MASSEY UNIVERSITY

SCHOOL OF FUNDAMENTAL SCIENCES

Web Application Requirements Analysis and Design

See also: David Parsons *et al* (2008 - 2010), *Dynamic Web Application Development Using (XML & Java, PHP & MySQL and ASP.Net),* Chapter 2

Topic Learning Objectives

1. To understand some of the techniques used in analysing web application system **requirements**
2. To be able to use some **notation from the UML** related to web applications
3. To understand some aspects of the processes involved in the **development lifecycle of web applications**
4. To be able to apply some common **design patterns** to the structure of web pages

Lecture Outline

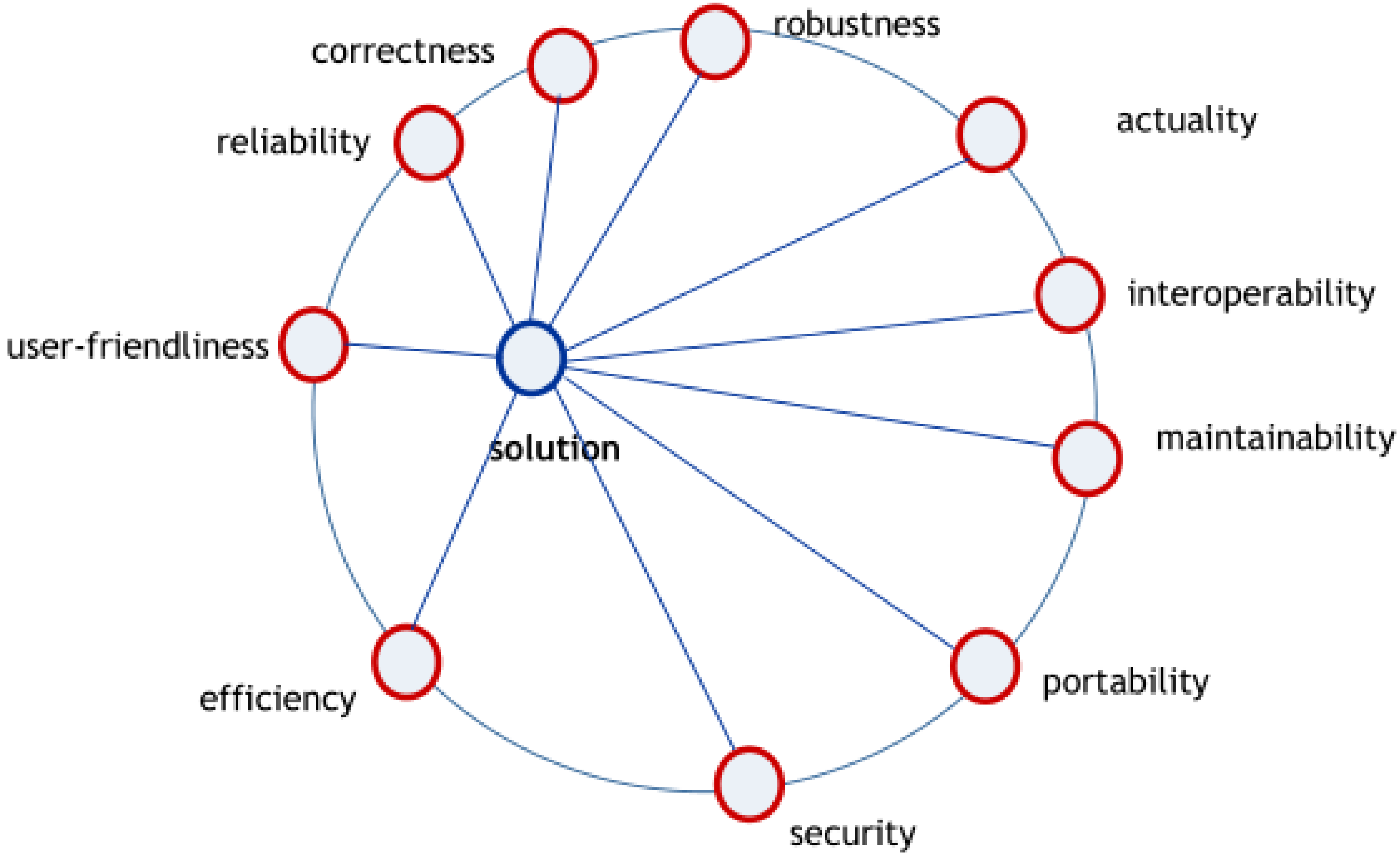
1. Whats different Web App Requirements;
2. SDLC’s;
3. UML & Unified Process;
4. Modelling Requirements
5. Analysis Tools:
   1. Domain Models,
   2. Use Cases,
   3. Storyboards;
6. Further Use cases;
7. From Analysis to Design;
8. Webflow Design;
9. Design Paterns for Web Page Structures

What’s Different..

About web based systems?

- ❑ Special kind of user interface;
- ❑ Large number of anonymous users;
- ❑ Data communications issues (speed, concurrency);
- ❑ Request-response model of iteration; and
- ❑ Client may be less rich than desktop.

Trade-off Circle for Web Applications



Trade-off Circle for Web Applications

External qualities are visible to the user and includes the following ones:

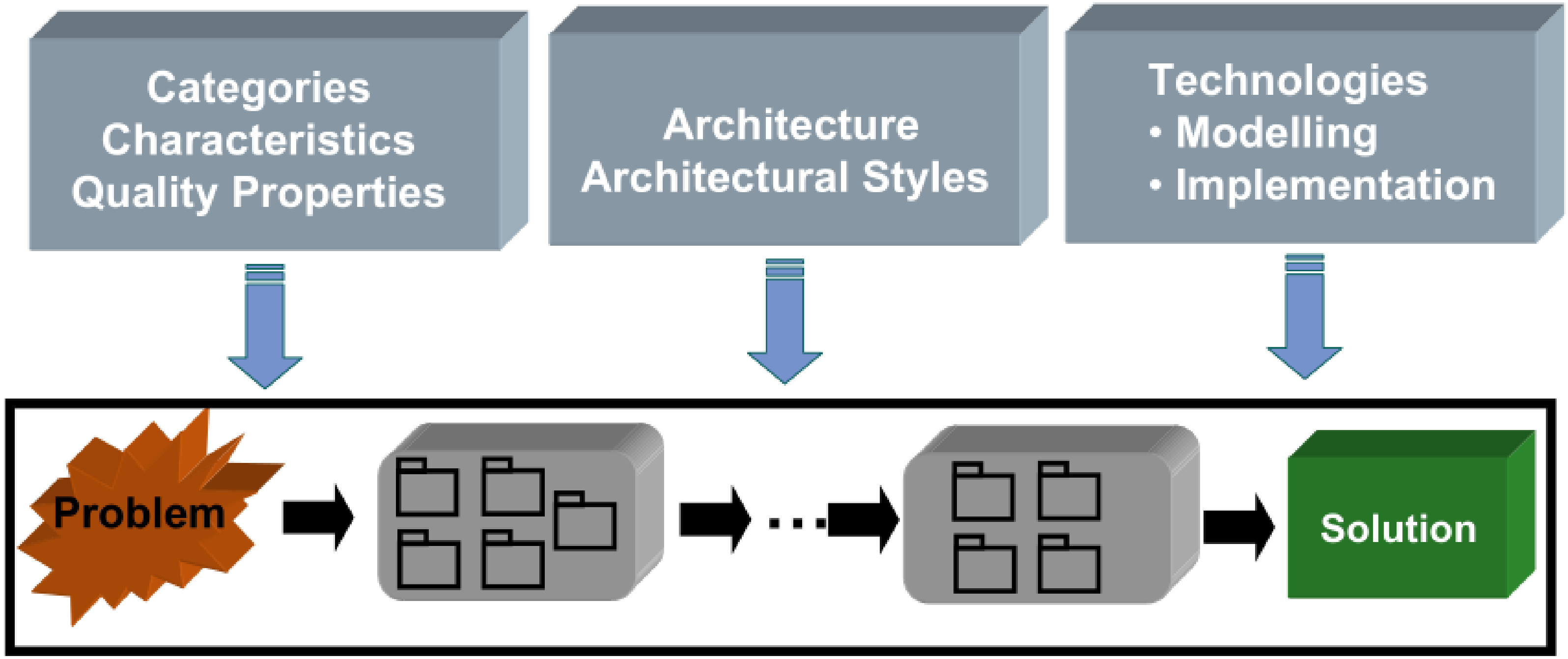
- ❑ **correctness:** a web application is functionally correct if it behaves according to the specification of the application
- ❑ **reliability:** the probability that the software will operate as expected, occurring software errors are not serious
- ❑ **robustness:** software behaves reasonably even in circumstances that were not anticipated in the requirements specification
- ❑ **actuality:** actuality of content must be guaranteed
- ❑ **user-friendliness:** easy to use by human (novice / experts)
- ❑ **efficiency:** economical handling of resources (time, storage space)
- ❑ **security:** system is protected from unauthorized access.

Trade-off Circle for Web Applications

Internal qualities are concerning and visible to the developer:

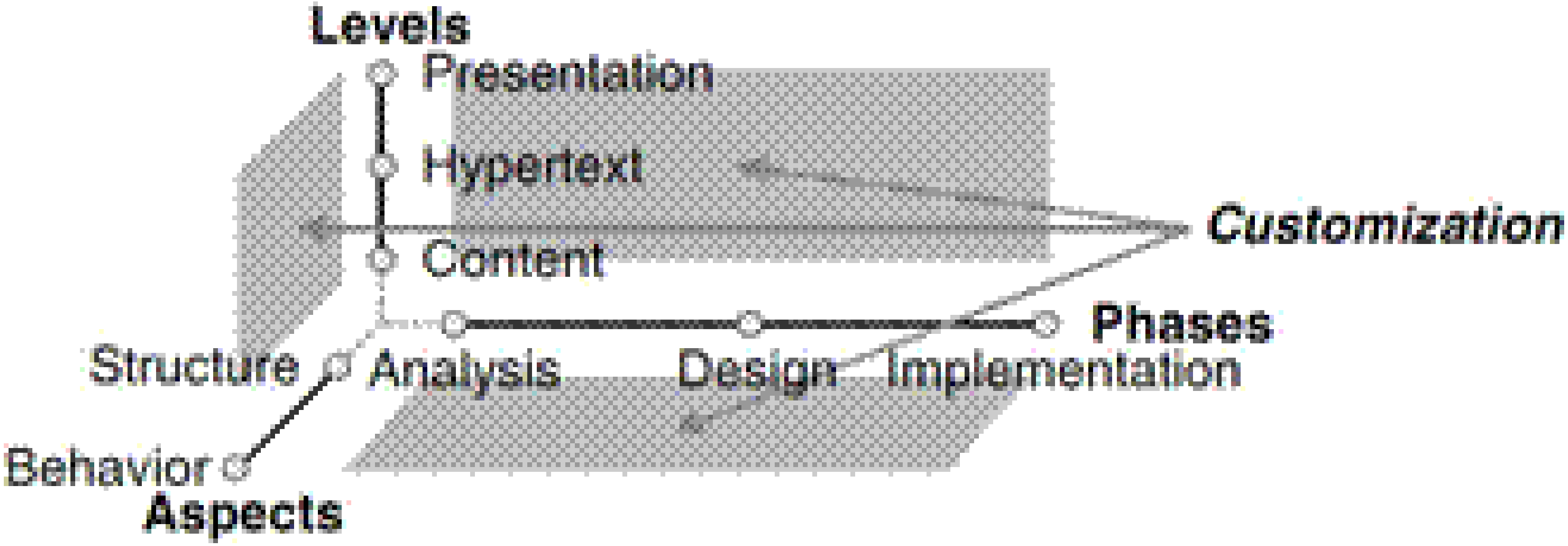
- ❑ **portability:** a web application is portable if it can run in different environments
- ❑ **interoperability:** refers to the ability of the web application to coexist and cooperate with other systems
- ❑ **maintainability:** ability to modify a web application after it has been deployed, to correct errors or extend the web application

