CONCEPTS

Elec 4309 Senior Design

Wendell H Chun

September 12, 2017

Overview

- Concepts
  - What concepts underpin design?
- Practices
  - Systems reflect principles and practices used in their construction

"The time has come," the guru said,
"To talk of many things:
Of use – and cases – and object models –
Of processes – and strings –
And why notations unify –
And if we need such things."

Introduction

- Conceptual Design
  - ♦ Show components at abstract level
  - ♦ Illustrate general interactions
  - ♦ Note: not completely specified
- Goals
  - ♦ Convince client of our ability to solve the problem
  - ♦ Jumpstart discussion

Goal of Conceptual Design

Arrive at an agreed-upon design that is ready for detailed design and implementation or fabrication.

- Design approach agreed upon by the sponsors and the team
- Presented at the Conceptual Design Review (aka Preliminary Design Review)
- Reduces risk of investment of time and effort into the wrong approach
- In industry, often triggers authorization for next round of funding.

Conceptual Design

- Identify data content and describe data at an *abstract*, or conceptual level (so it can be implemented in any system)
- Should represent real-world objects in the most *realistic* way possible
- Should reflect *what* can be done with this design with a clear understanding of engineering rules or constraints
- Ensure that all data *needed* are in the model, and that all data in the model are needed

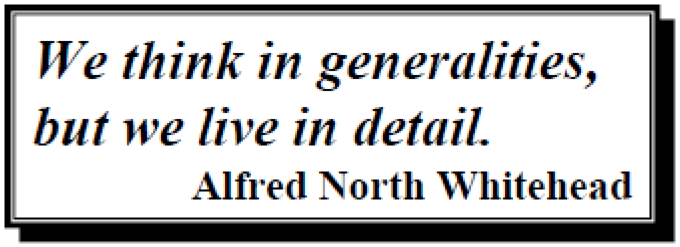
Conceptual Data Design

- Prepared at beginning of project
- High level view of how the client sees the data
- Top down process
- Not concerned with details
- Normally prepared using brainstorming approach
- Identification of Entities, Attributes and Relationships represented using Entity Relationship Diagram
- ERD is constantly modified as project progresses and more is learnt about application’s data

Dangers

- **Premature fixation** - focus on a solution without considering alternatives
- **Bounded ideation** - limitations that the design environment can place on generating ideas
- **Circumscribed thinking** - influence that the outside sources can have on design decision making
- **Paralysis by analysis** - inability to move forward with concept

Concepts



We think in generalities,
but we live in detail.
Alfred North Whitehead

Guiding Principles

- Discuss and get different perspectives
- Iterate, iterate, iterate!
- Consider many alternatives
- “To get a good idea, get lots of ideas”

- Empathize with the user

Techniques used in Conceptual Design

- Scenarios
  - ♦ Basis for the overall design
  - ♦ Basis for technical implementation
  - ♦ Means of cooperation within design teams
  - ♦ Means of cooperation across professional boundaries (multidisciplinary teams)

Conceptualizing

- Think of your project as the mini-system where you're playing multiple roles as a user, system designer, and so on as you already described in your proposal

3 Steps in Conceptual Design

- Requirement analysis
  - ♦ Engineering rules → constraint, entity, relationship
  - ♦ Users' requirement → data, functionalities
- Develop conceptual model
  - ♦ Use design tool such as ERD or UML
  - ♦ Normalization techniques

Decomposition of System

- Divide big problem into smaller problems
- Known systems have known decompositions
  - ♦ New products need new decomposition

Architecture

- An architecture defines the arrangement of structural elements in a system
  - Relates to form and function
  - Architectural style is the underlying structuring principle and philosophy

Architecture: where the rubber meets the sky.

