WP6: database engine

From XML to Database working with Castor

Cyprus Meeting, 23rd May 2007

Valentina Baglioni

WP6 overview

- Purposes of WP6:
  - □ To create BIRO database
  - □ To store XML documents produced by each partecipating region automatically

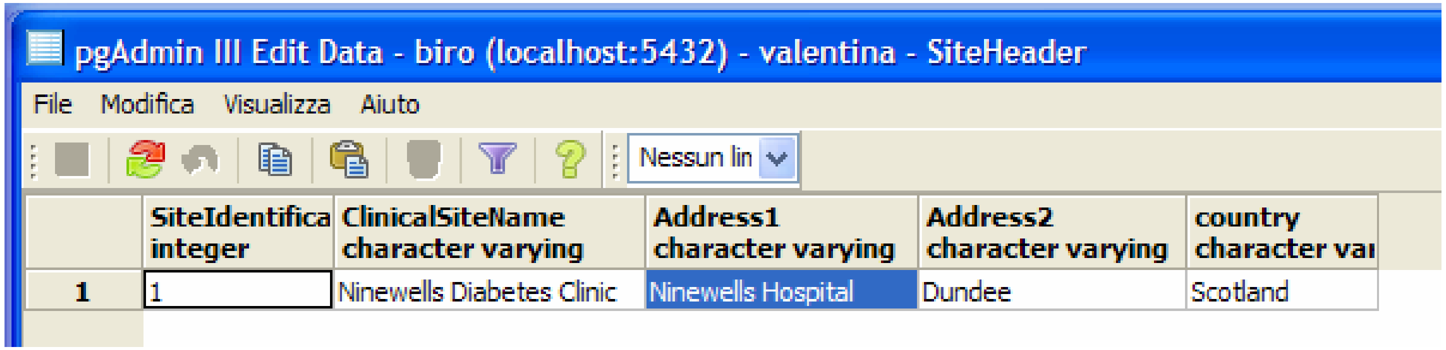
- Input and output of WP6:
  - □ WP6 uses the results of WP3 and WP4: BIRO XML Schema, common dataset and data dictionary
  - □ WP6 will feed the information required by the WP8 (statistical engine)

The problem

- To store information from an XML document into a database in a simple and effective way.

```
  <?xml version="1.0" encoding="UTF-8" ?>
-<ECDataExport xmlns:xsi="..."
   xsi:noNamespaceSchemaLocation="...">
- <SiteHeader>
  <SiteIdentification>1</SiteIdentification>
  <ClinicalSiteName>Ninewells Diabetes Clinic</ClinicalSiteName>
  <Address1>Ninewells Hospital</Address1>
  <Address2>Dundee</Address2>
  <Country>Scotland</Country>
  ...
</SiteHeader>
...
```

XML persistence

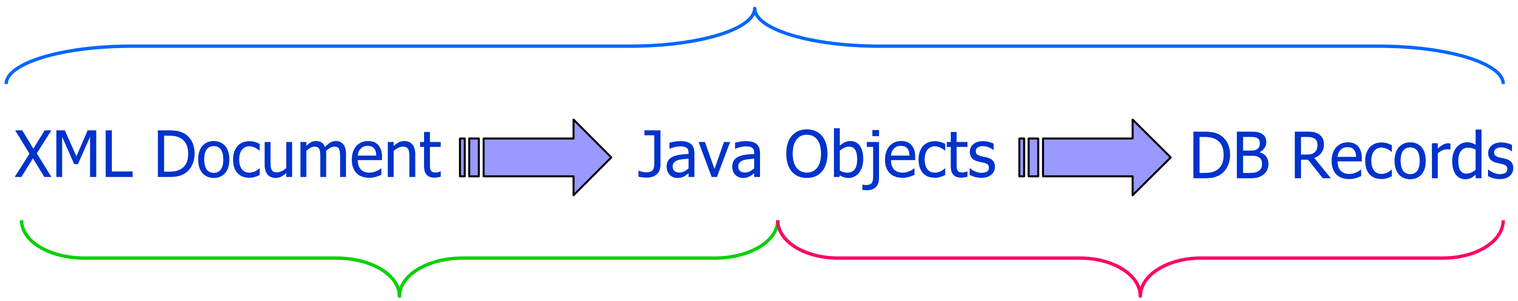

pgAdmin III Edit Data - biro (localhost:5432) - valentina - SiteHeader
File
Modifica
Visualizza
Aiuto
Nessun lin

	SiteIdentifica integer	ClinicalSiteName character varying	Address1 character varying	Address2 character varying	country character vai
1	1	Ninewells Diabetes Clinic	Ninewells Hospital	Dundee	Scotland

