

Frameworks

1.	Objectives	2
2.	Frameworks.....	3
3.	Classification.....	3
4.	Example: Components for Java (BC4J)	6
5.	Existing Frameworks.....	9
6.	Presistence Frameworks	11
7.	Content Management System Frameworks	11
8.	Cocoon Framework	12
9.	Summary.....	13

1.Objectives

- A framework is not a library
- Components in a library have little, if any, interaction with each other (e.g., string, file, date, list).
- Each component can be independently added to an application.
- Building complex applications from scratch is very expensive.
- Where are we in the software life cycle?

Requirements	Design	Implementation
	Software Architectures		
		Components	
	Software Component Architecture		
DSSA: Domain-Specific Software Architectures			
	Frameworks		
	Design Patterns		
		Which Programming language?	

2. Frameworks

- There are no hard and fast rules as to what constitutes a framework. In general, it should provide the skeleton of an application.
- A *framework* is a partially complete (sub-) system that is intended to be instantiated.
- It defines the architecture for a family of (sub-) systems and provides the basic building blocks to create them.
- It also defines the places where adaptations for specific functionality should be made.
- In an object-oriented environment a framework consists of abstract and concrete classes.

3. Classification

- There are two categories of frameworks:
 - Horizontal
 - Vertical
- Horizontal framework:
 - It is a more general framework and can be applied in several applications.
 - Horizontal frameworks provide the most flexibility.
 - Example: GUI toolkits, Master/detail coordination.
- Vertical framework:
 - It a more specific framework.
 - It can only be applied to a specific application domain.