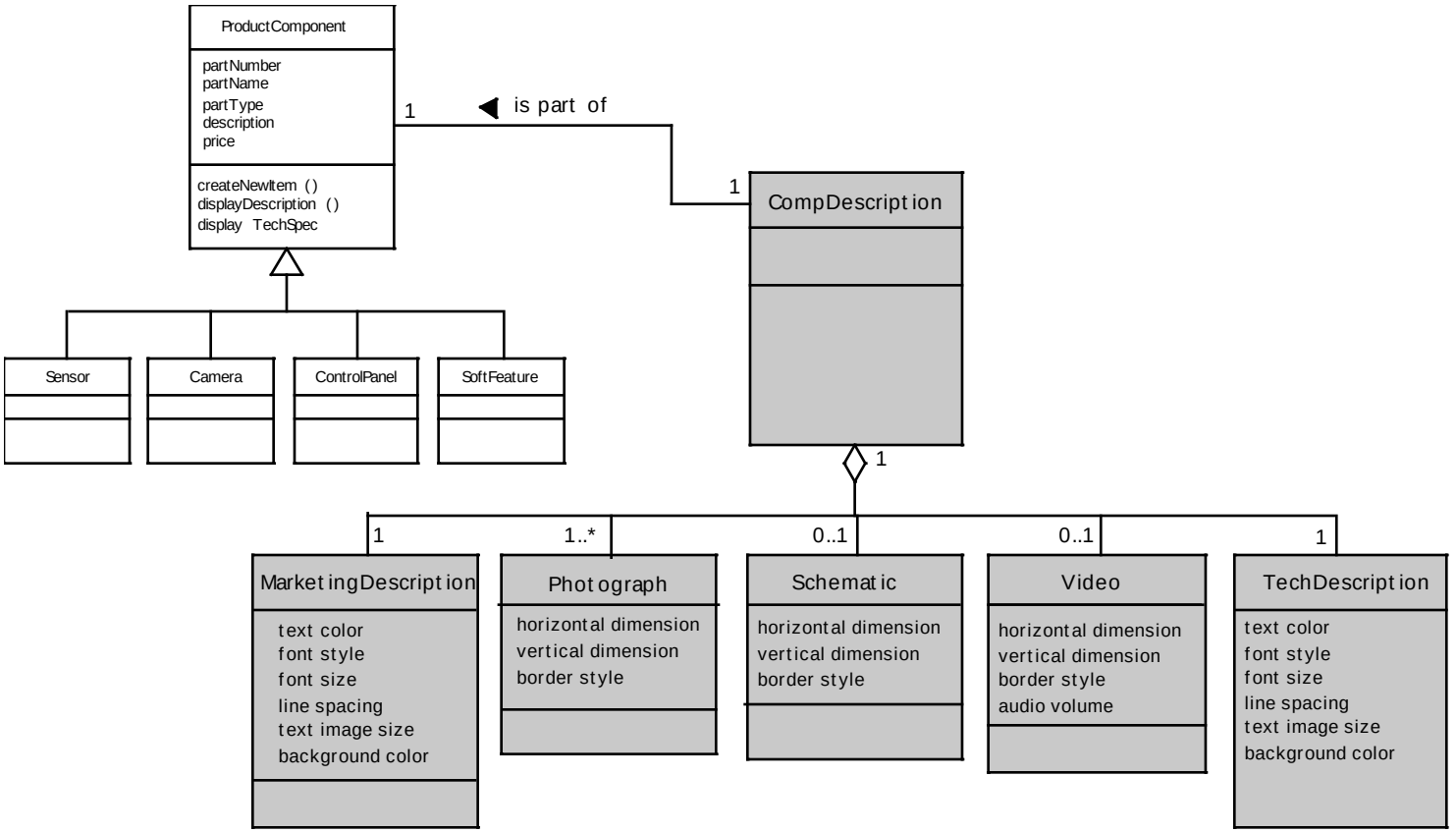
Cross Fade Transition

Yahoo - yPatterns

Content Design

- Develops a design representation for content objects
  - For WebApps, a content object is more closely aligned with a data object for conventional software
- Represents the mechanisms required to instantiate their relationships to one another.
  - analogous to the relationship between analysis classes and design components described in Chapter 11
- A content object has attributes that include content-specific information and implementation-specific attributes that are specified as part of design