Solution – main idea

- □ **Castor**
- □ Hibernate

XML Persistence



XML Document → Java Objects → DB Records

Marshalling/Unmarshalling

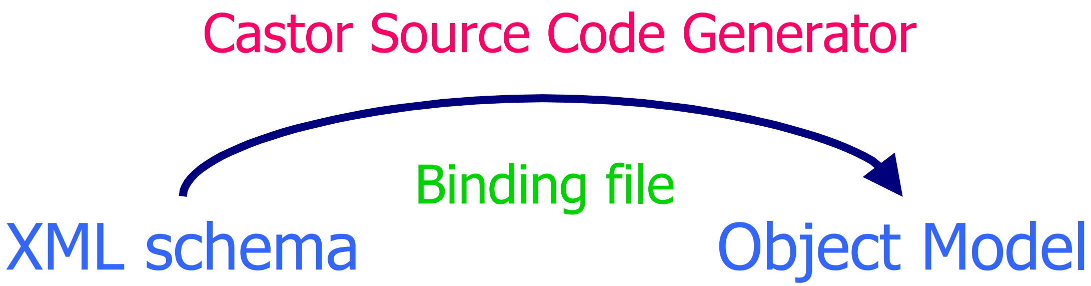
- □ JAXB
- □ XMLBeans
- □ Jibx

Object Persistence

- □ Apache Cayenne
- □ OJB

Working with Castor

- 1st step: XML Databinding



Castor Source Code Generator

Binding file

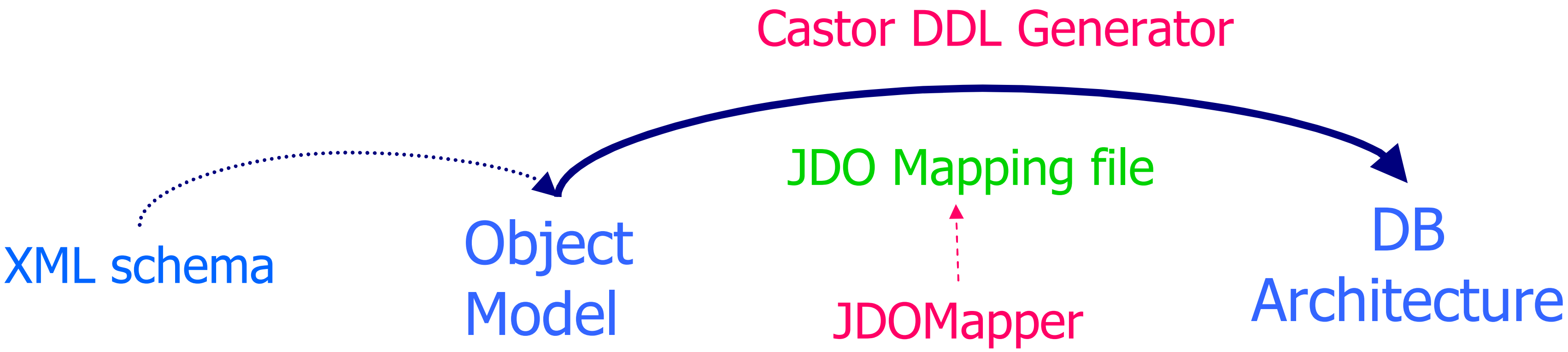
XML schema

Object Model

  - □ A Java class is created for each element in XML schema whose type is a complexType.
  - □ Classes are not created for elements whose type is a simpleType.
  - □ User can modify the construction of the object model through a binding file