Support for Web Application Development



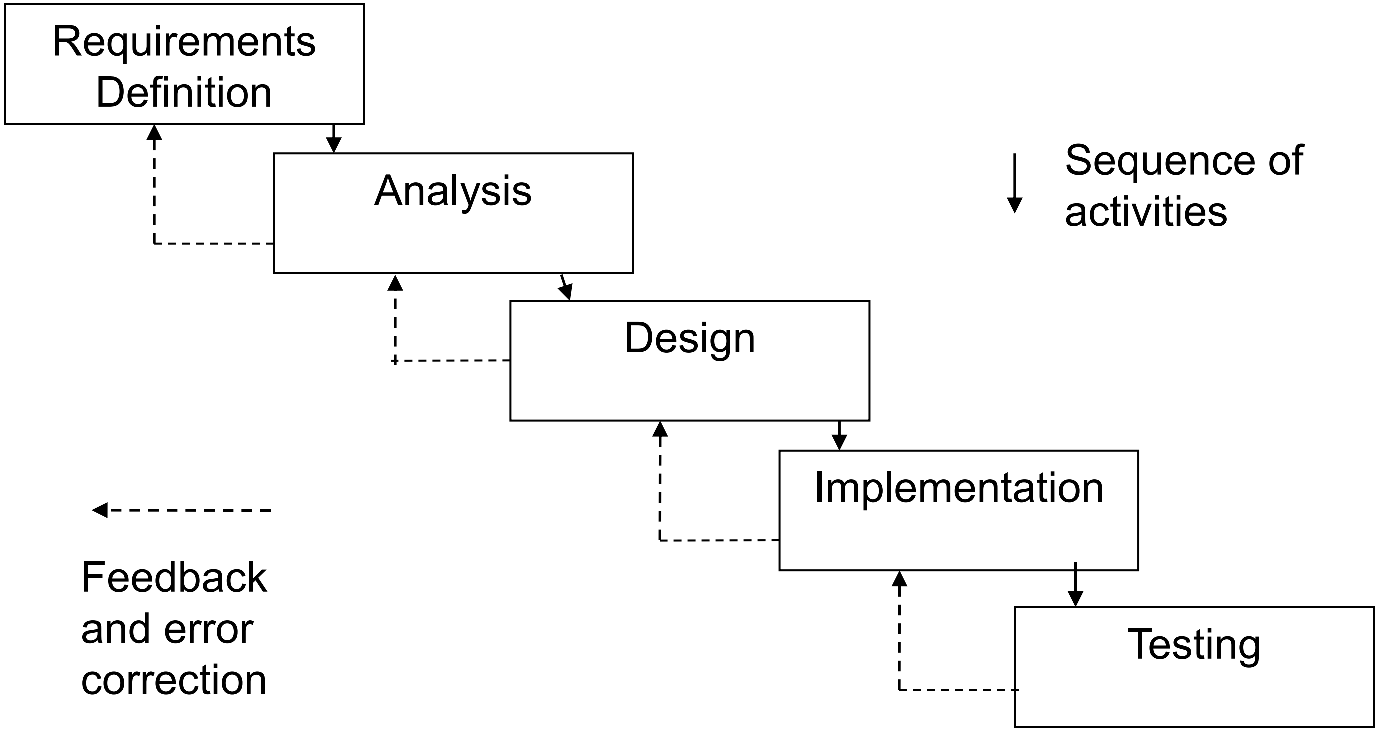
Model-based Development

Design dimensions of Web applications



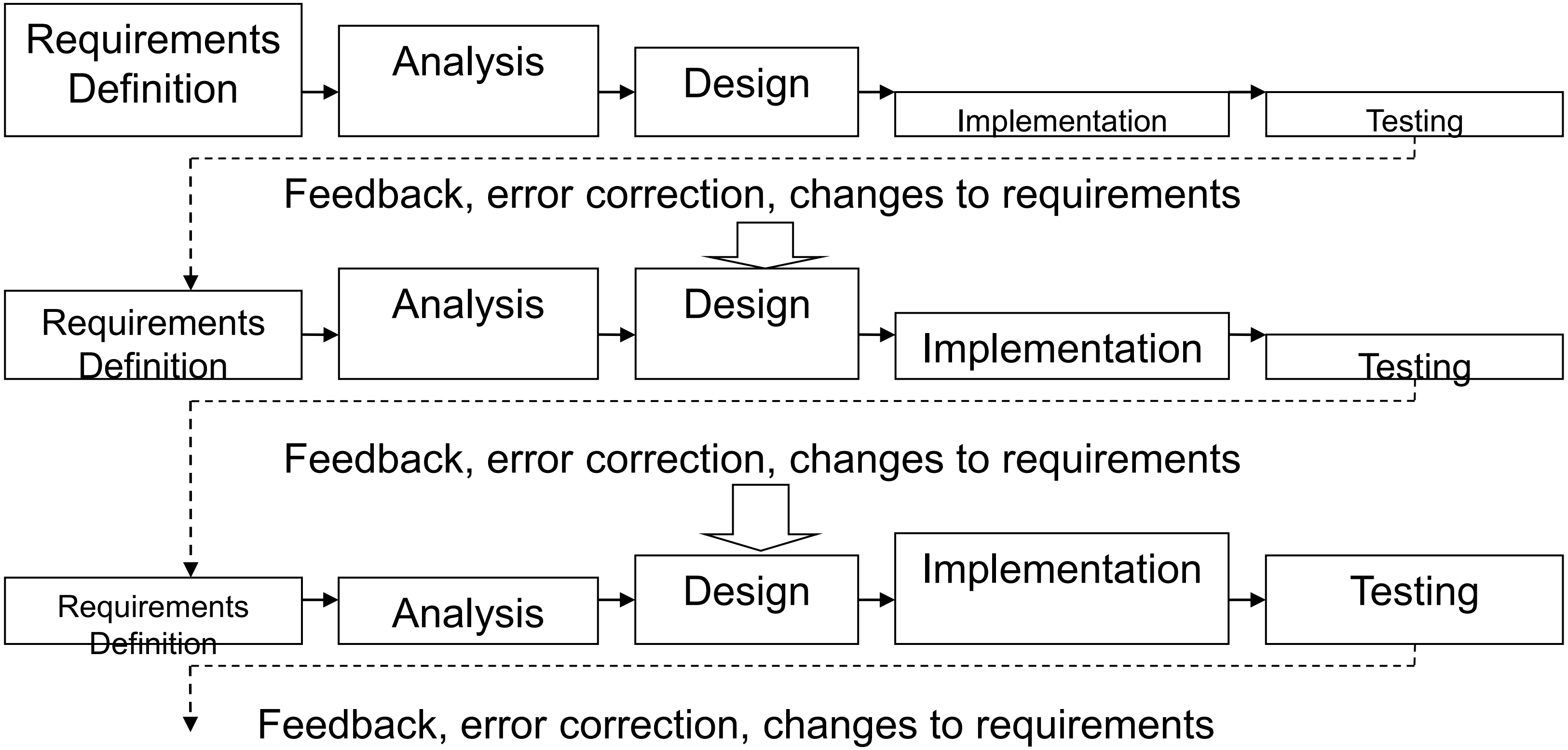
The Waterfall Model

An early approach to developing software



The Iterative Model

Better for dynamic systems

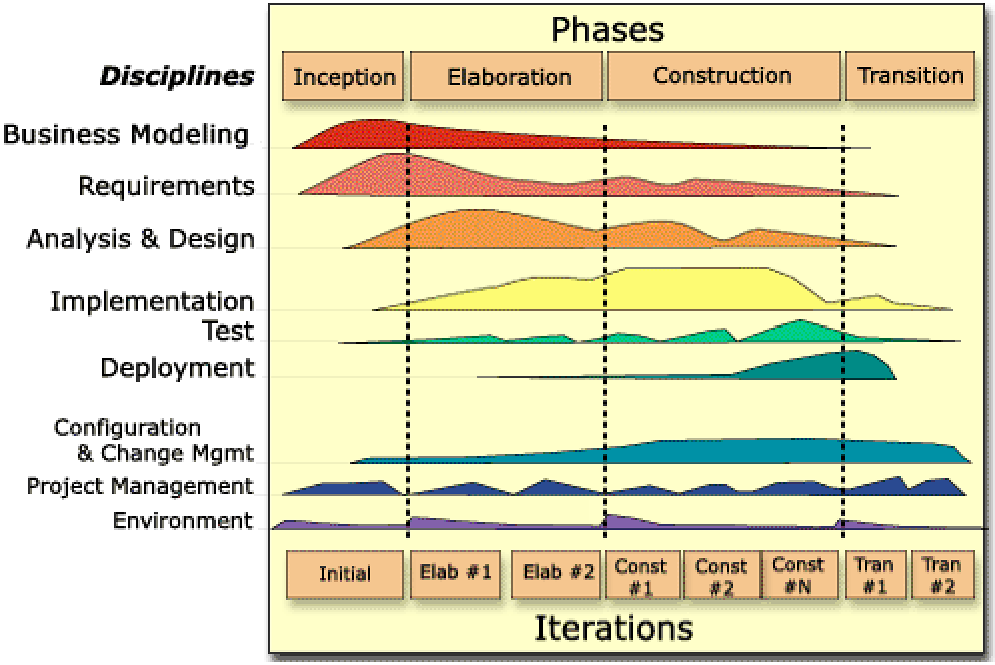


The Unified Modelling Language

- A standard set of diagrams for modelling object oriented software development;
- We do not have to use the entire language for every type of system; and
- Some ‘agile’ methodologies (e.g. Iconix) use a small subset of the UML.

The Unified Process

A standard process for developing object oriented software

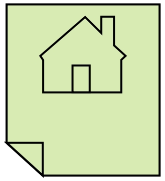


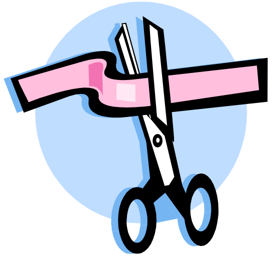
The UP

- 4 phases
- 9 disciplines
- Any number of iterations (at least 4!)
- Each iteration has a milestone (deliverable)

The UP as a building project

Plan and experiment → Construct the architecture → Add all the detail → Hand over to the new owner





Inception phase

Elaboration phase

Construction phase

Transition phase

Project Scenario

Web Home Cover is a new enterprise set up to provide home insurance over the web. The business case is based on providing a service that is entirely on-line and therefore highly efficient in terms of the initial capital investment required by the insurance company. Since the company will only operate via the web, it must have a web application that meets the needs of all its customers and staff. It must also be written to ensure that it will work for as many web clients as possible, from desktop computers to mobile devices

Mission Statement

To bring home insurance services to every corner of the web

Requirements gathering

- ❑ Focus group
- ❑ User profiles / personas
- ❑ Joint requirements workshop (of stakeholders)
- ❑ Brain storming / card storming
- ❑ 12 or so core requirements

Prioritizing Requirements

MoSCoW

- ❑ **M**ust have
- ❑ **S**hould have
- ❑ **C**ould have
- ❑ **W**ant to have

Can decide by voting, using multiple votes from different perspectives

Concept List

Identified from the core requirements

Remove concepts from the list that are:

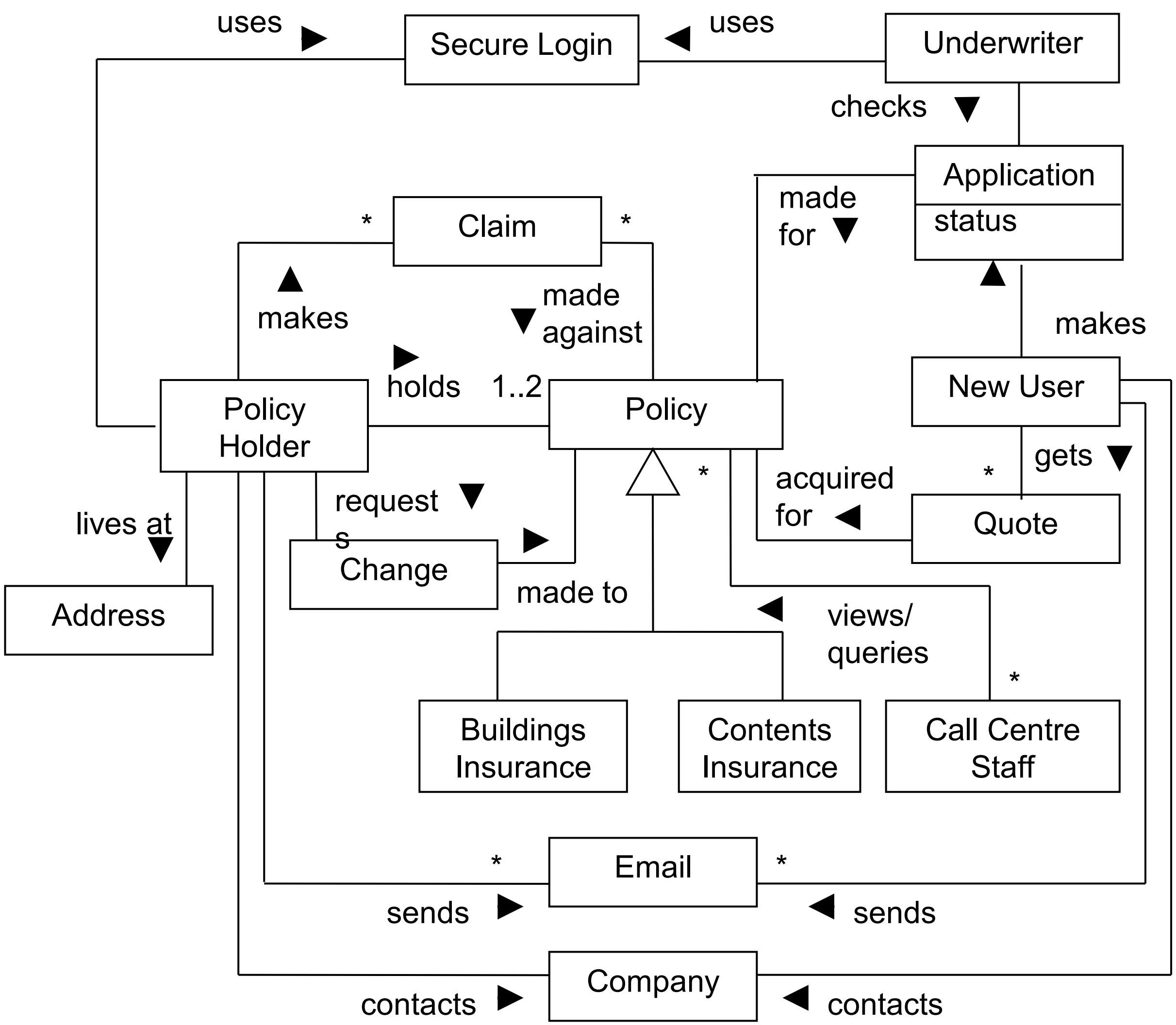
- ❑ Outside the system boundary
- ❑ The boundary itself
- ❑ Nouns for the system as a whole
- ❑ Synonyms
- ❑ Properties of other concepts

Domain Model

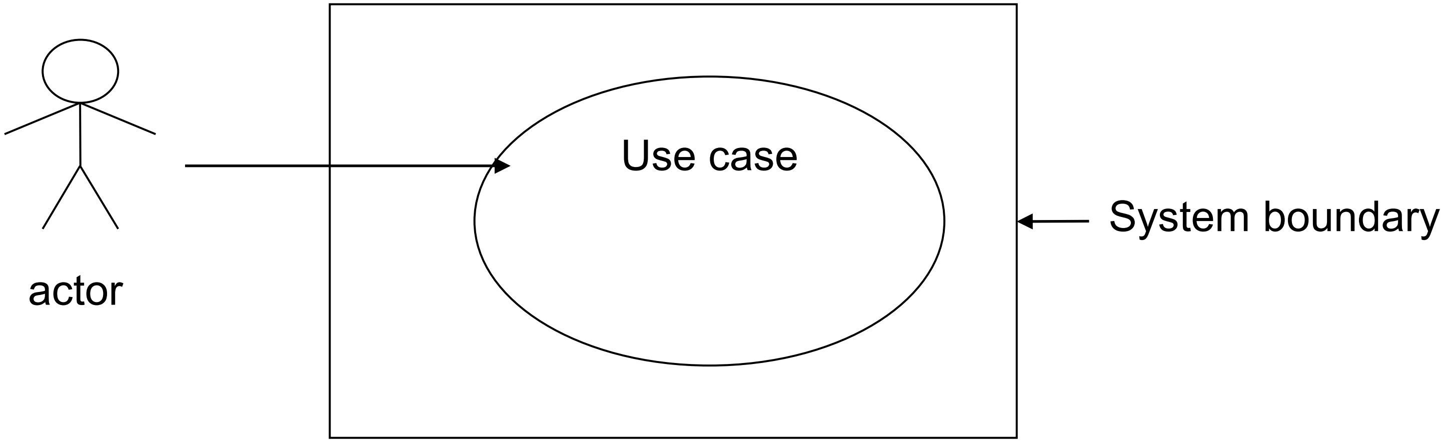
Describes the following:

- ❑ Key concepts from the domain
- ❑ Which components interact with each other
- ❑ How these relations can be described
- ❑ The cardinality of interaction
    - ❖ one to one,
    - ❖ one to many, and
    - ❖ many to many.

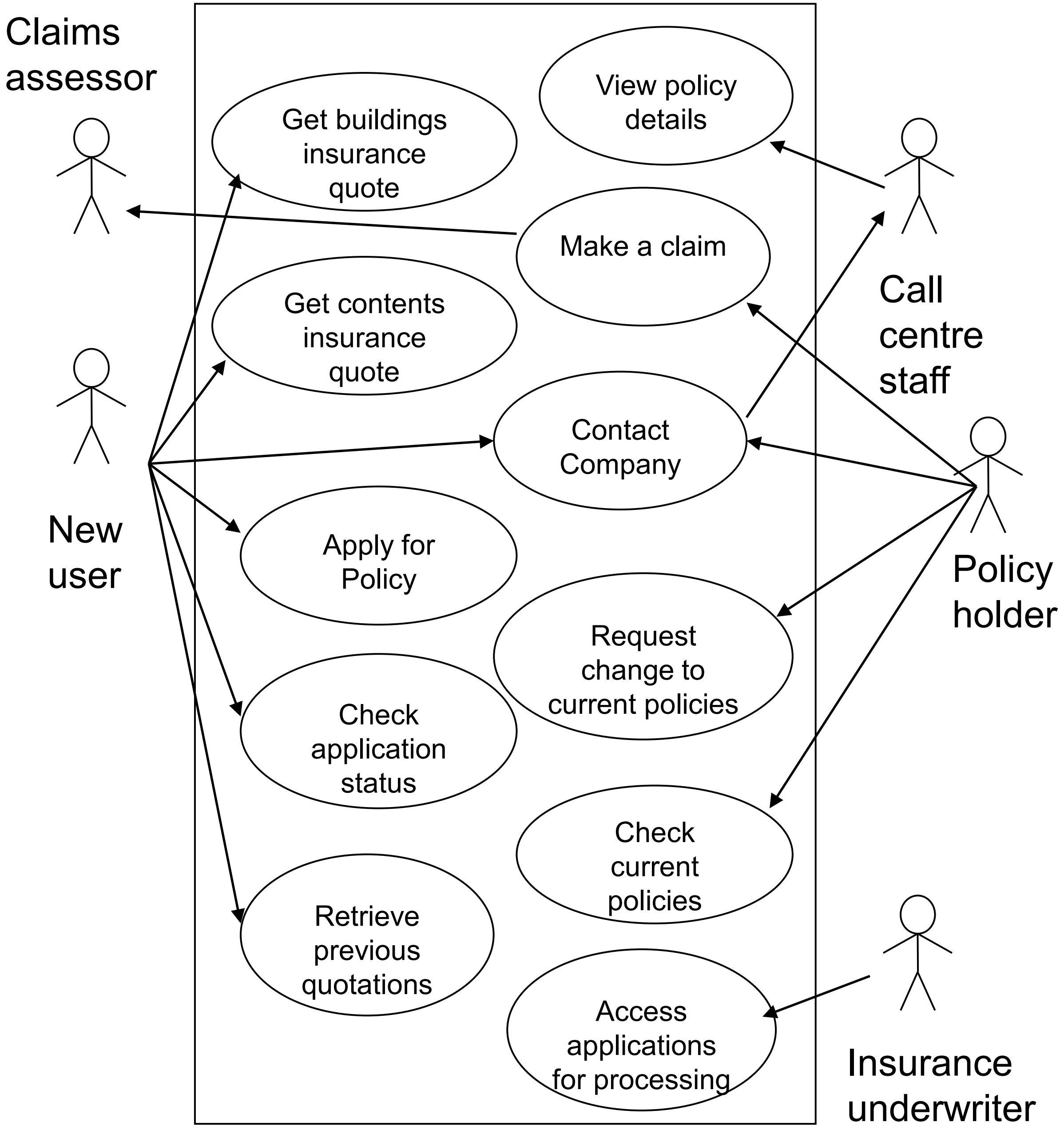
Domain Model



Use Case Diagrams



Use Cases


Claims assessor
Get buildings insurance quote
View policy details
Make a claim
Get contents insurance quote
Call centre staff
Contact Company
New user
Apply for Policy
Policy holder
Request change to current policies
Check application status
Check current policies
Retrieve previous quotations
Access applications for processing
Insurance underwriter

