AALBORG UNIVERSITY

Design and XP

Peter Dolog
dolog [at] cs [dot] aau [dot] dk
5.2.47
Information Systems
February 29, 2008

Goal

Design

Tutorial on Requirements Eng. and SCRUM reflections (D401b, s601b)

XP

What you should learn

To discuss and apply design concepts

Able to apply XP

Goal

Design

- Basic Concepts and Prinnciples
- Abstractions
- Properties
- Case Tools

Tutorial on Requirements Eng. and SCRUM reflections (D401b, s601b)

XP

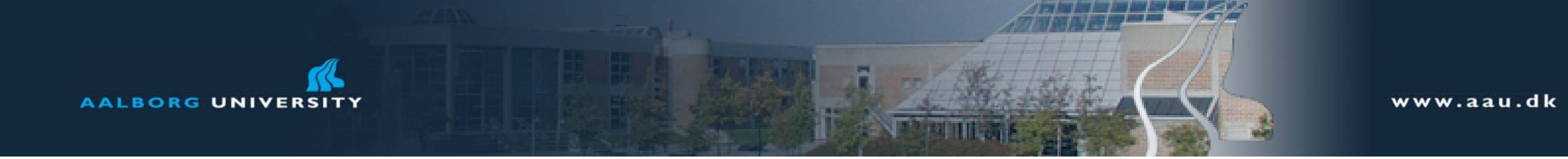
Software Design

Wikipedia:

Software design is a process of problem-solving and planning for a software solution. After the purpose and specifications of software is determined, software developers will design or employ designers to develop a plan for a solution. It includes low-level component and algorithm implementation issues as well as the architectural view.

Computer.org

Design -- (1) The process of defining the architecture, components, interfaces, and other characteristics of a system or component. (2) The result of the process in (1). [ANSI/IEEE Std 610.12 1990]



