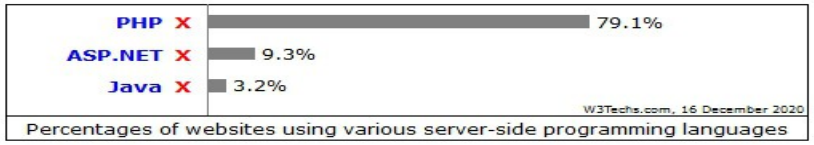
Herman Peeren, Strumenta Community, 17-12-2020

DOMAIN: extensions for Joomla Web Content Management System

Joomla: open source CMS with PHP

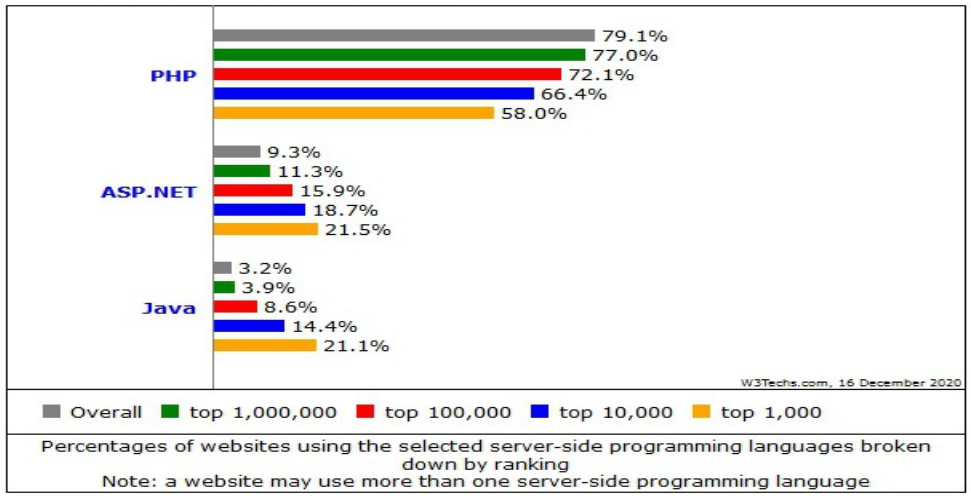


Percentages of websites using various server-side programming languages

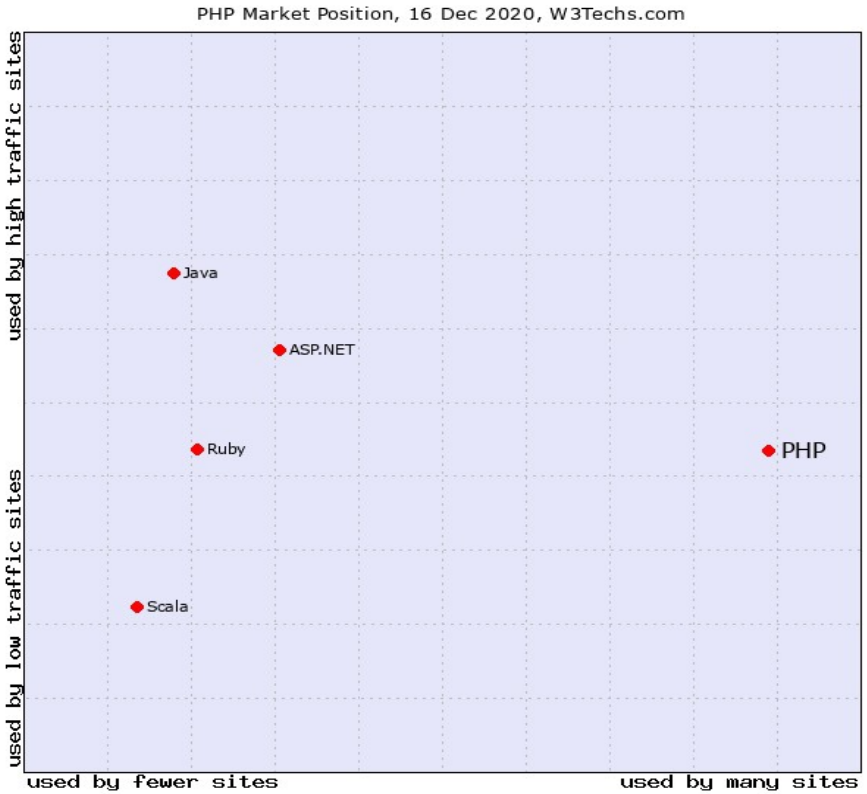