Use case description

Use Case Name : Get Buildings Insurance Quote

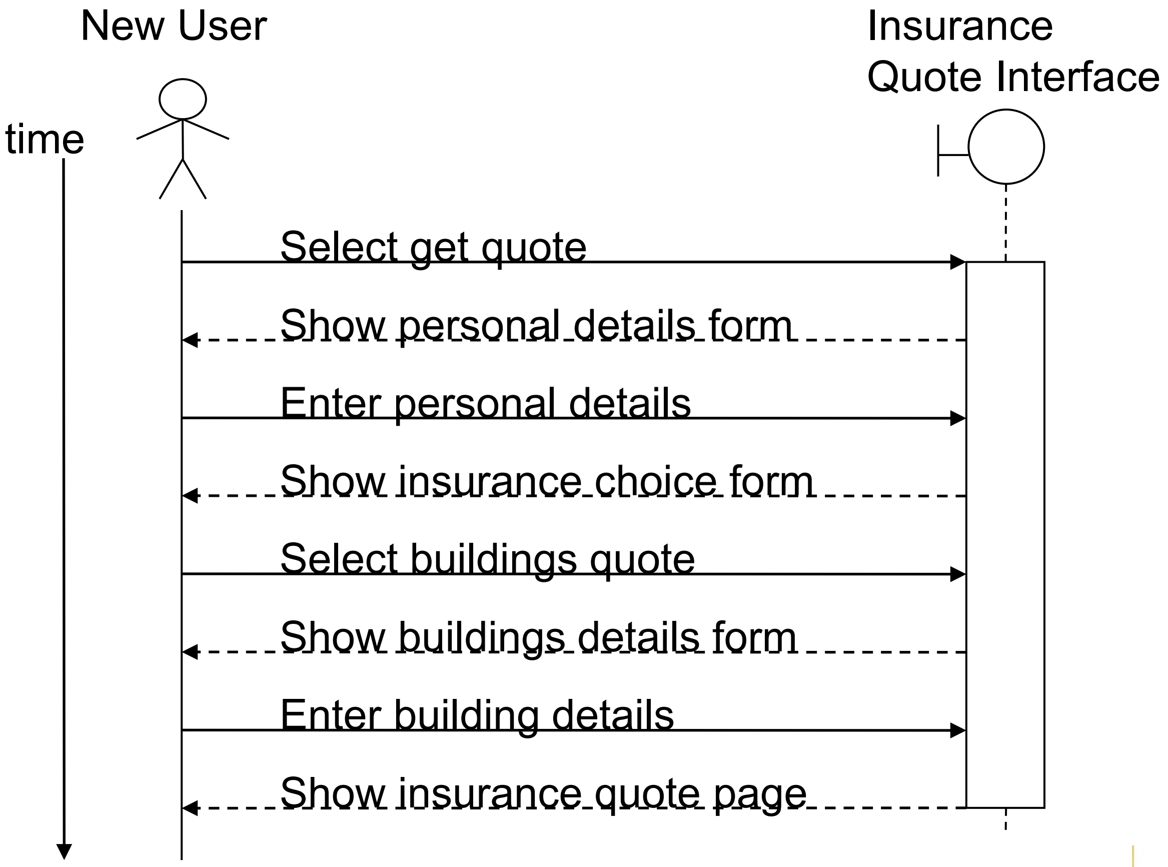
Actors: New user

Start page: Home page

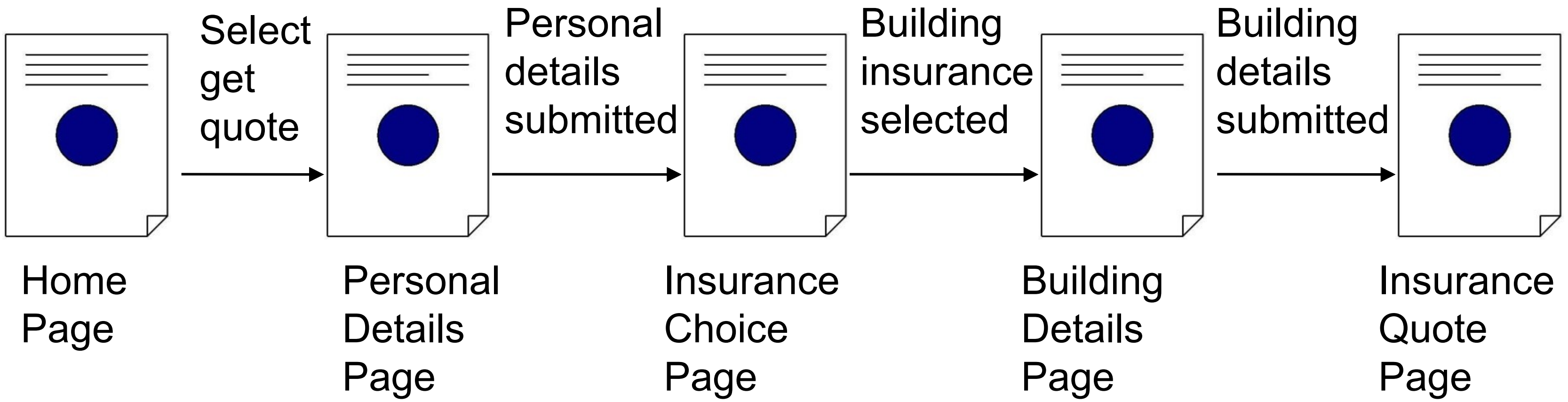
Use Case Description:

1. The actor chooses to get an insurance quote
2. The system requests the actor's personal details
3. The actor enters his/her personal details
4. The system displays a choice of available insurance quotes
5. The actor chooses to get a buildings insurance quote
6. The system requests information about the building to be insured
7. The actor enters data about the building
8. The system displays the buildings insurance quote.

System sequence diagram

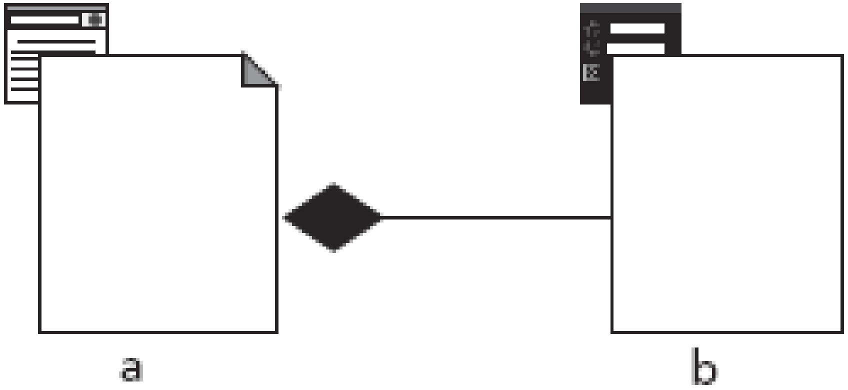

New User
Insurance Quote Interface
time
Select get quote
Show personal details form
Enter personal details
Show insurance choice form
Select buildings quote
Show buildings details form
Enter building details
Show insurance quote page

Basic storyboard



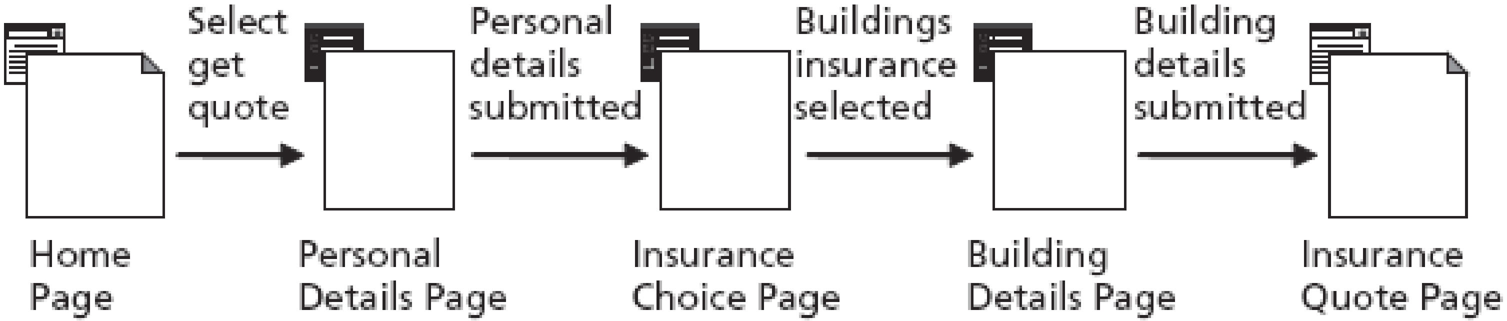
UML Extensions

Symbols for (a) client page and (b) form



Modified Storyboard

This version of the use case uses specific page types



Alternate Flow

Added to our existing use case

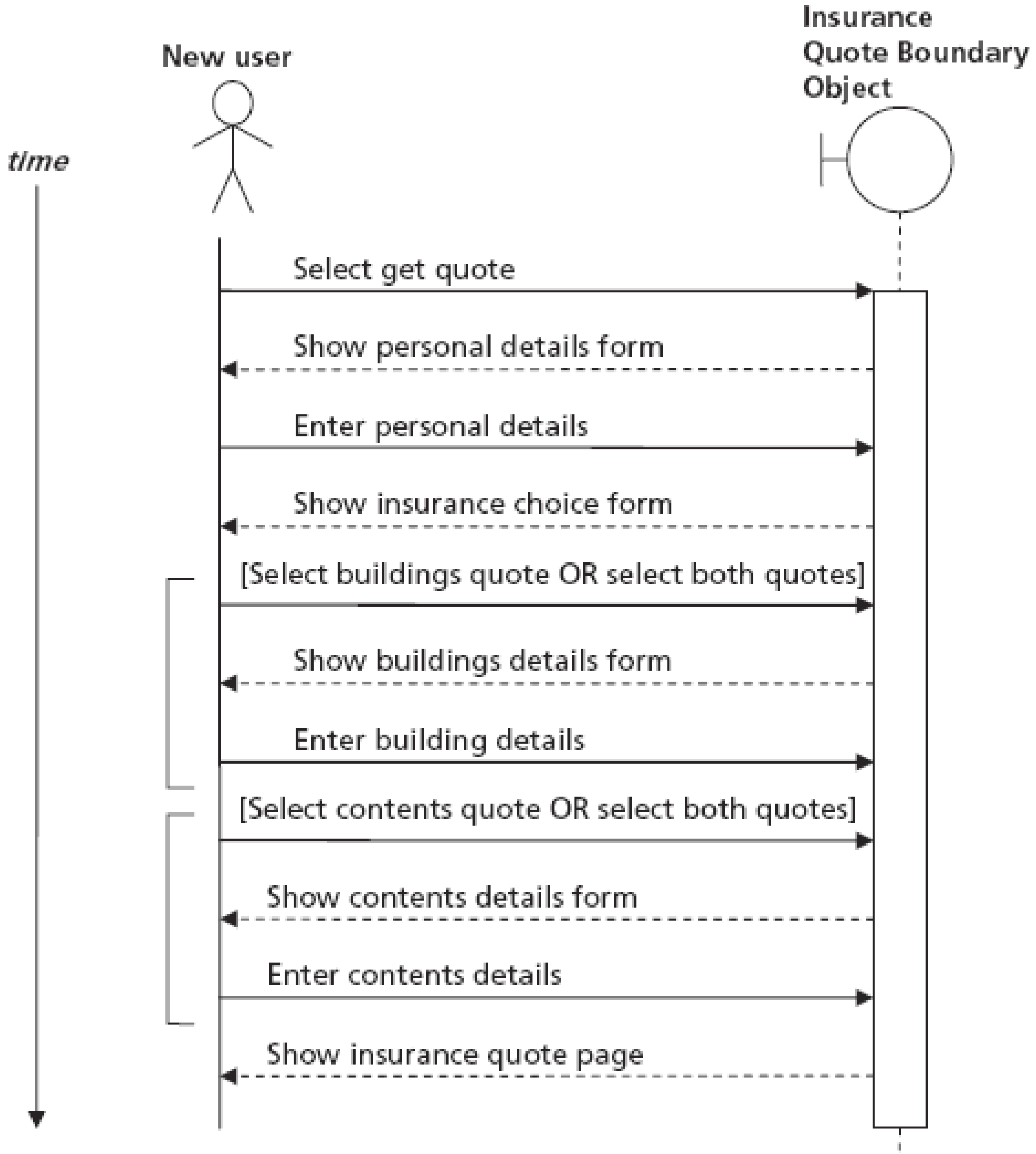
Alternate flow – contents insurance only

5a. The actor chooses to get a contents insurance quote.
6a. The system requests information about the contents to be insured.
7a. The actor enters data about the contents.
8a. The system displays the contents insurance quote.

