gesis
Leibniz-Institut für Sozialwissenschaften

Generating Database and Domain Models from DDI 4

Oliver Hopt, Brigitte Mathiak

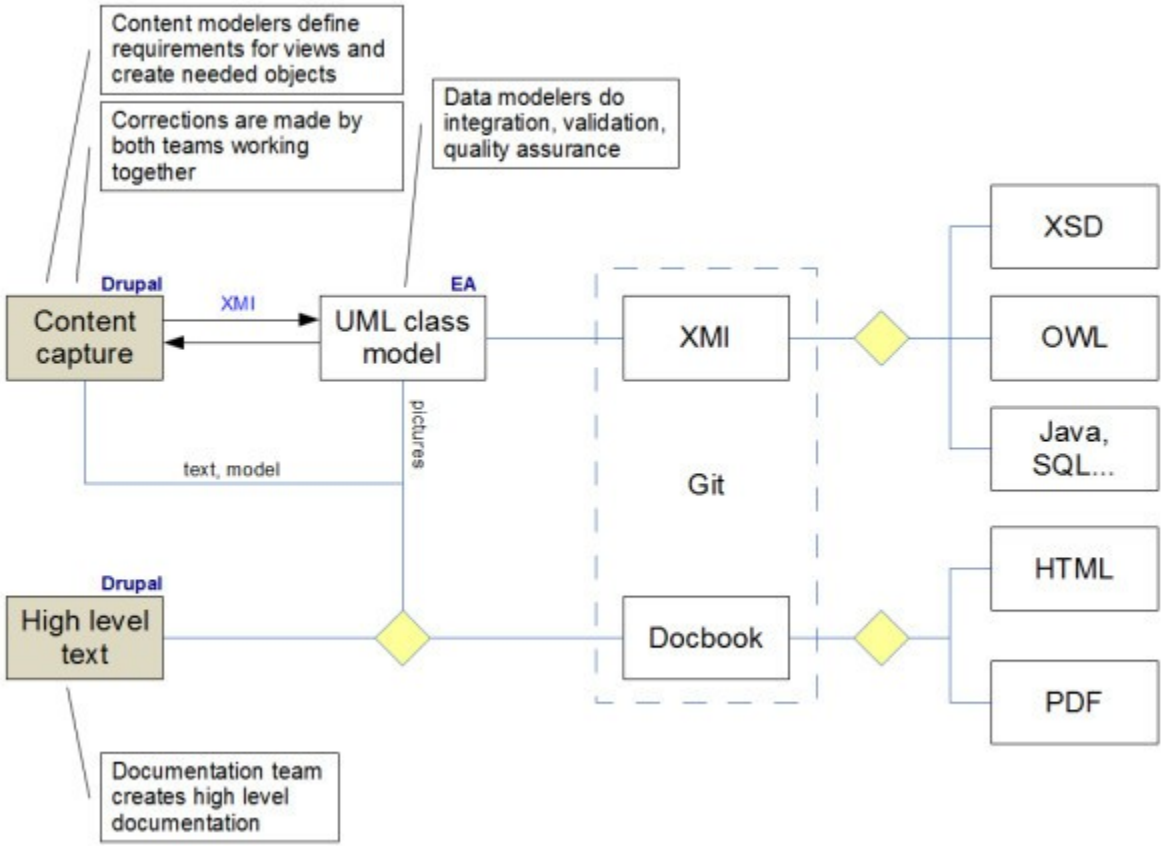
EDDI 2014, London

Mitglied der Leibniz-Gemeinschaft

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