Usage broken down by ranking

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

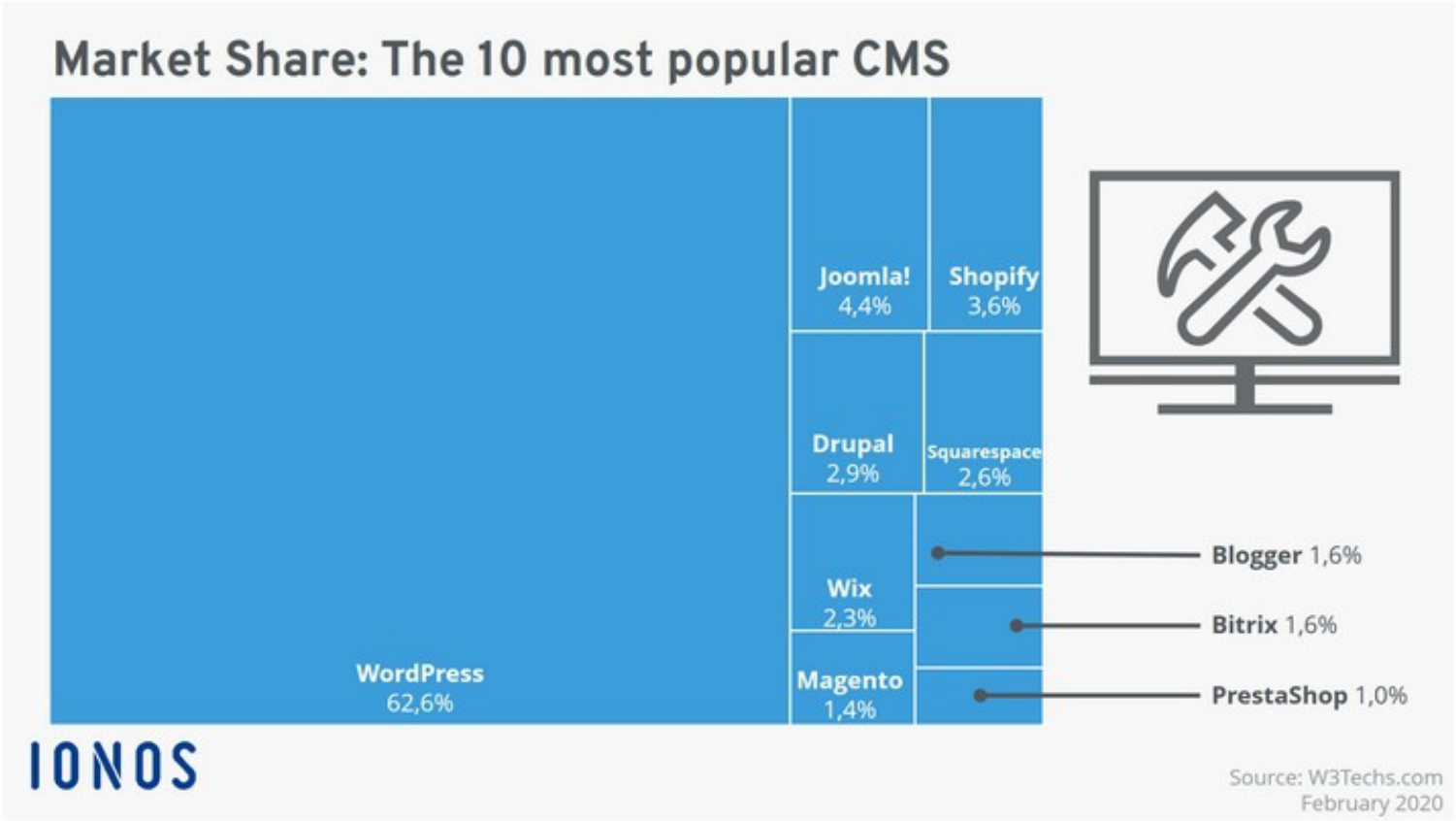
How to read the diagram:
PHP is used by 79.1% of all the websites whose server-side programming language we know.
PHP is used by 77.0% of all the websites whose server-side programming language we know and that rank in the top 1,000,000.



