

# Projectional Forms

Using HTML forms as model input interface

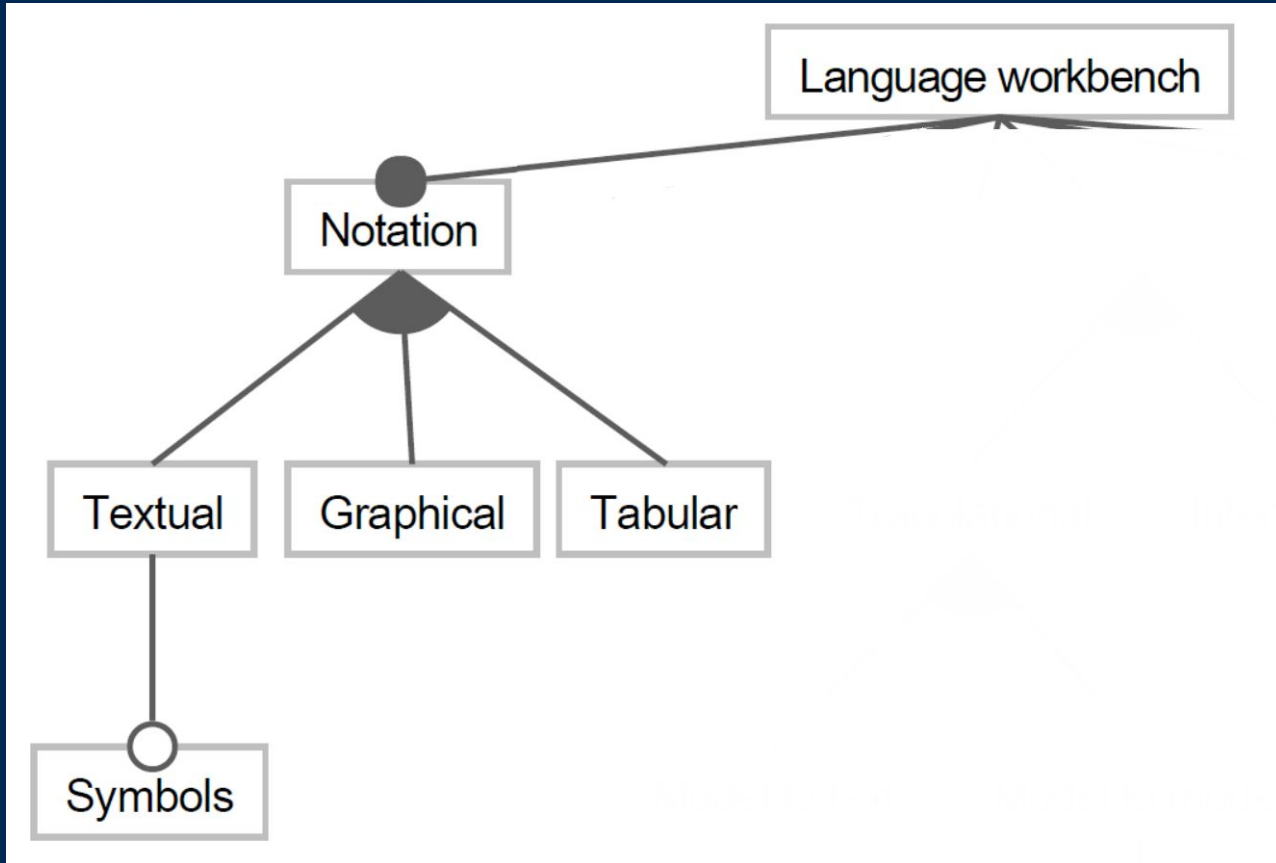
LangDev Meetup 2023  
Herman Peeren

```

        'role_id' => $role_details['id'],
        'resource_id' => $resource_details['id'],
    );
    if ( $this->rule_exists( $resource_details['id'], $role_details['id'] ) ) {
        if ( $access == false ) {
            // Remove the rule as there is currently no need for it
            $details['access'] = !$access;
            $this->_sql->delete( 'acl_rules', $details );
        } else {
            // Update the rule with the new access value
            $this->_sql->update( 'acl_rules', array( 'access' => $access ) );
        }
    }
    foreach( $this->rules as $key=>$rule ) {
        if ( $details['role_id'] == $rule['role_id'] && $details['resource_id'] == $rule['resource_id'] ) {
            if ( $access == false ) {
                unset( $this->rules[ $key ] );
            } else {
                $this->rules[ $key ]['access'] = $access;
            }
        }
    }
}

```

Code: textual input



User interfaces matter



Quest for online language workbenches

EXAMPLE

**Name**

**Email**

**Age**

**Country**

**Password**

**Resume**  Geen bestand geselecteerd.

**Hobbies** ☒ **Cricket** ☐ **Football**


**Gender** ☐ **Female** ☐ **Male**

**City**  ▾

**Address**

HTML form input interface

8 1/2 x 11 in.



WHOSE MUSEUM  
WWW.WHOSAMUSEUM.ORG

WHOSE MUSEUM INVITES YOU TO CONTRIBUTE TO THE COLLECTION!  
SIMPLY FILL OUT THIS FORM AND PASS IT IN WITH YOUR SUBMISSION.  
WE WILL MAKE SURE YOUR WORK IS EXHIBITED AND CARED FOR.  
THANK YOU.

**ABOUT YOU:**  
NAME: \_\_\_\_\_  
YEAR OF BIRTH: \_\_\_\_\_  
PLACE OF BIRTH: \_\_\_\_\_  
ADDRESS: \_\_\_\_\_  
EMAIL/PHONE: \_\_\_\_\_  
INTERESTS: \_\_\_\_\_

**ABOUT YOUR SUBMISSION:**  
TITLE/DATE: \_\_\_\_\_  
MATERIALS (WHAT IS IT MADE OF?): \_\_\_\_\_  
WHERE DID IT COME FROM? \_\_\_\_\_

WHO MADE IT? \_\_\_\_\_  
CURRENT LEGAL OWNER: \_\_\_\_\_  
PREVIOUS OWNERS? \_\_\_\_\_  
DO YOU AGREE TO GRANT THE RIGHT TO WHOSE MUSEUM TO (CIRCLE ONE)  
CREDIT LINE (HOW SHOULD WE CREDIT YOU FOR EXHIBIT LIST (CIRCLE ONE): \_\_\_\_\_

DO YOU AGREE TO ISSUE COPYRIGHT FOR NON-COMMERCIAL PURPOSES? THIS MEANS  
THE MUSEUM CAN REPRODUCE THE WORK FOR THINGS LIKE PUBLICITY MATERIALS,  
EXHIBITIONS, AND ON THEIR WEBSITE. YES NO (CIRCLE ONE)  
IS THERE A STORY TO YOUR SUBMISSION? (If so, you may use the back of this  
form)

I AM IN AGREEMENT WITH THESE TERMS: \_\_\_\_\_  
SIGNATURE: \_\_\_\_\_ DATE: \_\_\_\_\_

WM2009.052

My special case never fits



Not bad for programming...

