Proiectarea si Arhitectura Sistemelor Software Complexe

Design and Architecture of Complex Software Systems

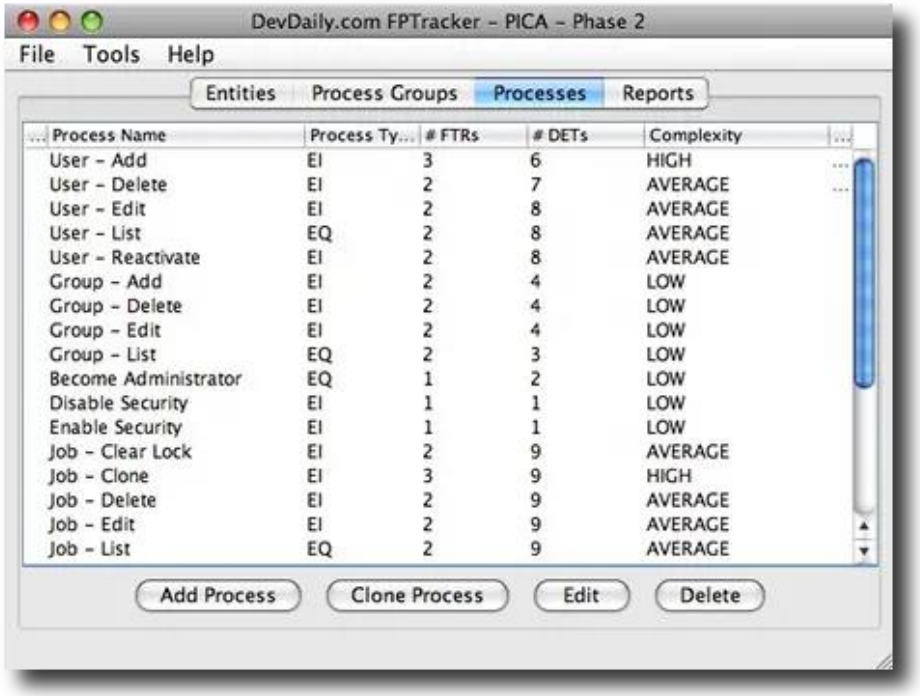
Conf.dr.ing. Ioana Şora

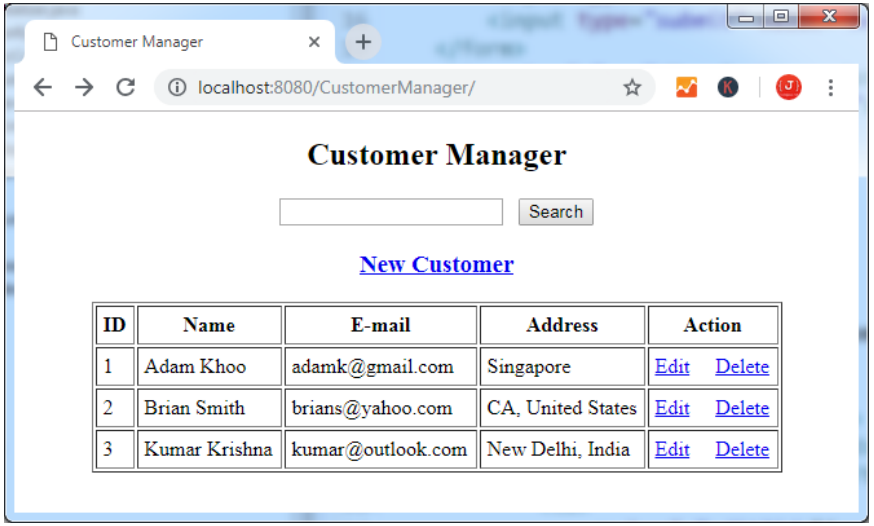
http://staff.cs.upt.ro/~ioana/arhit-engl/