Percentages of websites using the selected server-side programming languages broken down by ranking
Note: a website may use more than one server-side programming language



WordPress – Joomla



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

Extensions

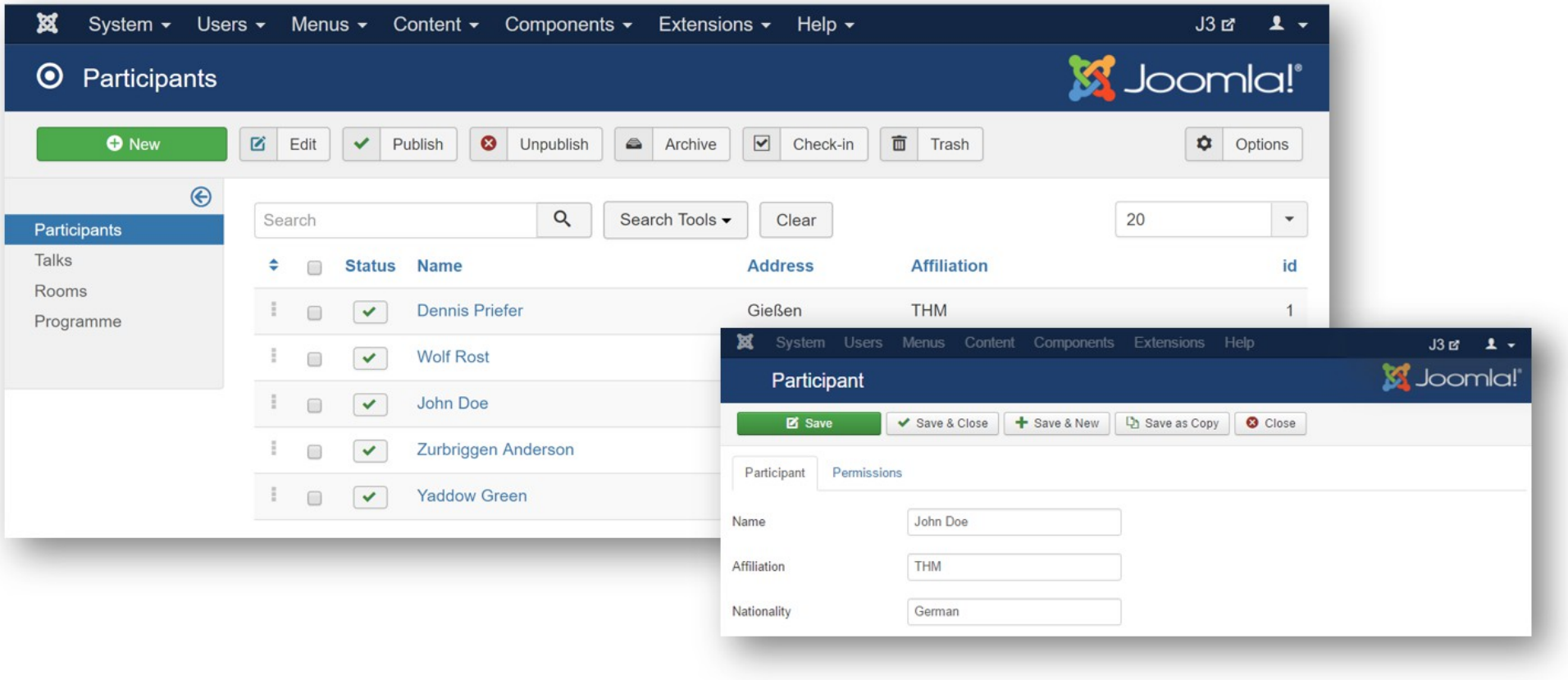
- WordPress plugins > 55.000 in official WP repo (nov. 2019)
- Joomla a.t.m. > 6.000 extensions on JED (was more; older versions)
- Joomla 3 → Joomla 4