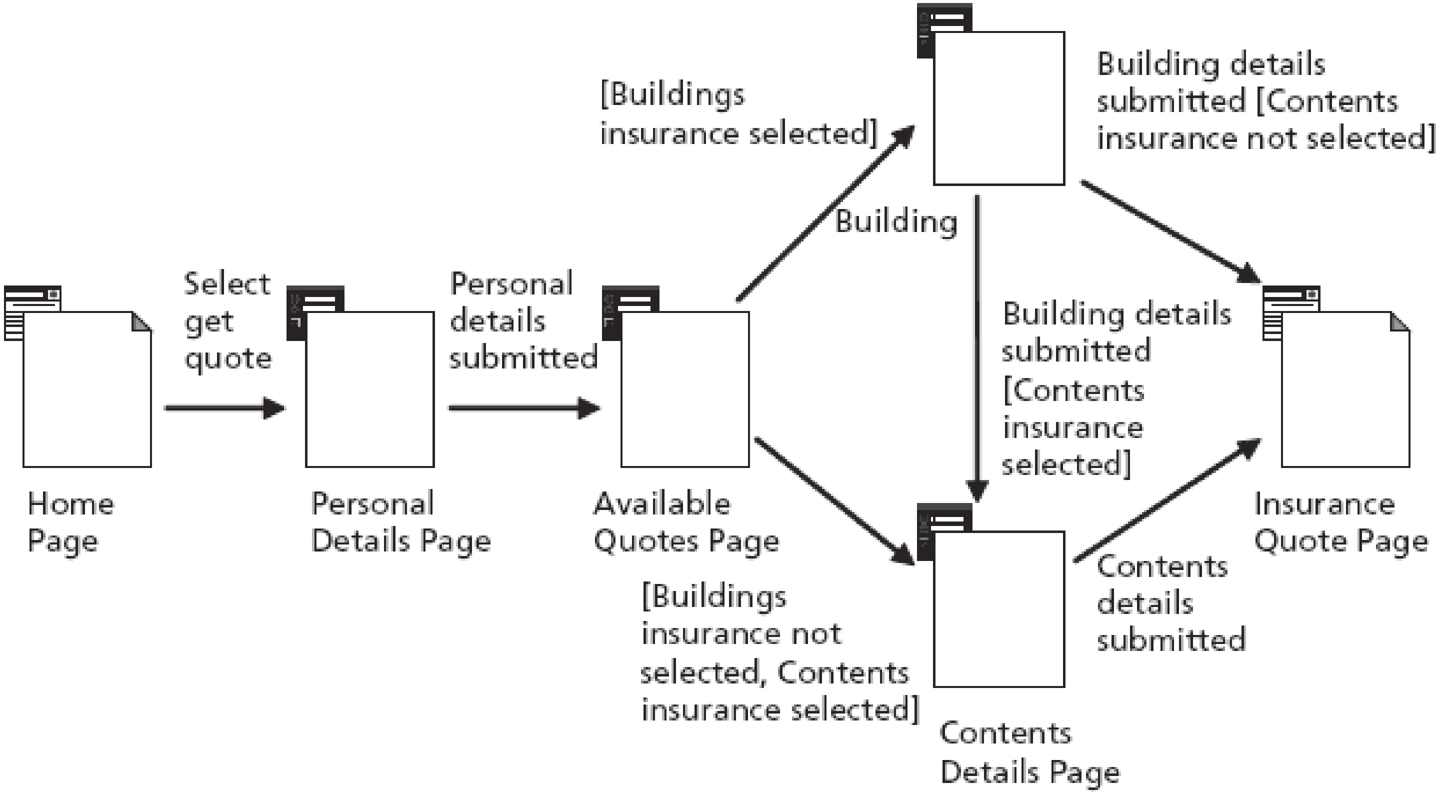
Alternate flow – both types of insurance

5a. The actor chooses to get both a buildings insurance quote and a contents insurance quote.
6, 7, 6a, 7a
8b. The system displays a contents insurance quote, a buildings insurance quote and a total.

Modified sequence diagram


New user
Insurance Quote Boundary Object
time
Select get quote
Show personal details form
Enter personal details
Show insurance choice form
[Select buildings quote OR select both quotes]
Show buildings details form
Enter building details
[Select contents quote OR select both quotes]
Show contents details form
Enter contents details
Show insurance quote page

Updated Storyboard


[Buildings insurance selected]
Building details submitted [Contents insurance not selected]
Building
Select get quote
Personal details submitted
Building details submitted [Contents insurance selected]
Home Page
Personal Details Page
Available Quotes Page
Insurance Quote Page
Contents details submitted
[Buildings insurance not selected, Contents insurance selected]
Contents Details Page

Analysis v Design

Analysis is about defining the problem domain and specifying how we anticipate the system will be used from the user perspective

- It is *technology agnostic*

Design is about how we plan the solution

- It is *technology aware*

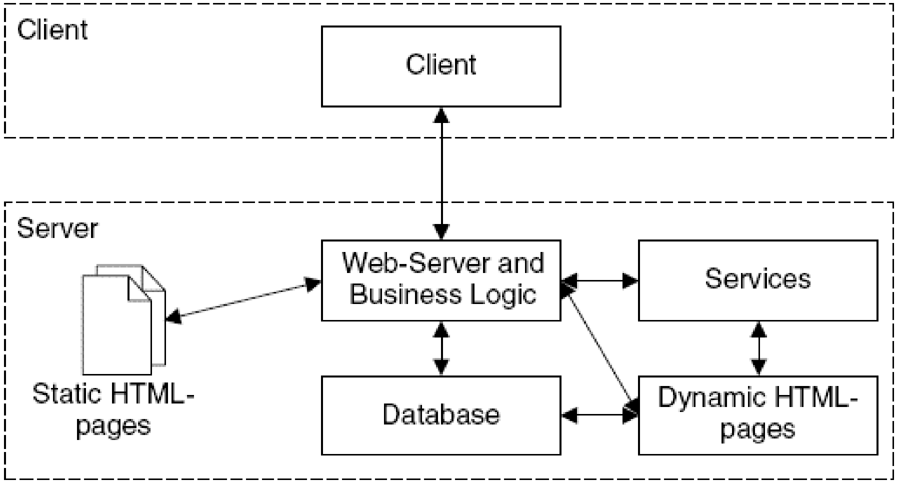
Design Detail (Break)

- In an iterative process, the transition from analysis to design is a gentle one
  - Not like going over the waterfall
- Design starts off at a high level and becomes more detailed
- To successfully design you need to understand the technology of the solution

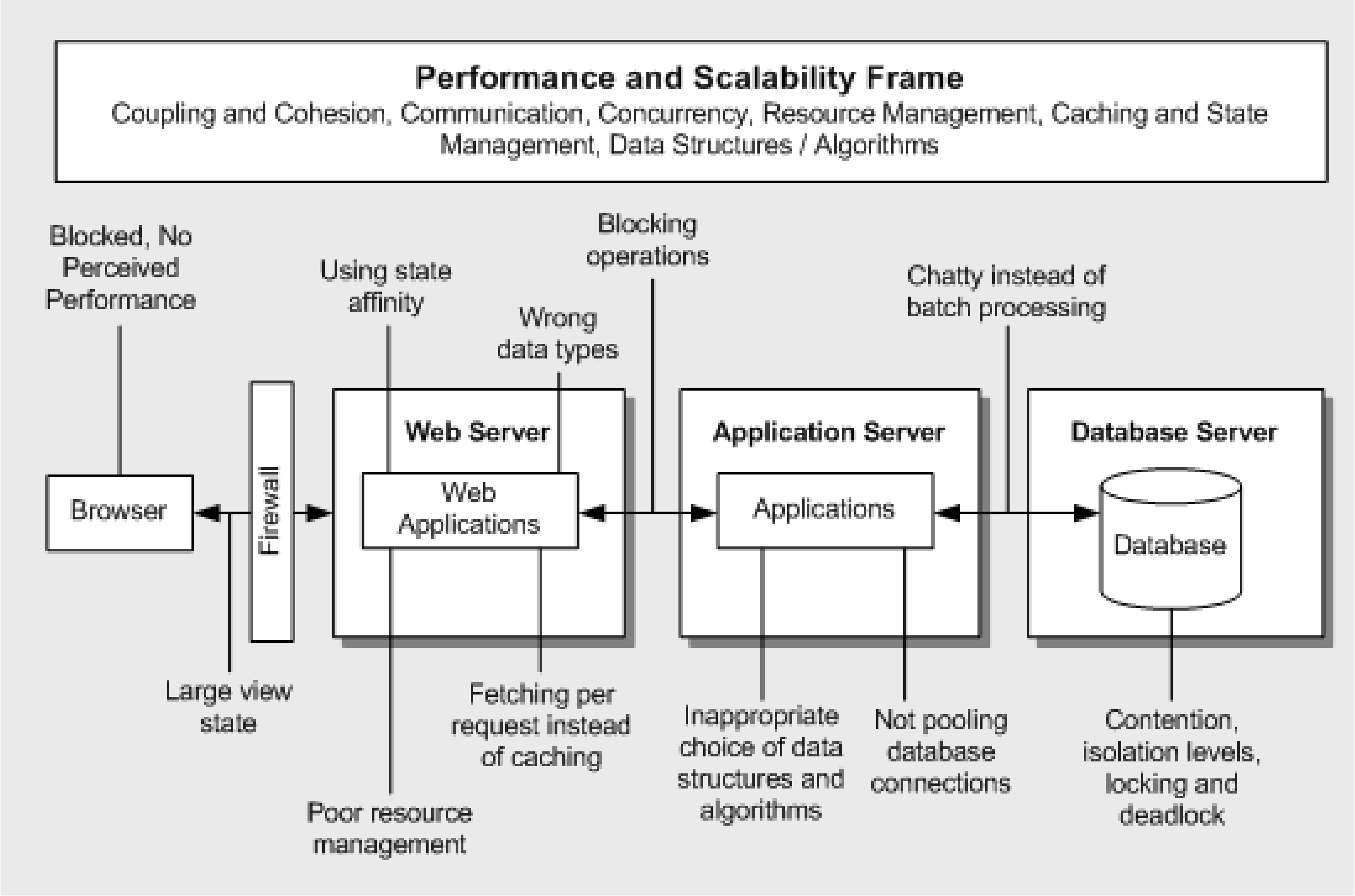
Design Aspects

- ❑ We will look at some architectural aspects of design from the server perspective
- ❑ We will also look at some aspects of design from the client perspective
- ❑ These are design patterns
  - ❖ Commonly used solutions to the problems of designing web applications