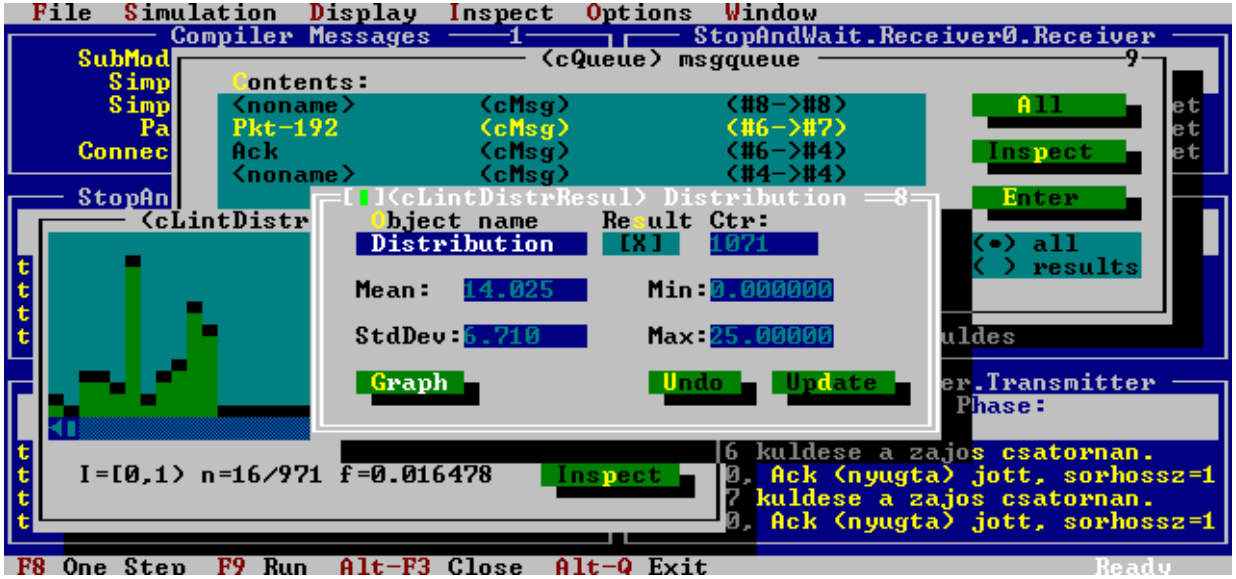
ioana.sora@cs.upt.ro

Quizz

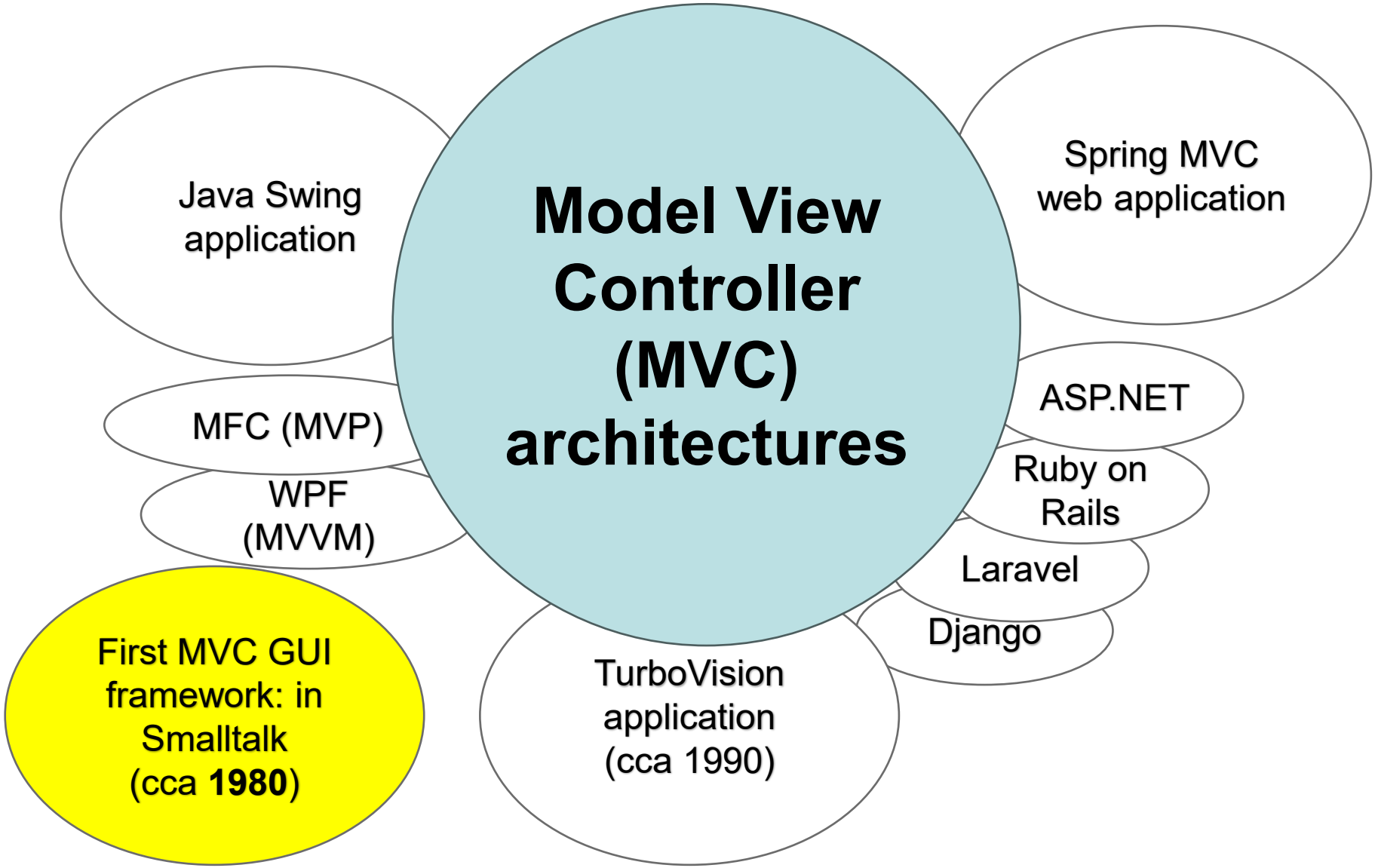
What do the following software systems have in common?


DevDaily.com FPTracker - PICA - Phase 2
File Tools Help
Entities Process Groups Processes Reports
Process Name | Process Ty... | # FTRs | # DETs | Complexity
User - Add | EI | 3 | 6 | HIGH
User - Delete | EI | 2 | 7 | AVERAGE
User - Edit | EI | 2 | 8 | AVERAGE
User - List | EQ | 2 | 8 | AVERAGE
User - Reactivate | EI | 2 | 8 | AVERAGE
Group - Add | EI | 2 | 4 | LOW
Group - Delete | EI | 2 | 4 | LOW
Group - Edit | EI | 2 | 4 | LOW
Group - List | EQ | 2 | 3 | LOW
Become Administrator | EQ | 1 | 2 | LOW
Disable Security | EI | 1 | 1 | LOW
Enable Security | EI | 1 | 1 | LOW
Job - Clear Lock | EI | 2 | 9 | AVERAGE
Job - Clone | EI | 3 | 9 | HIGH
Job - Delete | EI | 2 | 9 | AVERAGE
Job - Edit | EI | 2 | 9 | AVERAGE
Job - List | EQ | 2 | 9 | AVERAGE
Add Process
Clone Process
Edit
Delete

