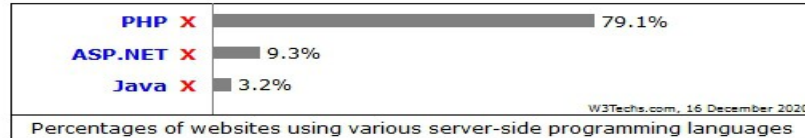


JOOMDD

Herman Peeren, Strumenta Community, 17-12-2020

DOMAIN: extensions for Joomla Web Content Management System

Joomla: open source CMS with PHP



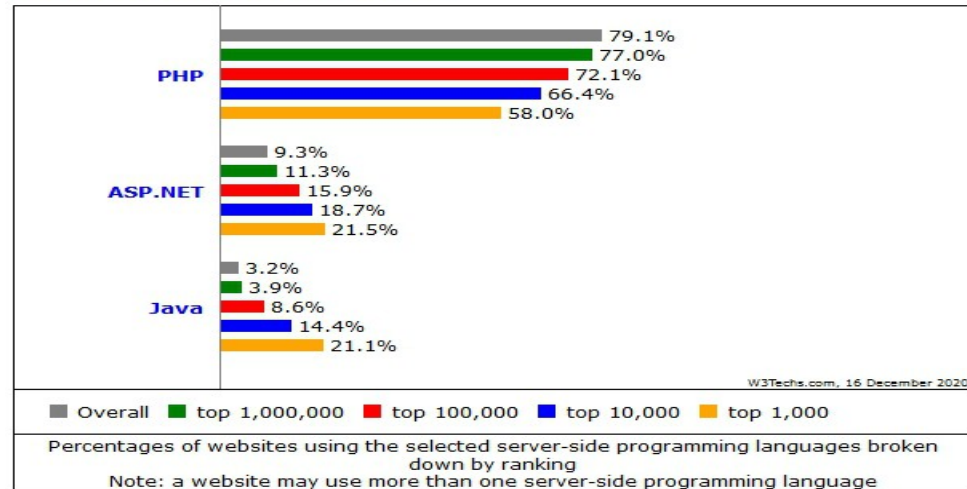
Usage broken down by ranking

This diagram shows the percentages of websites using the selected technologies broken down by ranking.

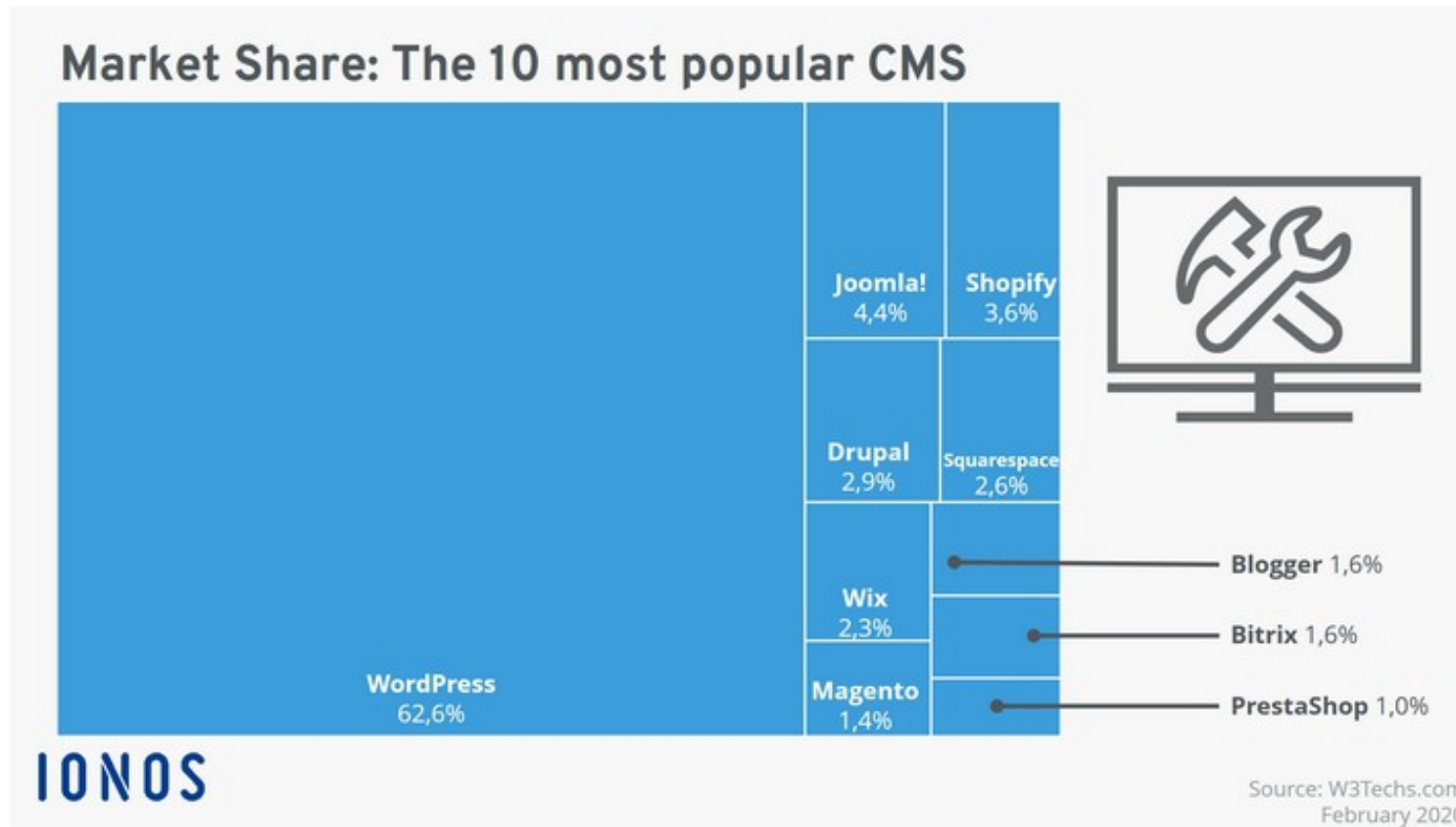
How to read the diagram:

PHP is used by 79.1% of all the websites whose server-side programming language we know.

PHP is used by 77.0% of all the websites whose server-side programming language we know and that rank in the top 1,000,000.



WordPress – Joomla



WordPress is the leading content management system by a large margin. / Source: https://w3techs.com/technologies/overview/content_management/all

Extensions

- **WordPress plugins > 55.000 in official WP repo (nov. 2019)**
- **Joomla a.t.m. > 6.000 extensions on JED (was more; older versions)**
- **Joomla 3 → Joomla 4**

Joomla default component extension

- **backend & frontend**
- **list view & detail view**
- **frontend (website): menu**
- **backend (administrator): editing, toolbar (new, save, edit, etc)**
- **standard CRUD-operations**
- **standard search, sort, select etc.**
- **default multi-language (> 70 languages for Joomla core)**
- **MVC**

Boilerplate code & conventions.

List view and detail view

The image shows two overlapping screenshots of the Joomla! administration interface. The background screenshot displays the 'Participants' list view, and the foreground screenshot shows the 'Participant' detail view.

Participants List View:

- Navigation: System, Users, Menus, Content, Components, Extensions, Help. Joomla! logo and version J3.
- Toolbar: New, Edit, Publish, Unpublish, Archive, Check-in, Trash, Options.
- Search: Search bar, Search Tools, Clear, 20 items per page.
- Table:

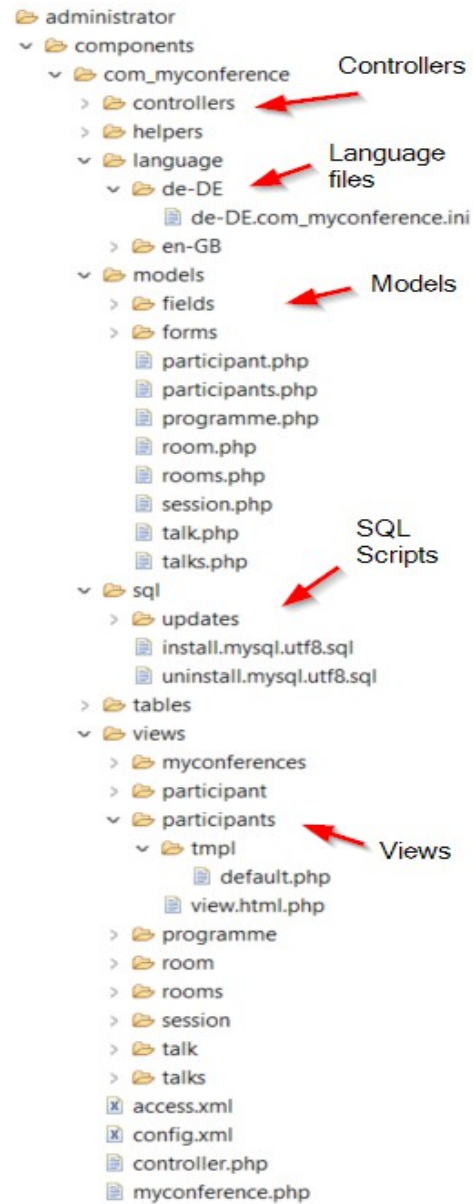
Status	Name	Address	Affiliation	id
<input checked="" type="checkbox"/>	Dennis Priefer	Gießen	THM	1
<input checked="" type="checkbox"/>	Wolf Rost			
<input checked="" type="checkbox"/>	John Doe			
<input checked="" type="checkbox"/>	Zurbruggen Anderson			
<input checked="" type="checkbox"/>	Yaddow Green			

Participant Detail View:

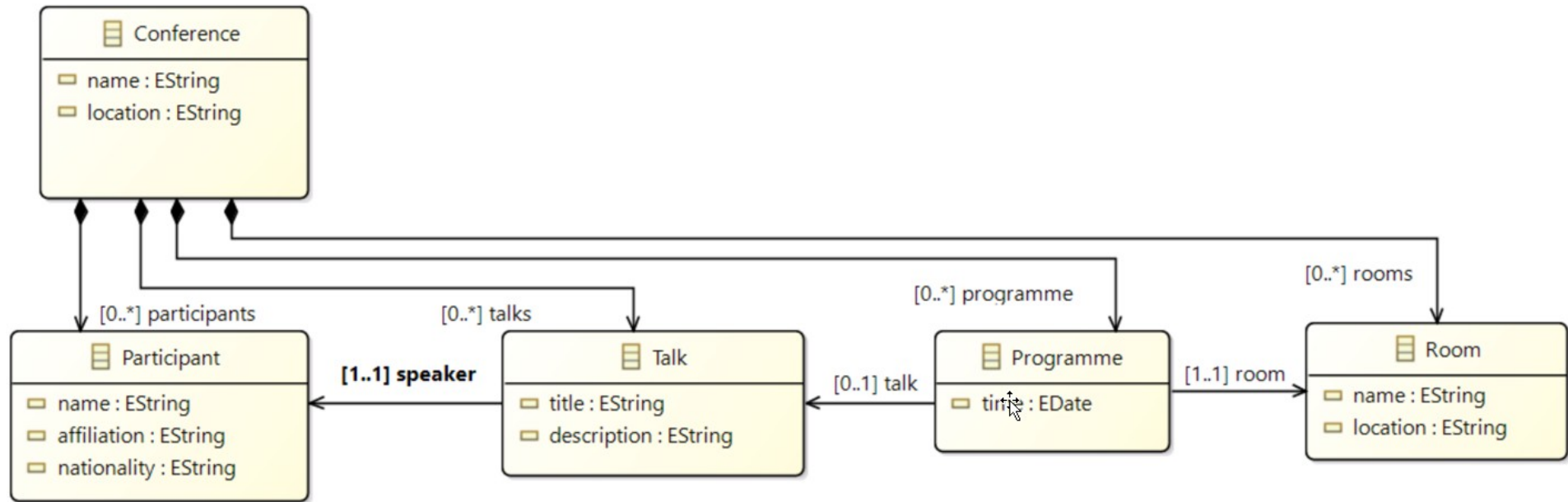
- Navigation: System, Users, Menus, Content, Components, Extensions, Help. Joomla! logo and version J3.
- Toolbar: Save, Save & Close, Save & New, Save as Copy, Close.
- Form Fields:

Name	John Doe
Affiliation	THM
Nationality	German

Component file structure



Example: conference model (ERM)



The DSL: eJSL (in Xtext)

```
eJSLModel "Conference" {
  eJSL part: CMS Extension {
    entities {...} // Data modelling
    pages {...} // Interaction modelling
    extensions {...} // Extension modelling
  }
}
```



Data modelling: define entities, attributes & references in eJSL

```
Entity Talk {
  attributes {
    Attribute title {
      type = Short_Text
    }
    Attribute ^description { // Some identifiers need the '^' prefix, since they are also used as language keywords
      type = Text
    }
    Attribute speaker {
      type = Short_Text
    }
  }
  references {
    Reference { // Reference definition
      entityAttribute = speaker // The reference attribute of the entity
      referencedEntity = Participant // Identifier of the referenced entity
      referencedEntityAttribute = Participant.name // Referenced attribute
      min = 1 // Reference multiplicity (lower) (optional)
      max = 1 // Multiplicity (upper)
    }
  }
}
```


What entities/attributes on what kind of page

```
pages {  
  IndexPage Participants {...}  
  DetailsPage Participant {...}  
  IndexPage Talks {...}  
  DetailsPage Talk {...}  
  IndexPage Rooms {...}  
  DetailsPage Room {...}  
  IndexPage Programme {...}  
  DetailsPage Session {...}  
}
```

```
IndexPage Participants {  
  *entities = Participant // Will be interpreted as list view  
  representationColumns = Participant.name, Participant.address, Participant.affiliation // Which entity will be presented in the list?  
  // (optional) selection of list columns  
  filters = Participant.name, Participant.affiliation // Filters for Search tools support  
  links { // Link definition to enable interaction between views  
    InternalContextLink Details {  
      target = Participant // Link to details page  
      linkedAttribute = Participant.name // Attribute which is used as link  
      linkParameters {  
        Parameter name = *Attribute "Participant.name" // Send participant's name as request param  
      }  
    }  
  }  
}
```

And define which pages you want in the extension

```
extensions {  
  Component MyConference {  
    Manifestation {...}  
    languages {...}  
    sections {...}  
  }  
  Module Talks {...}  
}
```

```
sections {  
  FrontendSection { // Here you specify the pages which will be used as frontend views  
    *Pages {  
      *Page : Participants  
      *Page : Talks  
      *Page : Rooms  
      *Page : Programme  
    }  
  }  
  BackendSection { // Here you specify the pages which will be used as backend views  
    *Pages {  
      *Page : Participants  
      *Page : Participant  
      *Page : Talks  
      *Page : Talk  
      *Page : Rooms  
      *Page : Room  
      *Page : Programme  
      *Page : Session  
    }  
  }  
}
```