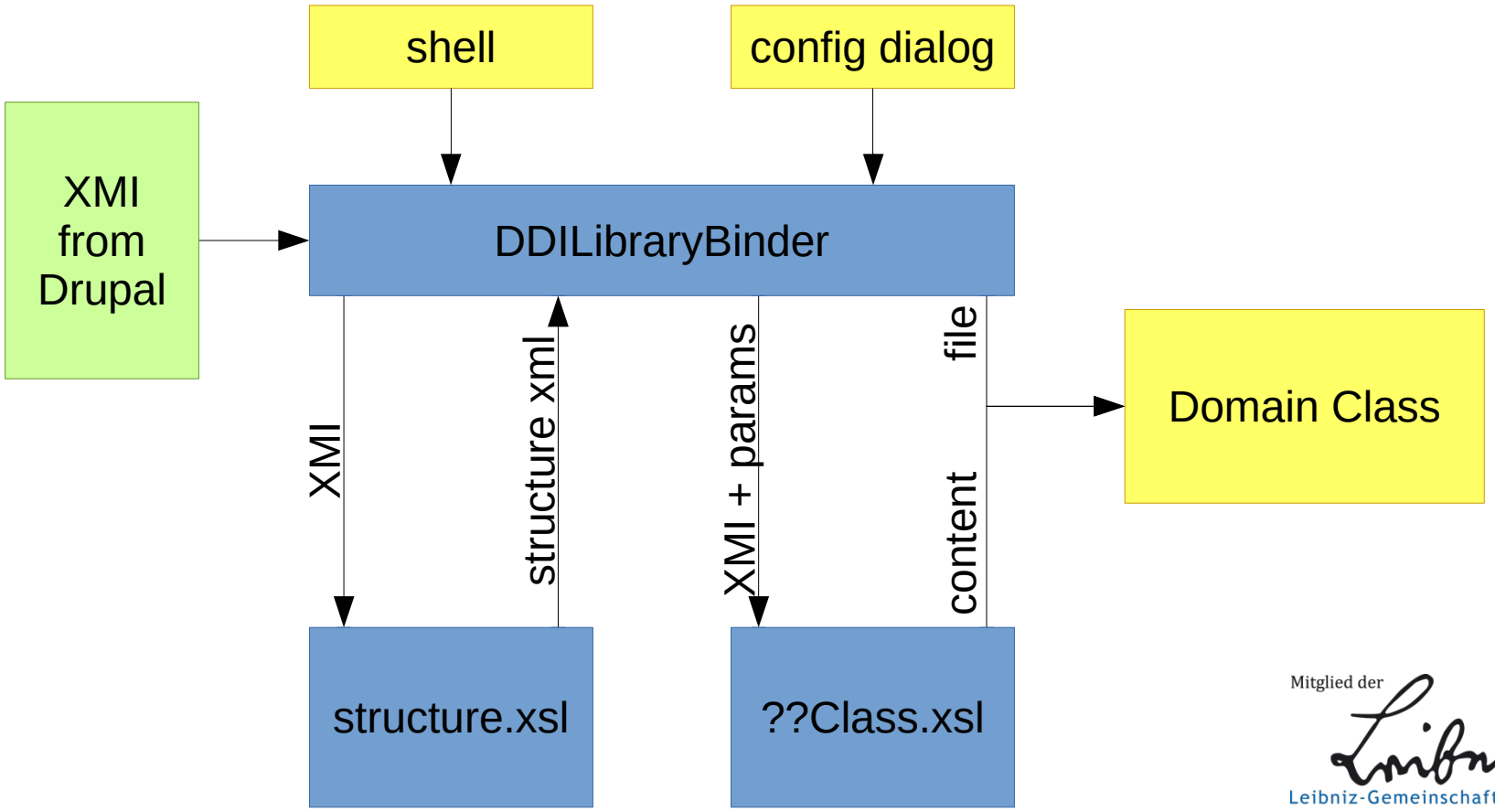


https://dditools.atlassian.net/wiki/display/DDI4/Roadmap+Production+Process

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



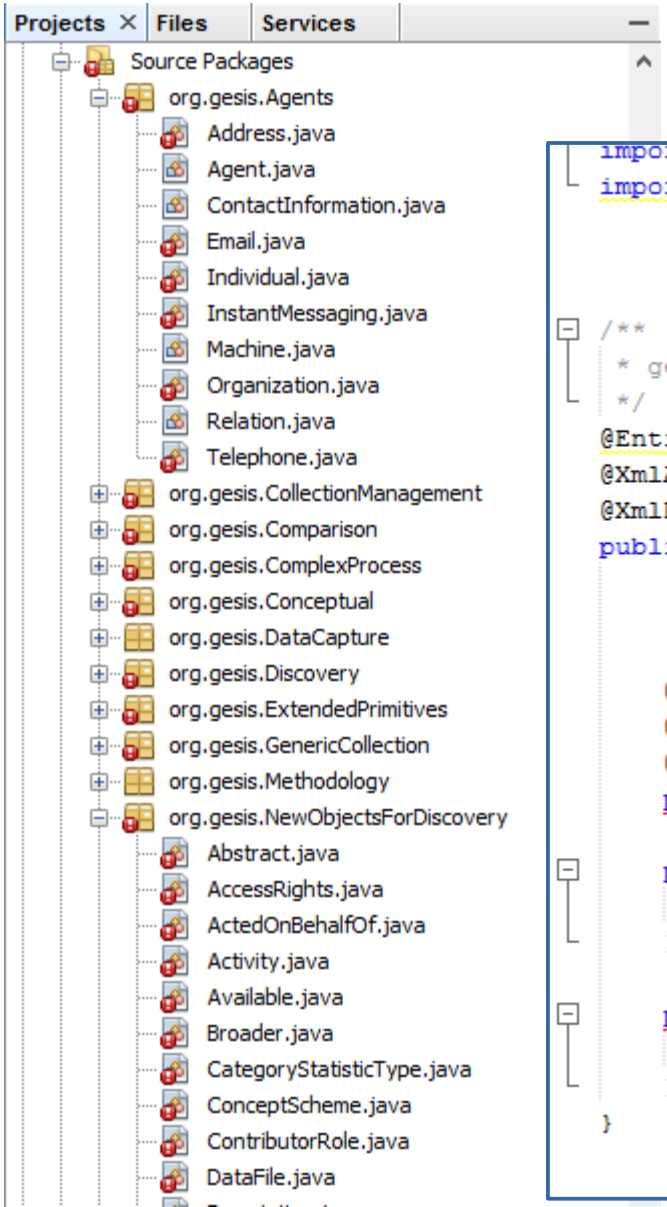
What is generated for Java?

