

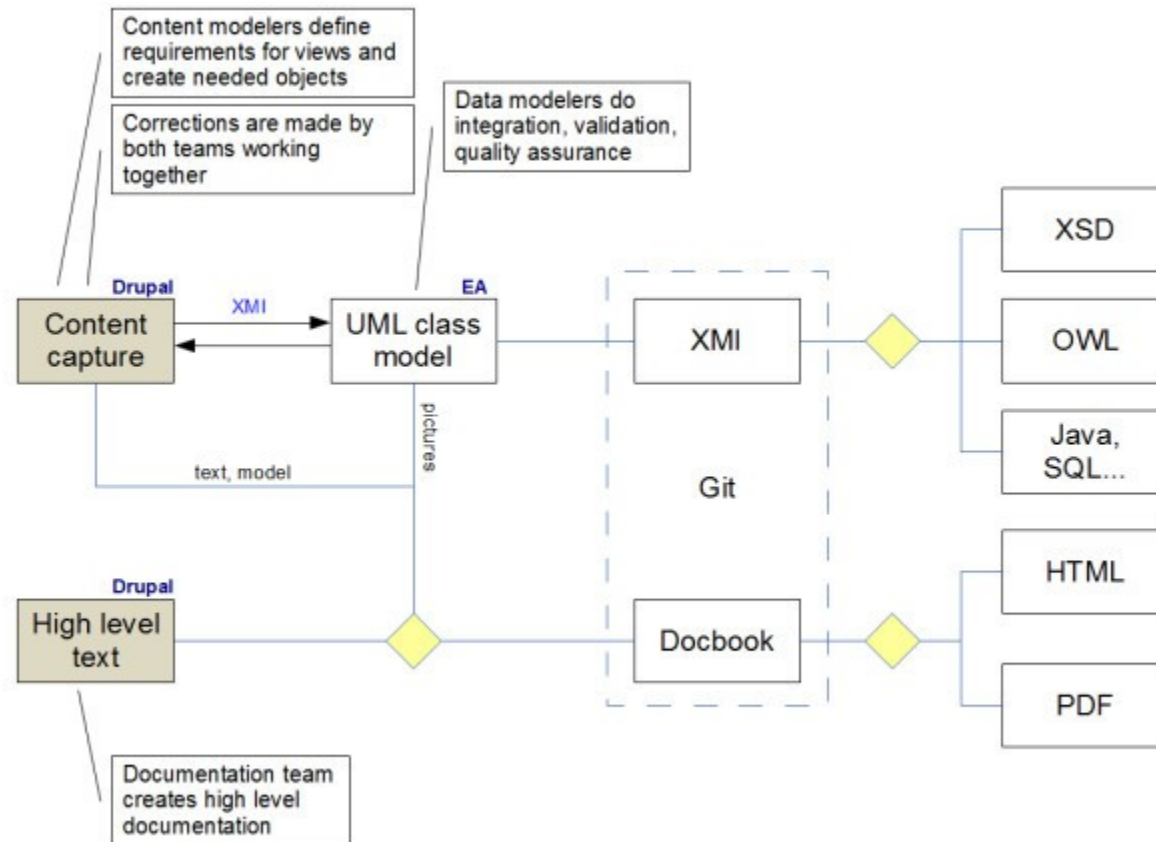
Generating Database and Domain Models from DDI 4

Oliver Hopt, Brigitte Mathiak
EDDI 2014, London

This talk is about

- Short overview of DDI 4 production process
- Generating domain classes for DDI 4
- Lessons learned
- Perspective