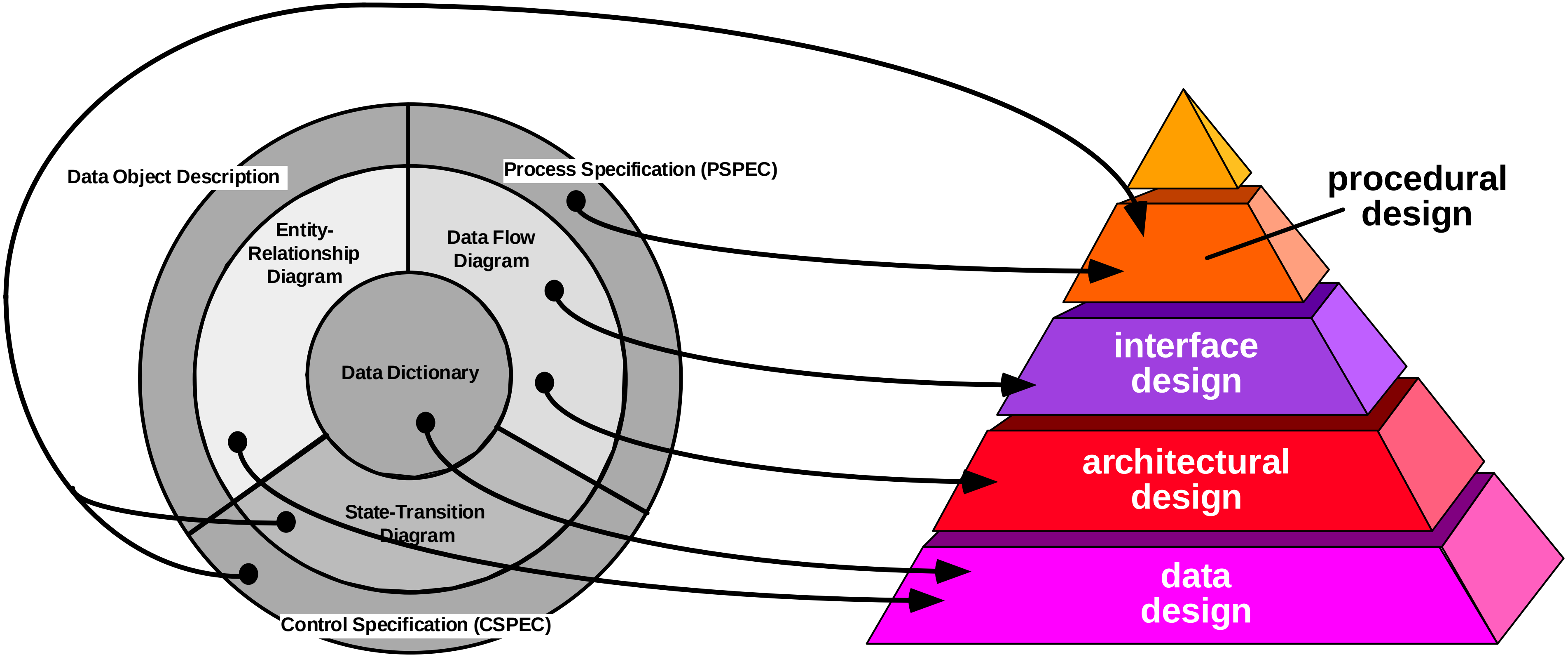
Key Design Principles

Abstraction (domain, solution, procedural, data, control)
Refinement
Modularity/Information Hiding

Key tasks [Unified Process]:

- Understand the technical environment (PL, UI, DB, Concurr.,…)
- Individual modules and subsystems;
- Use for smaller more manageable pieces of work;
- Capture major interfaces between subsystems;
- Use as much as possible a common notation for your design within a team;
- Decide for an appropriate design abstraction that the implementation is more or less straightforward refinement of the design without significant change of structure.