Xtend Code generator

Editor

- Eclipse
- also for JetBrains' PHPStorm & IntelliJ (but Xtext plugin is broken now)
- web-editor (also from Xtext) – ctrl-enter for suggestions

The screenshot shows the JooMDD Web Editor interface. At the top left is the logo 'JooMDD' and at the top right are links for 'Documentation' and 'LogIn'. The main header reads 'JooMDD Web Editor' and 'Create your Joomla! Extension on the fly...'. Below the header is a toolbar with buttons: 'Save Model', 'Generate Code', a dropdown menu currently showing 'Joomla 3', and 'Load Example Model'. To the right of the toolbar is a vertical list of actions: '+ Add Model', 'Download Node', 'Load Model to Editor', 'Upload Joomla! Extension', and 'Extract Extension Information'. The central area is a code editor titled 'main.eJSL' containing XML-like code for an eJSL model. The code defines an 'eJSLModel "Conference"' with an 'eJSL part: CMS Extension' containing two entities: 'Entity Participant' and 'Entity Talk'. 'Entity Participant' has four attributes: 'name' (Short_Text, unique), 'affiliation' (Text), 'nationality' (Text), and 'address' (Text). 'Entity Talk' has an 'attributes' section. A 'Resource Tree' on the right shows a project structure: 'My Workspace' containing 'reverse', 'src', 'src-gen', 'Extensions', and 'Conference'. 'Conference' contains a 'j4' folder. Red arrows point to the 'Generate Code' button with the annotation 'Code Generator for Joomla 3 and 4', the code editor with 'Model Editor', the 'Upload Joomla! Extension' button with 'Support for Legacy Code', and the 'j4' folder in the Resource Tree with 'Ressource Tree'.

JooMDD

Documentation LogIn

JooMDD Web Editor

Create your Joomla! Extension on the fly...

Save Model Generate Code Joomla 3 Load Example Model

+ Add Model
Download Node
Load Model to Editor
Upload Joomla! Extension
Extract Extension Information

Support for Legacy Code

main.eJSL

```
1 eJSLModel "Conference" {  
2   eJSL part: CMS Extension {  
3     entities {  
4       Entity Participant {  
5         attributes {  
6           Attribute name {  
7             type = Short_Text  
8             Unique attribute with ID  
9           }  
10          Attribute affiliation {  
11            type = Text  
12          }  
13          Attribute nationality {  
14            type = Text  
15          }  
16          Attribute address {  
17            type = Text  
18          }  
19        }  
20      }  
21      Entity Talk {  
22        attributes {  
23          ...  
24        }  
25      }  
26    }  
27  }  
28 }
```

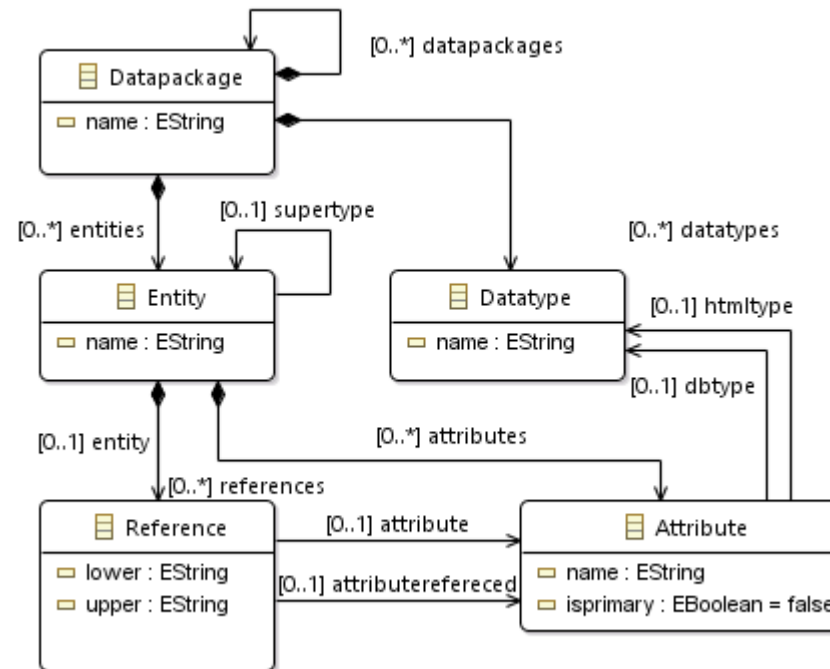
Model Editor

My Workspace
reverse
src
src-gen
Extensions
Conference
j4

Ressource Tree

eJSL

- Grammar in Xtext => ANTLR3 (no left recursion)



```
// data model
Type:
    DatatypeReference | StandardTypes
;

DatatypeReference:
    type=[Datatype|QualifiedName]
;

StandardTypes:
    type=StandardTypeKinds (nonnull?='Not Null')? ('Default =' default=STRING)? (autoincrement?='Auto Increment')?
;
```