## GETTING STARTED

### API

### EXAMPLES

[Getting Started Form](#)

[Basic Form](#)

[Field Types](#)

[Nested Fields](#)

### RECIPES

[FormField.js](#)

[ArrayFormField.js](#)

[FormFooter.js](#)

[Components.js](#)

### GITHUB

# Nested Fields

Firstname

Lastname

Email

Items

Item 0

Delete

Item 1

Delete

Add

Items Object

Title 0

Delete

Description 0

Title 1

Delete

Description 1

Add

```
{
```

Form Dirty

Form Invalid

# Developments in online-forms



- compact definition

*sample\_form.xml* File containing the XML for the form definition

```
<?xml version="1.0" encoding="utf-8"?>
<form>
  <field
    name="message"
    type="text"
    label="Enter message"
    size="40"
    class="inputbox"
    required="true" />
  <field name="email"
    type="email"
    label="Enter email"
    required="true"
    size="40"
    class="inputbox" />
  <field name="telephone"
    type="tel"
    label="Enter telephone number"
    required="true"
    size="40"
    class="inputbox"
    validate="tel" />
</form>
```

## Developments in online-forms

- compact definition
- nested fields

An example XML **field** definition for **single** mode:

```
<field name="field-name" type="subform"  
      formsource="path/to/exampleform.xml"  
      label="Subform Field" description="Subform Field Descripti
```

An example XML **field** definition for **multiple** mode:

```
<field name="field-name" type="subform"  
      formsource="path/to/exampleform.xml" multiple="true"  
      label="Subform Field" description="Subform Field Descripti
```

Example XML of *exampleform.xml*

```
<?xml version="1.0" encoding="UTF-8"?>  
<form>  
  <field name="example_text" type="text" label="Example Text"  
  <field name="example_textarea" type="textarea" label="Exam  
</form>
```

## Developments in online-forms

- compact definition
- nested fields
- custom fields

Type	Description
accessiblemedia	provides modal access to the media manager for insertion of images with upload permissions and a text field for adding a alternative text.
accesslevel	provides a drop down list of viewing access levels.
cachehandler	provides a list of available cache handling options.
calendar	provides a text box for entry of a date. An icon next to the text box provides a link to a calendar also be used to enter the date value.
captcha	provides the use of a captcha plugin.
category	provides a drop down list of categories for an extension.
checkbox	provides a single checkbox to be checked or unchecked
checkboxes	provides unlimited checkboxes that can be used for multi-select.
Chrome Style	provides a list of template chrome style options grouped by template.
color	provides a color picker when clicking the input box.

## Developments in online-forms

- compact definition
- nested fields
- custom fields
- multilingual

```
<field
  name="name"
  type="text"
  label="COM_EXTENGEN_PROJECTFORM_LANGUAGEENTITY_FIELD_NAME_LABEL"
  size="40"
  class="languageEntityName"
/>
```

```
COM_EXTENGEN_PROJECTFORM_LANGUAGEENTITY_LABEL="Language Entity"
COM_EXTENGEN_PROJECTFORM_LANGUAGEENTITY_FIELD_NAME_LABEL="LanguageEntity Name"
COM_EXTENGEN_PROJECTFORM_LANGUAGEENTITY_FIELD_KEY_LABEL="LanguageEntity Key"
COM_EXTENGEN_PROJECTFORM_FEATURE_FIELD_NAME_LABEL="Feature Name"
COM_EXTENGEN_PROJECTFORM_FEATURE_FIELD_KEY_LABEL="Feature Key"
COM_EXTENGEN_PROJECTFORM_ENUMERATION_LITERAL_FIELD_NAME_LABEL="EnumerationLiteral Nam
COM_EXTENGEN_PROJECTFORM_ENUMERATION_LITERAL_FIELD_KEY_LABEL="EnumerationLiteral Key"
COM_EXTENGEN_PROJECTFORM_ENUMERATION_FIELD LITERALS_LABEL="Literals"
COM_EXTENGEN_PROJECTFORM_PROPERTY_FIELD_TYPE_LABEL="Type"
```

## Developments in online-forms

- compact definition
- nested fields
- custom fields
- multilingual
- drag&drop ordering

The screenshot displays a web interface for editing a form titled "Features". At the top left of the form area is a green square button with a white plus sign. At the top right is a link that says "Show/Hide" with an eye icon. Below these are four feature entries, each consisting of a blue square button with a white plus sign, a yellow rectangular text field, and a red square button with a white minus sign. The text fields contain the following text: "Feature Name: name", "Feature Name: id", "Feature Name: fileExtension", and "Feature Name: document". To the right of each yellow field is a green square button with a white plus sign. The bottom-most entry, "Feature Name: document", is highlighted with a thick green border.

Developments in online-forms

- compact definition
- nested fields
- custom fields
- multilingual
- drag&drop ordering
- form validation

```
use Joomla\CMS\Form\FormRule;

/**
 * Form Rule class for the Joomla Platform.
 */
class LetterRule extends FormRule
{
    /**
     * The regular expression to use in testing a form field value.
     *
     * @var string
     */
    protected $regex = '^([a-z]+)$';

    /**
     * The regular expression modifiers to use when testing a form field value
     *
     * @var string
     */
    protected $modifiers = 'i';
}
```

Developments in online-forms

- compact definition
- nested fields
- custom fields
- multilingual
- drag&drop ordering
- form validation
- JSON output

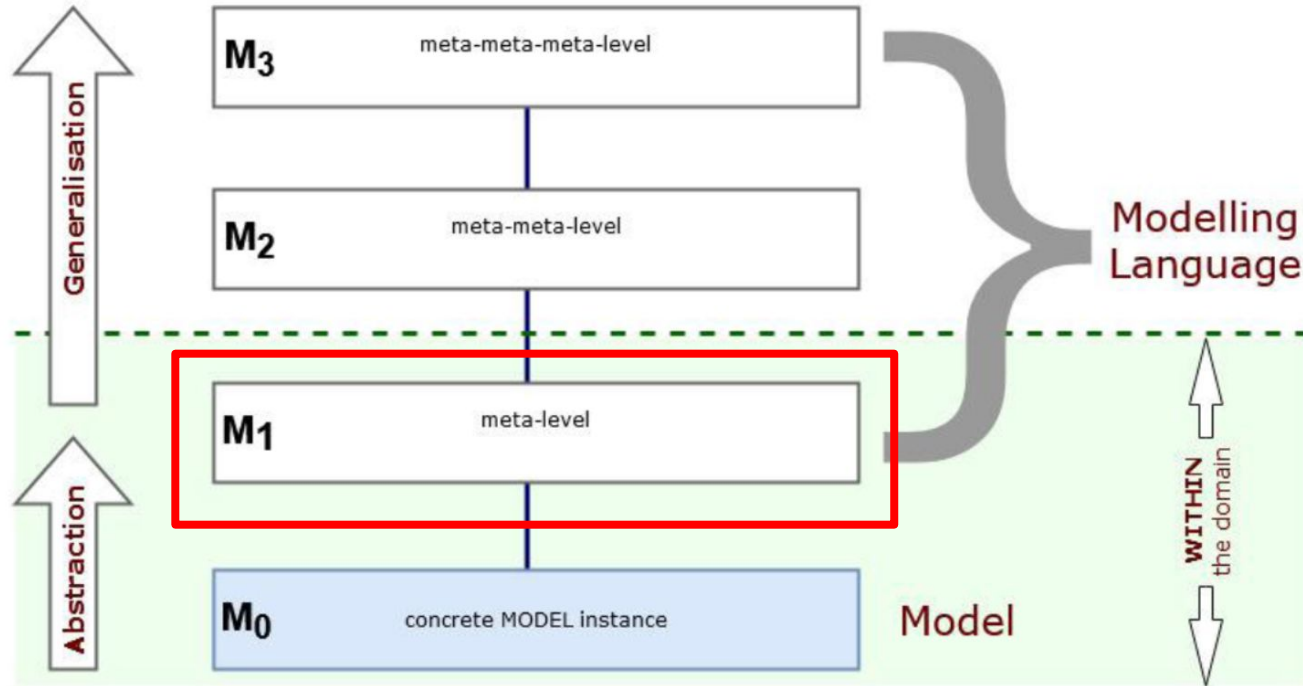
```
},  
"feature2": {  
  "name": "fields",  
  "key": "ER1_Entity_fields",  
  "is_optional": "1",  
  "feature_type": "Link",  
  "property": {  
    "type": "0",  
    "typeReference_key": "",  
    "LIonWeb_key": ""  
  },  
  "link": {  
    "type": "ER1_Field",  
    "is_multiple": "1",  
    "link_type": "Containment",  
    "containment": {  
      "extends": "",  
      "LIonWeb_key": ""  
    }  
  },  
}
```

Developments in online-forms



*Nested fields  
and  
JSON-output  
give  
an AST  
for free.*





**M1**: language to input the model.

https://extensions.joomla.org Zoeken

Joomla!® Download & Extend Discover & Learn Community & Support Developer Resources

Home Browse Extensions Search Community Support Vulnerable Extensions Log in Register