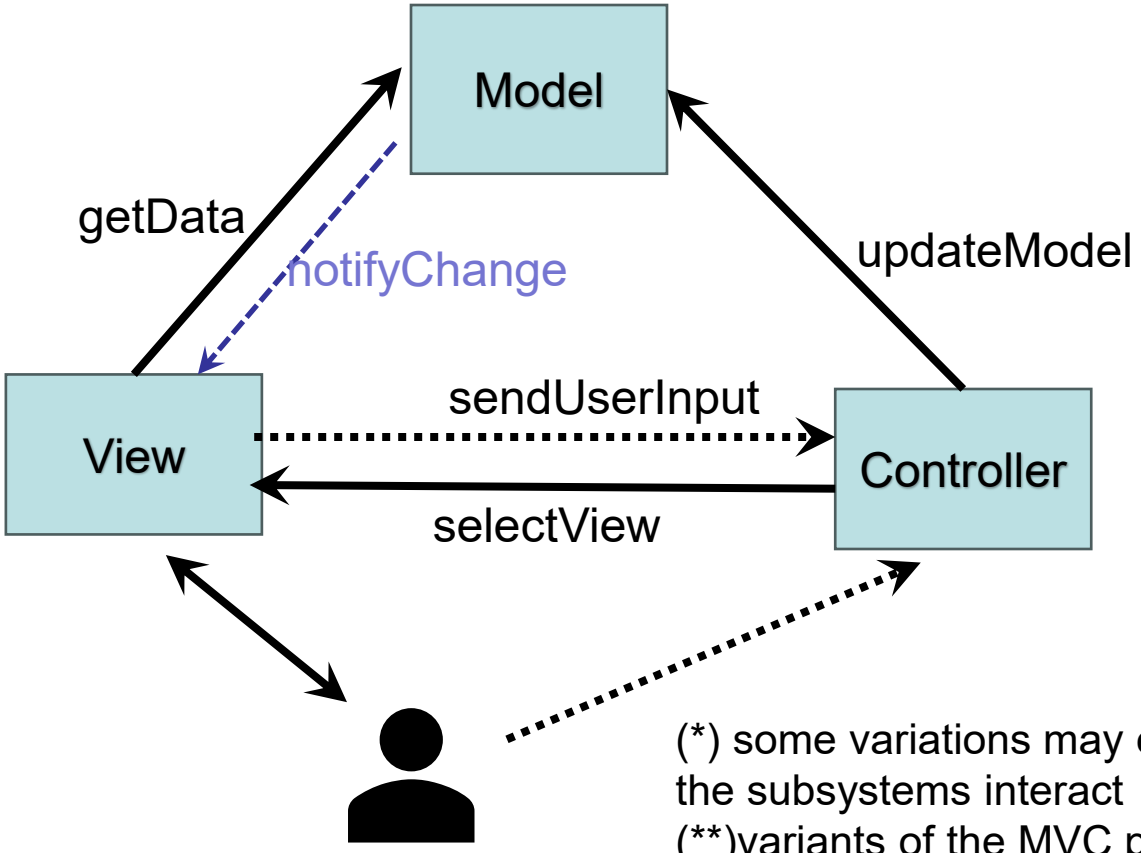

Customer Manager
localhost:8080/CustomerManager/
Customer Manager
Search
New Customer
ID | Name | E-mail | Address | Action
1 | Adam Khoo | adamk@gmail.com | Singapore | Edit Delete
2 | Brian Smith | brians@yahoo.com | CA, United States | Edit Delete
3 | Kumar Krishna | kumar@outlook.com | New Delhi, India | Edit Delete


File Simulation Display Inspect Options Window
Compiler Messages
StopAndWait.Receiver0.Receiver
(cQueue) msgqueue
Contents:
<noname> (cMsg) (#8->#8)
Pkt-192 (cMsg) (#6->#7)
Ack (cMsg) (#6->#4)
<noname> (cMsg) (#4->#4)
All
Inspect
Enter
(•) all
( ) results
(cLintDistrResul) Distribution
Object name Result Ctr:
Distribution [X] 1071
Mean: 14.025
Min: 0.000000
StdDev: 6.710
Max: 25.00000
Graph
Undo
Update
I=[0,1) n=16/971 f=0.016478
Inspect
6 kuldese a zajos csatornan.
0, Ack (nyugta) jott, sorhossz=1
7 kuldese a zajos csatornan.
0, Ack (nyugta) jott, sorhossz=1
F8 One Step F9 Run Alt-F3 Close Alt-Q Exit
Ready

What do they have in common?



