Instituto Politécnico Nacional
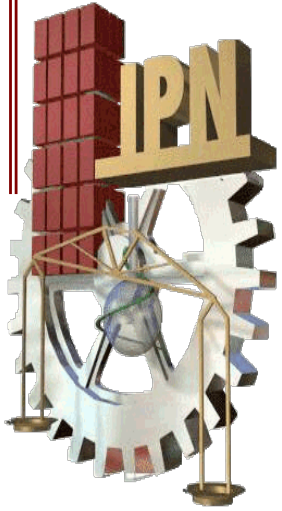
Escuela Superior de Computo

Patrones de diseño

M. en C. José Asunción Enríquez Zárate

asuncionez@gmail.com

Instituto Politécnico Nacional

Escuela Superior de Computo

DAO

M. en C. José Asunción Enríquez Zárate

asuncionez@gmail.com

Data Access Object(DAO): Problem

- The DAO pattern is a prominent concept in many data-driven application frameworks.

    - Premise: bundle data to expose it to clients.

- It promotes separation of concerns.

    - Business logic and data logic change at differente rates.

- Used to address the limits of some container-managed persistence technology (CMP)

3

Data Access Object(DAO): Forces

➤ The DAO Pattern addresses current conditions:

  ➤ You need a simple way to test data access.

  ➤ You need to decouple proprietary data sources form business logic.

  ➤ You want to clear line of demarcation between service-oriented applications and tehe data that they use.

  ➤ You want to separate complex query code from simple CRUD implementations
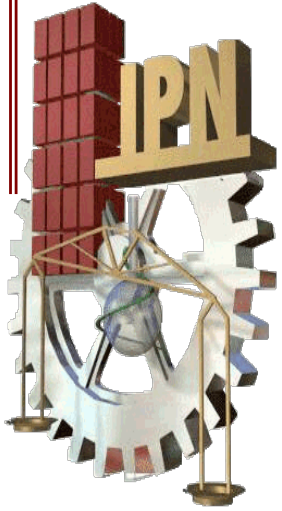
Data Access Object(DAO): Forces
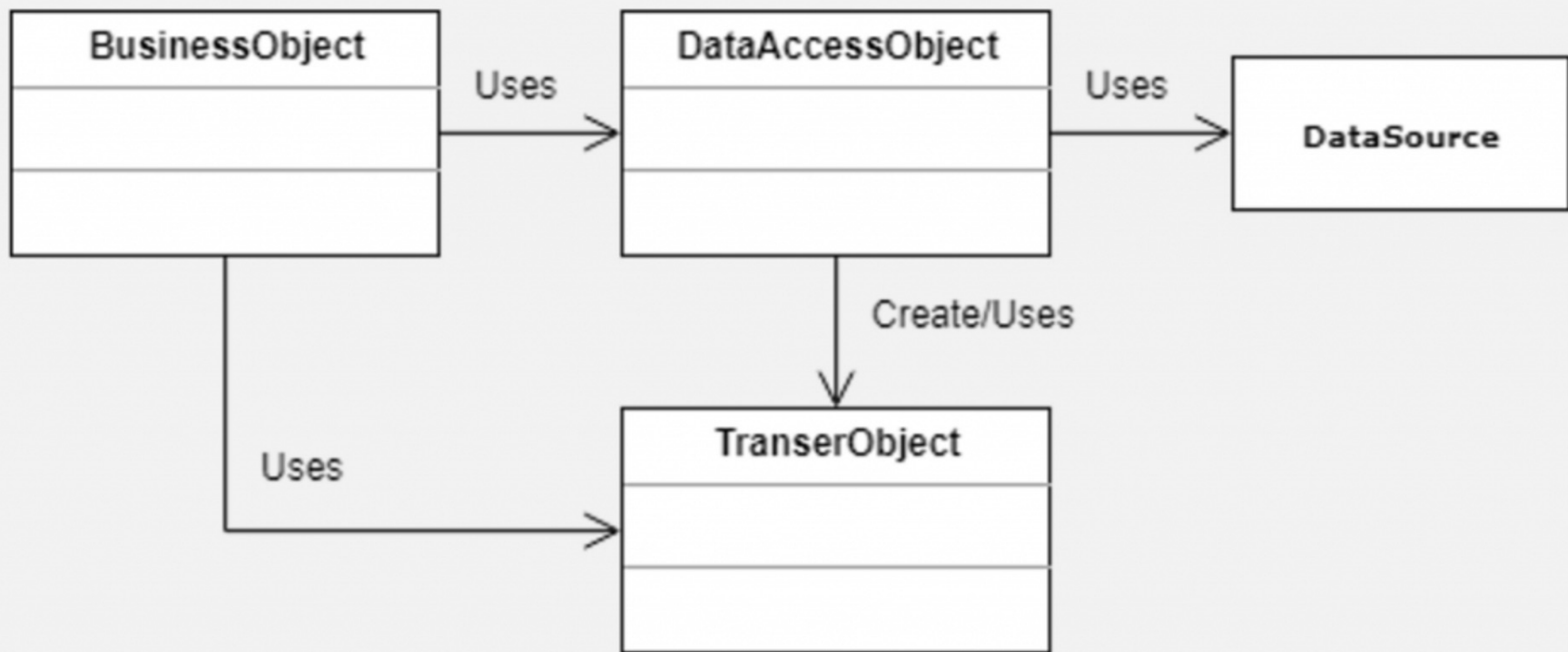
DAO: Consequences