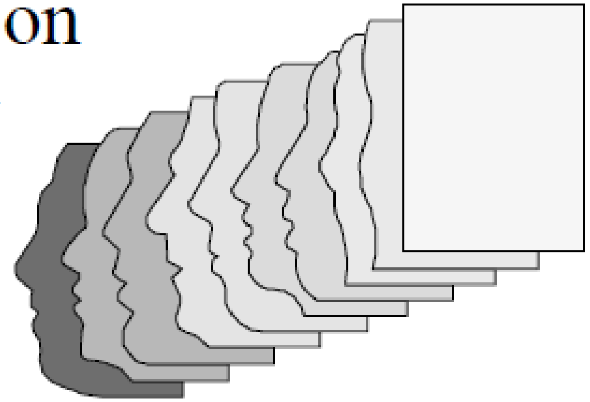
John Daniels

Structure

- Architecture defines the form and characteristics of a system
    - Properties include organizational structure and development process…
    - As well as the more visible features normally associated with structure

Communication

- Architecture also defines a model for communication between developers
  - Affects how a system is built and will evolve
  - Defines a vocabulary and framework for work
- Communication is founded on expressiveness, consistency and clarity



Documentation

- The words *documentation* and *communication* are not synonymous
  - Communication may include documentation
  - Documentation need not imply communication
- Producing design documentation is no substitute for doing design

Meaning

- Elements and combination of elements in a system have meaning
  - ♦ Meaning can come from the problem domain
  - ♦ Meaning can arise from the conventions and artifacts of construction

Requirements

Requirements establish a context from which and into which a system is built

Functional requirements focus on purpose

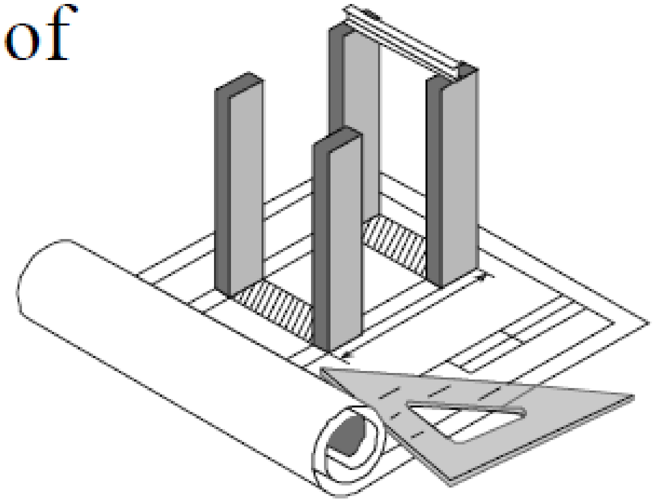
They are often the sole or main focus, because they are quantifiable

Non-functional requirements focus on qualities

They are not 'impure'

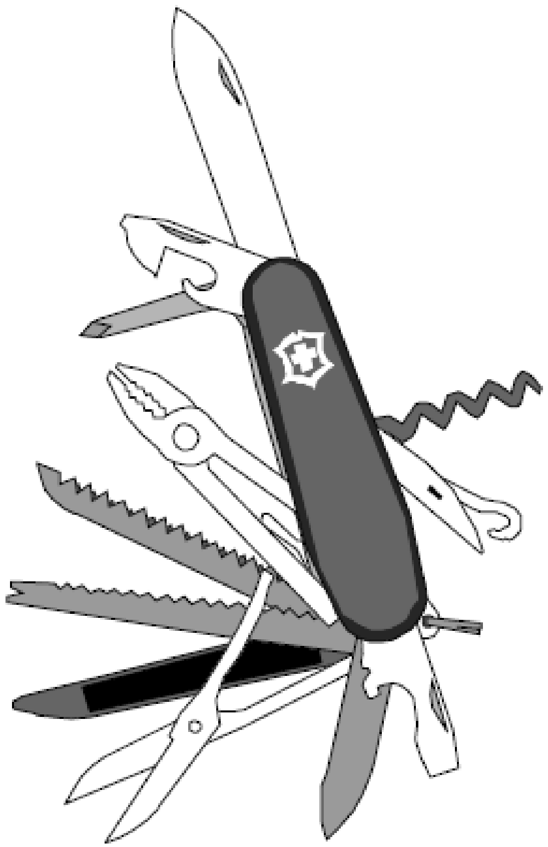
Models

- A model is an abstraction from a point of view for a purpose
  - Discover and document constraints
  - But don't confuse the map with the territory
- Modelarity is the degree of correspondence between problem and solution