Datatype **returns** *Datatype*:

```
{Datatype}
'Datatype' name=ID '=' type=STRING;
```

Parameter **returns** *Parameter*:

```
{Parameter}
'Parameter' name=ID
'{'
    //'type' '=' ((dtype=[Datatype|QualifiedName]) | ('jvmtype' jvmtype=[jvmTypes::JvmType|QualifiedName]))
    'type' '=' (dtype=HTMLTypes)
    ('defaultValue' defaultvalue=STRING)?
    ('label' label=STRING)?
    ('size' size=INT)?
    ('description' description=STRING)?
    ('values' '{' values+=KeyValuePair (',' values+=KeyValuePair)* '}')?
    ('fieldAttributes' '{' (attributes+=KeyValuePair (',' attributes+=KeyValuePair)*)? '}')?
'}
```

;

ParameterGroup **returns** *ParameterGroup*:

```
{ParameterGroup}
'ParameterGroup' name=ID
'{'
    ('label' label=STRING)?
    ('parameters' '{' ((globalparameters+=[Parameter]) | (parameters+=Parameter))* '}')
'}
```

;

PageAction **returns** *PageAction*:

```
{PageAction}
'PageAction' name=ID
'{'
    'type' '=' ((pageActionType=PageActionKind))
    'position' '=' ((pageActionPosition=PageActionPositionKind))
'}
```

;

Entitypackage **returns** *Entitypackage*:

```
{Entitypackage}
'Entitypackage' name=ID
```

```

    '{'
      ('entityPackages' '{' (entitypackages+=Entitypackage)* '}')?
      ('entities' '{' (entities+=Entity)* '}')?
      ('dataTypes' '{' (datatypes+=Datatype)* '}')?
    '}'
;

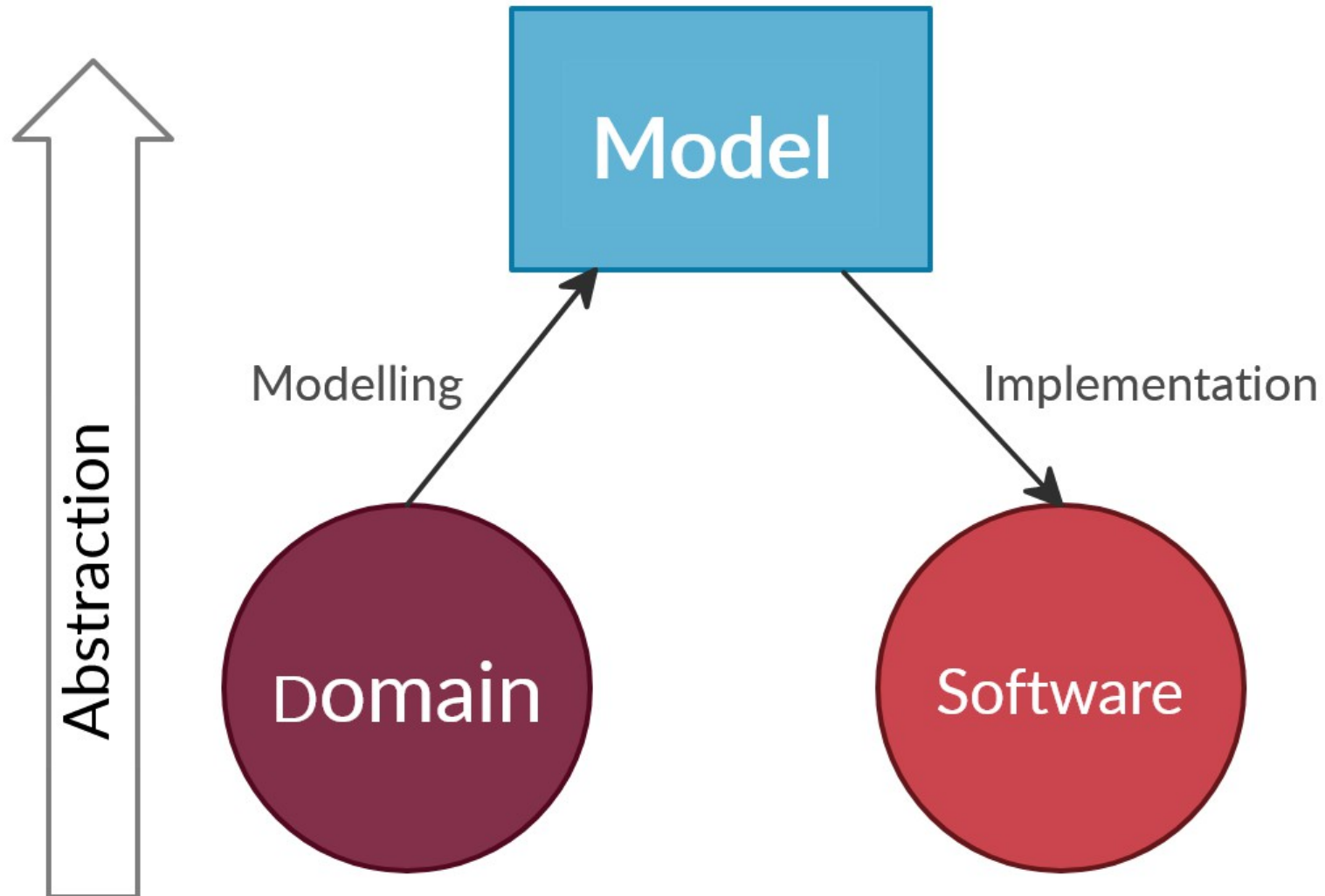
Entity returns Entity:
  {Entity}
  'Entity' name=MYID ('extends' supertype=[Entity|QualifiedName])? (preserve?=@preserve)?
  '{'
    ('attributes' '{' (attributes+=Attribute)* '}')?
    // Kardinalität
    ('references' '{' (references+=Reference)* '}')?
  '}'
;

Attribute returns Attribute:
  {Attribute}
  //'Attribute' name=ID ':' ((dtype=[Datatype|QualifiedName]) | ('jvmtype' jvmtype=[jvmTypes::JvmType|QualifiedName]))
  'Attribute' name=MYID (preserve?=@preserve)? '{'
    'type =' (type=Type)
    (isunique ?= 'Unique attribute' ('with' (withattribute=[Attribute|QualifiedName] | id?='ID'))?)?
    (isprimary ?= 'Primary attribute')?
  '}'
;

Reference returns Reference:
  {Reference}
  'Reference' (preserve?=@preserve)?
  '{'
    'entityAttribute = attribute+=[Attribute|QualifiedName] (',' attribute+=[Attribute|QualifiedName])*
    'referencedEntity = entity=[Entity|QualifiedName]
    'referencedEntityAttribute = (attributereferenced +=[Attribute|QualifiedName] | id?='ID') (attributereferenced
+=[Attribute|QualifiedName])*
    ('min =' lower=NUMBER)?
    'max = upper=NUMBER
  '}'
;

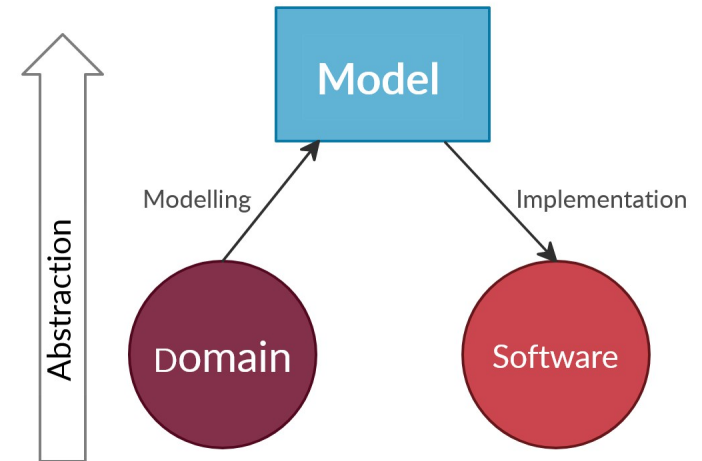
```