Joomla default component extension

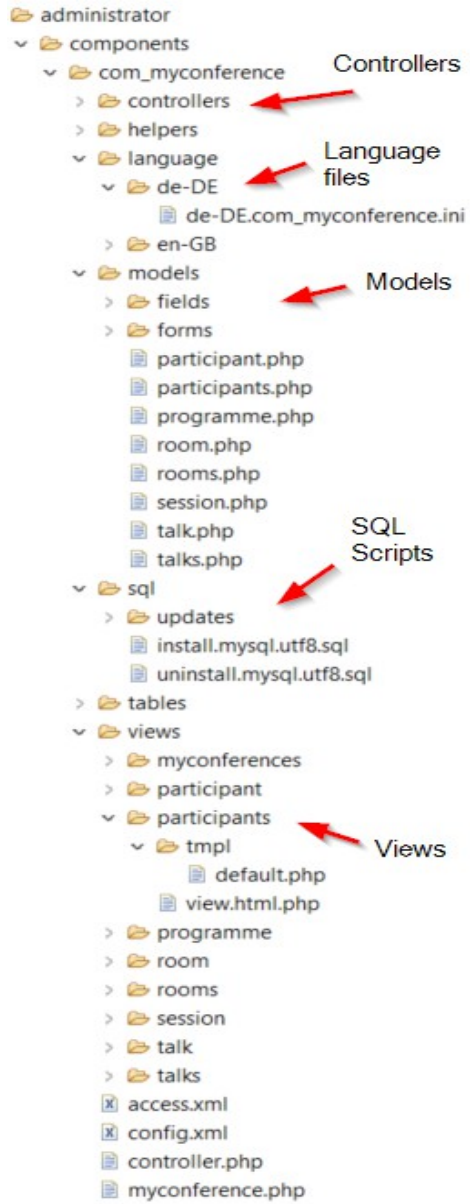
- backend & frontend
- list view & detail view
- frontend (website): menu
- backend (administrator): editing, toolbar (new, save, edit, etc)
- standard CRUD-operations
- standard search, sort, select etc.
- default multi-language (> 70 languages for Joomla core)
- MVC

Boilerplate code & conventions.

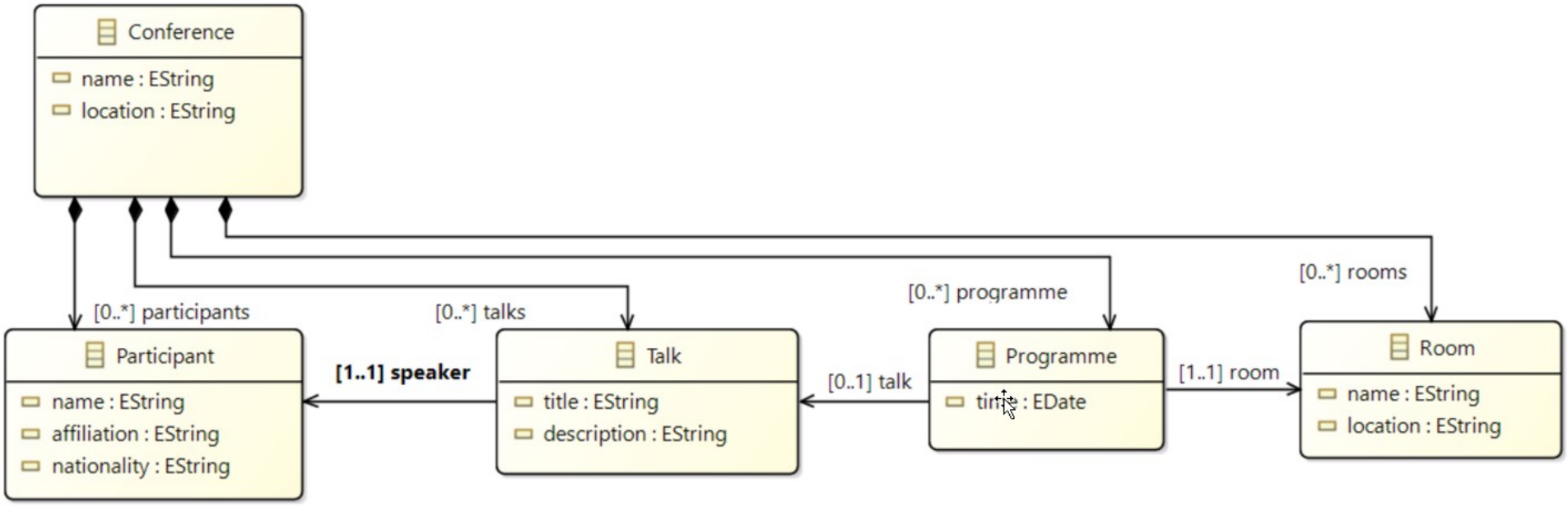
List view and detail view



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/attibutes on what kind of page

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

```
IndexPage Participants {                                    // Will be interpreted as list view
    *entities =  Participant                                // Which entity will be presented in the list?
    representationColumns = Participant.name, Participant.address, Participant.affiliation // (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
        }
    }
}
```