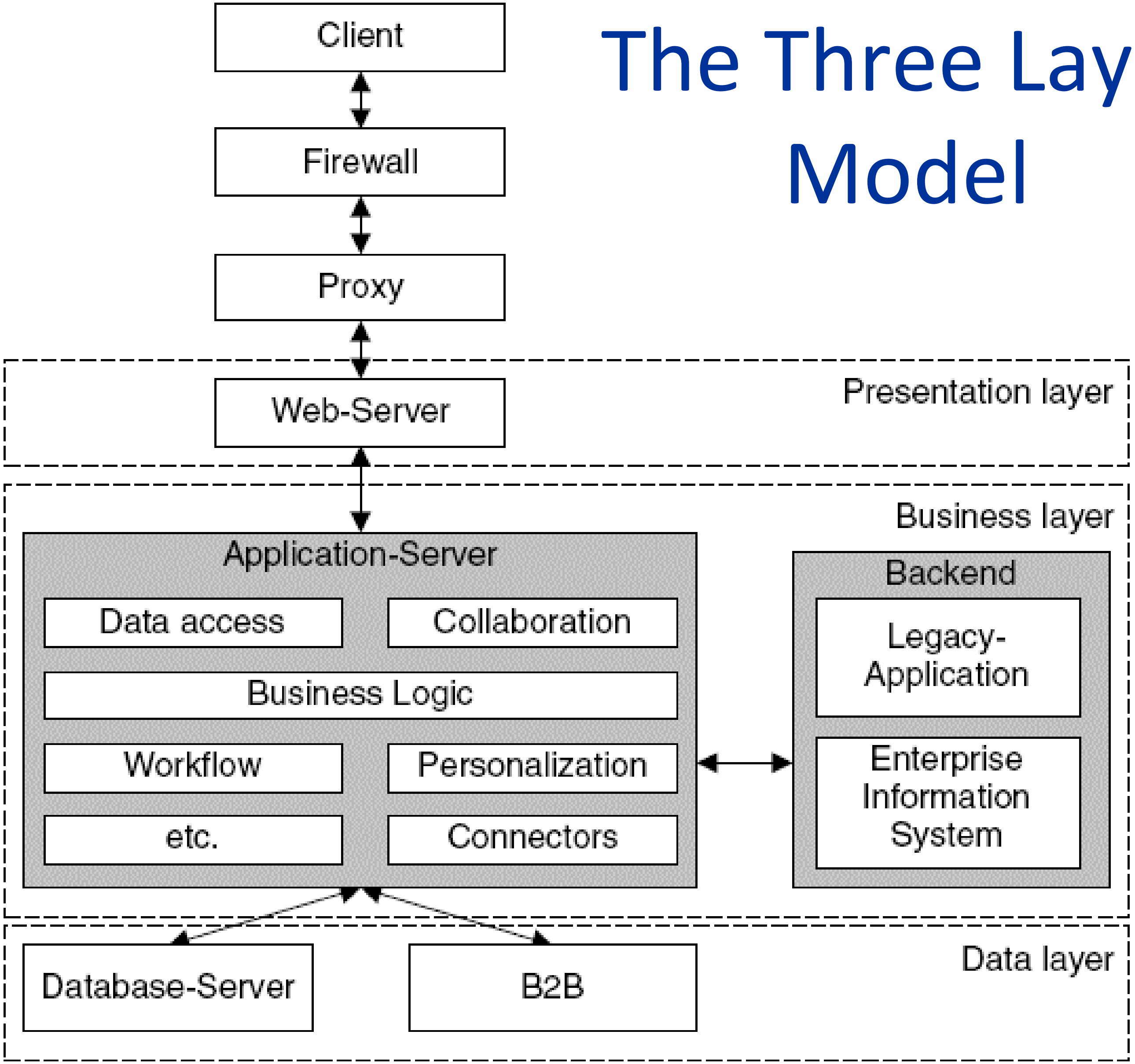
Client-Server



- ❑ An important feature of designing web based applications is allocating roles to the client and the server
- ❑ How ‘thin’ is the client?
- ❑ Should we use the browser’s ability to, for example, run JavaScript, Java Applets, Flash etc?
- ❑ What if the user has a mobile phone?


Performance and Scalability Frame
Coupling and Cohesion, Communication, Concurrency, Resource Management, Caching and State Management, Data Structures / Algorithms
Blocked, No Perceived Performance
Using state affinity
Blocking operations
Wrong data types
Chatty instead of batch processing
Browser
Firewall
Web Server
Web Applications
Application Server
Applications
Database Server
Database
Large view state
Poor resource management
Fetching per request instead of caching
Inappropriate choice of data structures and algorithms
Not pooling database connections
Contention, isolation levels, locking and deadlock

The Three Layer Model



Static and Dynamic Content

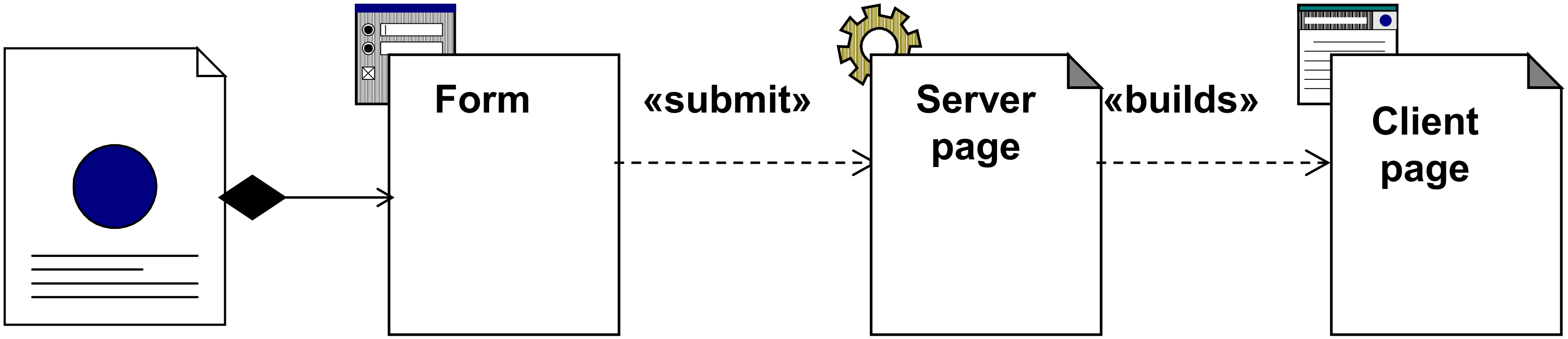
Our design has to take into account how much of the application will be static and how much dynamic

- ❑ How much can be just HyperText Markup Language (HTML) documents that are served to every client?
- ❑ How much will have to be generated on the fly for specific clients?

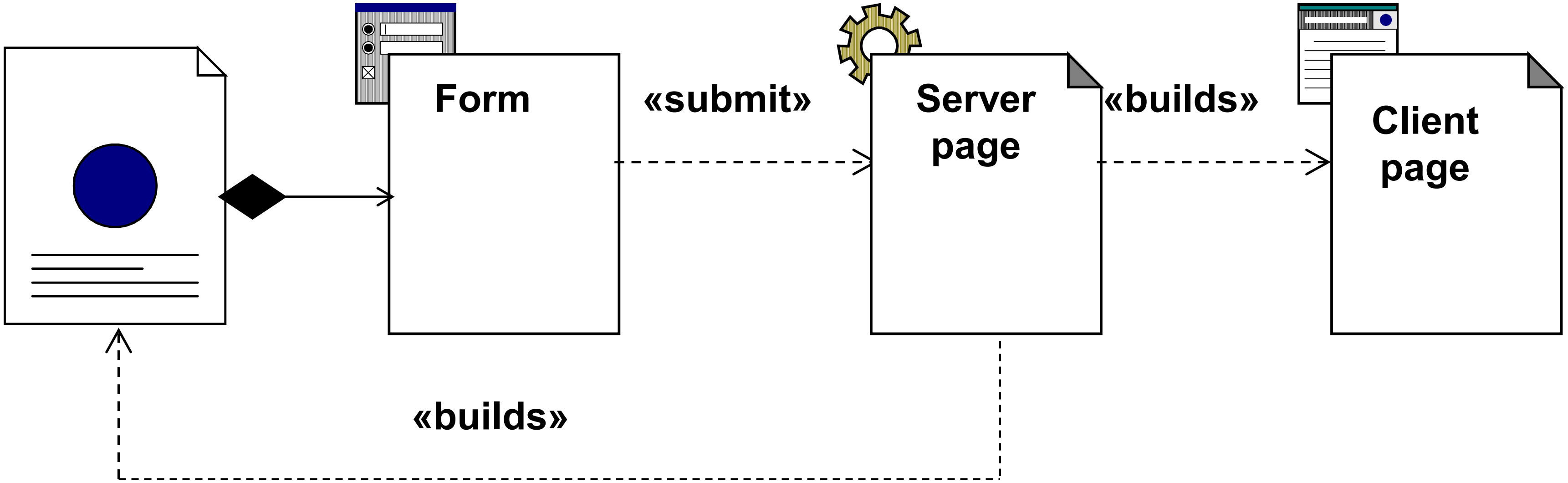
Server Pages

- ❑ Server pages are a technology for generating dynamic web pages on the server before sending them to the client as HTML
- ❑ They are programs that run on the server
  - – We will use server pages written in Java, but they can be written using other languages