What is a DSL?



Model is made with a modelling language. In JooMDD: eJSL.

- If domain = Joomla/CMS-extensions-building then eJML is a **Domain Specific** Language.
- Term “DSL” is mostly used to indicate the direct **modelling language**.
- If you would use an ERM language like eJSL to model e.g. a conference, then the modelling language would not be specific for that domain, not a DSML.
- But the **model** IS domain specific. Because the **semantics** are domain specific, using concepts from the domain.
- In DDD, focussing on business domains, modelling is often done with non-domain-specific concepts, like in an ERM and Business Process Modelling (and especially “events”).
- “ubiquitous language” is mainly modelled with domain specific **semantics**, not with domain specific (modelling) languages.
- Horizontal and vertical domains is 2-dimensional. But you can see it in **multiple dimensions** like business processes (order, sale, invoicing, payment etc), manufacturing processes (resources, products, storage, etc). Chr. Alexander: “A city is not a tree”. Gärdenfors: “the geometry of meaning”.
- Modular languages (DSL-engineering book).



- Not every domain specific Model needs a Domain Specific (Modelling) Language!
- A “real” DSL = a DSML.

You only need that if there are enough “moving parts”, variables: things that **change** over time, while other domain specific parts are invariant. You make a DSML to model the change. The language must have enough **domain specific expressivity**.

A language L_1 is *more expressive in domain D*
than a language L_2 ($L_1 \prec_D L_2$),
if for each $p \in P_D \cap P_{L_1} \cap P_{L_2}$, $|p_{L_1}| < |p_{L_2}|$.