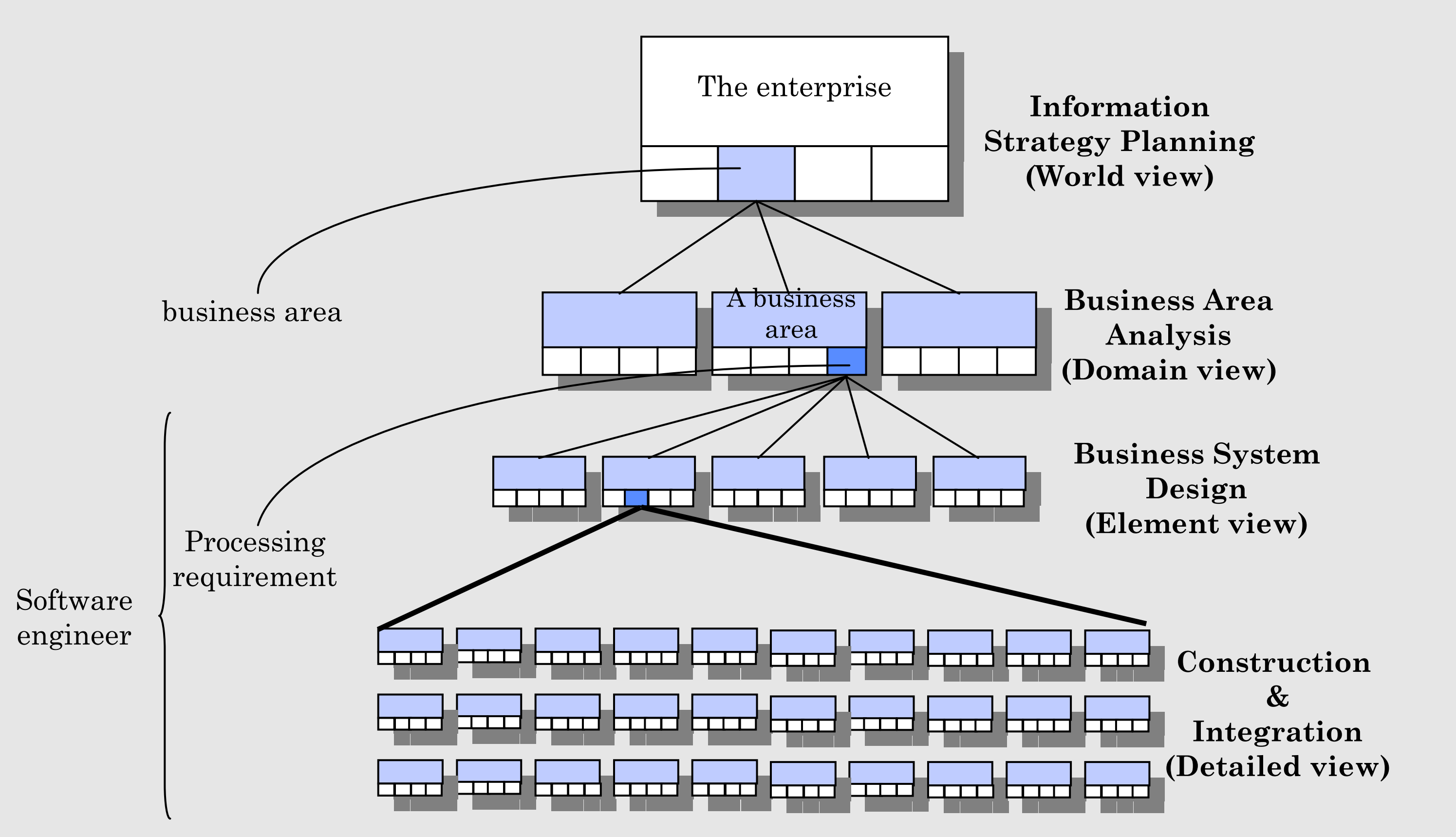
Analysis to Design



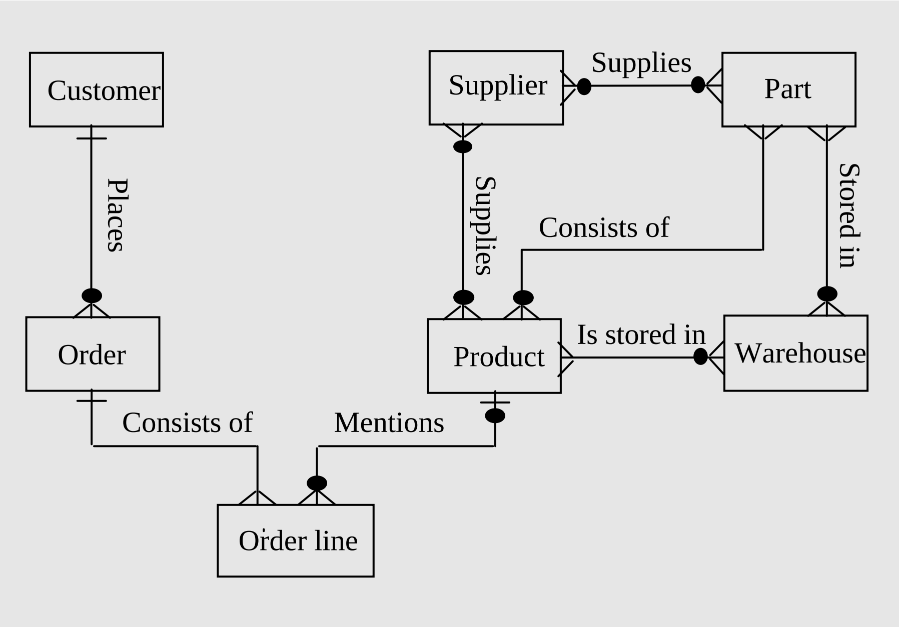
THE ANALYSIS MODEL

THE DESIGN MODEL

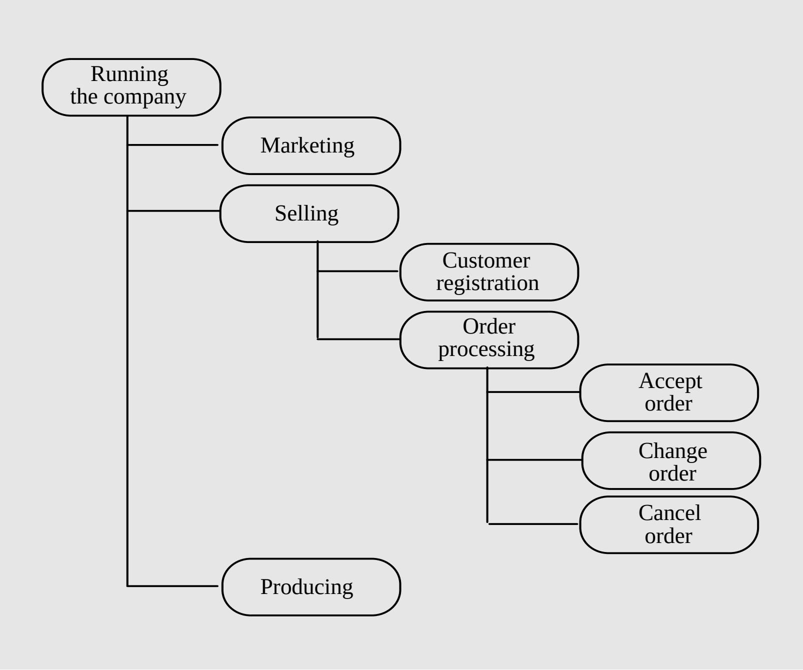
BPE Hierarchy


The enterprise
Information Strategy Planning (World view)
business area
A business area
Business Area Analysis (Domain view)
Business System Design (Element view)
Processing requirement
Software engineer
Construction & Integration (Detailed view)