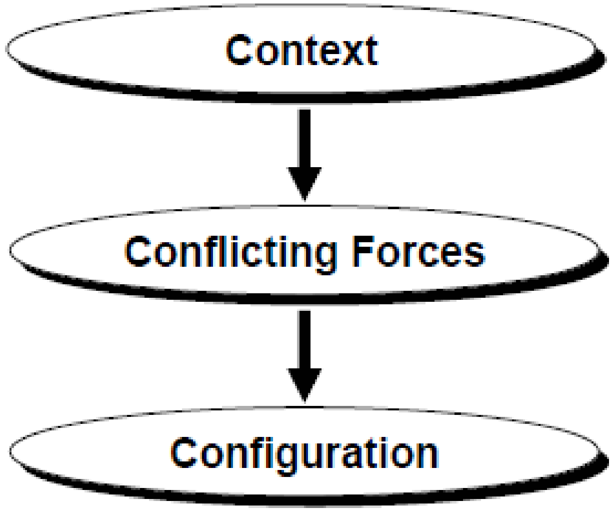
Simplicity and Complexity

- Essential complexity versus actual complexity
- *Simple* is not the same as *simplistic*
    - *Simple* implies a close fit between essential and actual complexity
    - *Simplistic* implies ignoring constraints and requirements



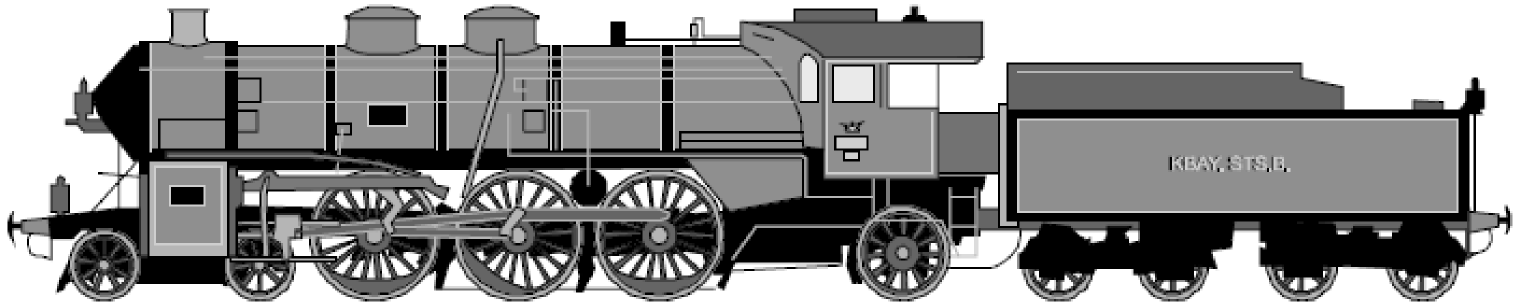
Patterns

- Patterns document recurring solutions
  - Act as a map to understand existing systems
  - Can be applied proactively in development of new systems
- Patterns have meaning
  - They distil successful experience
  - Allow clear access to complex structures



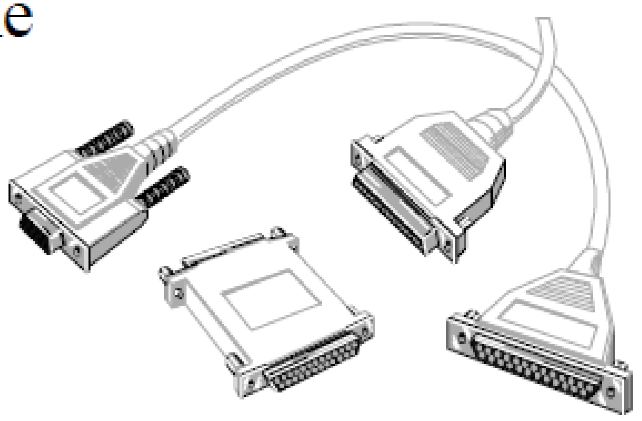
Partitioning

- Quality and qualities of separation
  - Coupling describes interconnectedness
  - Cohesion describes intraconnectedness
- Separation introduces connections
  - Connections can be directional or cyclic