**e-Commerce**
256

**Payment Gateway**
75

**e-Commerce Integrations**
60

**Shopping Cart**
31

**Donations**
23

**Paid Downloads**
13

**Membership & Subscriptions**
12

**Billing & Invoices**
9

**Payment systems**
9

**Gifts & Coupons**
6

View All

**Contacts & Feedback**
207

**Contact forms**
62

**Testimonials & Suggestions**
39

**Forms**
27

**Contact Details**
15

**Email**
13

**Articles comments**
11

**Polls**
11

**Opening Hours**
7

**Surveys**
7

View All

**Core Enhancements**
194

**Performance**
66

**Coding & Scripts Integration**
57

**File Management**
25

**Data Reports**
20

**Libraries**
5

**Multiple Sites**
4

**Installers**
3

**Flash Management**
2

**Authoring & Content**
191

**Content Construction**
63

**Custom Fields**
58

**Content Submission**
12

**Article Preview**
9

**Timed Content**
9

**Blog**
8

**Admin Publishing**
4

**Archive**
3

**Microblogging**
2

**Structure & Navigation**
177

**Menu Systems**
102

**Site Navigation**
38

**Site Map**
16

**Content Links**
14

**Sections & Categories**
5

**Multi Categorization**
2

**Communication**
156

**Chat**
39

**Live Support**
33

**Live Support - hosted**
22

**Forum**
11

**Video conference**
8

**Question & Answers**
7

**Forum Bridges**
5

**Instant Messaging**
5

**Online Status**
5

MDE-project: extensions for Joomla 3, 4 and 5.



I<sup>2</sup>

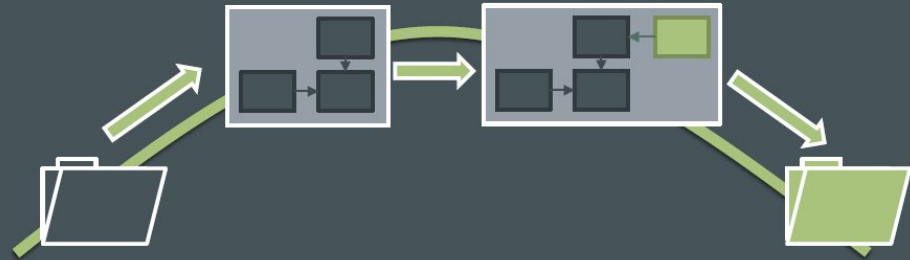
Institut für  
Informationswissenschaften

# CREATE A JOOMLA 4 COMPONENT WITH JOOMDD (JOOMLA 4 COMPONENT BOUWEN MET JOOMDD)

DENNIS PRIEFER, WOLF ROST

JDNL '18, EINDHOVEN, THE NETHERLANDS

1. CHALLENGES
2. JOOMDD
3. DEVELOPMENT USE CASES WITH JOOMDD
4. DEMO



JooMDD, an XText-project.

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')?
'}'
;
```

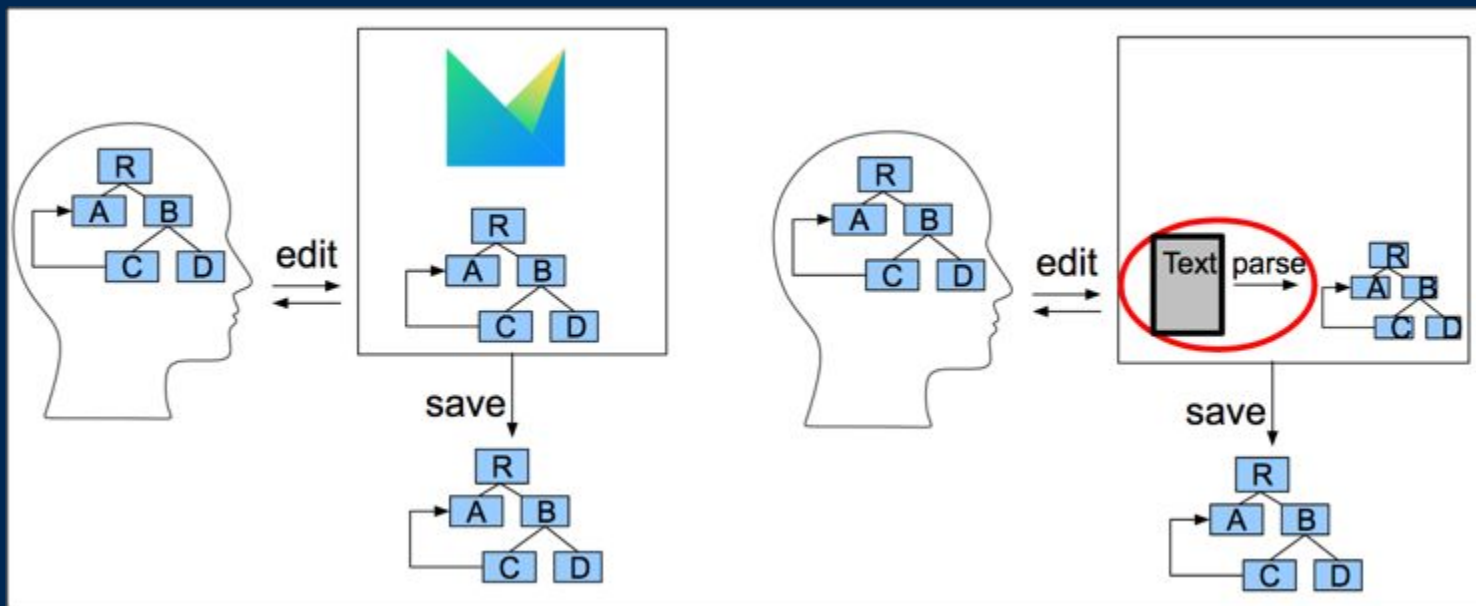
So, based on a DSL-grammar (eJSL).

```

1 eJSLModel "Conference" {
2     eJSL part: CMS Extension {
3         entities {
4             Entity Participant {
5                 attributes {
6                     Attribute name {
7                         type = Short_Text
8                     }
9                     Attribute affiliation {
10                        type = Text
11                    }
12                    Attribute nationality {
13                        type = Text
14                    }
15                    Attribute address {
16                        type = Text
17                    }
18                }
19            }
20            Entity Talk {
21                attributes {
22                    Attribute title {

```

Resulting in an Eclipse editor, generating Joomla 3 & 4 code.



I've ported JooMDD to MPS: projectional editing.

The screenshot displays the MPS IDE interface. On the left, the 'Logical View' pane shows a project tree for 'eJSL-MPS'. The 'Entity' concept is selected under the 'yepr.mps.ejsl' package. The right pane shows the definition of the 'Entity' concept, which extends 'BaseConcept' and implements 'IIdentifierNamedConcept'. The definition includes properties like 'instance can be root: false', 'alias: <no alias>', and 'short description: <no short description>'. It also defines 'attributes' and 'references' as collections of 'Attribute' and 'Reference' respectively.