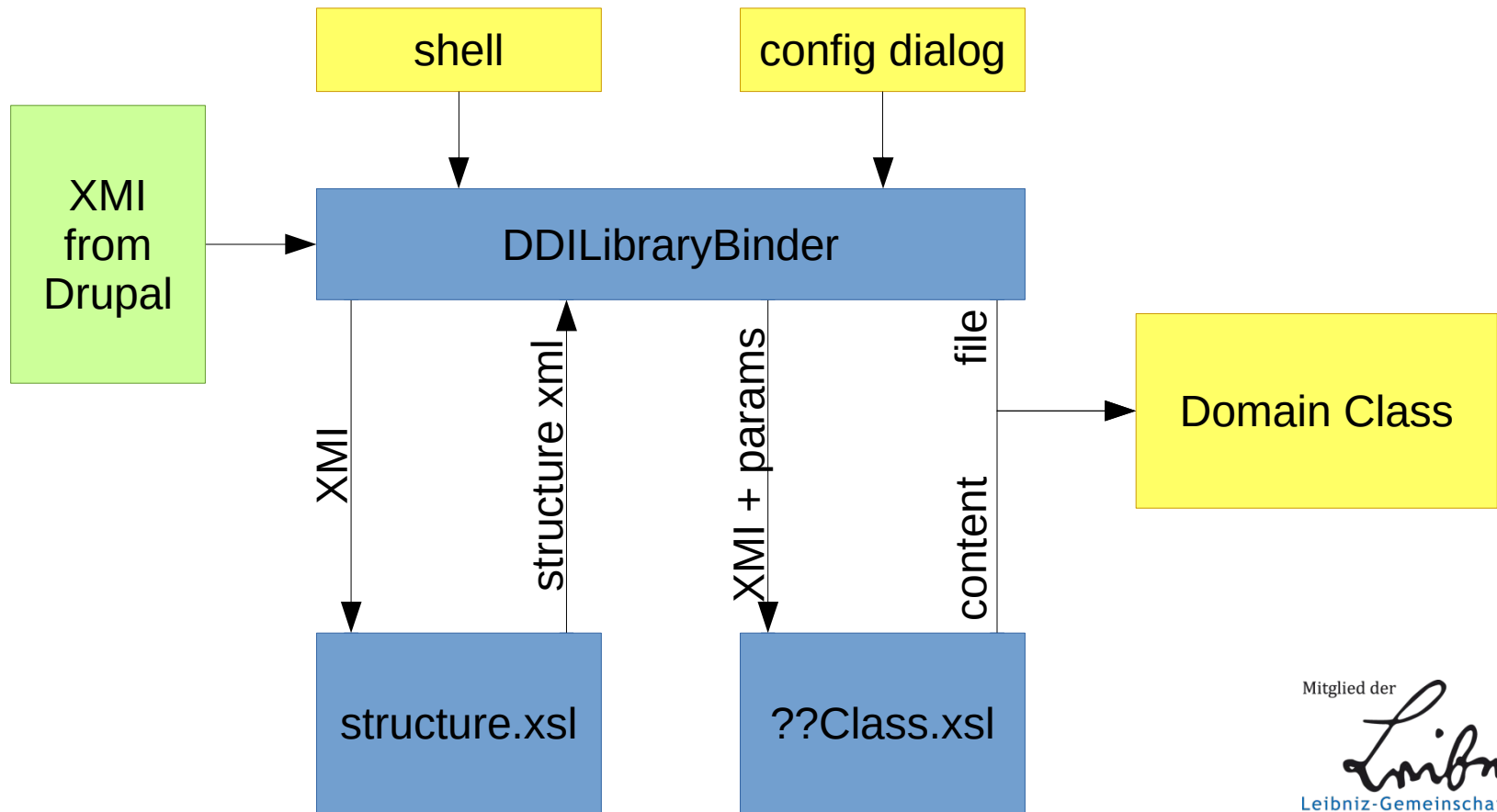
DDI 4: technical production process



Why no standard code generator?

- Code generators are written general purpose
 - Specialties for complex data types
 - It appeared sort of underdefined
 - Different set of rules
- Use of annotations without being defined in the model

Our way of generating code



What is generated for Java?

- Directories for packages (#13)
- One file per class (#387) containing
 - Class definition with parent class
 - Properties
 - Correct data types/classes
 - With List as needed
 - Imports for nested classes from other packages
 - Getter and setter (add/remove) methods
 - Annotations for JAXP and JPA for class and properties

Generic class example

```
@Entity
@XmlAccessorType(XmlAccessType.FIELD)
@XmlRootElement(name="MyClass")
public class MyClass extends org.gesis.core.DDIOObject{
    @XmlElement(name="Property")
    @Column
    private String property;
    public String getProperty() {
        return property;
    }
    public void setProperty(String property) {
        this.property = property;
    }
}
```

According to the handcrafted
example from last year

The screenshot displays an IDE interface with a project explorer on the left and a code editor on the right.

Project Explorer (Left):

- Source Packages
 - org.gesis.Agents
 - Address.java
 - Agent.java
 - ContactInformation.java
 - Email.java
 - Individual.java
 - InstantMessaging.java
 - Machine.java
 - Organization.java
 - Relation.java
 - Telephone.java
 - org.gesis.CollectionManagement
 - org.gesis.Comparison
 - org.gesis.ComplexProcess
 - org.gesis.Conceptual
 - org.gesis.DataCapture
 - org.gesis.Discovery
 - org.gesis.ExtendedPrimitives
 - org.gesis.GenericCollection
 - org.gesis.Methodology
 - org.gesis.NewObjectsForDiscovery
 - Abstract.java
 - AccessRights.java
 - ActedOnBehalfOf.java
 - Activity.java
 - Available.java
 - Broader.java
 - CategoryStatisticType.java
 - ConceptScheme.java
 - ContributorRole.java
 - DataFile.java

Code Editor (Right):

```
import javax.xml.bind.annotation.*;
import org.gesis.SimpleProcess.ProcessStep;

/**
 * generated by DDI4LibraryBinder
 */
@Entity
@XmlAccessorType(XmlAccessType.FIELD)
@XmlRootElement(name="ControlConstruct")
public class ControlConstruct extends org.gesis.SimpleProcess.ProcessStep{

    @XmlIDREF
    @XmlElement(name="")
    @ManyToOne
    private ProcessStep ;

    public ProcessStep get() {
        return ;
    }

    public void set(ProcessStep ) {
        this. = ;
    }
}
```


Even after cleanup

It won't run!