- 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.DDIObject{
    @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




```
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!

gesis
Leibniz-Institut für Sozialwissenschaften

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

Oliver Hopt, Brigitte Mathiak

EDDI 2014, London

Mitglied der Leibniz-Gemeinschaft

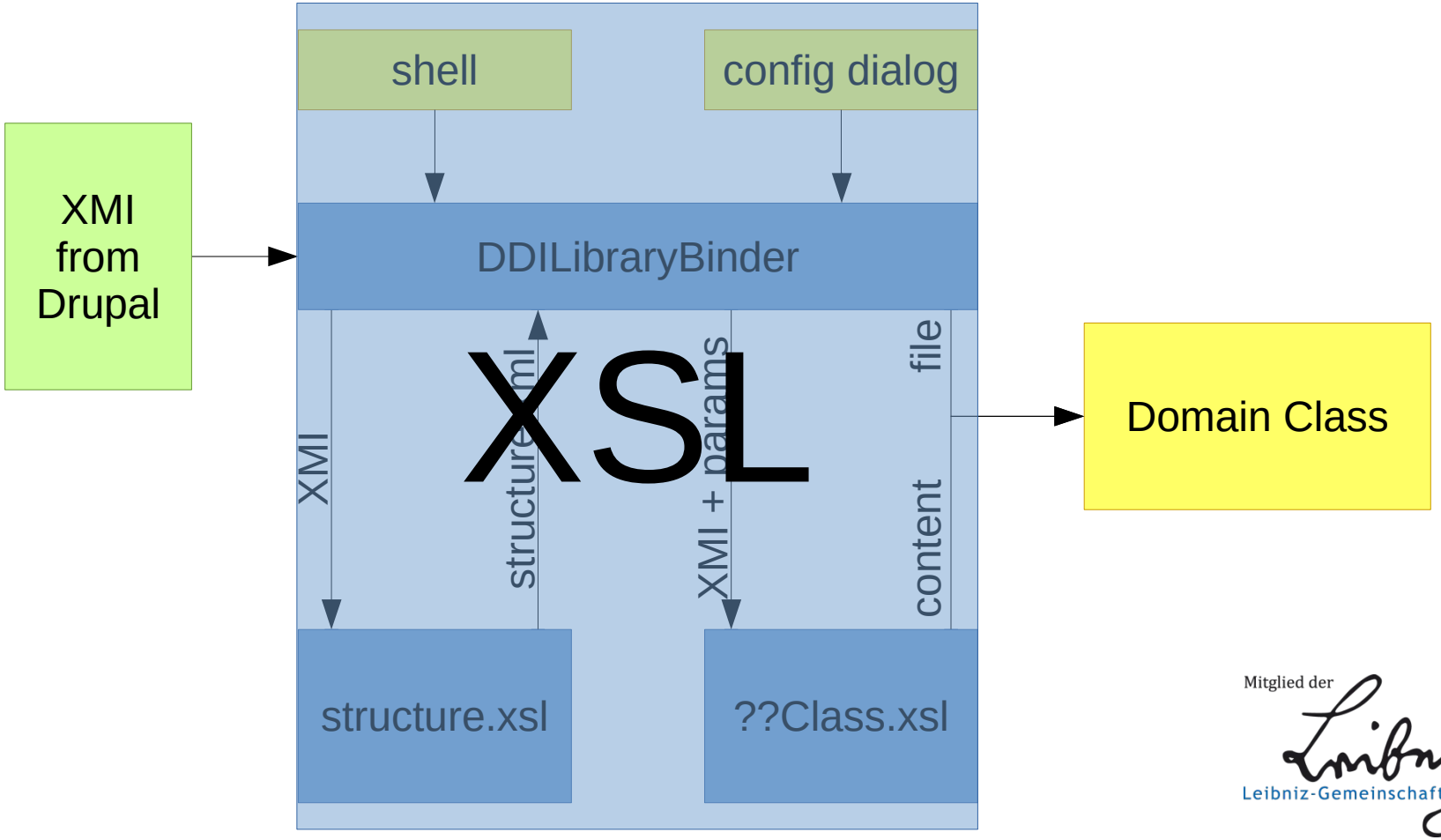
Lessons learned - general

- Generating was easier than expacted
- 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