

PROGRAMMATION ORIENTÉE OBJETS II

420-P33-SU

Model-View-Controller



Model-View-Controller (MVC) is a architectural pattern that separates data (model), user interface (view), and logical controller (controller).

Model-View-Controller

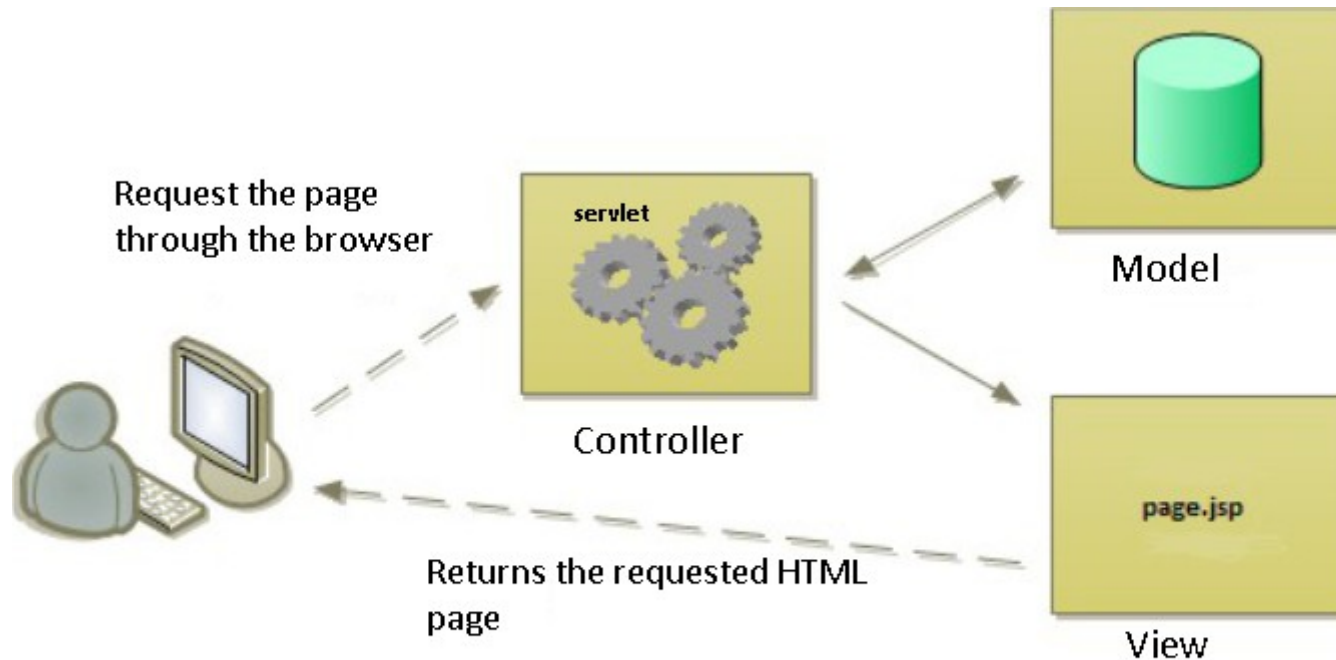
- ◆ Object oriented design pattern
- ◆ Renamed by Trygve Reenskaug in 1979
- ◆ Formalized by Steve Burbeck
- ◆ Relies on a separation of concepts (layers)
- ◆ MVC can be implemented and interpreted in different ways

Design Pattern



- A design pattern is a concept that :
 - Solves recurring problems which occur in the object oriented paradigm
 - Describes best in practice solutions for architecture and software design issues
- It is generally encouraged that developers employ design patterns even if they may sometimes constrain your code.

Model-View-Controller



The client's request arrives at the controller, which then returns the view to the client.

Model-View-Controller

- ♦ Model : represents the application's data. It also defines interactions with databases and how the data is processed.
- ♦ View : represents the user interface, and the user would interact with it. The view does not deal with any processing, but rather is only concerned with displaying data provided by the model.
- ♦ Controller : represented the interface between the model and view. It interprets the client request and returns the appropriate view. The controller is also responsible for synchronization between model and view.