Interfaces

- Interfaces define the seams in a system
  - Legal contracts between connected components
  - Separation of intent from realization
- Separation implies a need for unification
  - Separation is no good in the absence of connection



Time

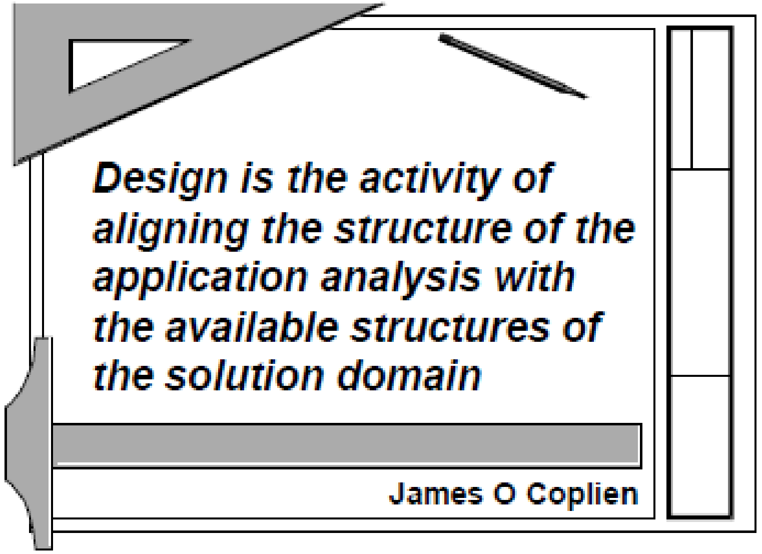
- The architecture of a system determines how it will evolve and adapt
  - Architecture is not simply about the present, but also influences the shape of things to come

Design involves assumptions about the future of the object designed, and the more the future resembles the past the more accurate the assumptions are likely to be. But designed objects themselves change the future into which they will age.

Henry Petroski [Petroski1992]

Design

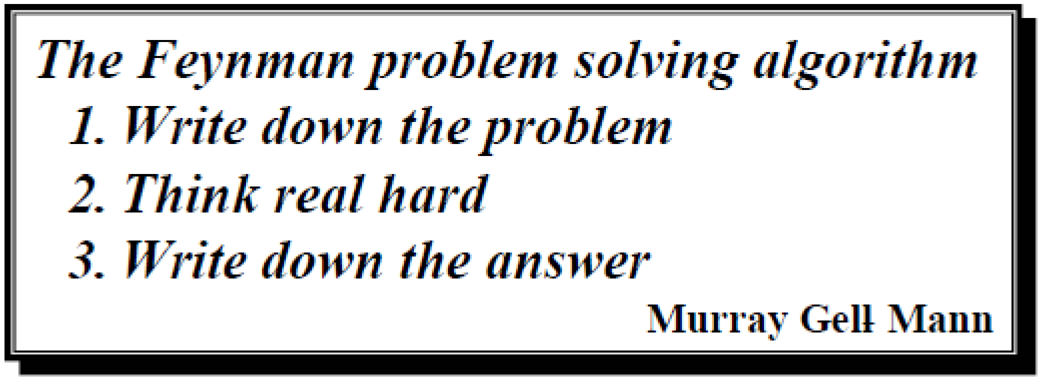
- Design is a creational and intentional act
  - Conception and construction of a structure on purpose for a purpose



Design is the activity of aligning the structure of the application analysis with the available structures of the solution domain

James O Coplien

Practices



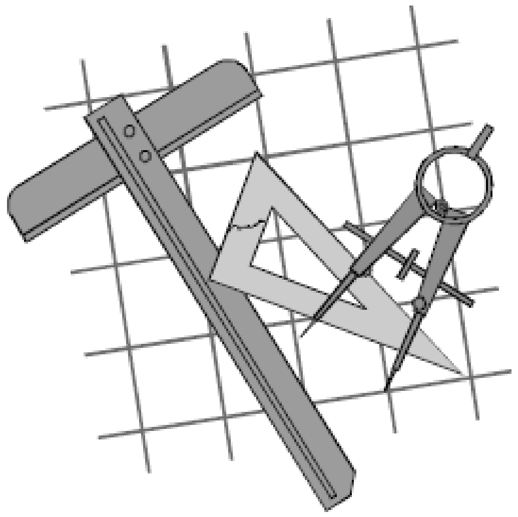
The Feynman problem solving algorithm

1. ***Write down the problem***
2. ***Think real hard***
3. ***Write down the answer***