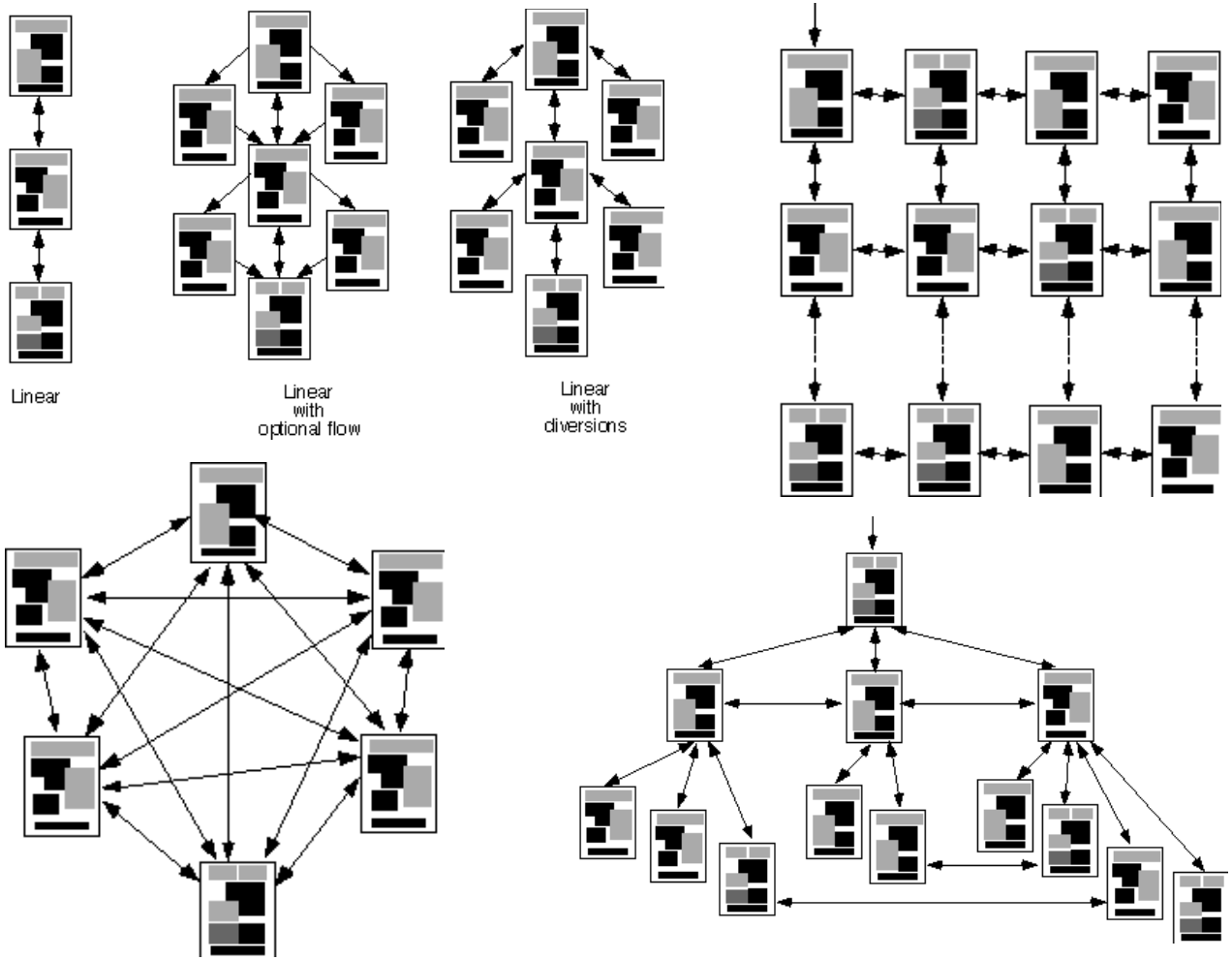
Design of Content Objects



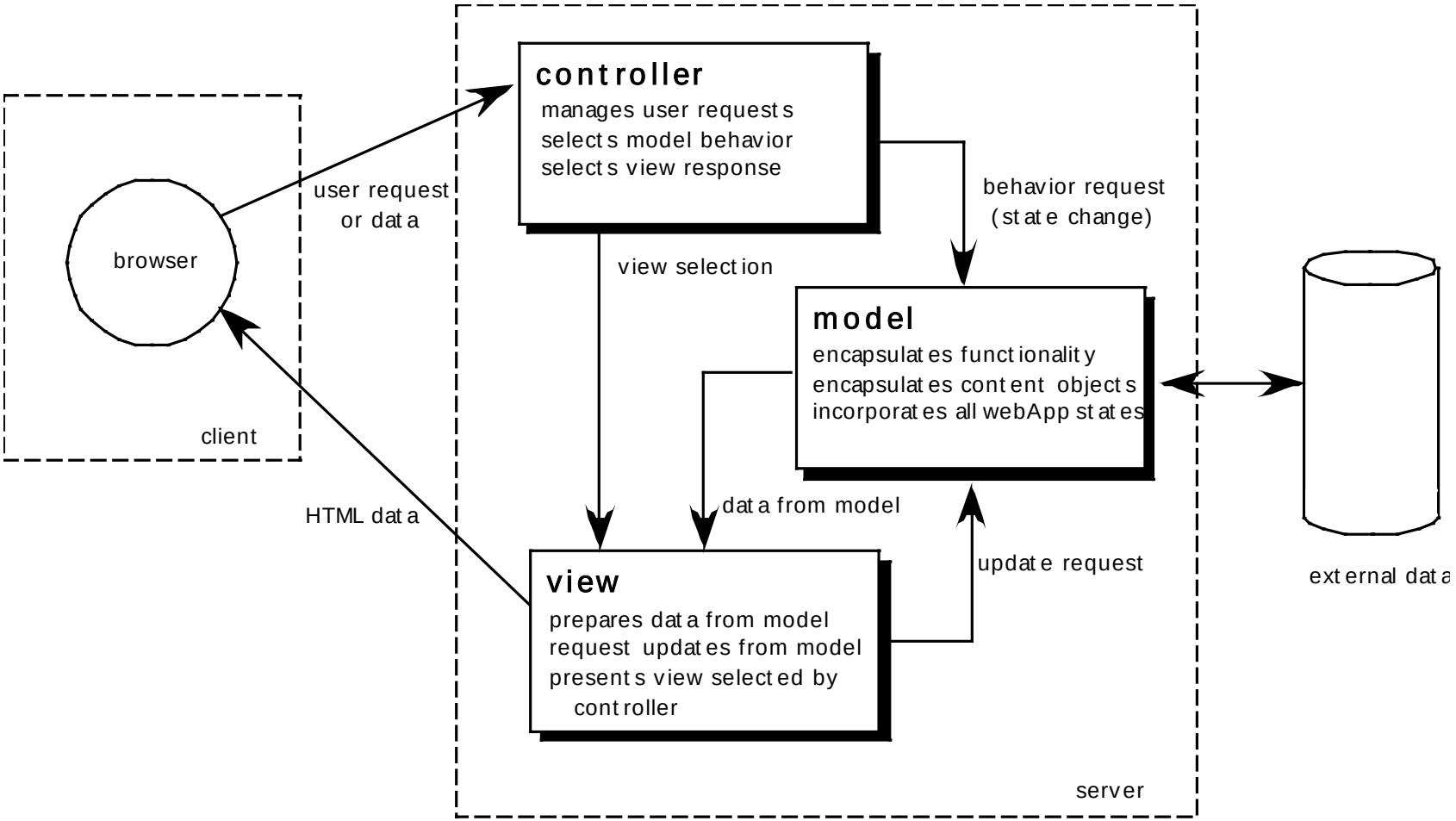
Architecture Design

- *Content architecture* focuses on the manner in which content objects (or composite objects such as Web pages) are structured for presentation and navigation.
  - The term information architecture is also used to connote structures that lead to better organization, labeling, navigation, and searching of content objects.
- *WebApp architecture* addresses the manner in which the application is structured to manage user interaction, handle internal processing tasks, effect navigation, and present content.
- Architecture design is conducted in parallel with interface design, aesthetic design and content design.