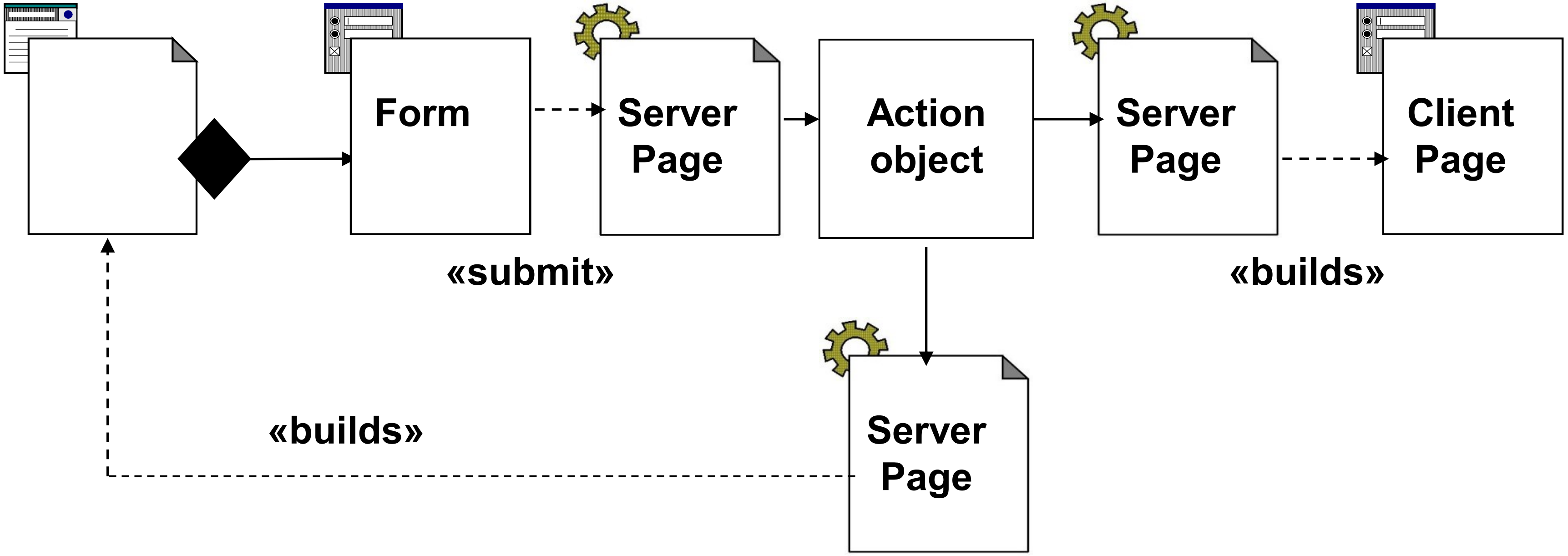
Server Pages and Client Pages



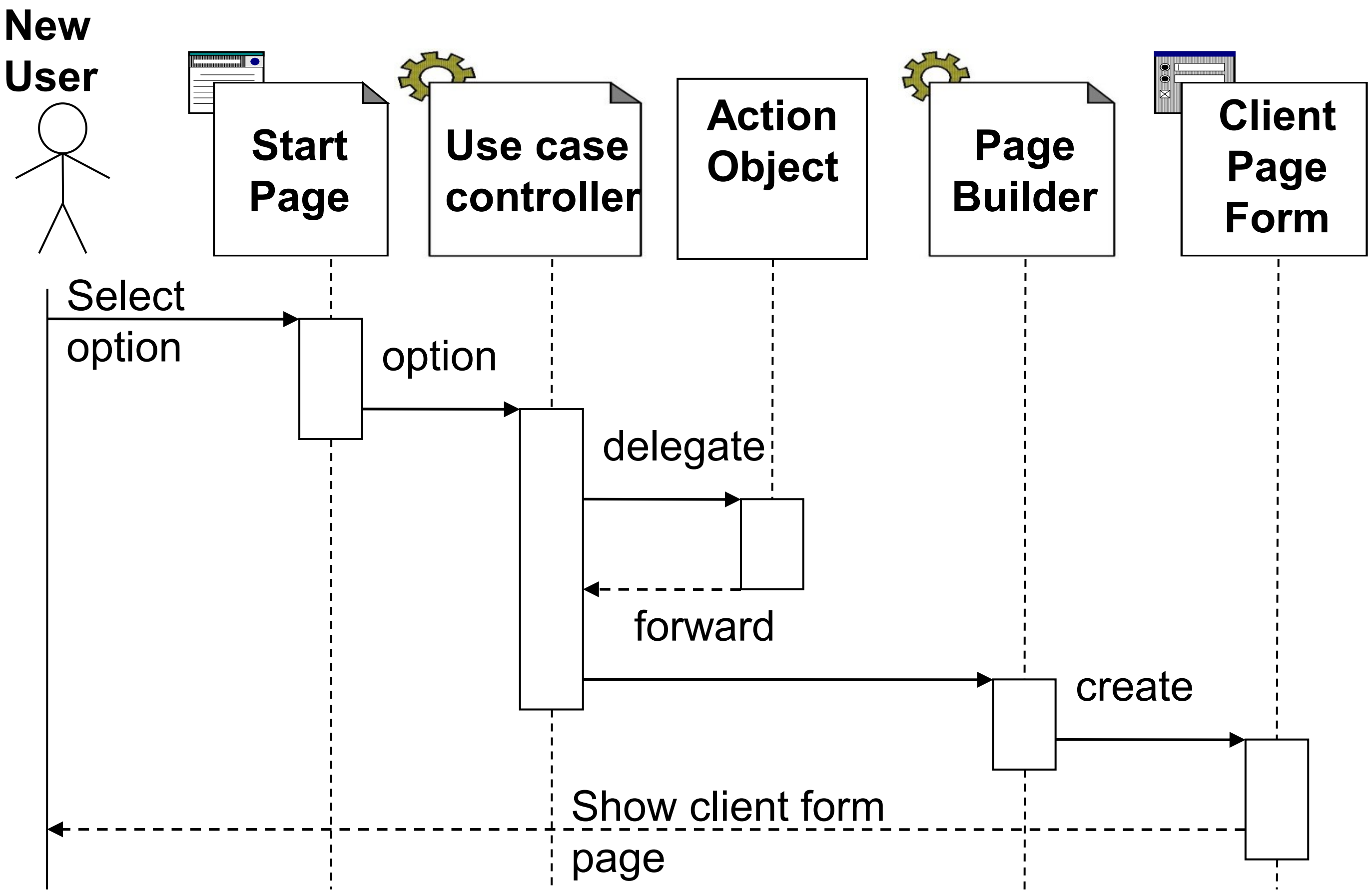
Dynamic Form Generation



Specialized Server Pages



Dynamic Webflow Sequence Diagram

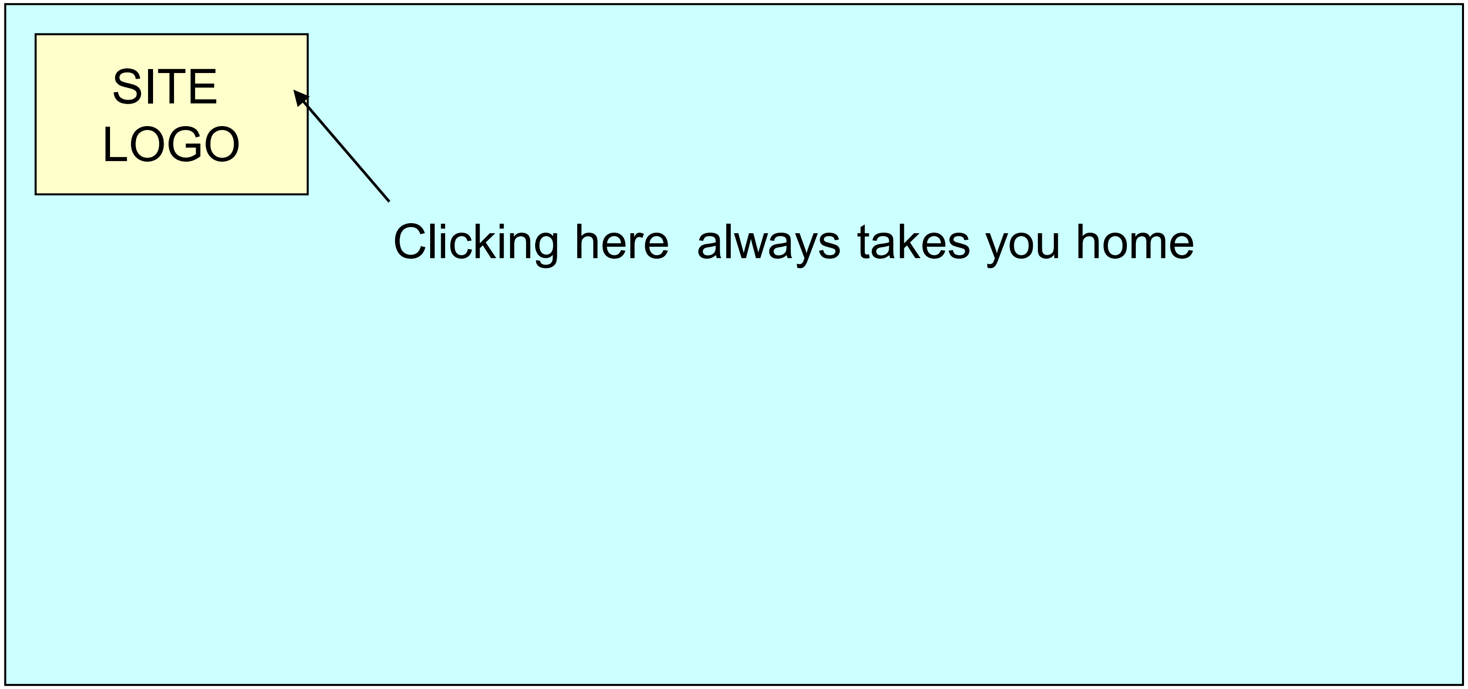


Design Patterns for Web Page Structure

We will look at some web design patterns that help us to structure our applications

- ❑ Site logo at top left
- ❑ Navigation bar
- ❑ Breadcrumbs
- ❑ Three region layout
- ❑ Home page
- ❑ Site map
- ❑ Store content in the database

Site Logo at Top Left



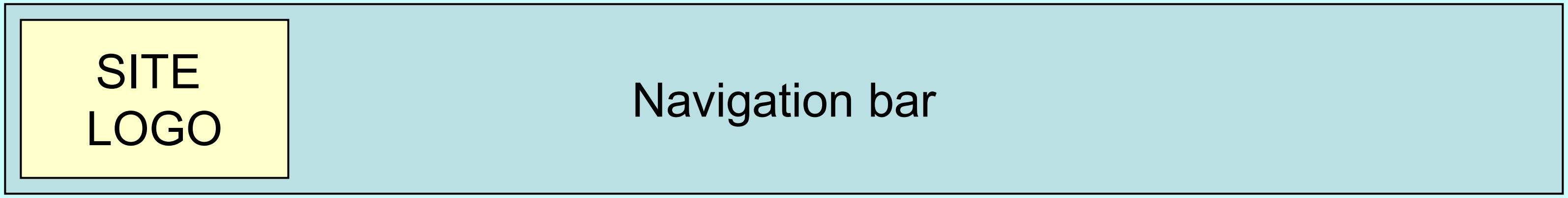
SITE LOGO

Clicking here always takes you home

Use Cases

- ❑ Our use cases will be starting points for user navigation
- ❑ They will frequently appear in a navigation bar across the top of the page
- ❑ The left hand side can be used for service navigation (i.e. what is inside the current use case)

Navigation Bar



The navigation bar includes:

- The site logo (home page link)
- Information about the organisation / company
- Privacy policy
- Contact information

For workflow / sales sites:

- Registration and log-in
- Checkout
- Shopping basket
- Account information

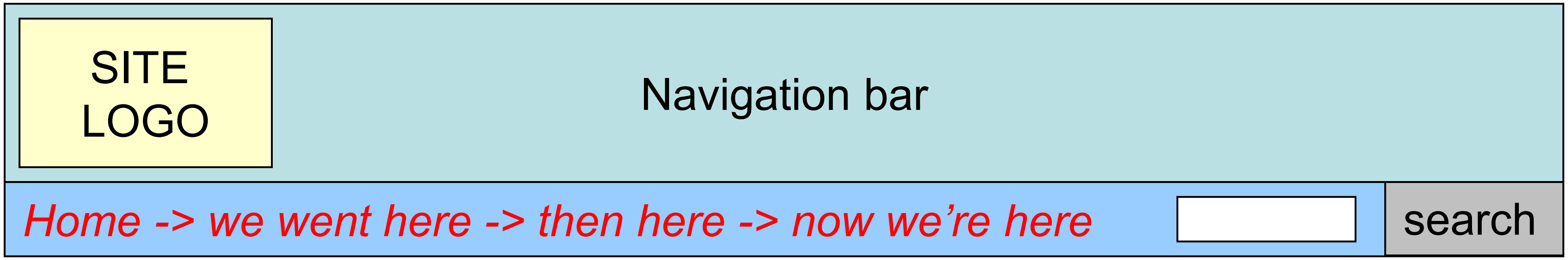
Other possibilities:

- Downloadable items
- Site map
- Communities
- Frequently asked questions
- News and press releases
- Jobs

etc…

Breadcrumbs

- ❑ Tell the user where they are relative to the home page
- ❑ Secondary to navigation bar
- ❑ May include a search box