Murray Gell Mann

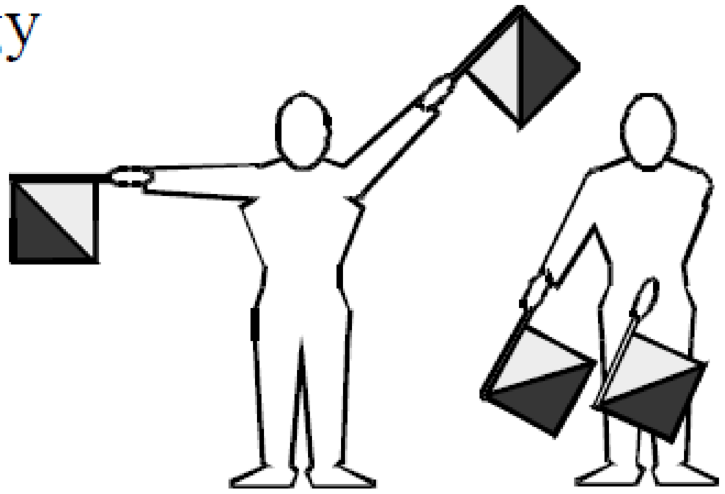
Context Sensitivity

- Solution structure is sensitive to details of purpose and context
  - Problem and solution feed forward and back
- Context free design is meaningless
  - No universal or independent model of design
  - Context can challenge and invalidate assumptions



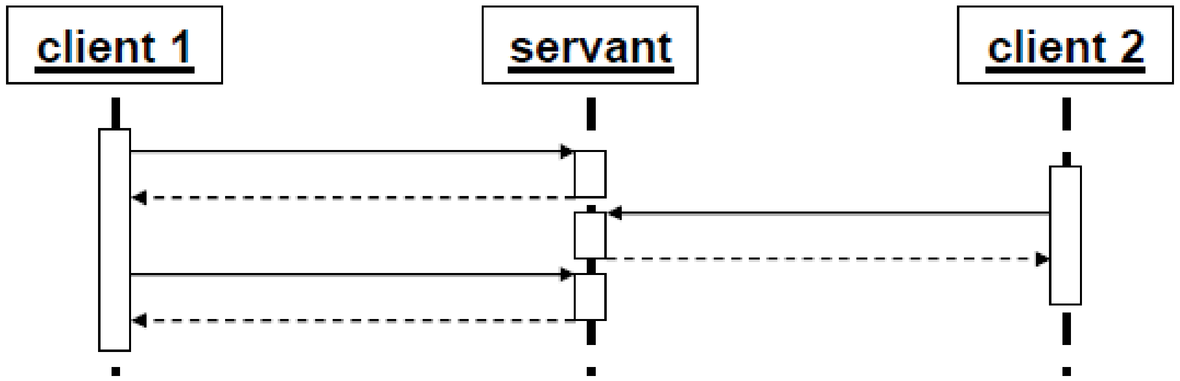
Idioms

- Idioms are language, language model, or technology specific patterns
  - Common conventions of style and usage
  - Dependency on or originating from specific features of a technology