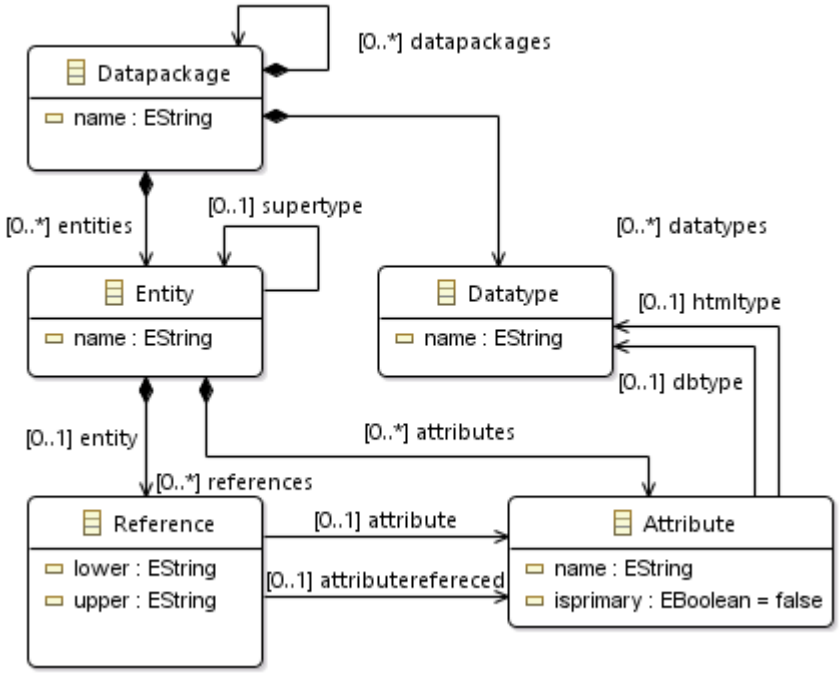
Xtend Code generator

Editor

- **Eclipse**
- **also for jetbrains' PHPstorm & IntelliJ (but Xtext plugin is broken now)**
- **web-editor (also from Xtext) – ctrl-enter for suggenstions**

eJSL

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

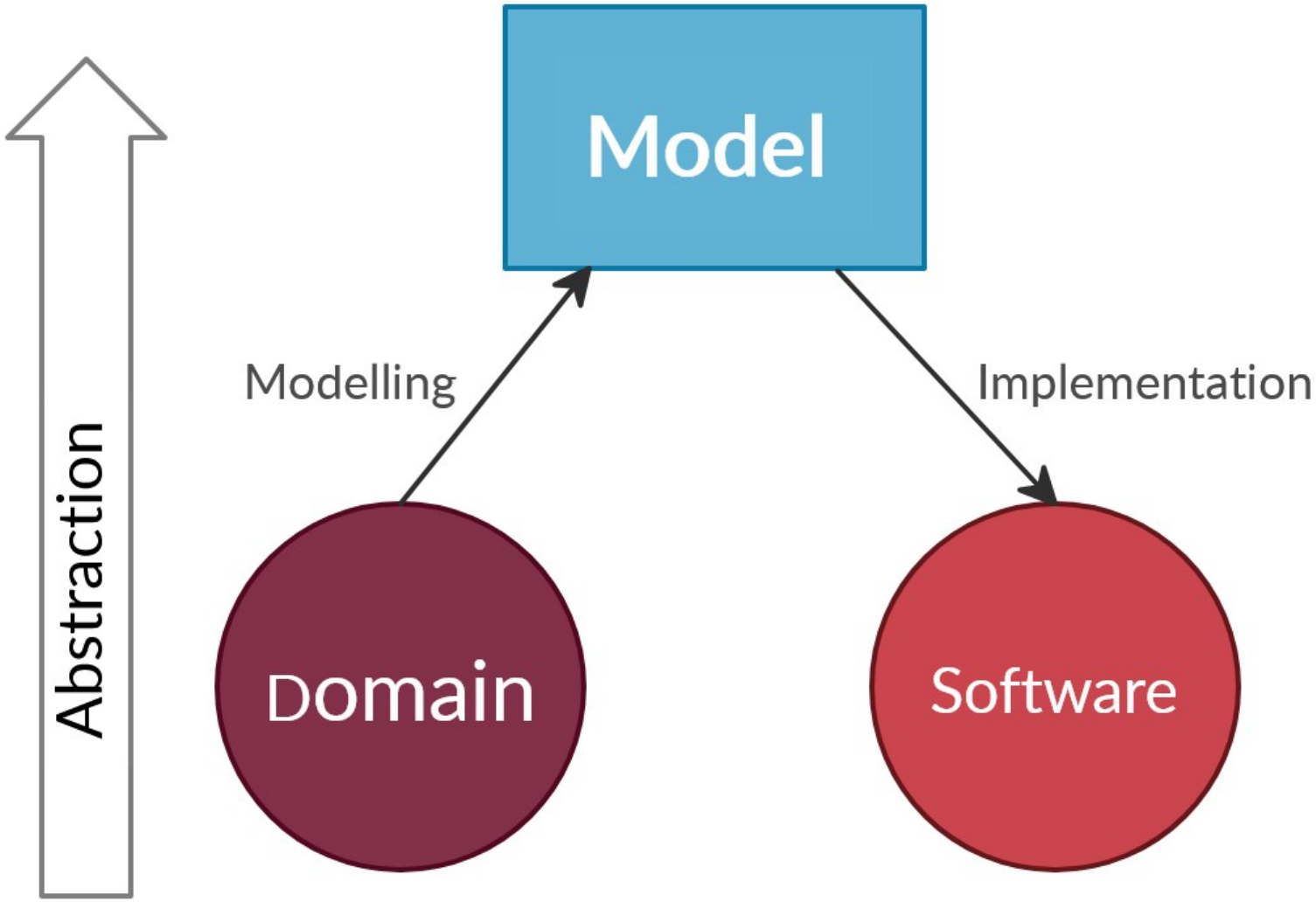



```
// data model
Type:
	DatatypeReference | StandardTypes
;

DatatypeReference:
	type=[Datatype|QualifiedName]
;

StandardTypes:
	type=StandardTypeKinds (notnull?='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 =' descripton=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
```

