Aspects of the Design of a Web Application
- software architecture
- database schema design
- software design
- detailed graphical layout of pages
- information design

Software Architecture

What is a software architecture?

*The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them.*

L. Bass, 1998

It is an abstract representation of a system that hides internal details of specific components.

Elements of an architectural model:

*Components:* represent key computational entities within a system. Granularity of a component can vary. Typically a component represents multiple objects, and sometimes entire systems. For example, in a Web application architecture each database is typically an entire component, even though it may represent hundreds of thousands of lines of code. At the same time, components inside the Web application server can have many fewer objects.

*Connectors:* represent communication paths among components

A typical Web application architecture has a browser, possibly components inside the browser (JavaScript), Web application server, application-specific components inside the Web app. server, one or more databases.

Software architectures are typically represented using boxes and arrows diagrams, where the boxes are components, and the arrows are connectors.

Database Schema Design

This was covered in CS 180. In short, the set of database tables, and for each table, the name and type of each column. Specify which columns are indexed.
Software Design

Typically performed using Unified Modeling Language (UML).

There are multiple diagram types in UML. For software design work, UML structure and sequence diagrams are the most useful. Only UML structure diagrams will be covered in class (see the readings page for additional readings on UML diagrams).


Classes:

```
Car
+ registration number : String
+ date : CarData
+ speed : Integer
+ direction : Direction

+ drive(speed : Integer, direction : Direction)
+ getData() : CarData
```

*Figure 4.10 The class Car has attributes and operations. The operation drive has two parameters, speed and direction. The operation getData has a return type, CarData.*

Relationships

Association

```
Author

Uses

Computer
```

*Figure 4.15 An author uses a computer. The Author class has an association to the Computer class.*

```
Person

1: *

owns

Car

0: *

ownsby

Car
```

*Figure 4.16 A person owns many (zero-to-many) cars. A car can be owned by many (one-to-many) people.*

Aggregation

Shared aggregation (referential containment)
**Figure 4.34** A team is composed of team members. One person could be a member of many teams. The model shows an example of a shared aggregation, where the people are the shared parts.

Composition aggregation (inclusion containment)

**Figure 4.36** The diamond shows the composition aggregate; the window contains (is aggregated of) many menus, buttons, listboxes, and texts. All types of aggregation can have a name.

Generalization (class/subclass)

**Figure 4.41** Vehicle is a general class (superclass) derived to specific classes (subclasses) via inheritance (generalization – specialization).