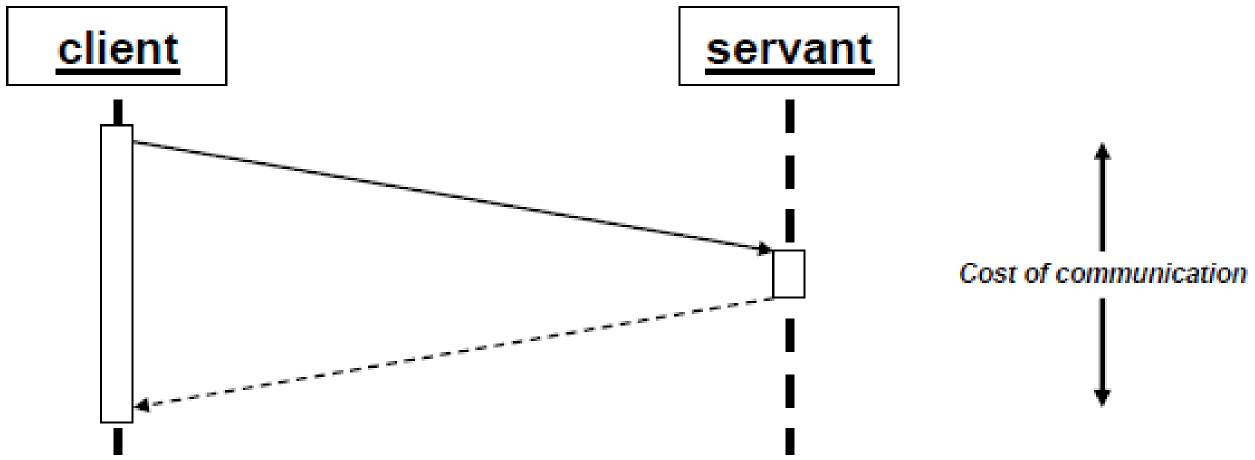
Concurrency as a Context

- Synchronization is required to ensure consistent and coherent state
- Property style programming is inappropriate
    - ♦ E.g. MIDL properties, OMG IDL *attributes*, *set* and *get* operation pairs, etc.



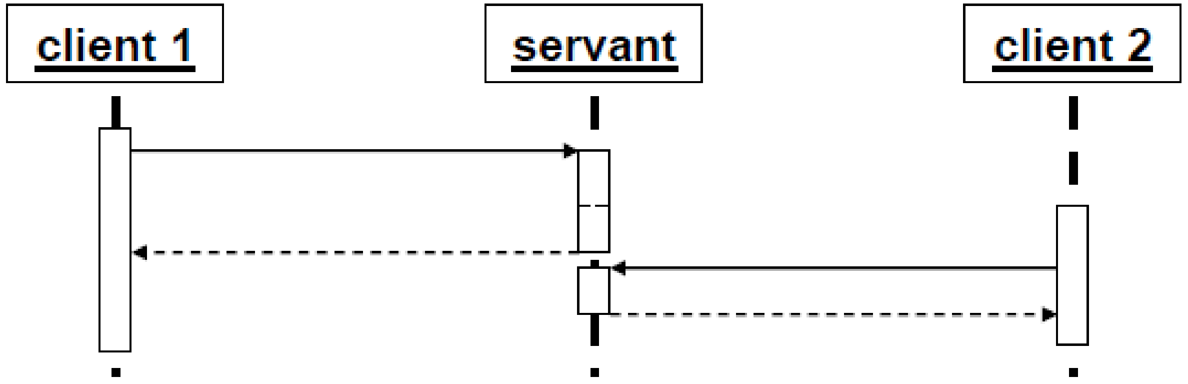
Distribution as a Context

- Concurrency is implicit
- Operation invocations are no longer trivial
  - Communication can dominate computation
  - Partial failure is almost inevitable



Context Can Affect Interface

- Define compound rather than primitive operations based on common usage
  - Combined Operation for operation sequences
  - Combined Attributes for attribute groups
  - Batch Operation handles repetition