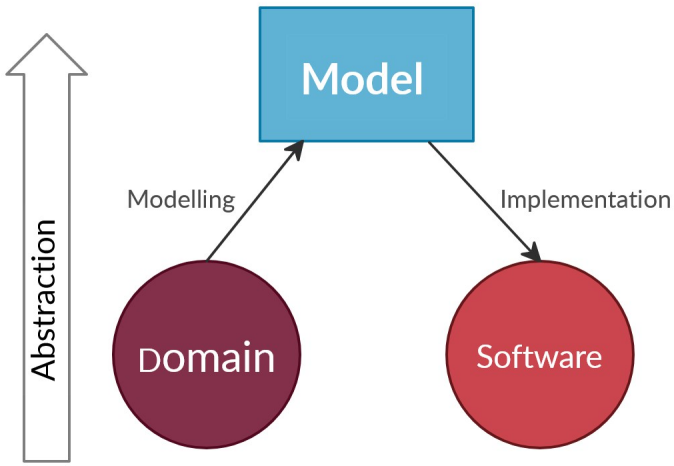
```
	'{'
		('entityPackages' '{' (entitypackages+=Entitypackage)* '}')?
		('entities' '{' (entities+=Entity)* '}')?
		('dataTypes' '{' (datatypes+=Datatype)* '}')?
	'}'
;

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 =' (attributerefereced +=[Attribute|QualifiedName]| id?='ID') (attributerefereced