```
concept Entity extends BaseConcept
    implements IIdentifierNamedConcept

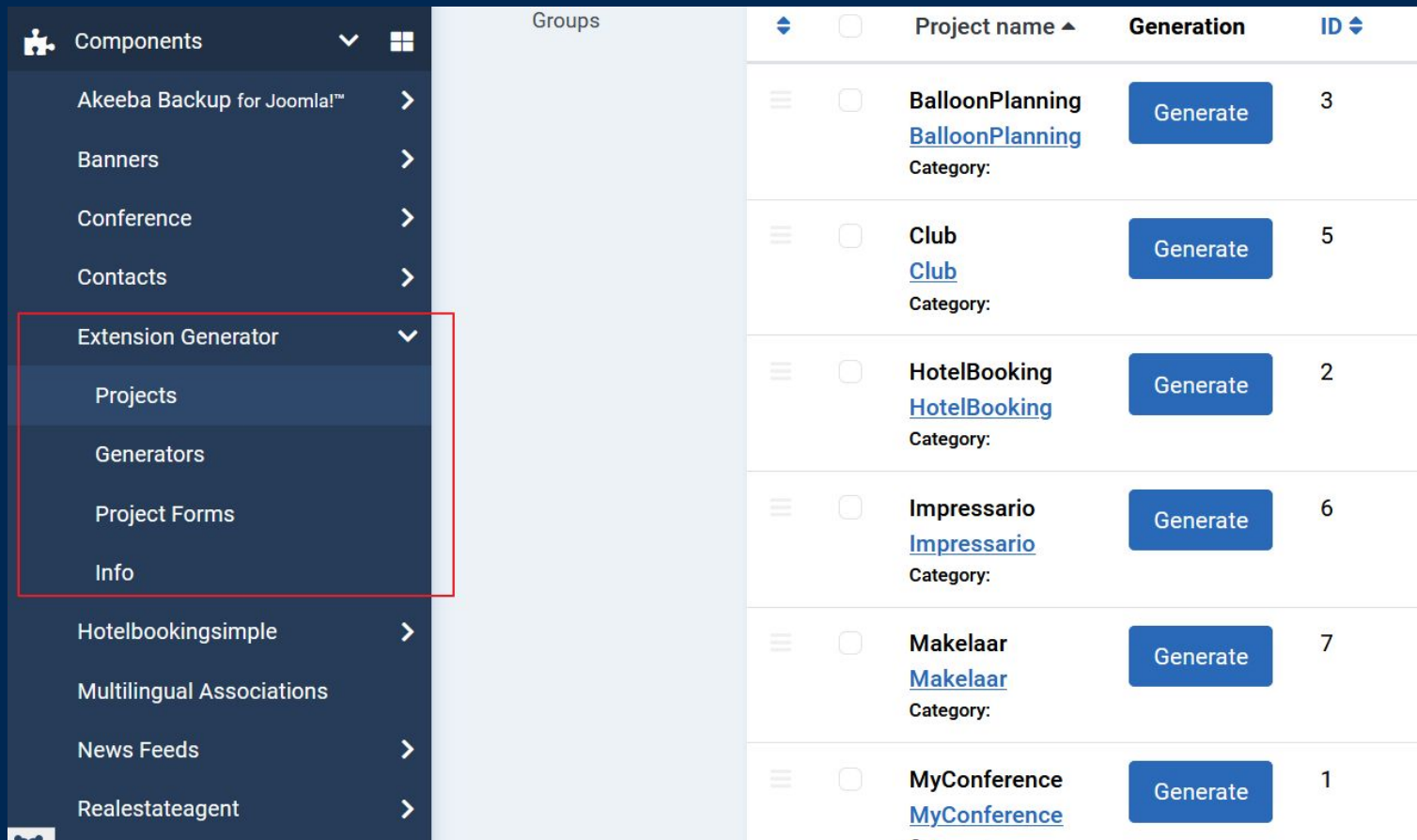
    instance can be root: false
    alias: <no alias>
    short description: <no short description>

    properties:
    << ... >>

    children:
    attributes : Attribute[0..n]
    references : Reference[0..n]