Working with Castor

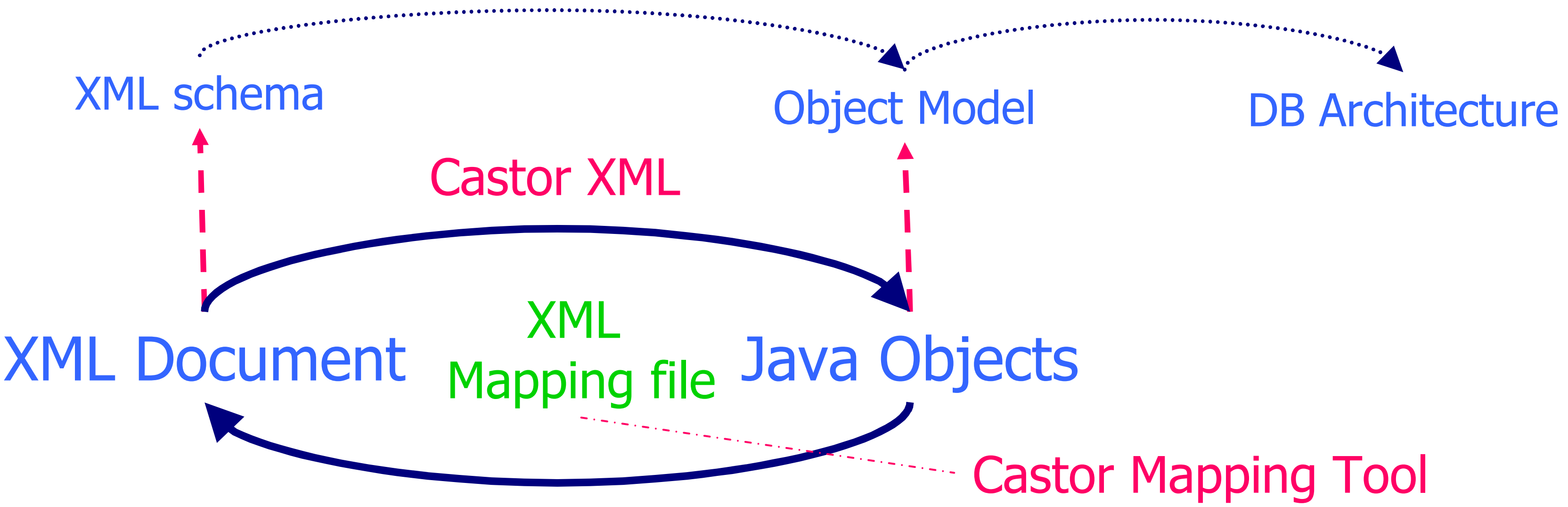
- 2nd step: DDL Generation



- JDO Mapping file specifies the links between classes and fields in object model and tables and columns in database
- JDO (Java Data Object): high-level API which defines standard methods for Objects persistence
- JDO Mapping file is not optional.
- JDOMapper: 3rd party tool which automatically constructs the basic mapping and provides the user with a GUI for fine tuning

Working with Castor

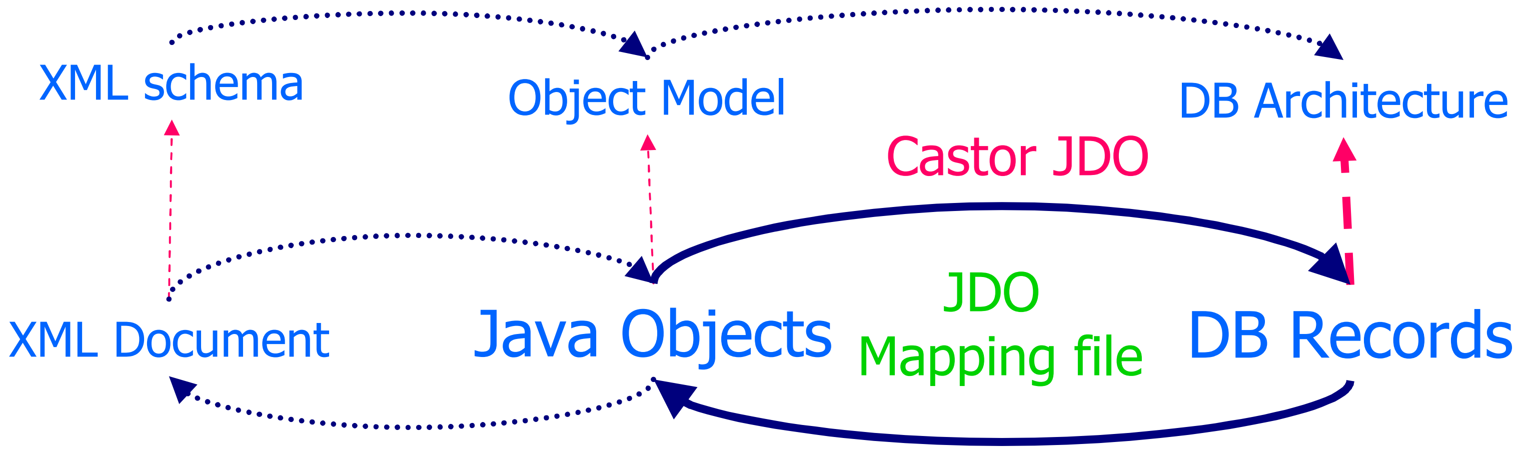
- 3rd step: unmarshalling



- Mapping file may be necessary in order to have more control over Castor XML behaviour
- It is possible to use the mapping file and Castor default behaviour in conjunction