+=[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)

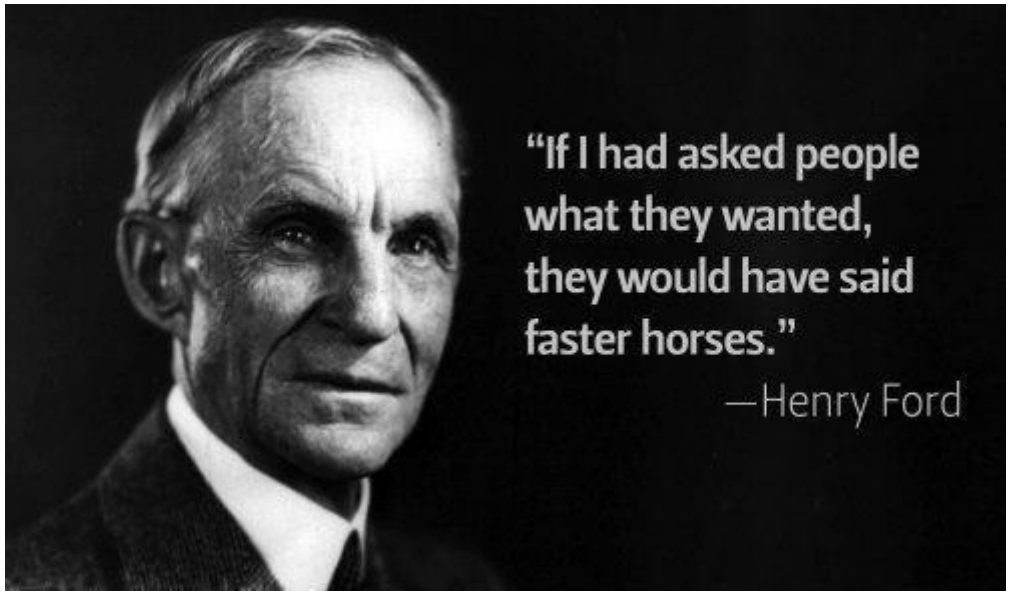
Translation: a language is more expressive in a domain
if you can mostly write shorter programs with it to express the same things.

Educational / research project

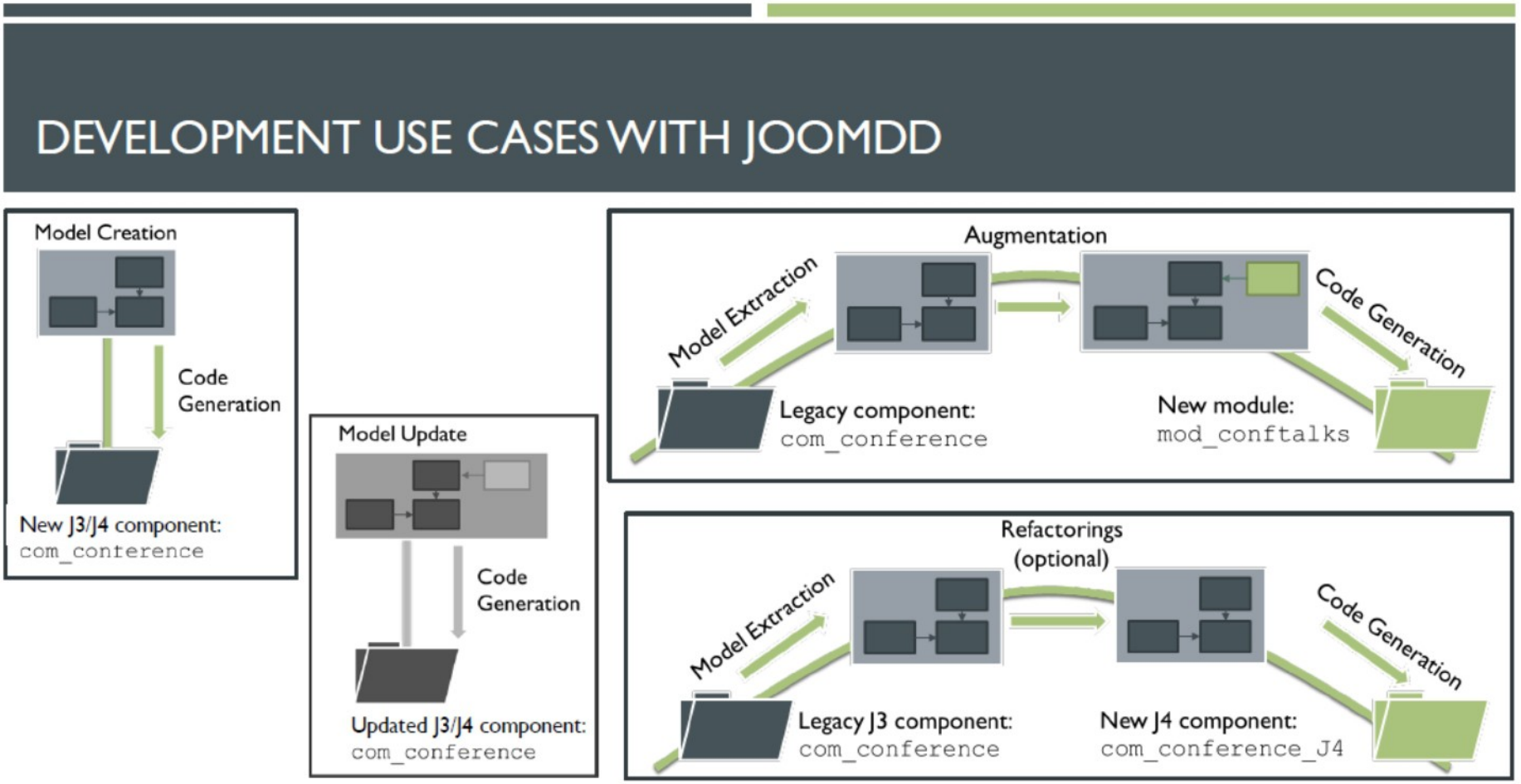