Content Architecture



MVC Architecture

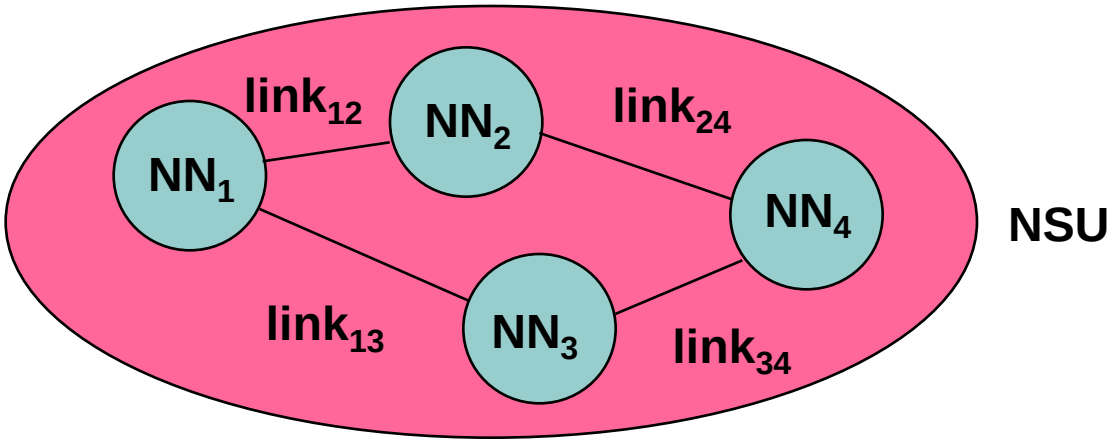


Navigation Design

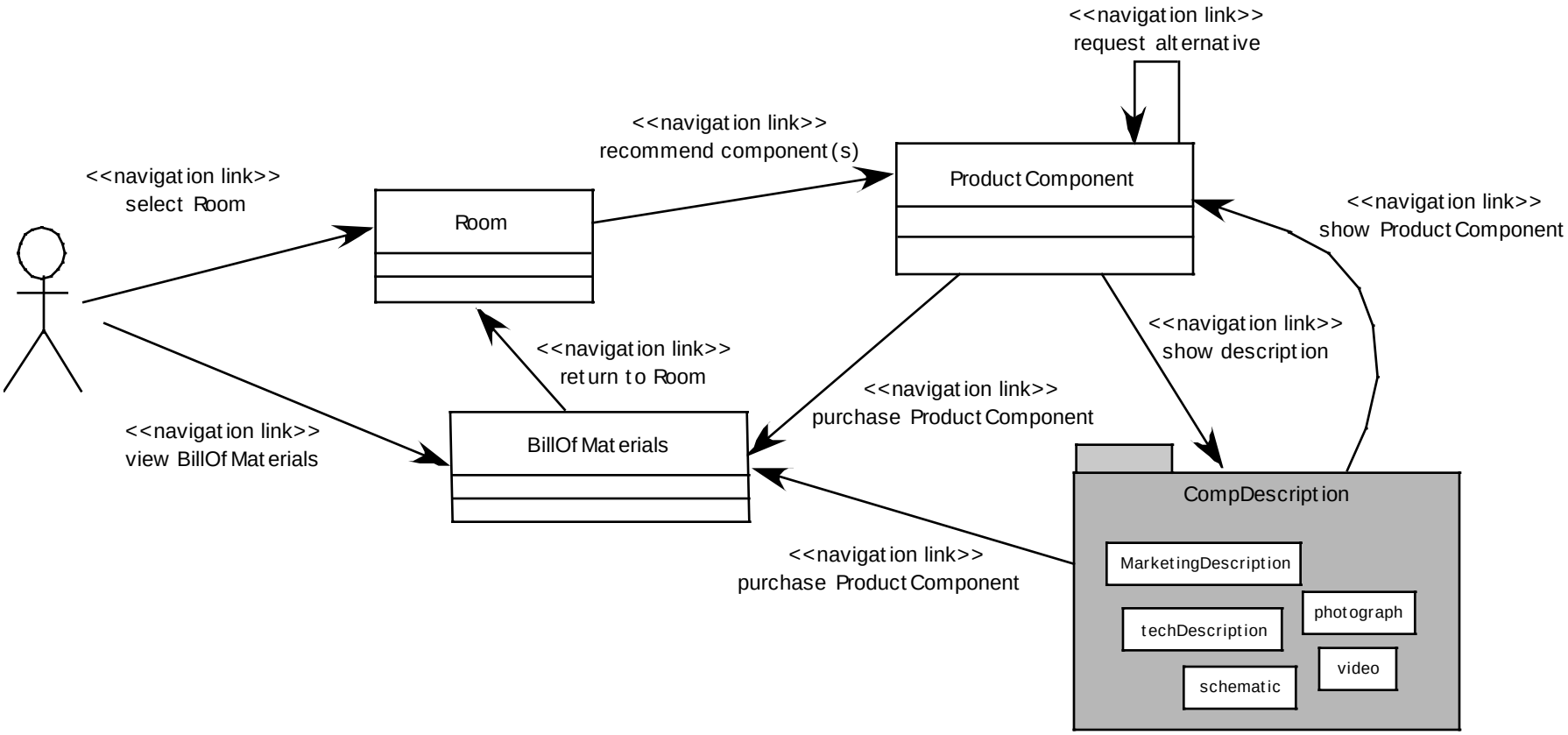
- Begins with a consideration of the user hierarchy and related use-cases
  - Each actor may use the WebApp somewhat differently and therefore have different navigation requirements
- As each user interacts with the WebApp, she encounters a series of *navigation semantic units* (NSUs)
  - NSU—"a set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