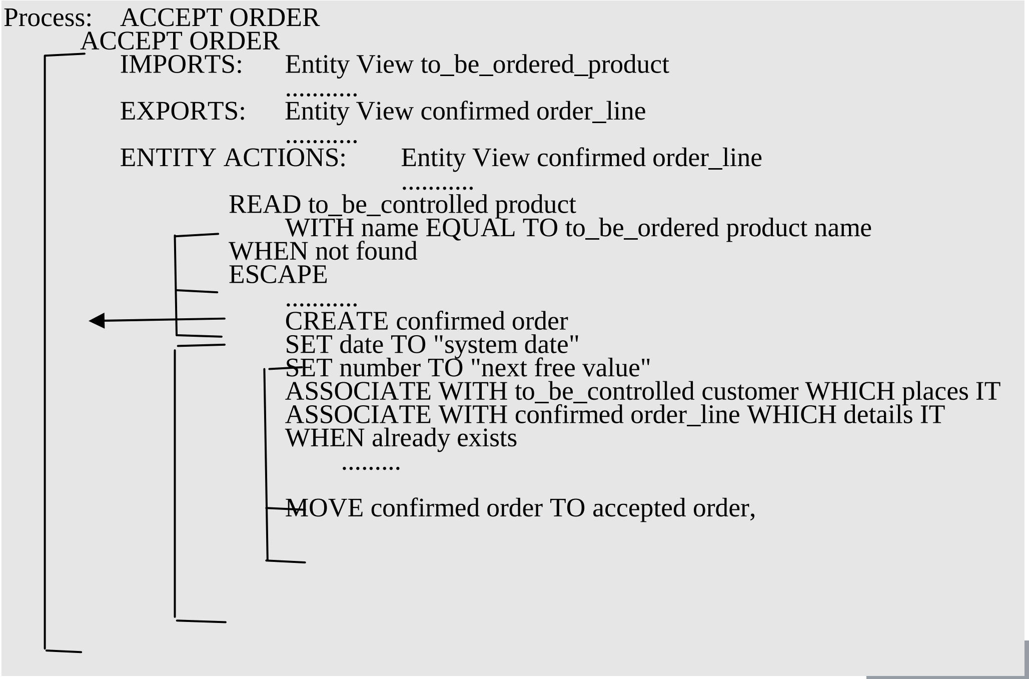
IEF - Analysis: ERD



IEF - Analysis: Process Hierarchy



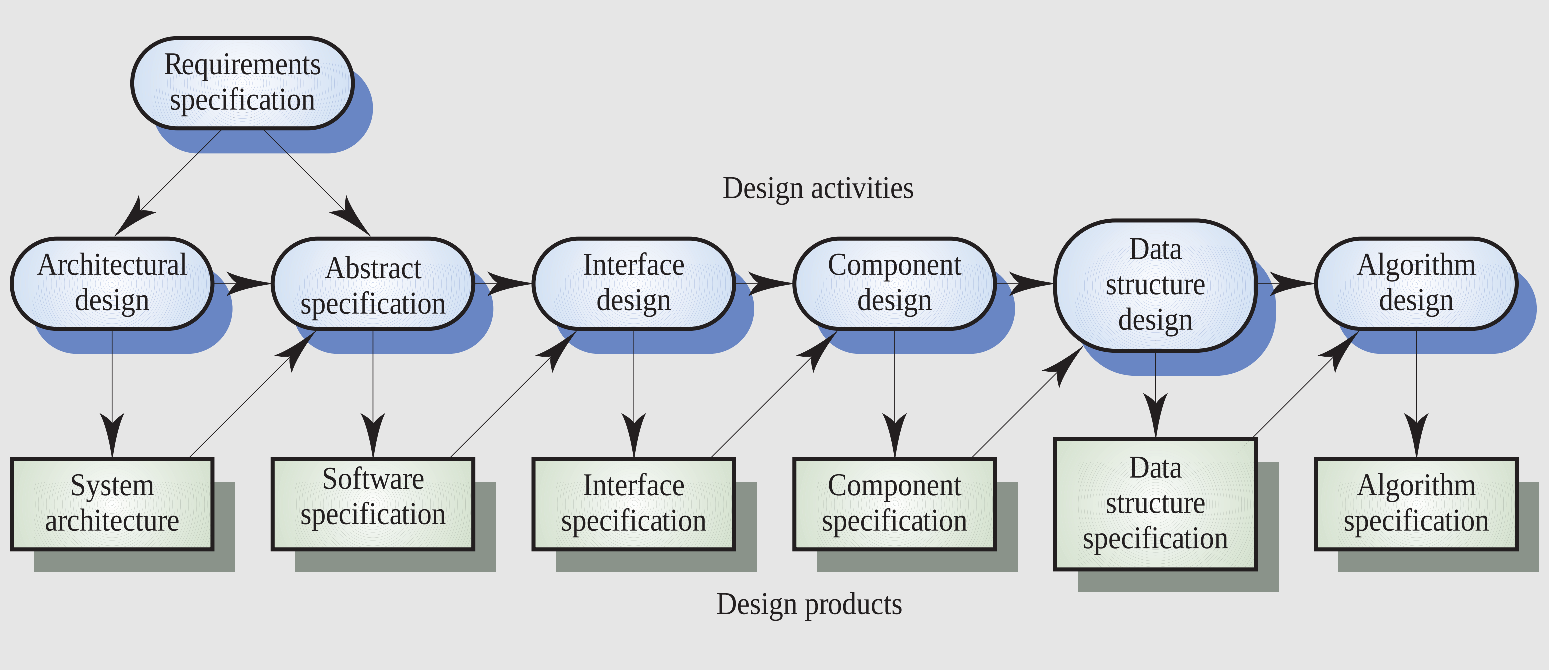
IEF - Analysis: Process Handling



```
Process:    ACCEPT ORDER
      ACCEPT ORDER
         IMPORTS:      Entity View to_be_ordered_product
                       ...........
         EXPORTS:      Entity View confirmed order_line
                       ...........
         ENTITY ACTIONS:       Entity View confirmed order_line
                               ...........
                 READ to_be_controlled product
                       WITH name EQUAL TO to_be_ordered product name
                 WHEN not found
                 ESCAPE
                       ...........
                       CREATE confirmed order
                       SET date TO "system date"
                       SET number TO "next free value"
                       ASSOCIATE WITH to_be_controlled customer WHICH places IT
                       ASSOCIATE WITH confirmed order_line WHICH details IT
                       WHEN already exists
                             .........

                       MOVE confirmed order TO accepted order,
```