Constraints

- Constraints bound the meaningful behavior of a system
  - ♦ Can be realized in types, exceptions, language features, etc.
- Constraints can be liberating
  - ♦ Ensuring what's true is true and what's not is not frees rather than binds a developer

$$F = G\,\frac{m_1 m_2}{r^2}$$

Minimalism

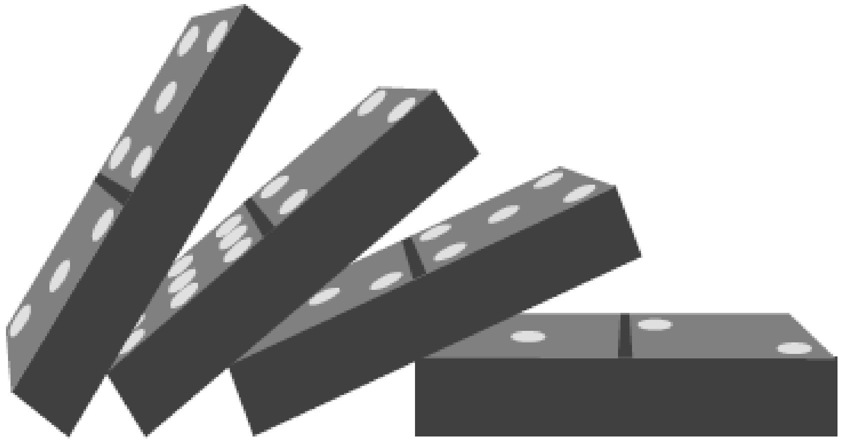
- Additional options and features can lead to confusion rather than clarity
  - Overachieving interfaces are weaker and more complex not stronger and simpler

The difference between a good and a poor architect is that the poor architect succumbs to every temptation and the good one resists it.

Ludwig Wittgenstein

Dependency Management

- Partition to minimise dependencies
  - Low coupling and high cohesion
- Put things together that change together
  - Where dependencies exist they should be on stable elements