MVC architectural pattern for interactive systems



(*) some variations may exist in the way the subsystems interact
(**)variants of the MVC pattern: MVP, MVVM, etc

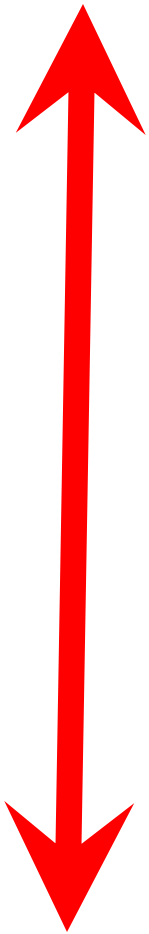
Technologies vs Patterns

Pattern

- In software:
- **Pattern = *general* solution to a *recurring* problem in a specific context**
- **Pattern =** a form of ***reuse***:
  - Reuse of ***wisdom***, ***experience*** (**no direct reuse of code!**)
  - Kind of partial ***reuse of design solutions***:
    - At different levels of abstraction
    - At different levels of detail
- **Pattern introduce a domain-specific *vocabulary* (a language of patterns)**

Pattern

Abstraction



Detail

"*An **architectural style** defines a **family** of software systems in terms of their **structural organization**. An architectural style expresses **components** and the **relationships** between them, with the **constraints** of their application, and the associated composition and design **rules** for their construction.*"

"*An **architectural pattern** expresses a **fundamental structural organization schema** for software systems. It provides a set of **predefined subsystems**, specifies their responsibilities, and includes **rules** and guidelines for organizing the **relationships** between them.*"

"*A **design pattern** provides a **scheme for refining the subsystems** of a software system, or the relationships between them. It describes a commonly-recurring structure of communicating components that solves a **general design problem** within a **particular context**.*"

Why is software architecture important?

- Communication between stakeholders
- Early design decisions
- Transferable, reusable model

Terminology, vocabulary ?

Starting points ? Reuse?

Established terminology, good practices, standards, success stories ….
PATTERNS

Course goals. Course topics

- Fundamental architectural styles and patterns
  - Laters; Pipes&Filters; Repository
  - Event-Driven Architectures. Publish-Subscribe infrastructures. (greenrobot; RabbitMQ)
- Architectural patterns for some relevant application domains:
  - Adaptive systems: The Reflection pattern. Meta Layer Architectures. Reflective languages. Applications of reflective languages.
  - Remote Procedure Call. Remote Method Invocation: Patterns (Remote Proxy); Middleware technologies examples (gRPC; RMI)
  - Data access patterns:
    - Decoupling data access - DAO pattern. Technology example – Spring Data
    - Mapping OO program model on Data models (Data binding, ORM)

Practical assignments

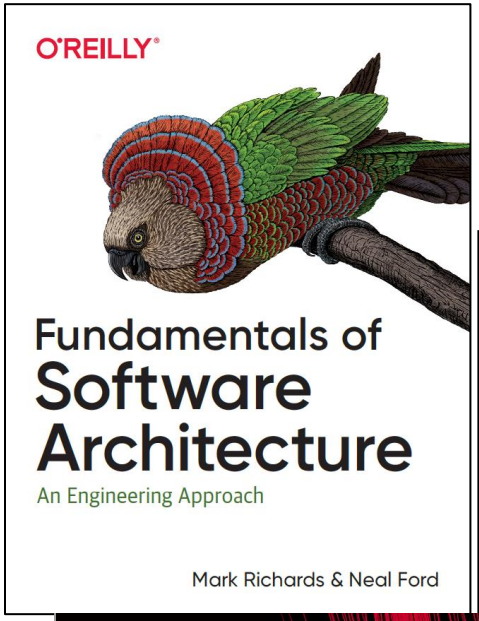
- Small projects where you have to apply/implement the patterns discussed in the lectures
- 4 main topics => 4 assignments. Each assignment 2-3 weeks
- Every assignment has multiple options, with various degrees of complexity, from easy to more difficult
- Can add bonus points for exam.
- By default, final grade = (lab + written exam +bonus points)/2
- Enough points (6 points) -> no more written exam at all
- Need *good* OO programming knowledge background. Can use implementation language of your choice, but examples in lectures are mostly Java-related
- PDSS (SSD): nice to have but not an absolute prerequisite