    references:
    << ... >>
```

So, defining concepts in MPS

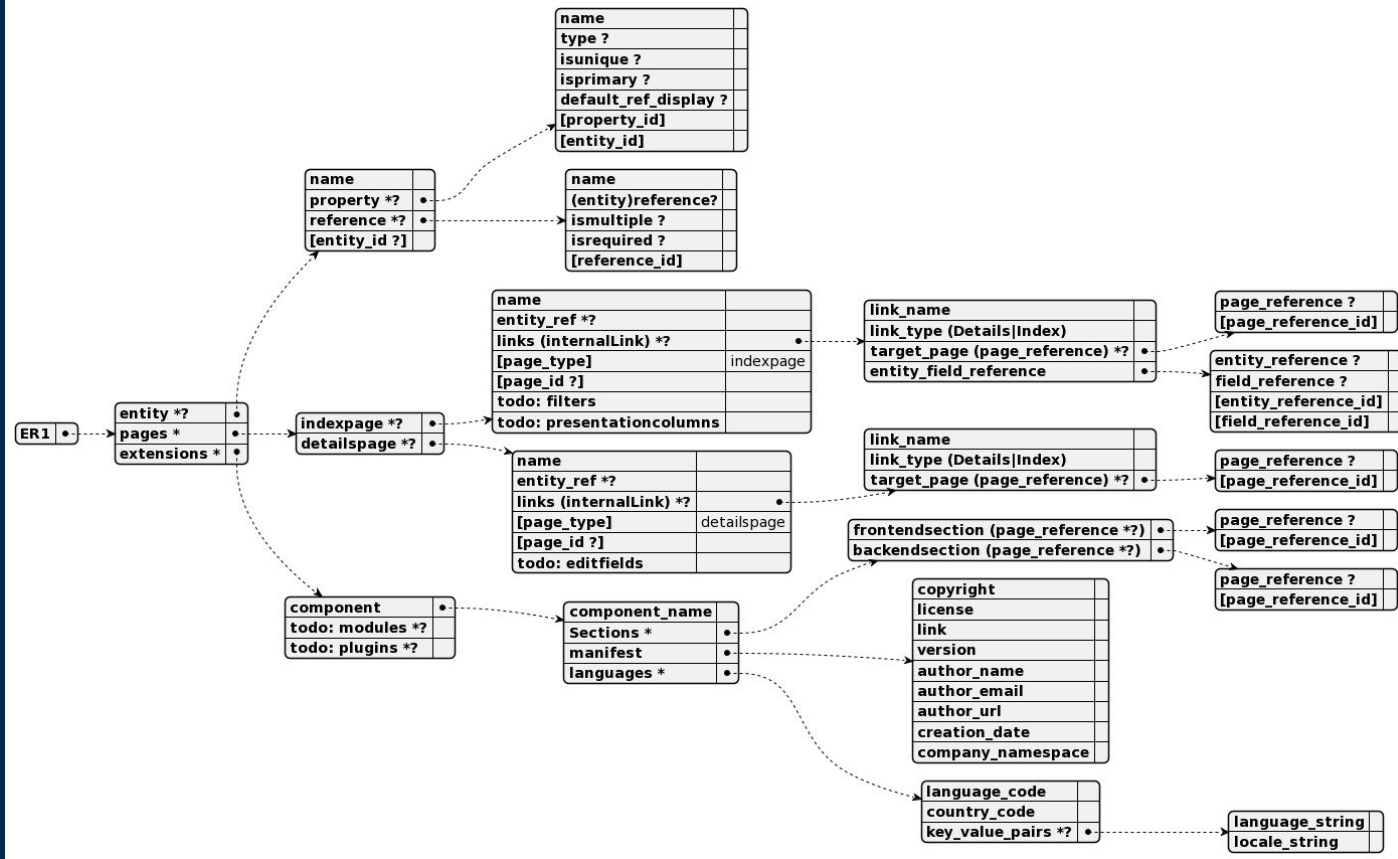


The image shows a Joomla! interface. On the left, the 'Components' menu is open, with the 'Extension Generator' sub-menu highlighted by a red box. The sub-menu items are 'Projects', 'Generators', 'Project Forms', and 'Info'. On the right, the 'Groups' table is displayed, showing a list of projects with their names, links, categories, and IDs. Each row has a 'Generate' button.

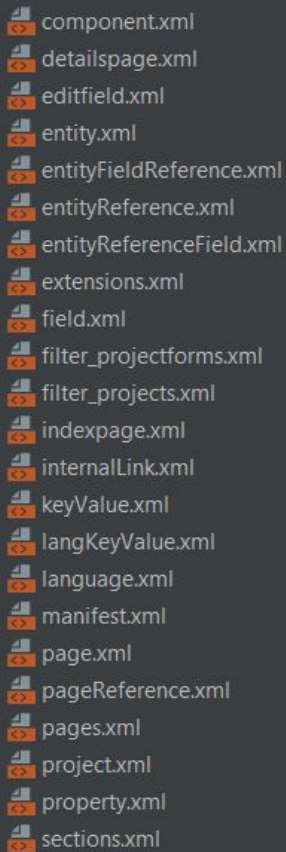
		Project name ▲	Generation	ID ◆
☰	<input type="checkbox"/>	BalloonPlanning <a href="#">BalloonPlanning</a> Category:	<button>Generate</button>	3
☰	<input type="checkbox"/>	Club <a href="#">Club</a> Category:	<button>Generate</button>	5
☰	<input type="checkbox"/>	HotelBooking <a href="#">HotelBooking</a> Category:	<button>Generate</button>	2
☰	<input type="checkbox"/>	Impressario <a href="#">Impressario</a> Category:	<button>Generate</button>	6
☰	<input type="checkbox"/>	Makelaar <a href="#">Makelaar</a> Category:	<button>Generate</button>	7
☰	<input type="checkbox"/>	MyConference <a href="#">MyConference</a>	<button>Generate</button>	1

And finally to Extension Generator (a Joomla component).





So, based on (nested) forms.



- component.xml
- detailspage.xml
- editfield.xml
- entity.xml
- entityFieldReference.xml
- entityReference.xml
- entityReferenceField.xml
- extensions.xml
- field.xml
- filter\_projectforms.xml
- filter\_projects.xml
- indexpage.xml
- internalLink.xml
- keyValue.xml
- langKeyValue.xml
- language.xml
- manifest.xml
- page.xml
- pageReference.xml
- pages.xml
- project.xml
- property.xml
- sections.xml

Forms defined in xml-files.

SaveSave & CloseClose

Help

Project name \*MyConference

EntitiesPagesExtensions

Entity Relationship Diagram

Entities+Show/Hide

Entity nameParticipant

Fields+Show/Hide

Field namename

Property or reference?

Property

Reference

Is this field native property or a reference to another entity?

Property

TypeShort Text

Is unique?☐

Resulting in forms to input the model

Project name \*

**Entities** Pages Extensions ←

Entity Relationship Diagram

Entities +

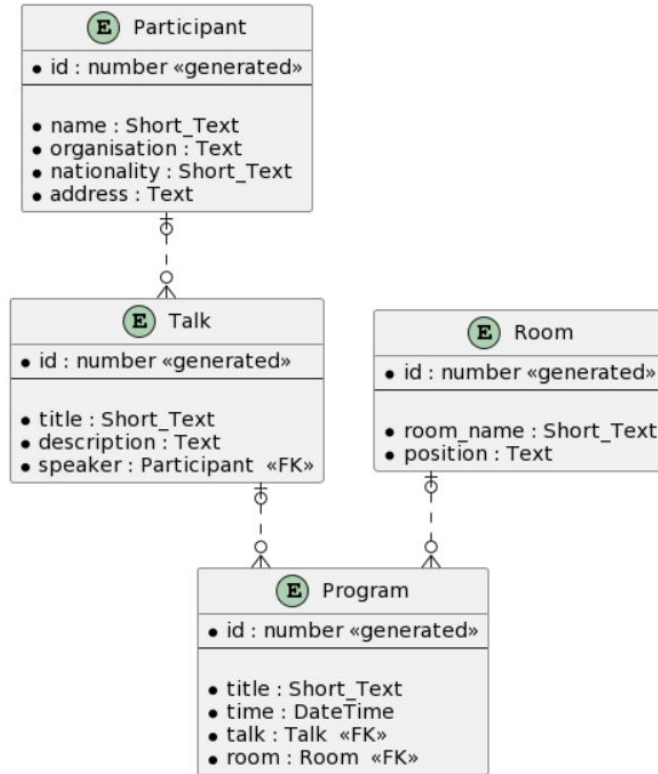
Entity name \*

Properties

Property name *	Type	Is unique?	Is primary?	Default ref display?
<input type="text" value="name"/>	Short Text <input type="button" value="v"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="text" value="affiliation"/>	Text <input type="button" value="v"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="nationality"/>	Short Text <input type="button" value="v"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="address"/>	Text <input type="button" value="v"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Model (project) in 3 parts: data, interaction, extensions.

## Entity Relationship Diagram for MyConference



Example in JooMDD: conference.

Field name

speaker

Property or reference?

☐ Property

☒ Reference

Is this field native property or a reference to another entity?

Reference

Reference

Participant

Multiple?

☐

Not yet available in this version! See roadmap for more information.

Required?

☐

At least 1 item required?

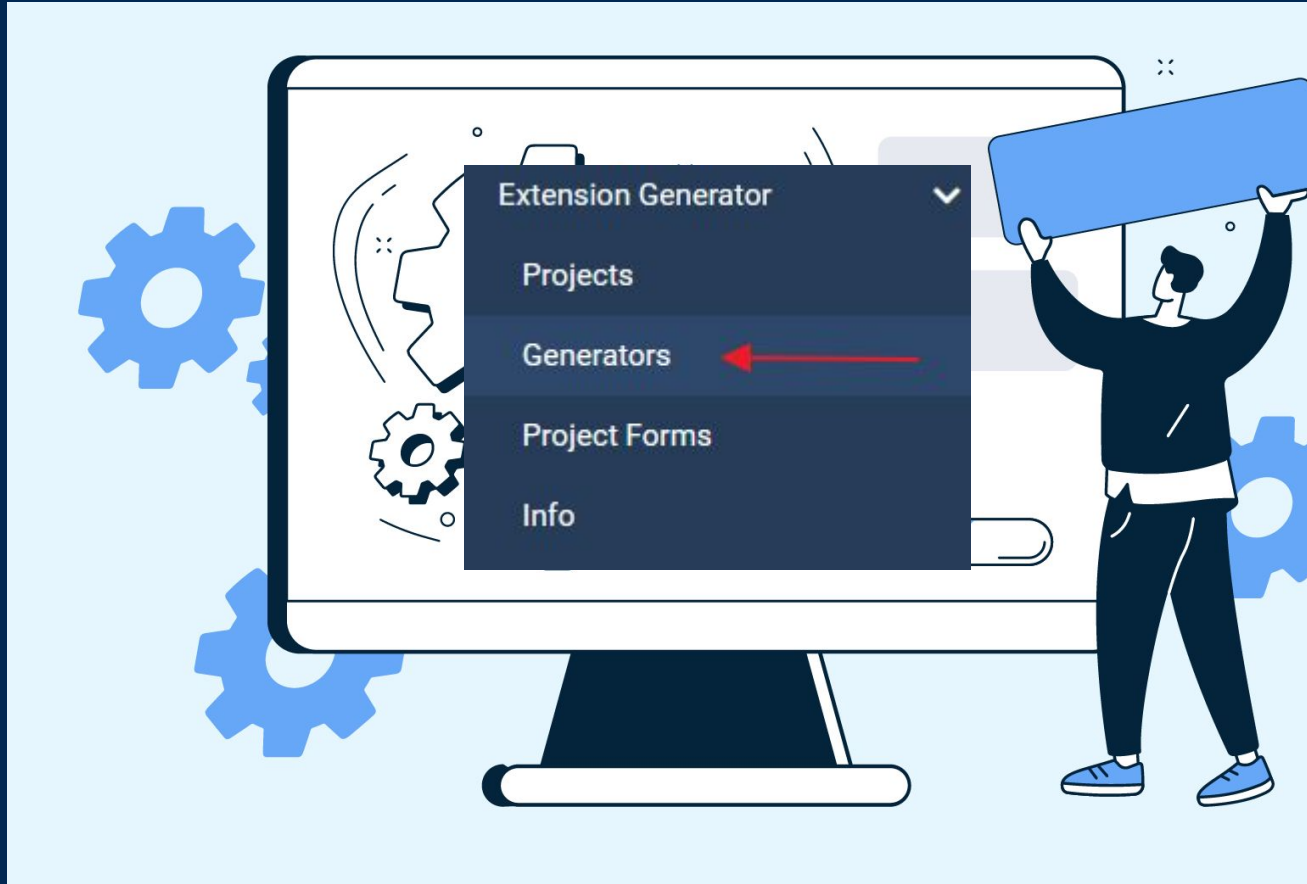
Conference example: a speaker references to a Participant