Generating ~~Database and~~ Domain Models from DDI 4

Oliver Hopt, Brigitte Mathiak
EDDI 2014, London

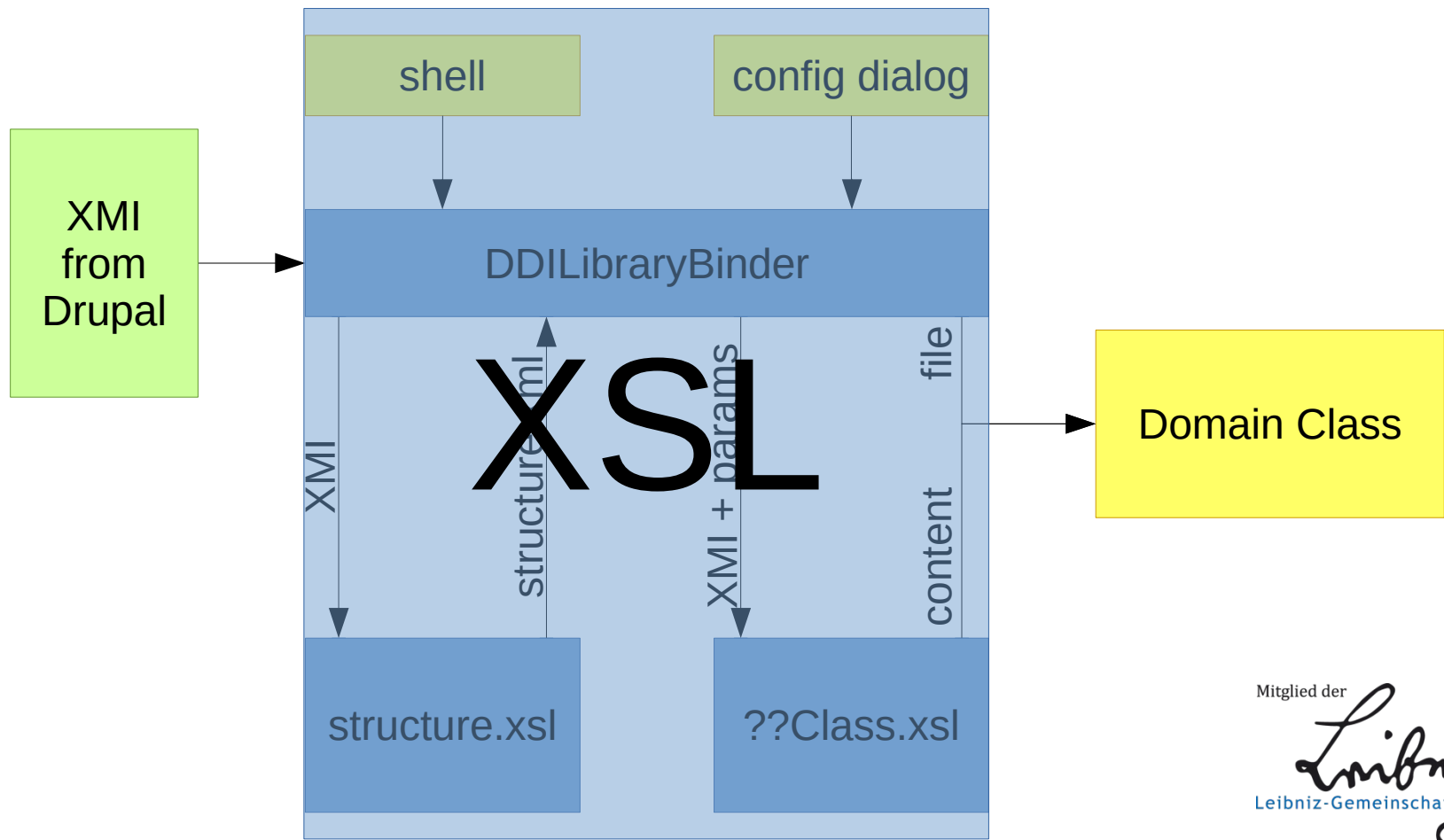
Lessons learned - general

- Generating was easier than expected
- Getting it running therefore even harder
- Good decision to go our own way
 - Price
 - Flexibility
 - Possibly supported platforms
- Strategy for storage needed

Going further

- Adopting the extended rules for generating the schema
- Making use of views
- Additional platforms
 - Grails
 - Ruby on rails
 - Some .Net
 - Some PHP

Makeover for code generation



Discussion and participation