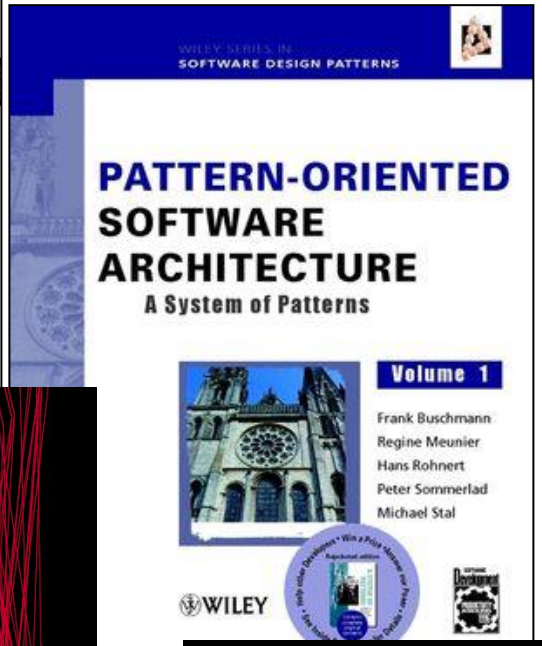
Course Webpage

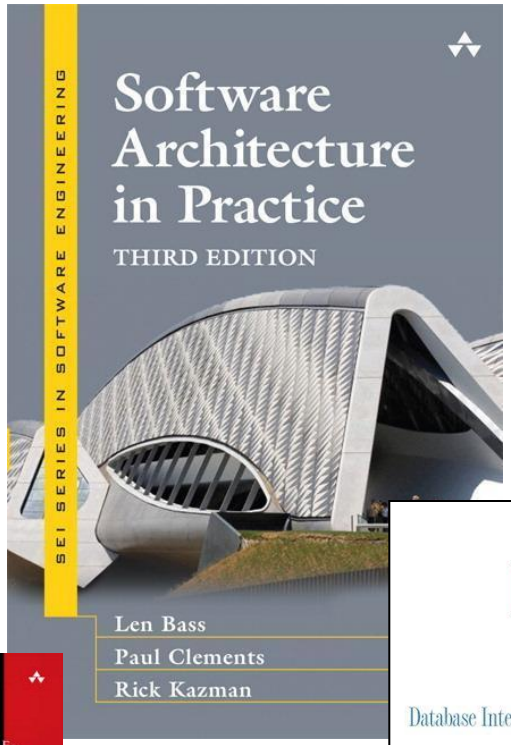
- On Campus Virtual
- PASSC https://cv.upt.ro/course/view.php?id=1604
- DACSS https://cv.upt.ro/course/view.php?id=1661