```

155     public function save($data)
156     {
157         {% if foreign|length > 0 %}
158             // Add foreign key column names
159             {% for reference in foreign %}
160                 $data['{{ reference.columnName }}'] = $data['{{ reference.fieldName }}'];
161             {% endfor %}
162         {% endif %}
163
164         return parent::save($data);
165     }
166
167     // Override getTable to be sure the right table name is used (especially when page name is dif
168     // name).
169     public function getTable($name = '{{ entityName }}', $prefix = '', $options = [])
170     {
171         return parent::getTable($name, $prefix, $options);
172     }
173 }

```

Generator is mostly mapping of AST to (Twig)templates.



Mapping is now still (PHP-)scripted, will be forms-based



## Generation log

### === BACK-END ===

access.xml generated  
config.xml generated  
LICENSE.txt generated  
conference.xml generated  
provider.php generated  
ConferenceComponent.php generated  
generated install.mysql.utf8.sql sql-file  
generated uninstall.mysql.utf8.sql sql-file  
generated CREATE TABLE sql statement for #\_\_conference\_participant in sql-file  
ParticipantTable.php generated  
generated CREATE TABLE sql statement for #\_\_conference\_talk in sql-file  
TalkTable.php generated  
generated CREATE TABLE sql statement for #\_\_conference\_room in sql-file  
RoomTable.php generated  
generated CREATE TABLE sql statement for #\_\_conference\_program in sql-file  
ProgramTable.php generated  
ParticipantsController.php generated  
ParticipantsModel.php generated  
HtmlView.php generated  
default.php generated  
ParticipantController.php generated  
ParticipantModel.php generated  
HtmlView.php generated  
edit.php generated  
TalksController.php generated  
TalksModel.php generated  
HtmlView.php generated  
default.php generated  
TalkController.php generated  
TalkModel.php generated  
HtmlView.php generated