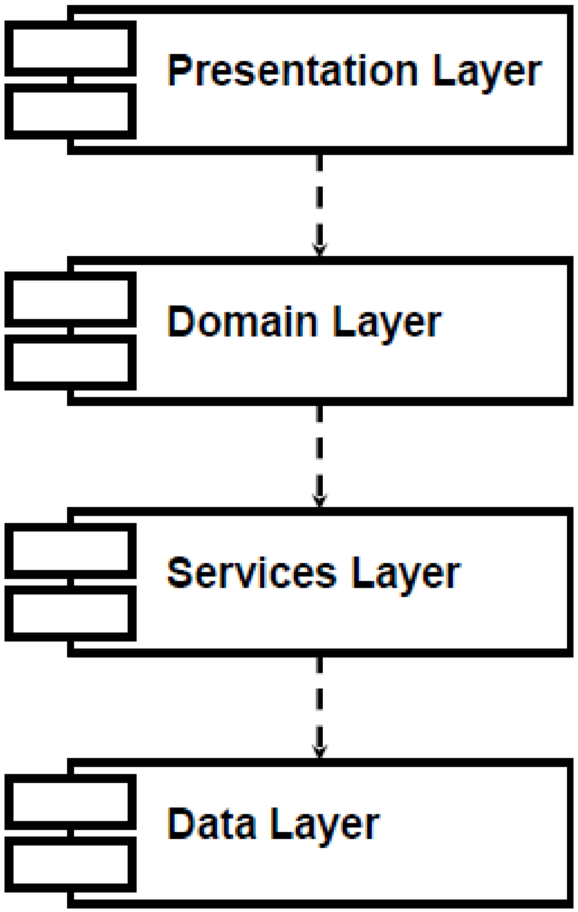


Decoupling

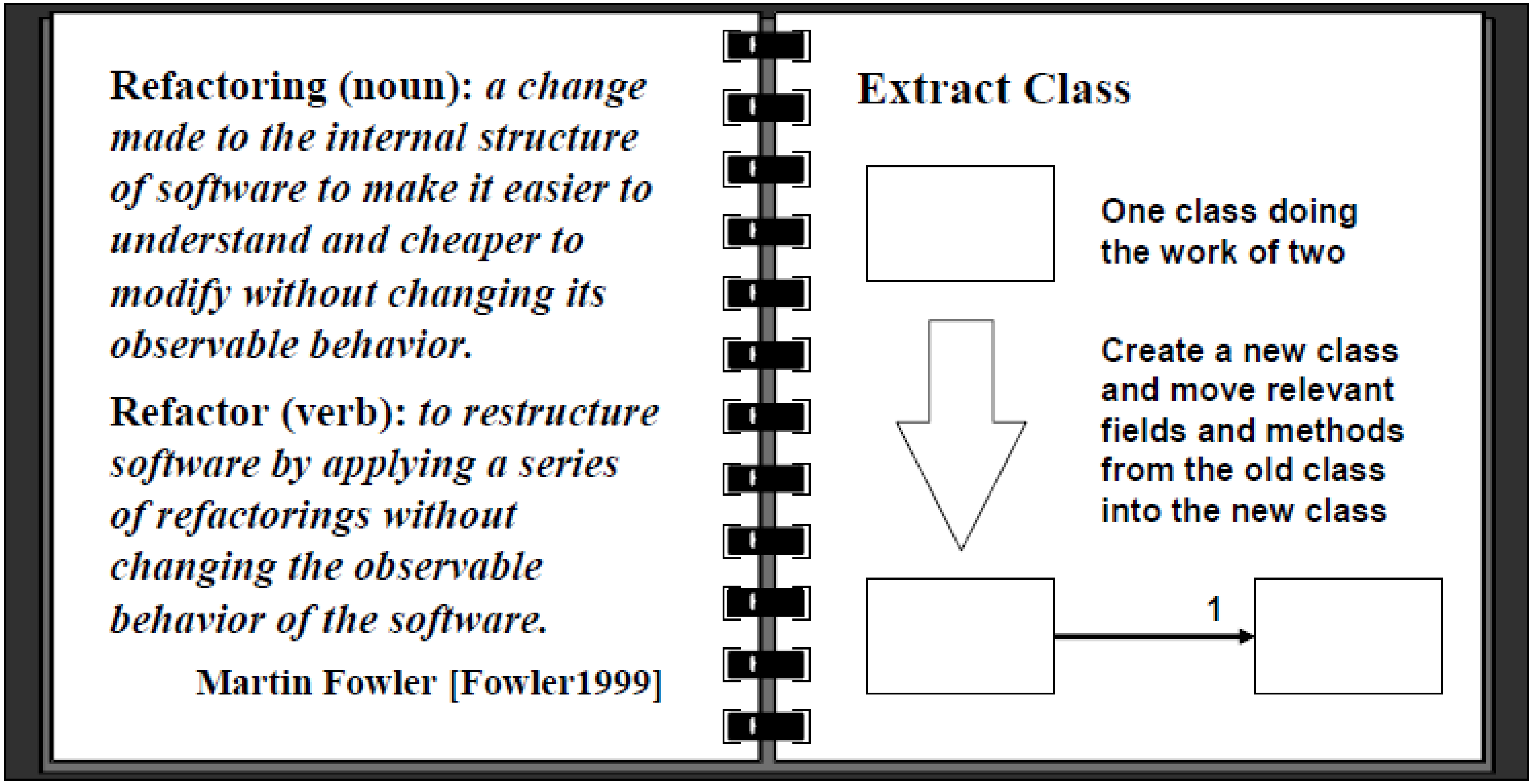
- Interface Decoupling
  - Separate client usage interface from creational implementation class
- Role Decoupling
  - Depend on the interface, the whole interface, and nothing but the interface
- Dependency Inversion
  - Can be used to break cyclic dependencies

Layering

- Layers are encapsulated levels in a system
- Systems may be layered with respect to…
  - Levels of abstraction
  - Rate of change
  - Development skills
  - Organizational structure



Refactoring



Refactoring (noun): *a change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behavior.*

Refactor (verb): *to restructure software by applying a series of refactorings without changing the observable behavior of the software.*

Martin Fowler [Fowler1999]

Extract Class

One class doing the work of two

Create a new class and move relevant fields and methods from the old class into the new class

Iteration

- Iterative development provides a framework in which a system can grow
  - Accommodates change and inspiration
  - Offers time for building and testing
- Empirically based
  - Tests assumptions as well as code
  - Accepts that change happens

Summary

- Analysis is the first act of design and implementation the last
  - Design includes conception and construction
- Simplicity should bound complexity
  - Identification and preservation of constraints
  - Management and reduction of dependencies

> ***Design is getting from here to there***
>
> **Henry Petroski [Petroski1992]**