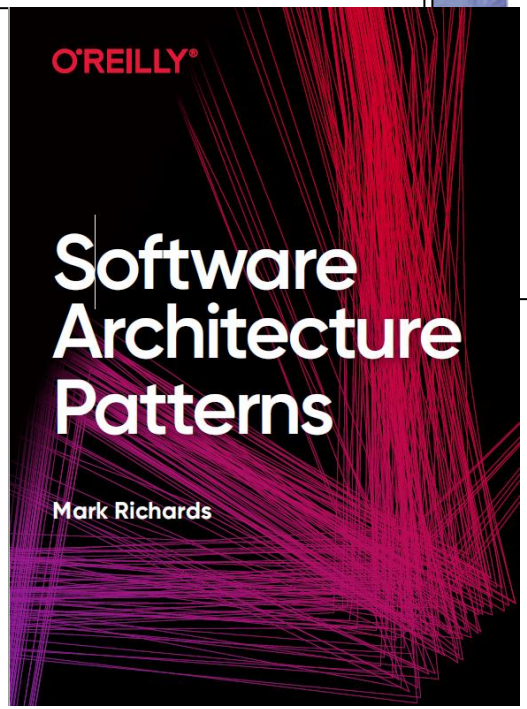
- http://staff.cs.upt.ro/~ioana/arhit-engl/
  - Lectures slides and schedule
  - Lab assignments
  - Bibliographic pointers and additional resources

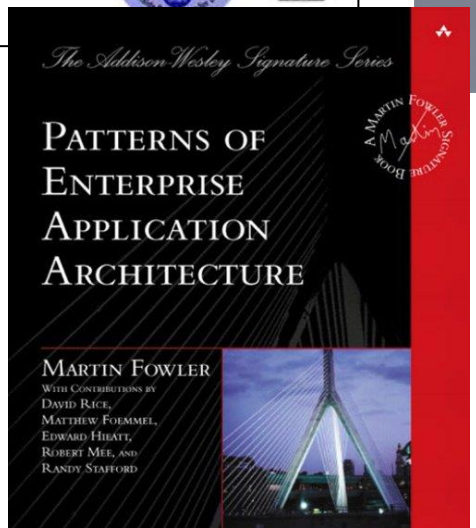
Resources

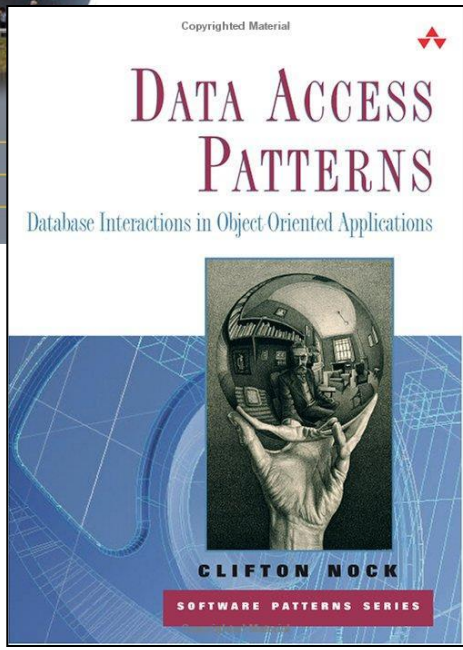

O'REILLY
Fundamentals of
Software
Architecture
An Engineering Approach
Mark Richards & Neal Ford


WILEY SERIES IN
SOFTWARE DESIGN PATTERNS
PATTERN-ORIENTED
SOFTWARE
ARCHITECTURE
A System of Patterns
Volume 1
Frank Buschmann
Regine Meunier
Hans Rohnert
Peter Sommerlad
Michael Stal
WILEY


SEI SERIES IN SOFTWARE ENGINEERING
Software
Architecture
in Practice
THIRD EDITION
Len Bass
Paul Clements
Rick Kazman


O'REILLY
Software
Architecture
Patterns
Mark Richards


The Addison Wesley Signature Series
PATTERNS OF
ENTERPRISE
APPLICATION
ARCHITECTURE
MARTIN FOWLER
WITH CONTRIBUTIONS BY
DAVID RICE,
MATTHEW FOEMMEL,
EDWARD HIEATT,
ROBERT MEE, AND
RANDY STAFFORD


Copyrighted Material
DATA ACCESS
PATTERNS
Database Interactions in Object Oriented Applications
CLIFTON NOCK
SOFTWARE PATTERNS SERIES