edit.php generated

DisplayController.php generated

### === FORM FILES ===

participant.xml generated

talk.xml generated

room.xml generated

session.xml generated

### === LANGUAGE FILES ===

administrator/components/com\_conference/language/en-GB/com\_conference.ini generated

administrator/components/com\_conference/language/nl-NL/com\_conference.ini generated

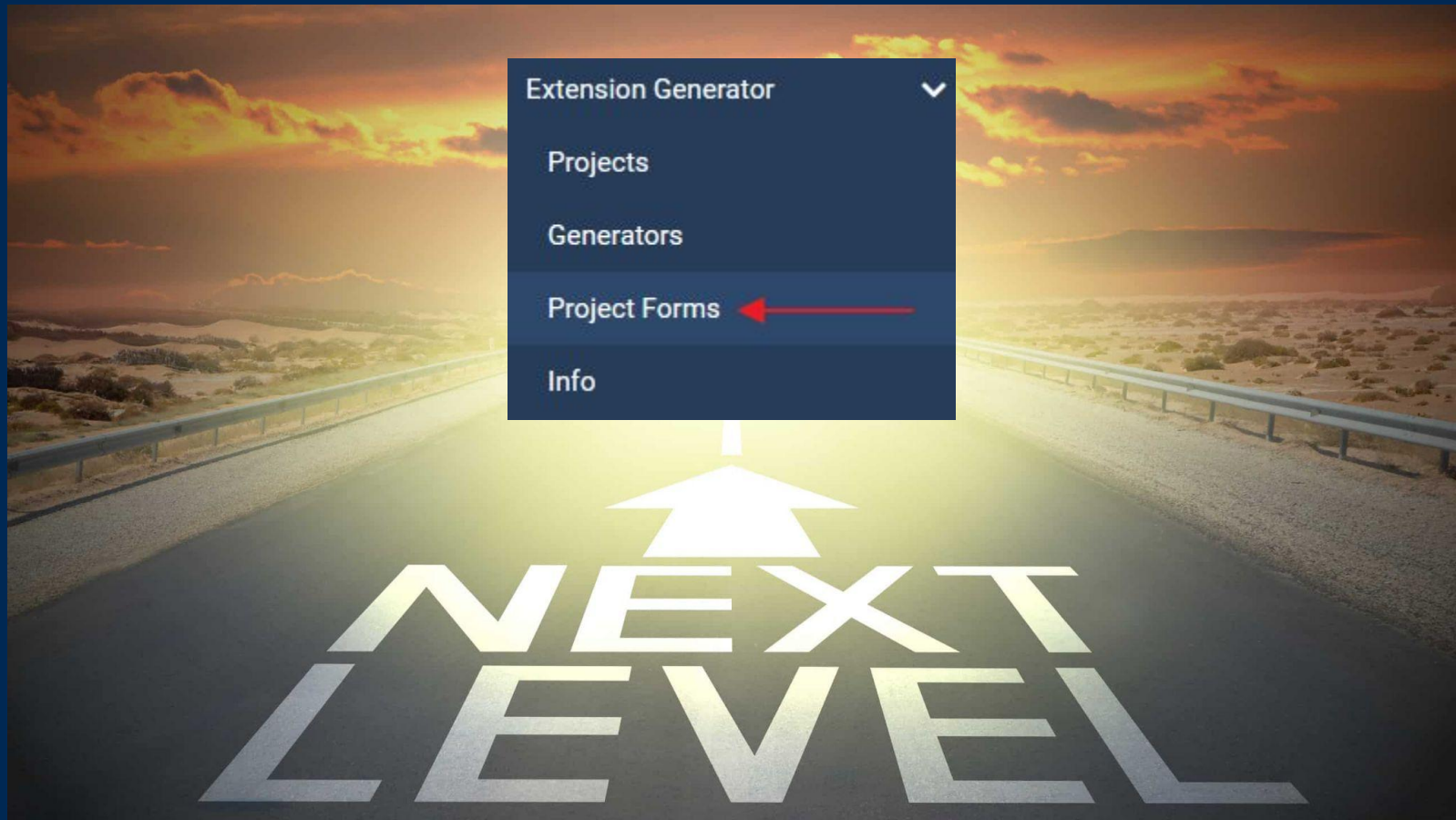
administrator/components/com\_conference/language/en-GB/com\_conference.sys.ini generated

administrator/components/com\_conference/language/nl-NL/com\_conference.sys.ini generated

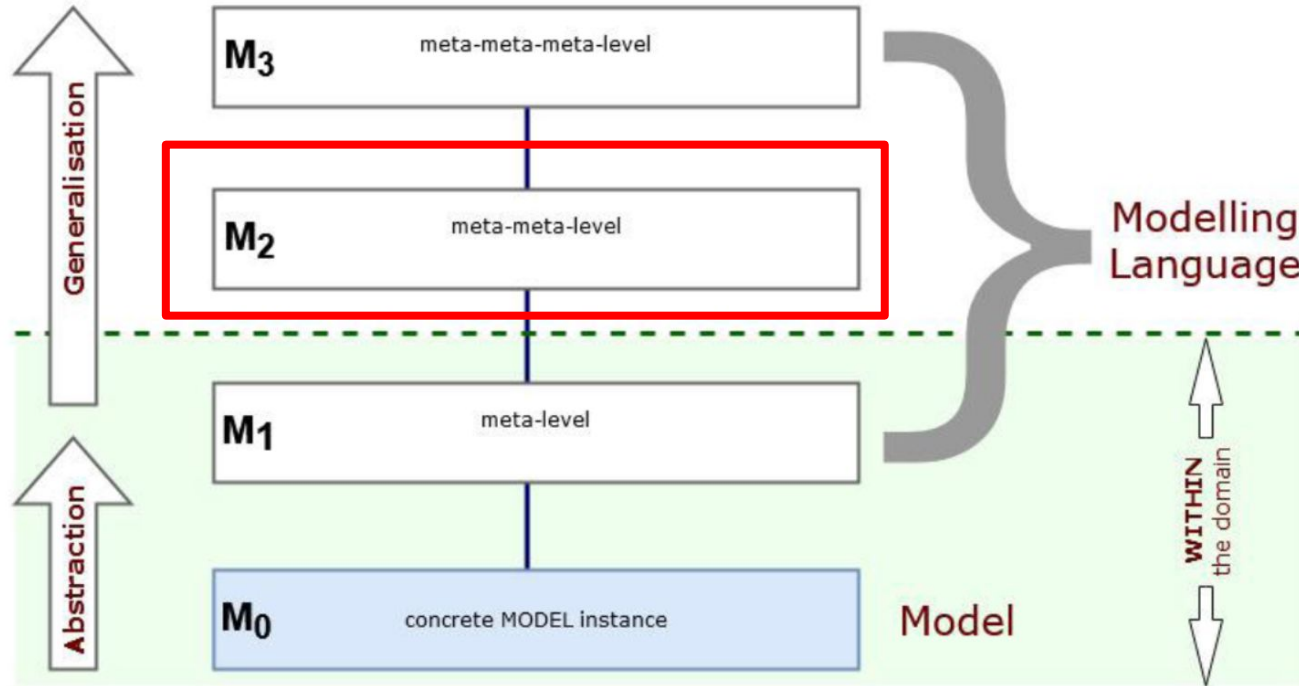
components/com\_conference/language/en-GB/com\_conference.ini generated

components/com\_conference/language/nl-NL/com\_conference.ini generated

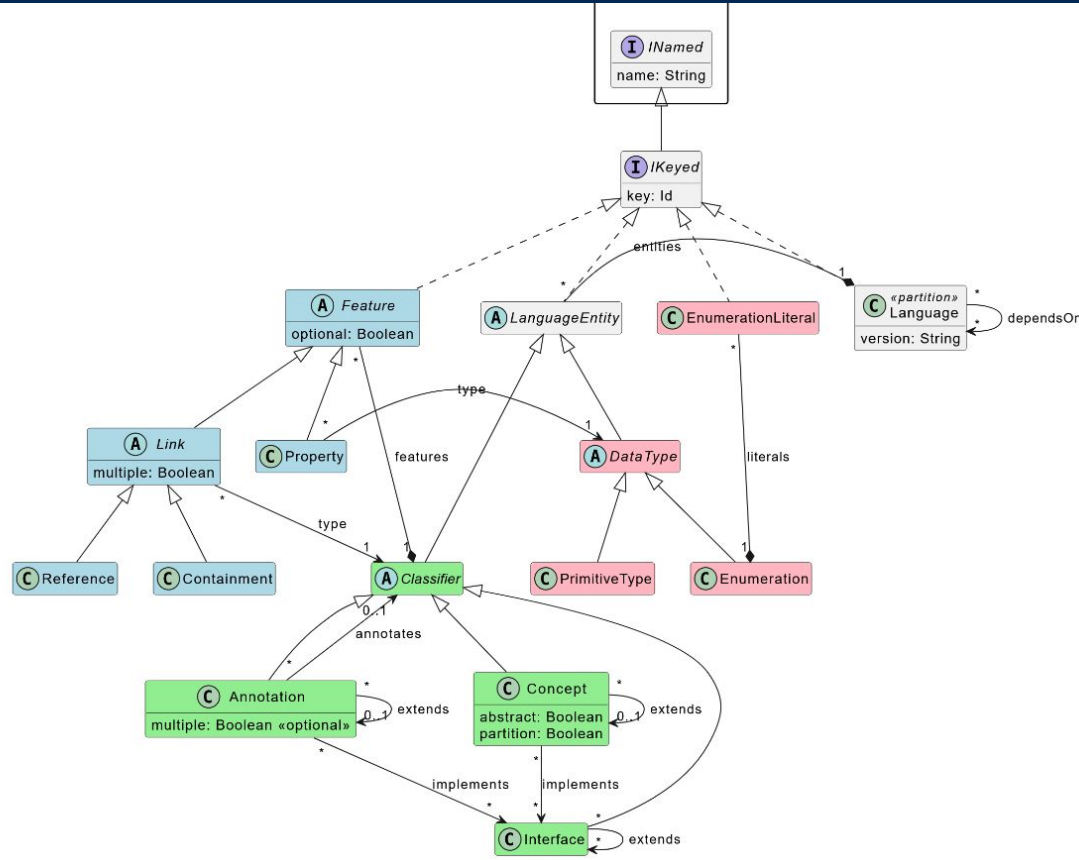
# Files for the Joomla extension are generated.



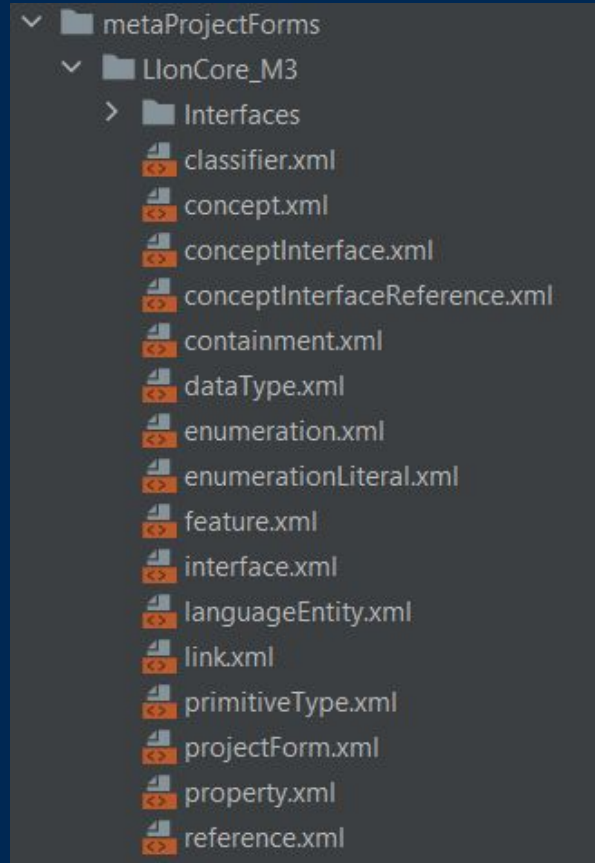
Project forms can be defined with forms!



**M<sub>2</sub>:** language to input the model  
for the language to input the model (meta-meta-level)



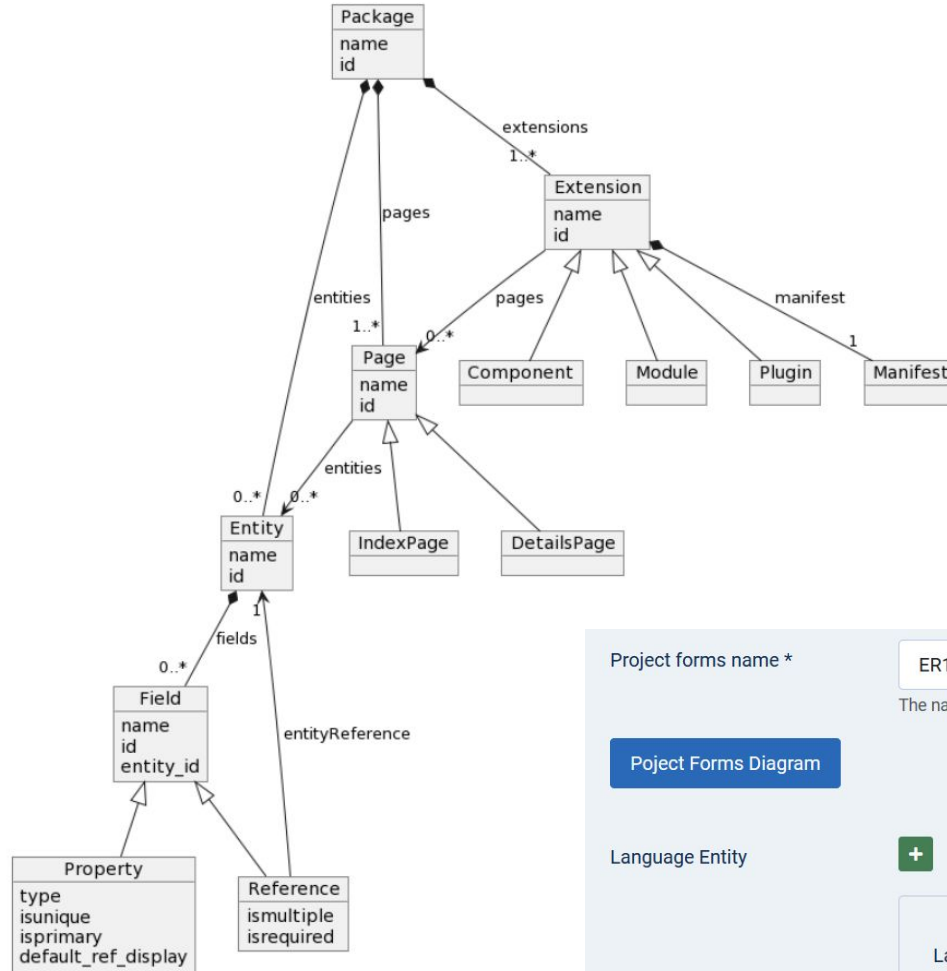
I use LionCore\_M3 as M3-language



Forms defined in xml-files

## Project Form Diagram for ER1

(my eJSL-implementation)



Project forms name \*

ER1

The name of the forms that are used to model the project.

Project Forms Diagram

Language Entity



LanguageEntity Name

Package

LanguageEntity Name

XmlDocument

LanguageEntity Key

XmlDocument

Type of Language Entity?

☒ Classifier

☐ DataType

Classifier or DataType?

Classifier

Classifier Type

☒ Concept

☐ ConceptInterface

☐ Annotation

A Classifier is either a Concept, a ConceptInterface or an Annotation

Concept

Concept is abstract?

☐

Concept can be root?

☐

Extends

Implements

+

Show/Hide

Features

+

Show/Hide

Feature Name

prolog

Feature Key

XmlDocument-prolog

Optional?

☒

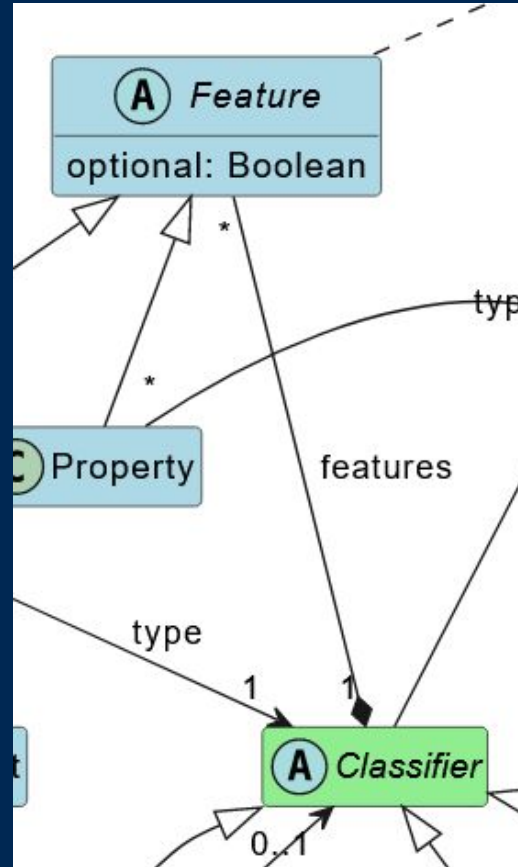
Can we have zero of this?

Property or link?

☐ Property