Advantages of MVC Architecture

- ♦ Suited for graphical applications.
- ♦ Suited for applications where there are multiple interface.
- ♦ Modularity :
 - Reduces complexity during the design phase
 - Facilitates maintenance and modification

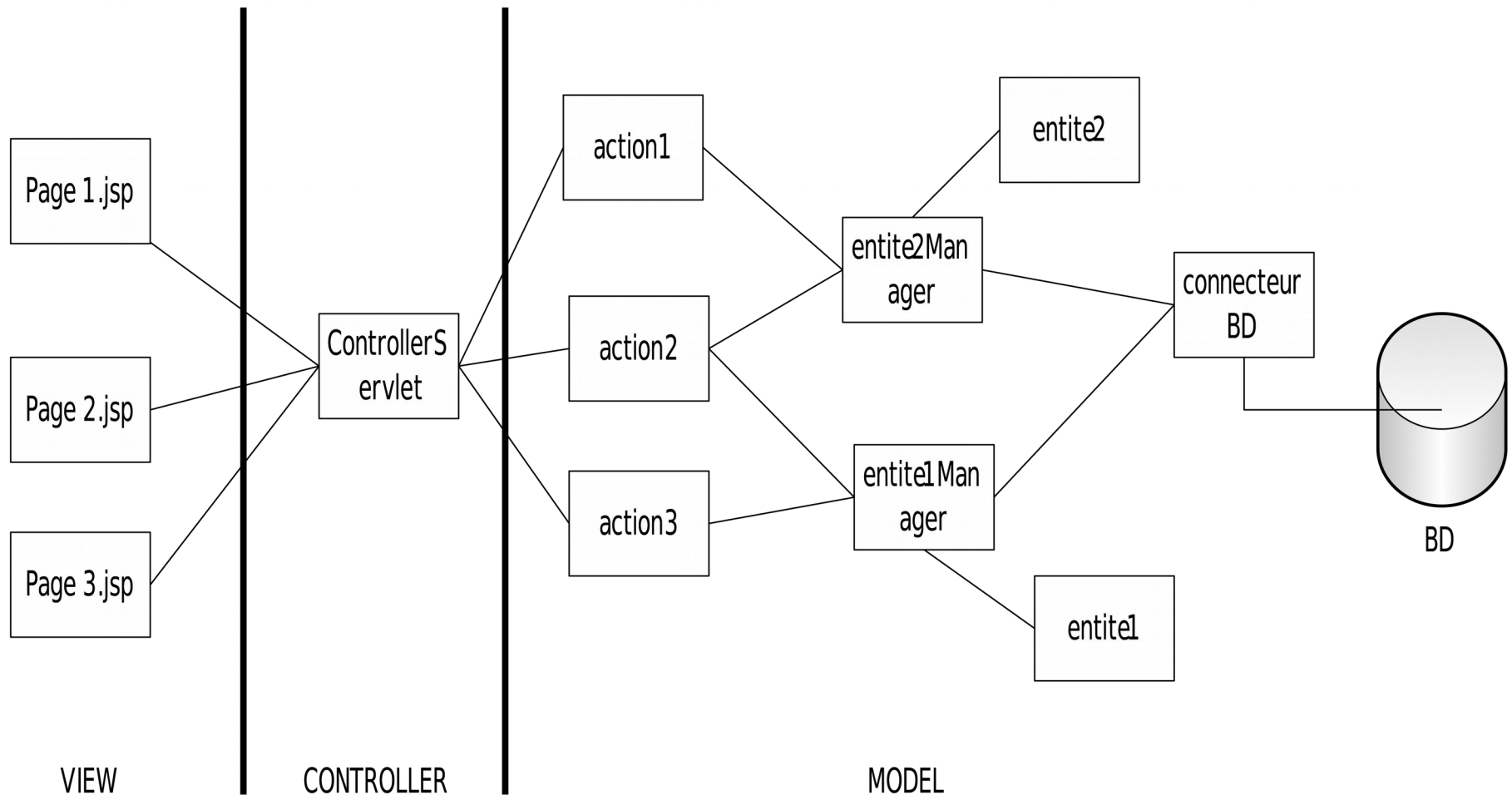
Disadvantages of MVC Architecture

- ♦ Modularity:
 - / Increases complexity during the implementation phase.
 - / Potential for fragmentation of developers

Recurring issues related to MVC and web

- ♦ Should the controller layer taking into account any particular processing?

MVC Architecture Elaborated



Example

Parts of MVC Architecture:

- ♦ Service
- ♦ Entity
- ♦ Manager
- ♦ Action
- ♦ Controller
- ♦ The view that requests the action and the view that displays the result.