(DSL Engineering book, Ch. 4.1)

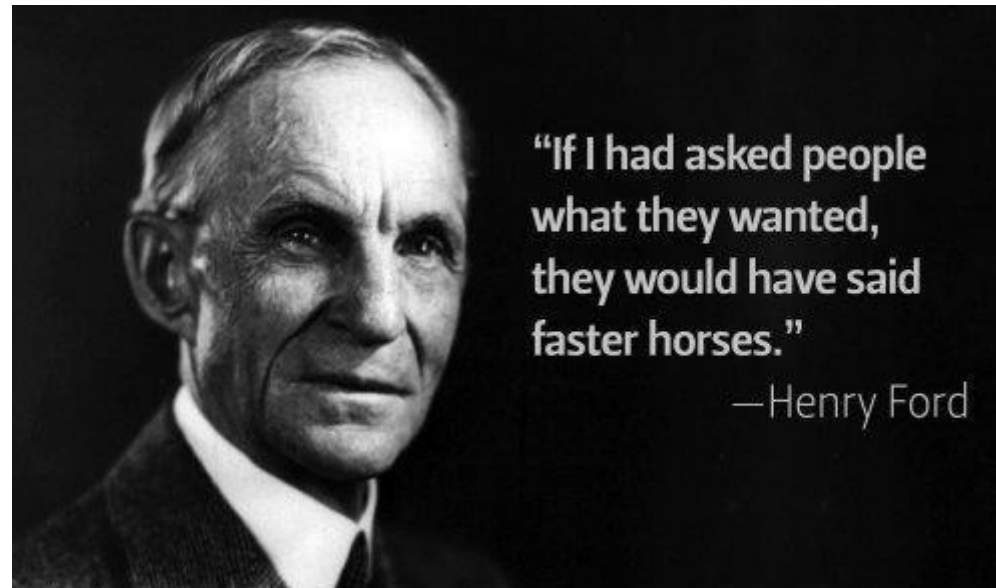
Translation: a language is more expressive in a domain
if you can mostly write shorter programs with it to express the same things.

Educational / research project

What the “customer” needs:

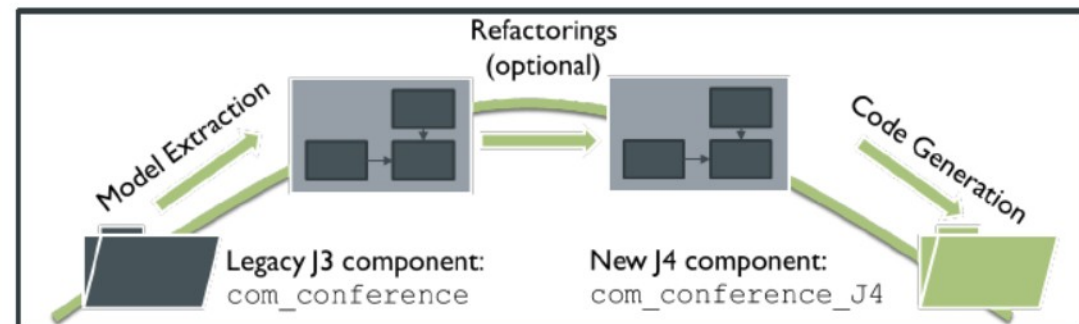
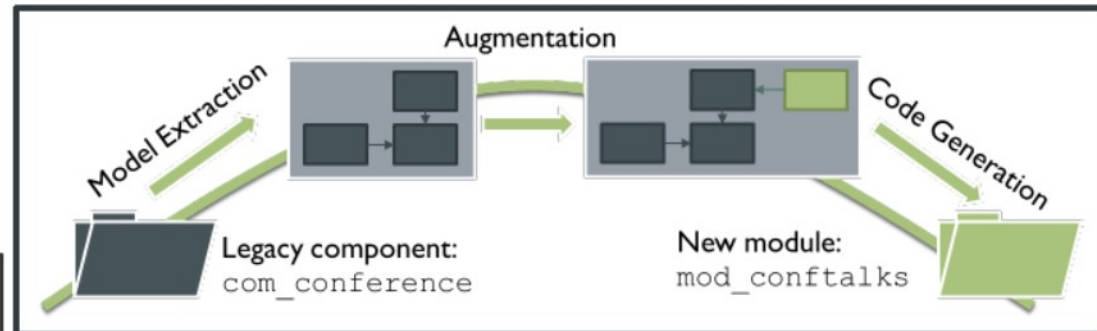
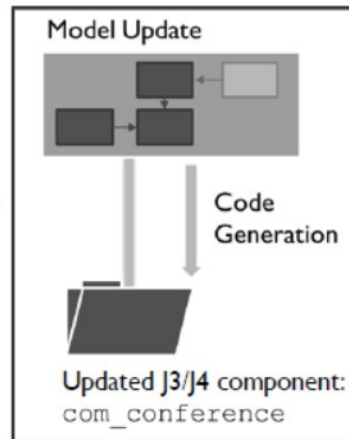
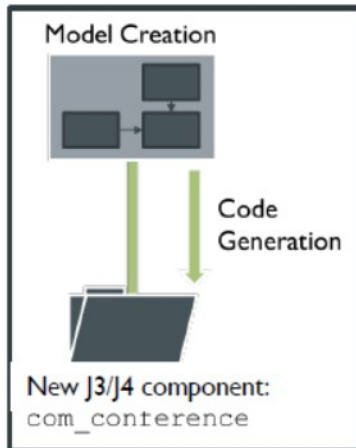
- Joomla devs using PHPstorm → IntelliJ-plugin
- webeditor(needs a Java-server) – SAAS
- reverse engineering