What the "customer" needs:

- **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.

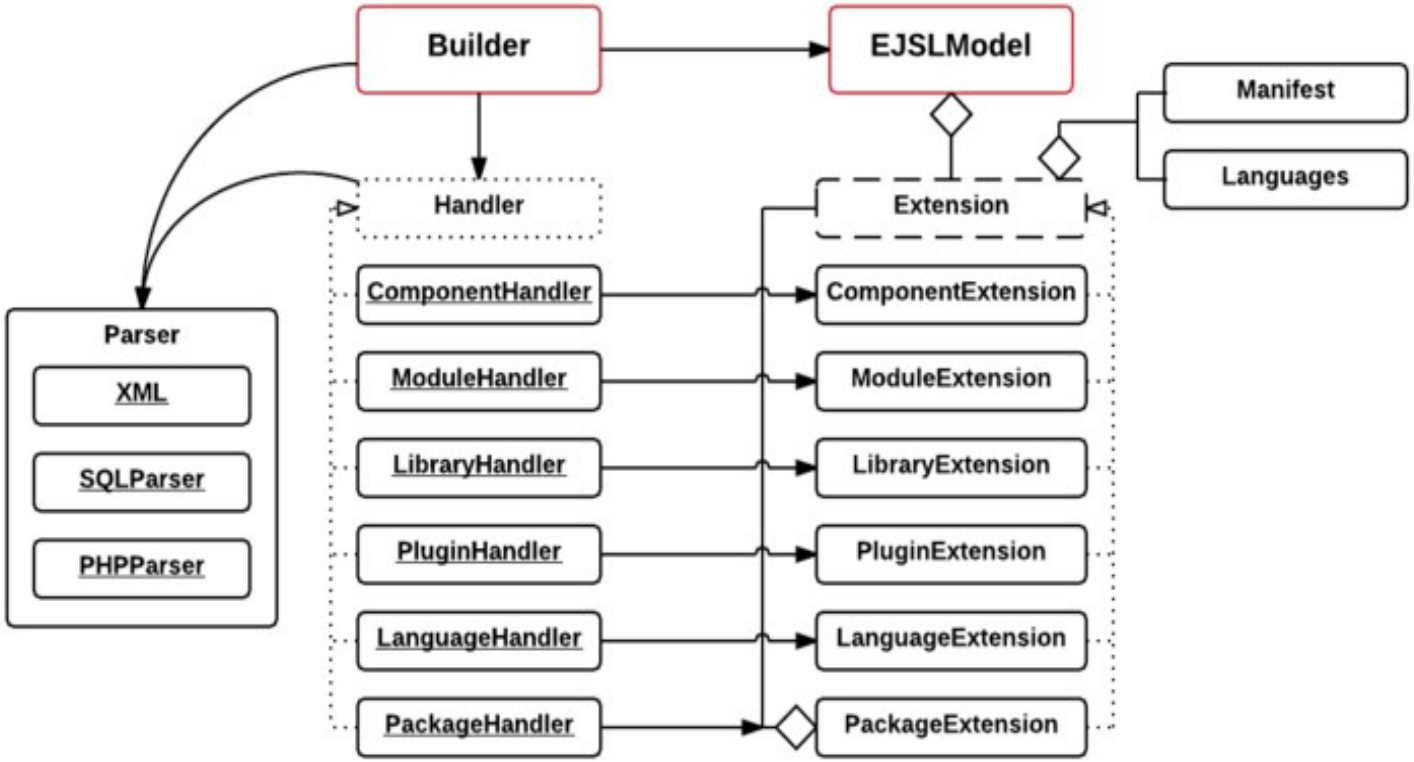


Reverse engineering



Joomla Extensions --> jext2ejsl --> eJSL Model

- **Built parsers for PHP and SQL in Scala**
- 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