Working with Castor

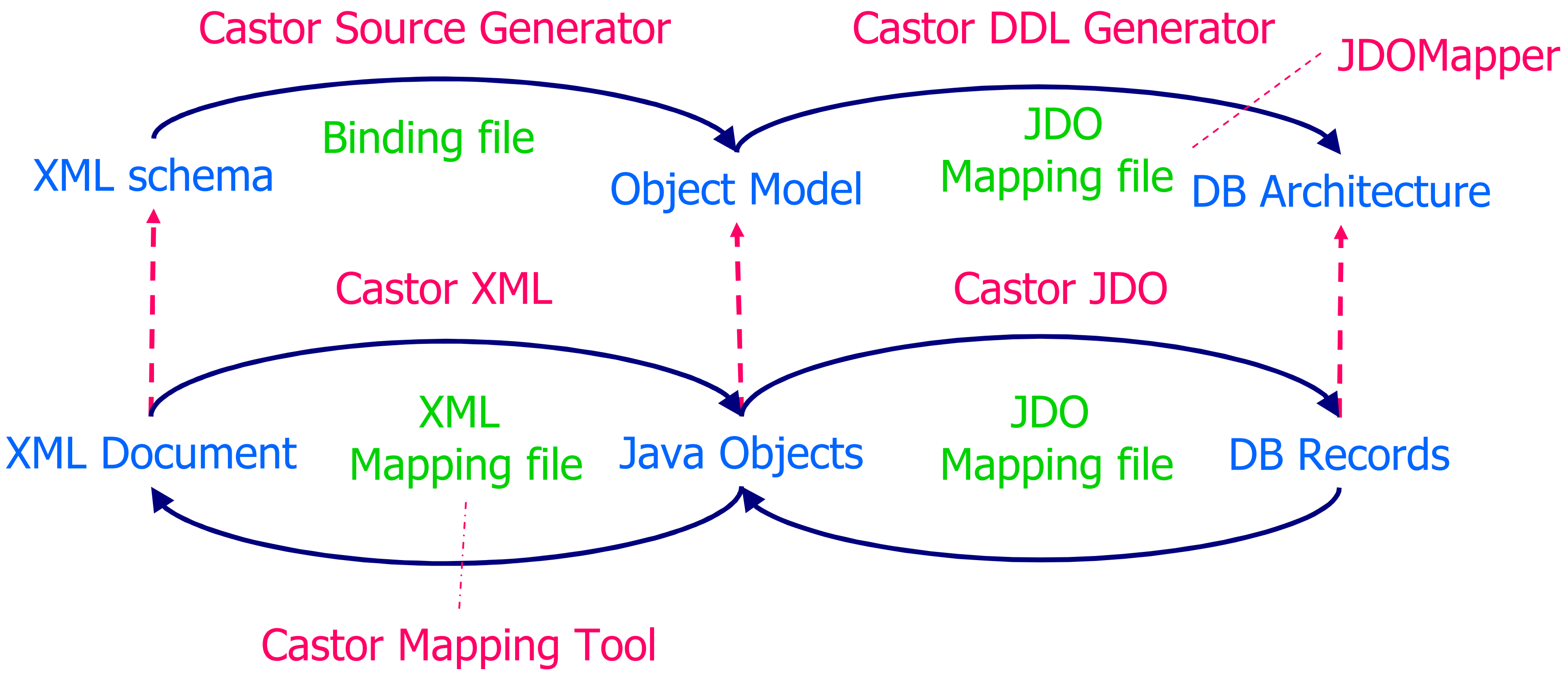
- 4th step: objects persistence



- JDO mapping file is the same used by Castor DDL Generator
- Castor JDO supports one-one and one-many relation types, SQL/Java inheritance, multiple column primary keys and automatic type conversion
- retrieving, updating and deleting the information can be done with OQL queries instead of SQL queries

Working with Castor

- The complete process



Castor Source Generator

Castor DDL Generator

JDOMapper

Binding file

JDO Mapping file

XML schema

Object Model

DB Architecture

Castor XML

Castor JDO

XML Mapping file

JDO Mapping file

XML Document

Java Objects

DB Records

Castor Mapping Tool

Implementation

- We applied the process to “ECDataExtractDefinition.xsd” and “ECClinicalDefinitions.xsd”
- Source Code Generation: the binding file is necessary to avoid name collisions

- ECClinicalDefinitions
  - □ DataItem
    - Definition → DataItemDefinition
      - □ Definition → DataItemDescription
- ECClinicalDefinitions
  - □ OutcomeIndicator
    - Definition → OutcomeIndicatorDefinition
      - □ Definition → OutcomeIndicatorDescription
- ECDataExport
  - □ Data → PatientData
    - Patient
      - □ EpisodeData
        - Data