Selling DSLs: not only look at what nice product you have, but also **what the customer needs.**



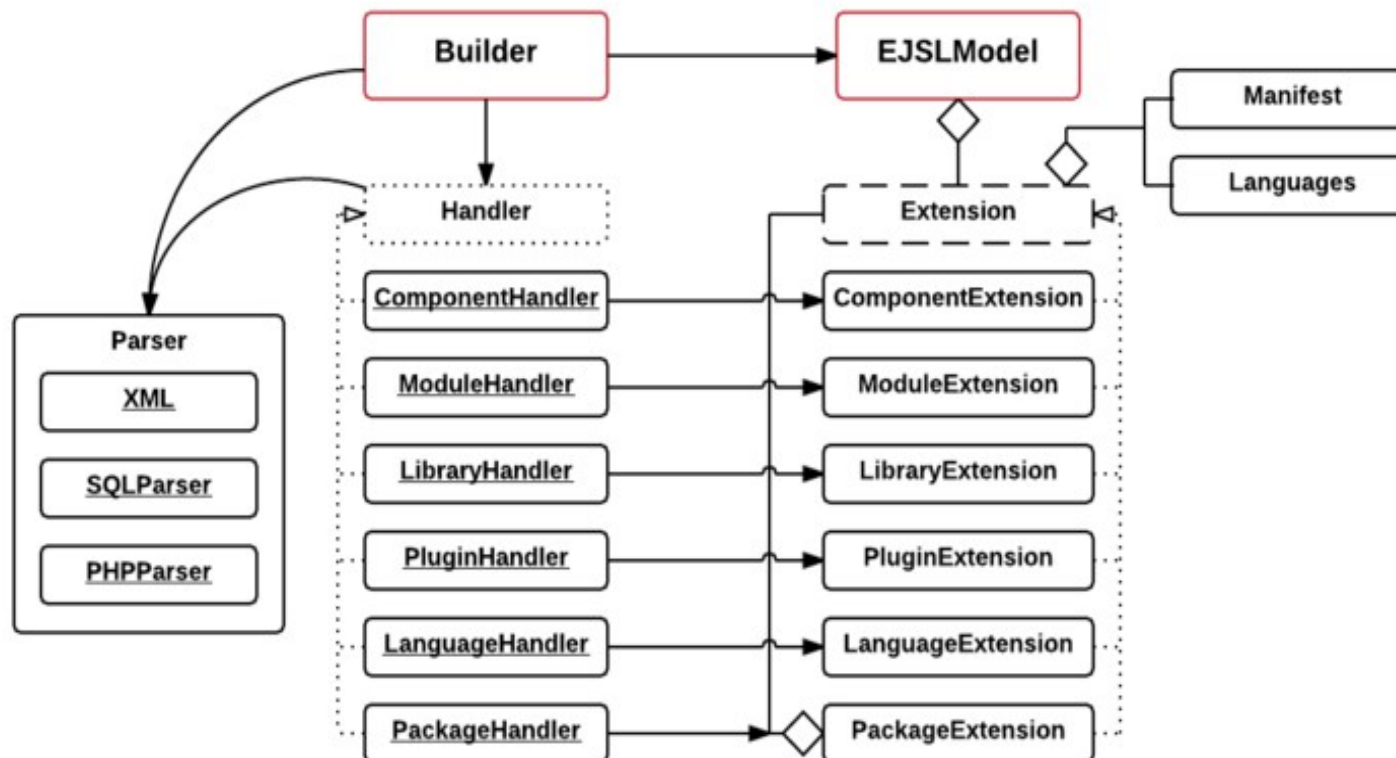
Reverse engineering

DEVELOPMENT USE CASES WITH JOOMDD



Joomla Extensions --> jext2ejsl --> eJSL Model

- Built parsers for PHP and SQL in Scala
- [https://wiki.thm.de/Reverse-Engineering_\(Joomla-Code_zu_eJSL-Instanzmodell\)](https://wiki.thm.de/Reverse-Engineering_(Joomla-Code_zu_eJSL-Instanzmodell)) (German)



Details of the extension get lost, especially when coding was different from “standard” Joomla coding. Some things are guessed to fit the model.

Reverse engineering very interesting as research-project.

Repo:

<https://github.com/thm-mni-ii/JooMDD>

Web-editor:

<https://icampus.thm.de:9443/>

Thank you!

herman@yepr.nl

hermanpeeren.nl