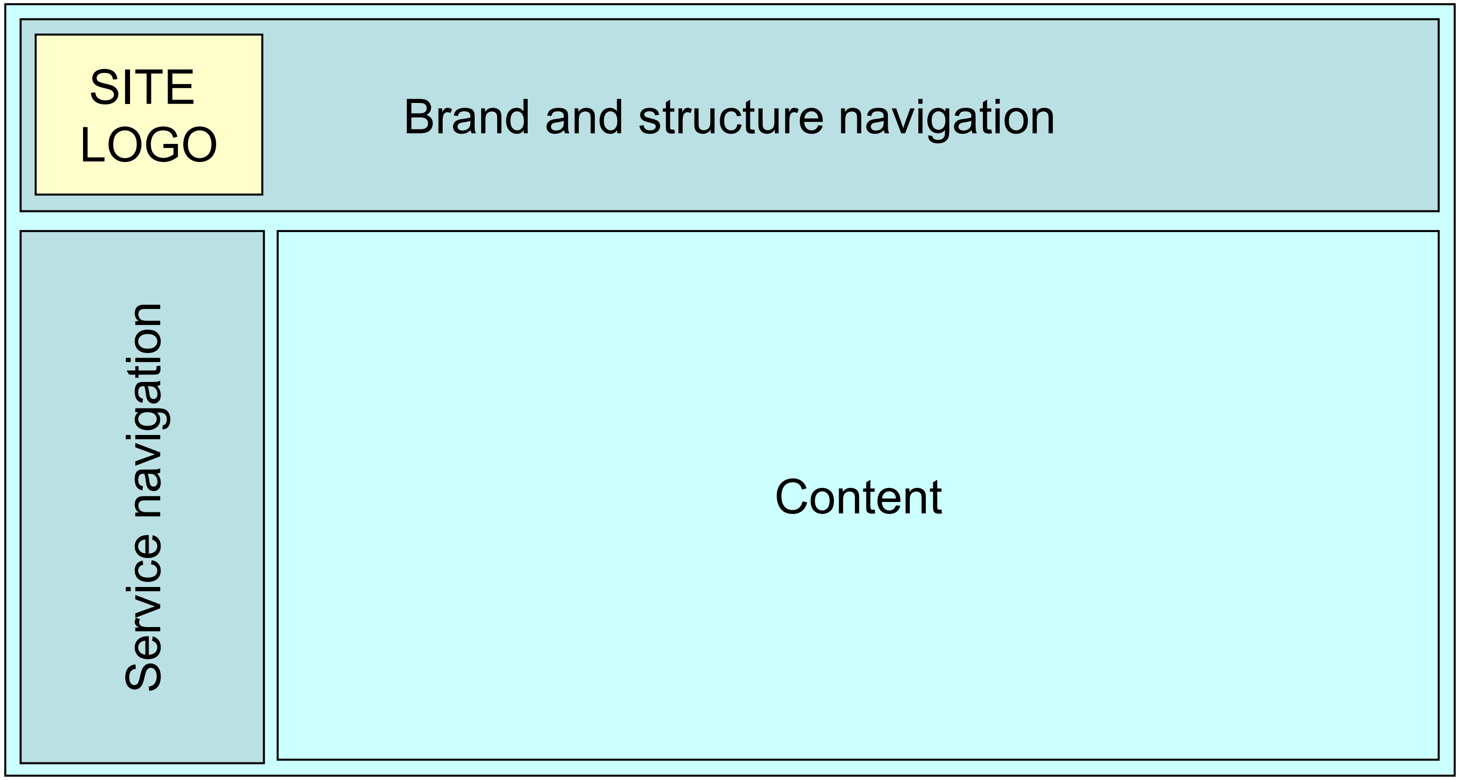
SITE LOGO

Navigation bar

Home -> we went here -> then here -> now we're here

search

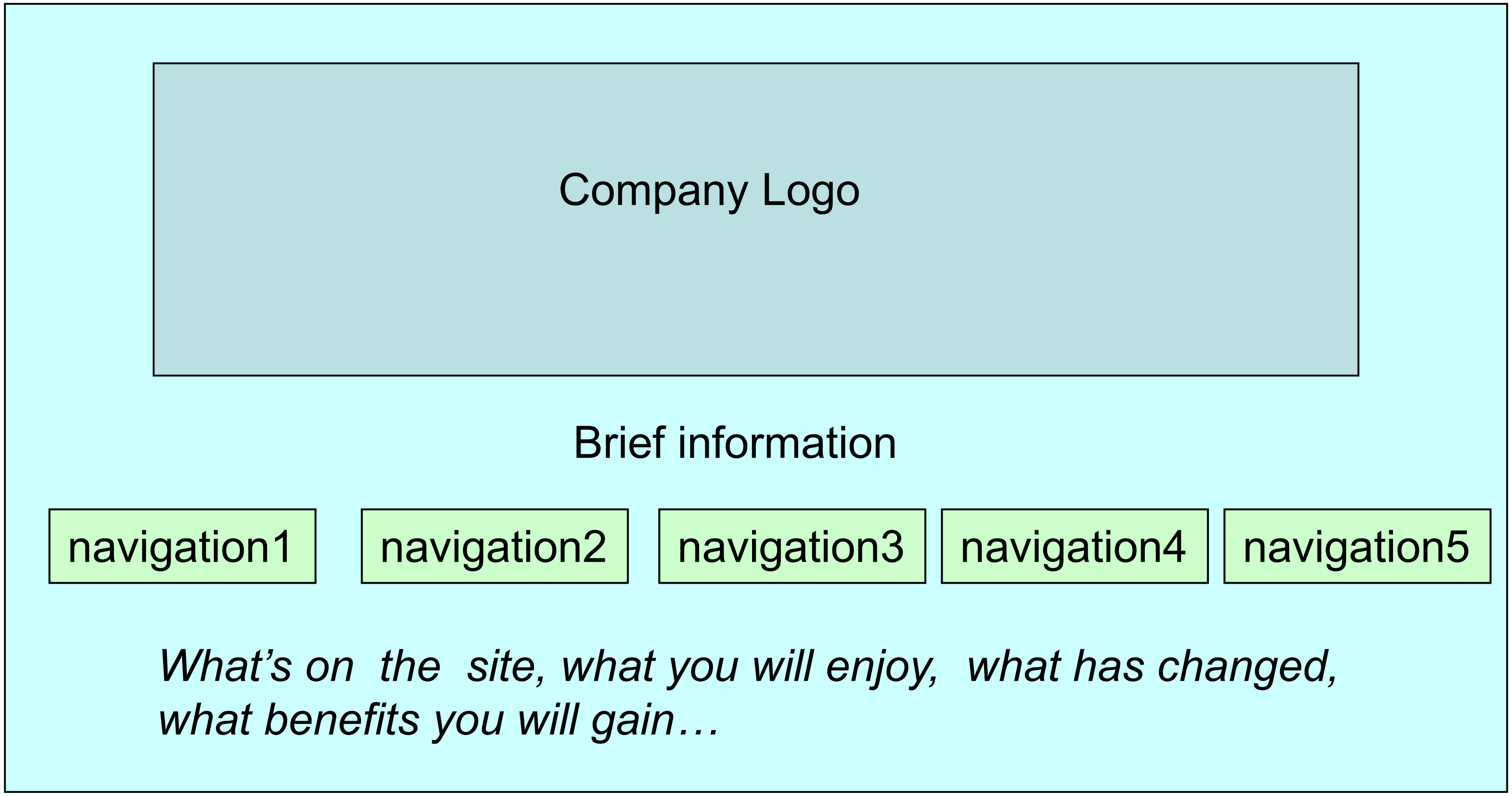
Three-Region Layout



Home Page

- ❑ Can be an exception to the three-region layout
- ❑ But not a splash screen – needs to include navigation to the main use cases

Home Page Example Layout



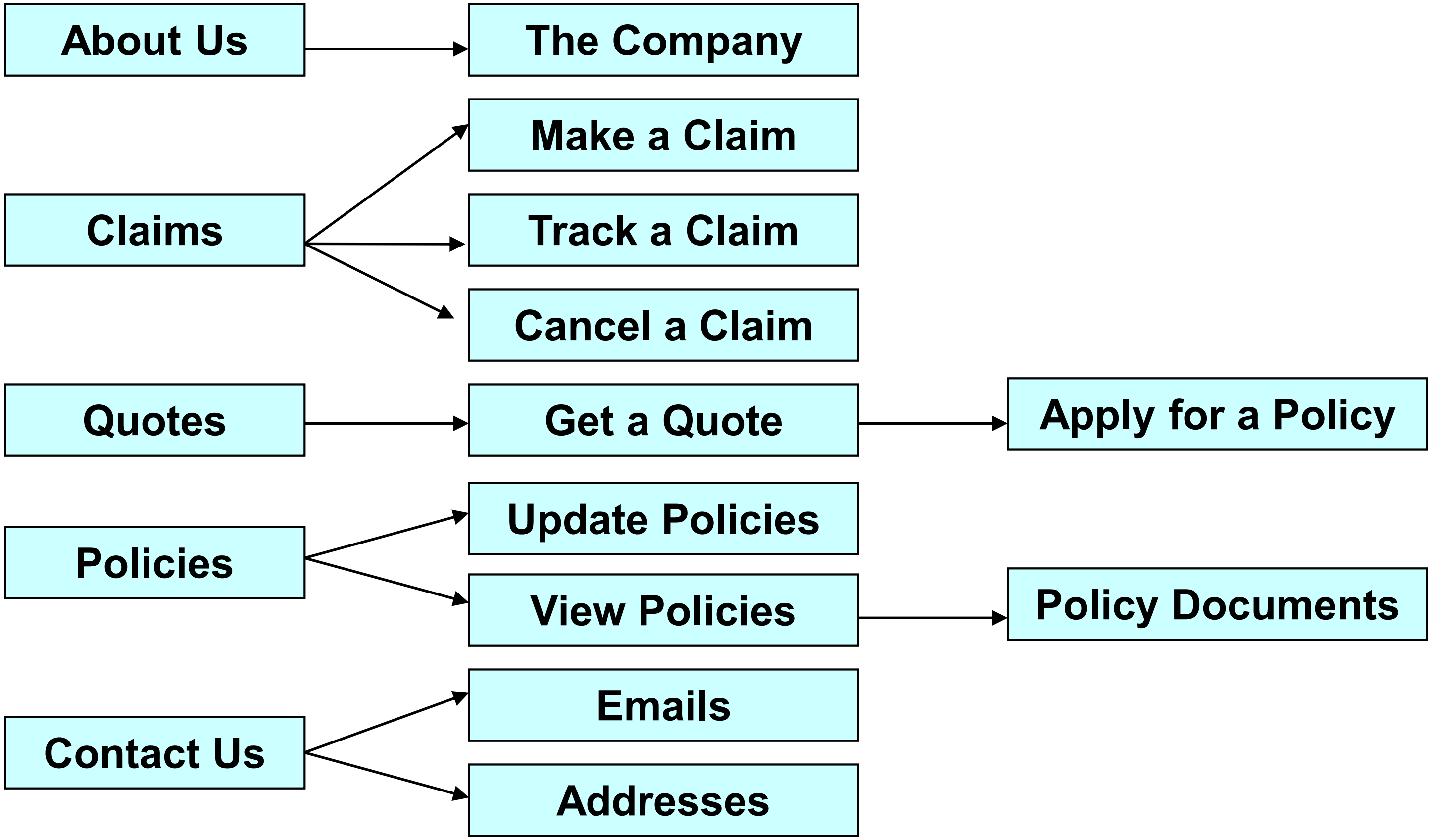
Company Logo

Brief information

navigation1 navigation2 navigation3 navigation4 navigation5

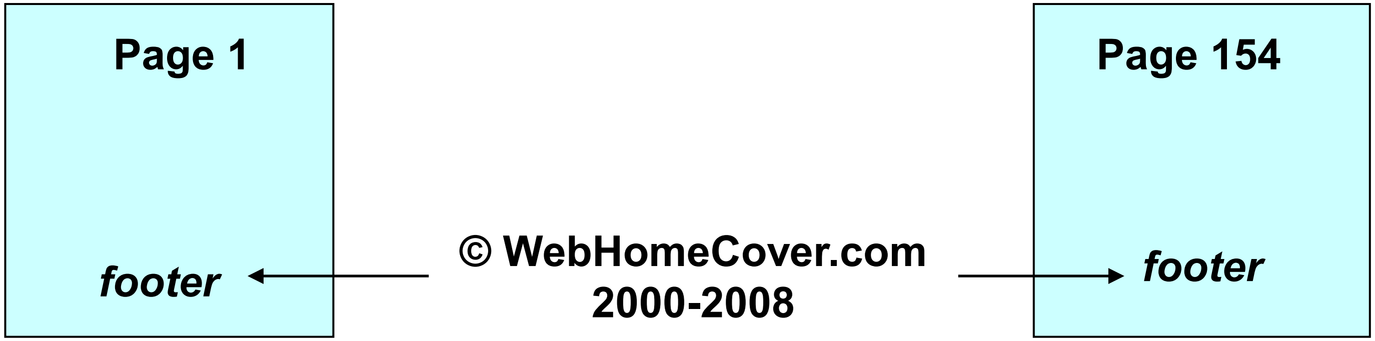
What's on the site, what you will enjoy, what has changed, what benefits you will gain…

Site Map


About Us
The Company
Claims
Make a Claim
Track a Claim
Cancel a Claim
Quotes
Get a Quote
Apply for a Policy
Policies
Update Policies
View Policies
Policy Documents
Contact Us
Emails
Addresses

Store Content in the Database

Some content is used in many different places and/or can change frequently



Page 1

footer

© WebHomeCover.com 2000-2008

Page 154

footer

General Design Guidelines

- ❑ Design around existing content, not future content
- ❑ Avoid unnecessary images
- ❑ Exploit hyperlinks
- ❑ Use Cascading Style Sheets
- ❑ Make navigation flow
- ❑ Visit your own site regularly

MASSEY UNIVERSITY

SCHOOL OF FUNDAMENTAL SCIENCES

End of Lecture

158.256 Web Application