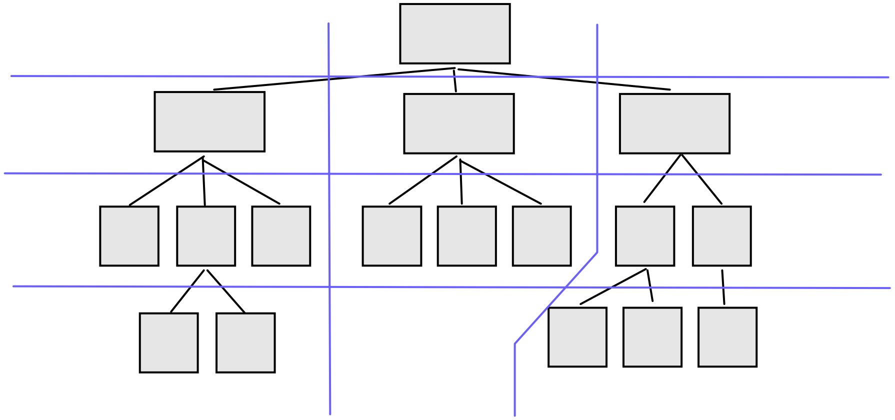
The Software Design Process



Source: Ian Sommerville, Software Engineering

Partitioning the Architecture

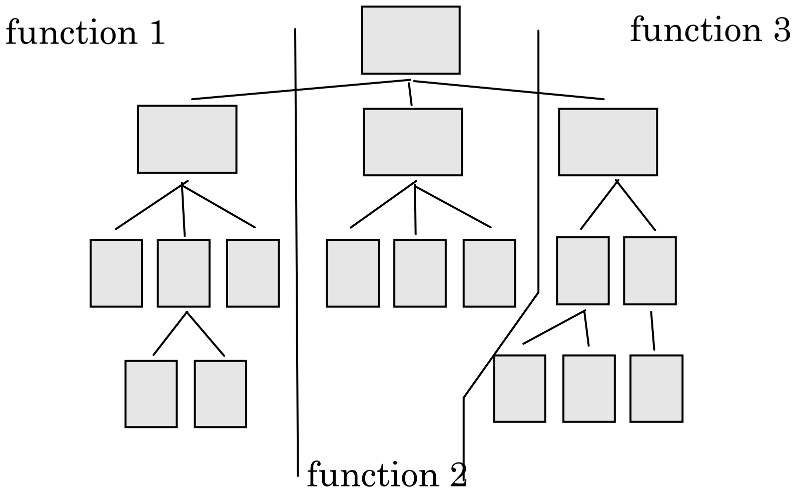
"horizontal" and "vertical" partitioning are required



Horizontal Partitioning

define separate branches of the module hierarchy for each major function

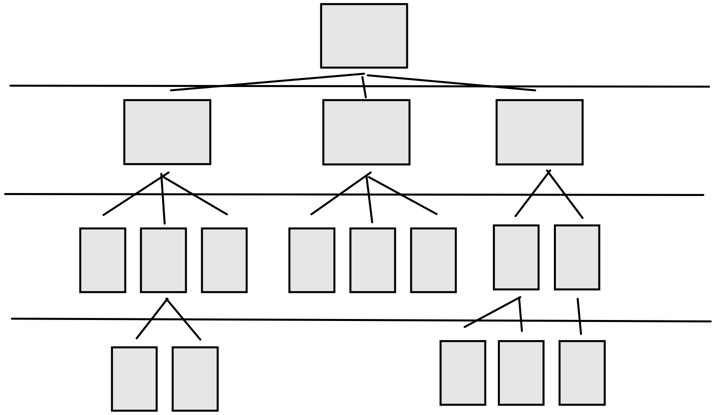
use control modules to coordinate communication between functions



Vertical Partitioning - Factoring

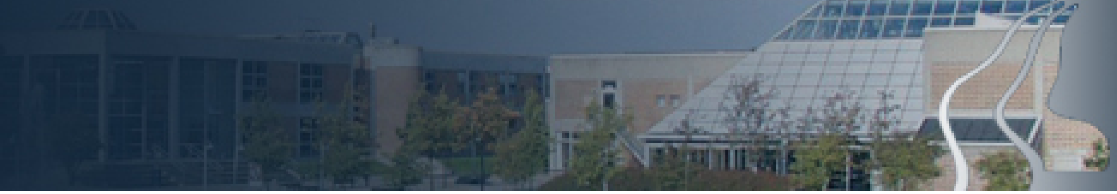
design so that decision making and work are stratified