- Adss an interface and its implementation to the code

- Must be maintained in parallel with entity logic

- Puts reusable data access logic in a central place

- Case for reuse may prove scarce or unrealistic

- Is the least awkward approach to propietary data APIS

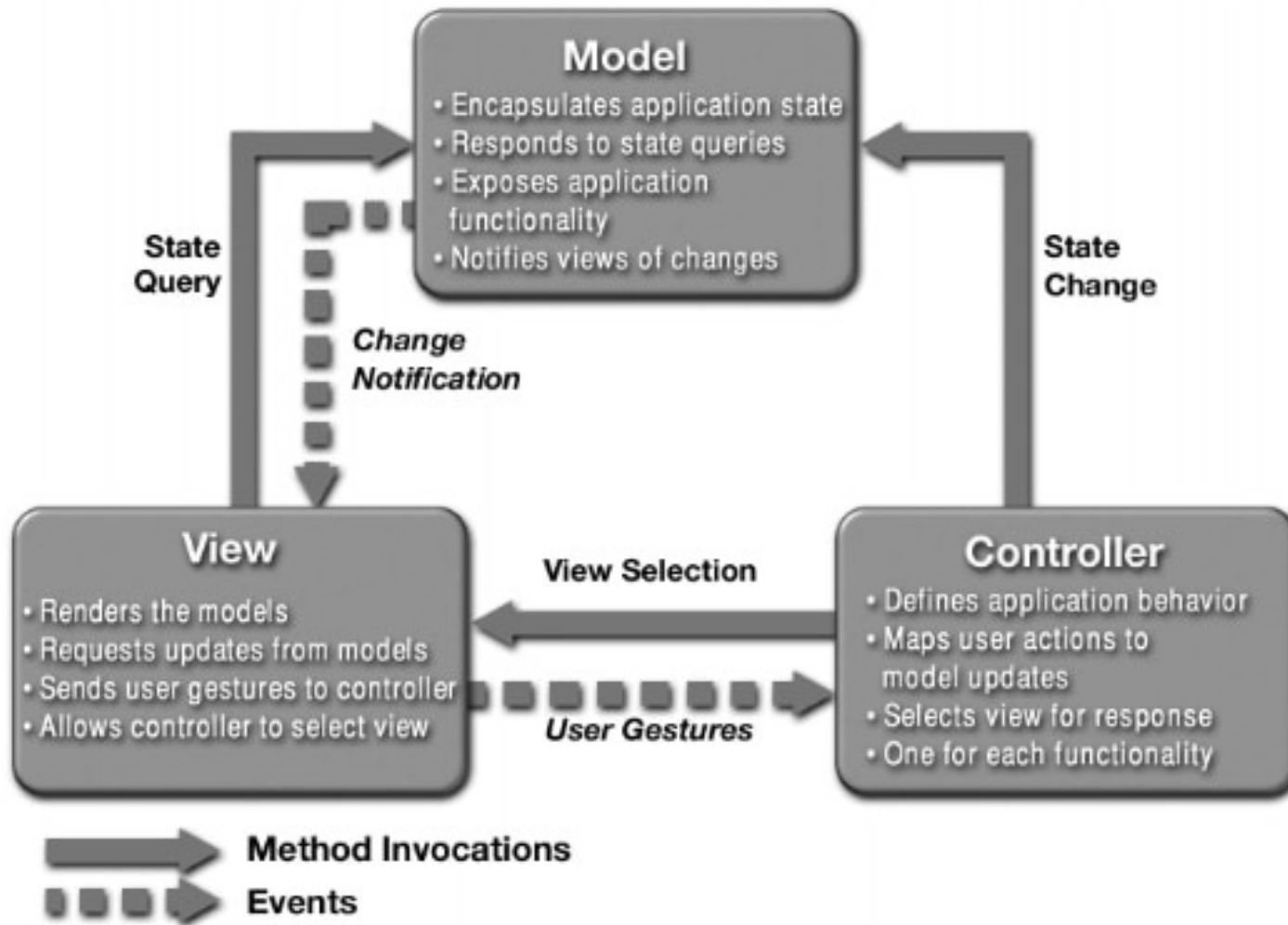# Instituto Politécnico Nacional

Escuela Superior de Computo

MVC

M. en C. José Asunción Enríquez Zárate

asuncionez@gmail.com

MVC Architecture

Model
- Encapsulates application state
- Responds to state queries
- Exposes application functionality
- Notifies views of changes

State Query

Change Notification

State Change

View
- Renders the models
- Requests updates from models
- Sends user gestures to controller
- Allows controller to select view

View Selection

User Gestures

Controller
- Defines application behavior
- Maps user actions to model updates
- Selects view for response
- One for each functionality

Method Invocations

Events

MVC Architecture

➤ MVC is a widely known development metaphor.

  ➤ It originated with the SmallTalk programming language.

  ➤ It is a well-know approach for UI-centric frameworks

    ➤ Swing

    ➤ Struts2

    ➤ Spring

  ➤ It is useful as a general example for separating concerns.

    ➤ Delegate command request

    ➤ Uses event notificaction for flexible communication

Patrones de Diseño   15/12/21

Benefits of the MVC Model

➢ In the M-V-C, the concept separates the data (Model), its presentation (View), and its commands (Controller) into interdependient concerns.

➢ The MVC model promotes the underlying assumption that several developers are participating in a project that uses it.

➢ Separation of concerns, as a design concept , impose a variety of interfaces that allow the concerns to connect with each other event though the internal development remains independent.

Applying the MVC Pattern: Problem Forces

➢ User interfaces and bussines logic change at diferent rates.

➢ Some clients will support multiple interfaces:

  ➢ You can display the same information in different views.

  ➢ Some views update to reflect the changes in the model data.

  ➢ The model data must reflect the changes initiated from view immediately.

  ➢ The user interface is loosely coupled to the data and logic.

# Applying the MVC Pattern: Problem Forces

➢ The MVC pattern divides the system into three areas:

> ➢ Controller: Accepts user requests, invokes processing on the Model components, and determines which View component should be displayed.

> ➢ View: Display a GUI with the Model component data. It may also pass GUI-driven events to the Controller components.

> ➢ Model: Contains business data, processing and rules. It is unaware of any user interface type.