Navigation Semantic Units

- Navigation semantic unit
  - Ways of navigation (WoN)—represents the best navigation way or path for users with certain profiles to achieve their desired goal or sub-goal. Composed of …
    - Navigation nodes (NN) connected by Navigation links



Creating an NSU



Component-Level Design

- WebApp components implement the following functionality
  - perform localized processing to generate content and navigation capability in a dynamic fashion
  - provide computation or data processing capability that are appropriate for the WebApp's business domain
  - provide sophisticated database query and access
  - establish data interfaces with external corporate systems.

Patterns Repositories for web

- **Hypermedia Design Patterns Repository**
  - http://www.designpattern.lu.unisi.ch/
- **InteractionPatterns by TomErickson**
  - http://www.pliant.org/personal/Tom_Erickson/InteractionPatterns.html
- **Web Design Patterns by Martijn vanWelie**
  - http://www.welie.com/patterns/
- **Improving Web Information Systems with Navigational Patterns**
  - http://www8.org/w8-papers/5b-hypertext-media/improving/improving.html
- **An HTML 2.0 Pattern Language**
  - http://www.anamorph.com/docs/patterns/default.html
- **Common Ground - A Pattern Language for HCI Design**
  - http://www.mit.edu/~jtidwell/interaction_patterns.html
- **Patterns for Personal Web Sites**
  - http://www.rdrop.com/~half/Creations/Writings/Web.patterns/index.html
- **Indexing Pattern Language**
  - http://www.cs.brown.edu/~rms/InformationStructures/Indexing/Overview.html

Summary

- Web Engineering & Web applications
  - growth and Complexity of Web
  - Evolution of Engineering Discipline
- Web engineering process & Management
  - Scenario
  - Process
  - Modle: WSDM, OOHDM
  - Metrics

Summary

- Requirements for WebApps analysis (a case study)
  - requirement spec, finding actors, specifying use cases, business modeling, content model, domain model, RNA, configuration model
- Design Modeling for WebApps
  - WebApp Quality
  - WebE Design Pyramid: interface, aesthetic, content, navigation, architecture, component
  - Patterns Repositories

Further Readings

- http://en.wikipedia.org/wiki/Web_engineering
- Pressman, R.S., 'Applying Web Engineering', Part 3, Chapters 16-20, in Software Engineering: A Practitioner's Perspective, Sixth Edition, McGraw-Hill, New York, 2004. http://www.rspa.com/'
- "Handbook of Research on Web 2.0, 3.0, and X.0: Technologies, Business, and Social Applications", San Murugesan (Editor), Information Science Research, Hershey – New York, October 2009, ISBN 978-1-60566-384-5

Thank you!