decision making modules should reside at the top of the architecture



decision-makers

workers

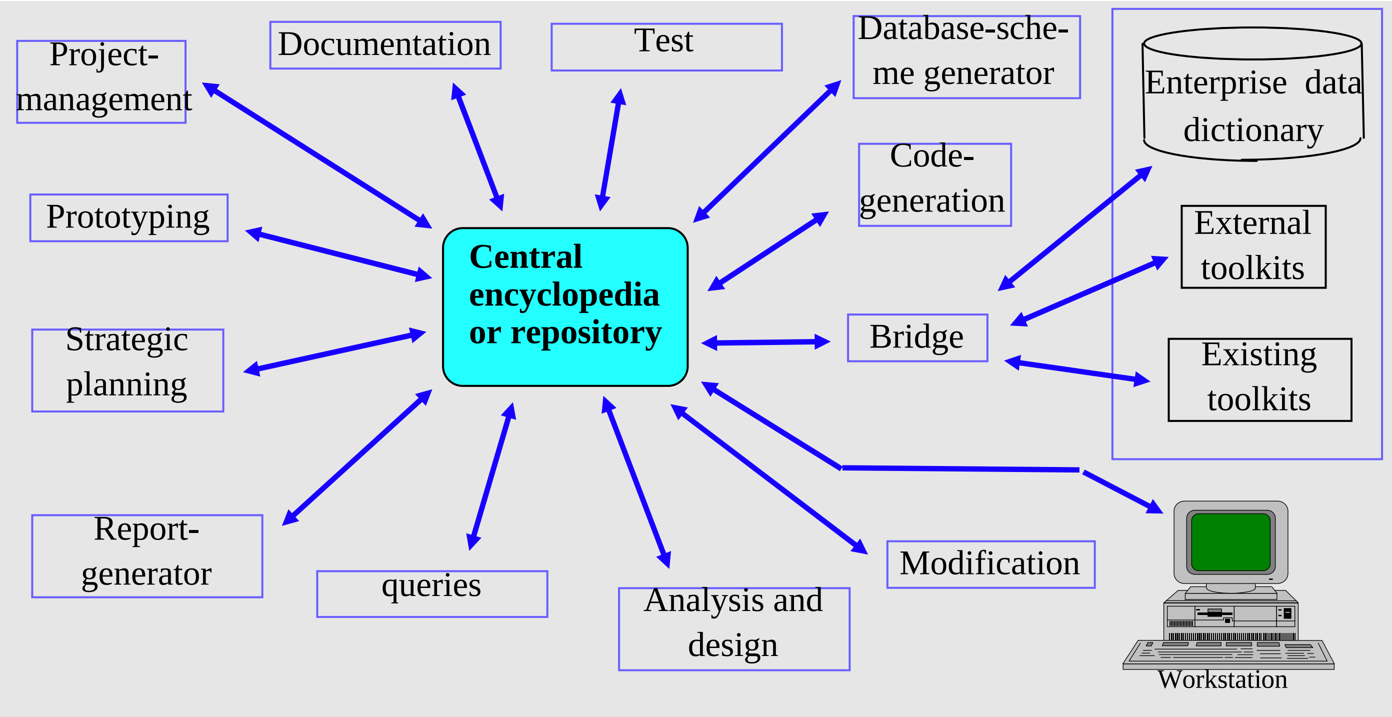


Properties

Information Hiding vs.

- Functional Independence (Each module addresses single-minded function)
- Cohesion (a module performs a single task requiring little interaction with modules on the other part of program) – high
- Coupling (measure of interconnection among modules) – low

Integrated Case — I-CASE



Some adopotion results (Butler 2000)

SE Process si strongly influenced by using CASE (Gyorkos and Rozman 92)

CASE tools increase development productivity (Banker and Kauffman 91)

Quality enhancements, effectiveness of indivudual developers (Finlay and Mitchell 94)

70% of case tools are not used after 1 year (Kemerer 92)

30% companies who adopt CASE abondon it after 2 years (Isoda et al 95)

…

Goal

Design

- Basic Concepts and Prinnciples
- Abstractions
- Properties
- Case Tools

Tutorial on Requirements Eng. and SCRUM reflections (D401b, s601b)

XP