Patrones de Diseño 15/12/21
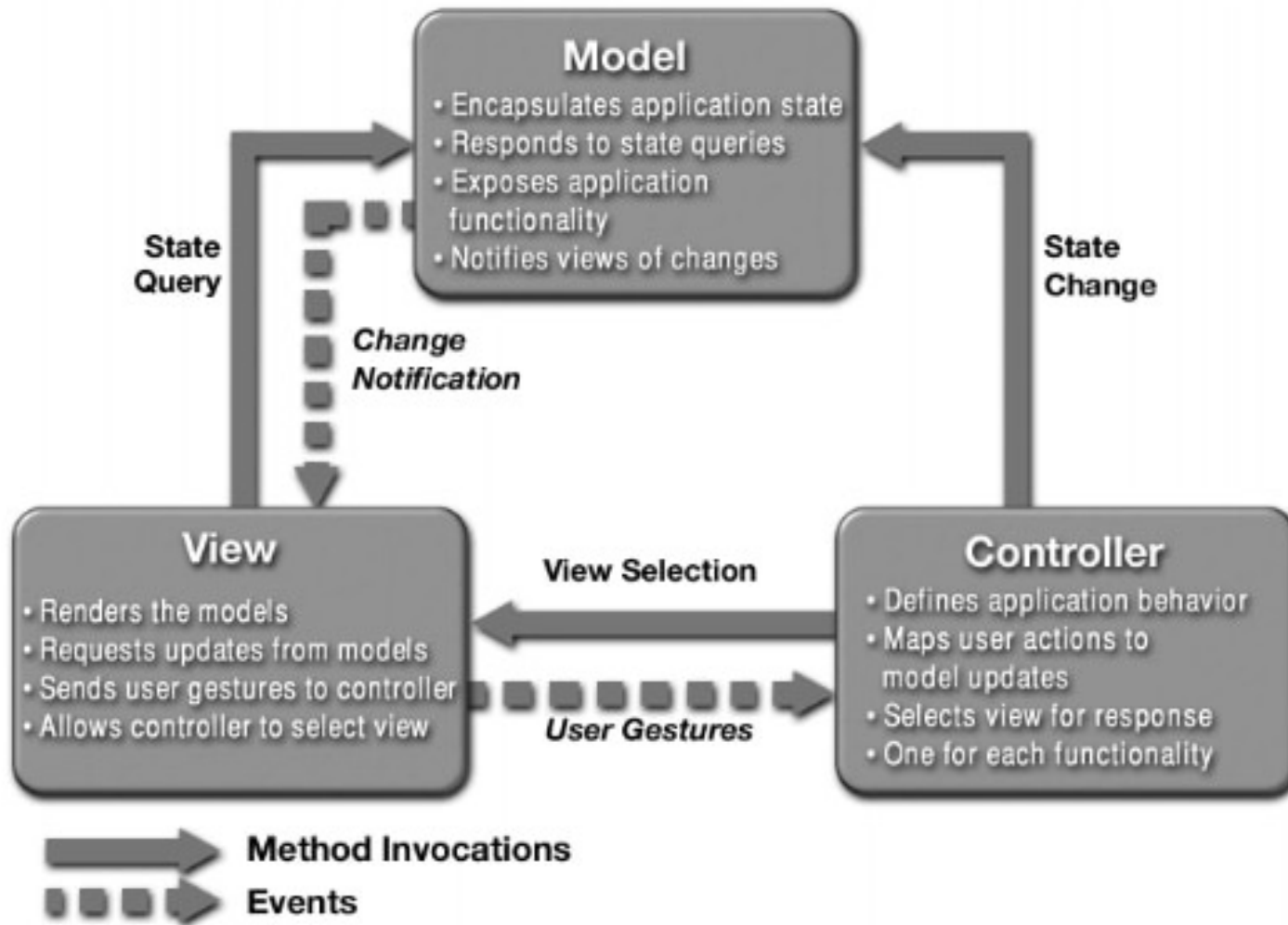
Applying the MVC Pattenr: Consequences

- Advantages:

  - Multiple views offer simple or detailed perspectives.

  - It is easier to modify the Model, View and Controller components separately.

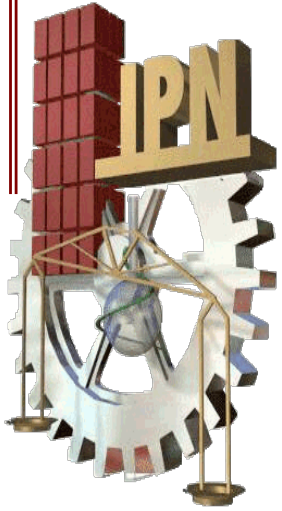 - Some developers specialize on the UI, whereas other specialize on the model.

- Disadvantages;

  - A very simple implementation of subsystems may greatly increase the communication overhead

Instituto Politécnico Nacional

Escuela Superior de Computo

Patrones de diseño

M. en C. José Asunción Enríquez Zárate

asuncionez@gmail.com