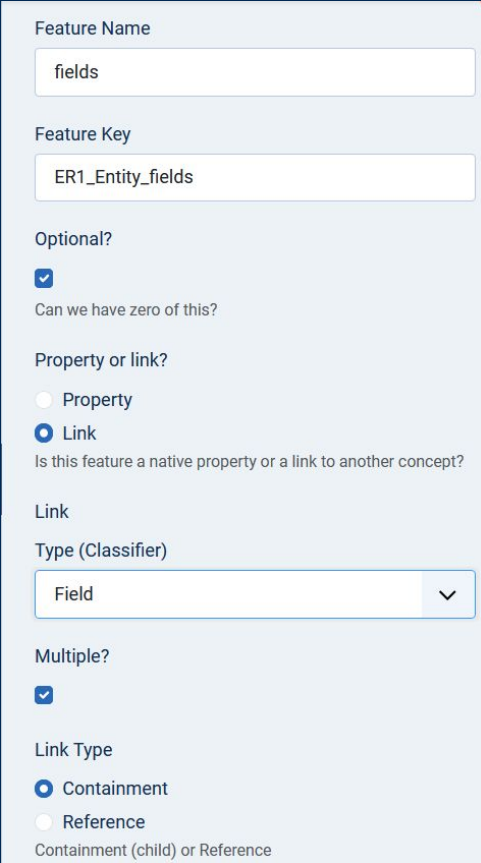
Implementation: from M2-AST to M1-form

- feature / containment





- feature / containment:  
subform



A screenshot of a web form for configuring a feature. The form is titled 'Feature Name' and contains several sections: 'Feature Name' with a text input field containing 'fields'; 'Feature Key' with a text input field containing 'ER1\_Entity\_fields'; 'Optional?' with a checked checkbox; 'Can we have zero of this?' with a text input field; 'Property or link?' with radio buttons for 'Property' and 'Link' (selected); 'Is this feature a native property or a link to another concept?' with a text input field; 'Link' with a text input field; 'Type (Classifier)' with a dropdown menu showing 'Field'; 'Multiple?' with a checked checkbox; 'Link Type' with radio buttons for 'Containment' (selected) and 'Reference'; and 'Containment (child) or Reference' with a text input field.

Feature Name

fields

Feature Key

ER1\_Entity\_fields

Optional?

☒

Can we have zero of this?

Property or link?

☐ Property

☒ Link

Is this feature a native property or a link to another concept?

Link

Type (Classifier)

Field

Multiple?

☒

Link Type

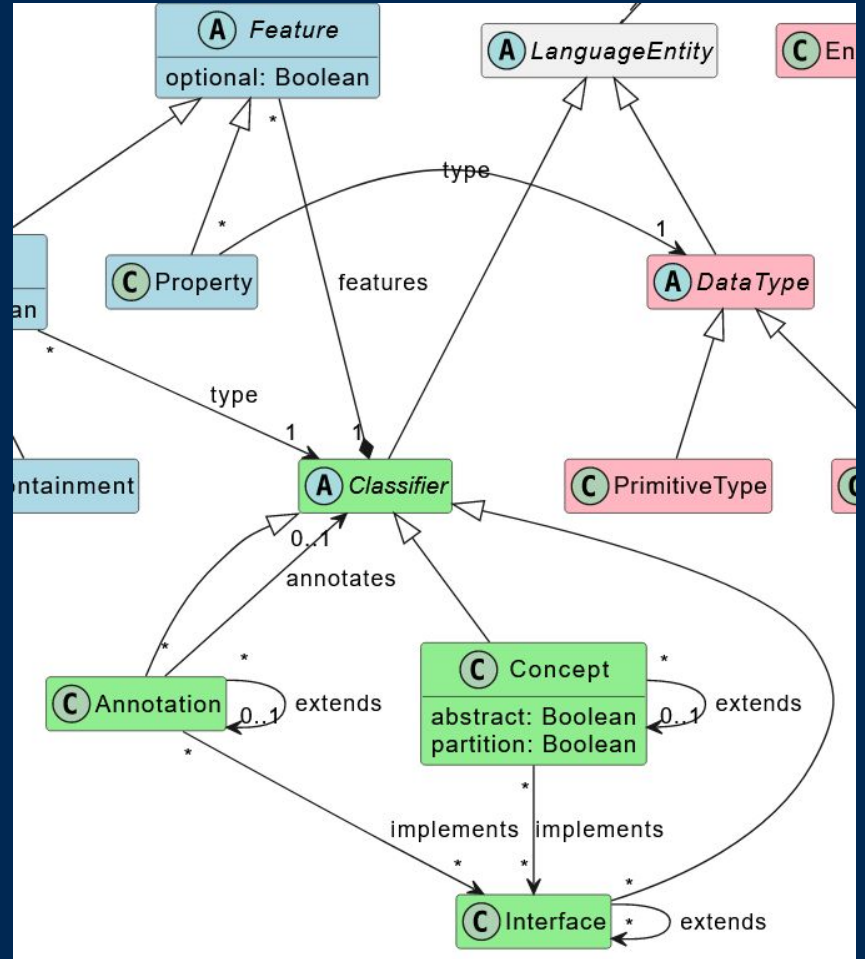
☒ Containment

☐ Reference

Containment (child) or Reference

Implementation: subform always visible

- feature / containment:  
subform
- inheritance



- feature / containment:  
subform
- inheritance:  
switch + subform show

LanguageEntity Key	ER1_Entity
Type of Language Entity?	<input checked="" type="radio"/> Classifier <input type="radio"/> DataType Classifier or DataType?
Classifier	Classifier Type <input checked="" type="radio"/> Concept <input type="radio"/> ConceptInterface <input type="radio"/> Annotation A Classifier is either a Concept, a ConceptInterface or an Annotation
	Concept Concept is abstract? <input type="checkbox"/>
	Concept can be root? <input type="checkbox"/>
	Extends <div></div>
	Implements <div>+</div>
	Features

Show/Hide

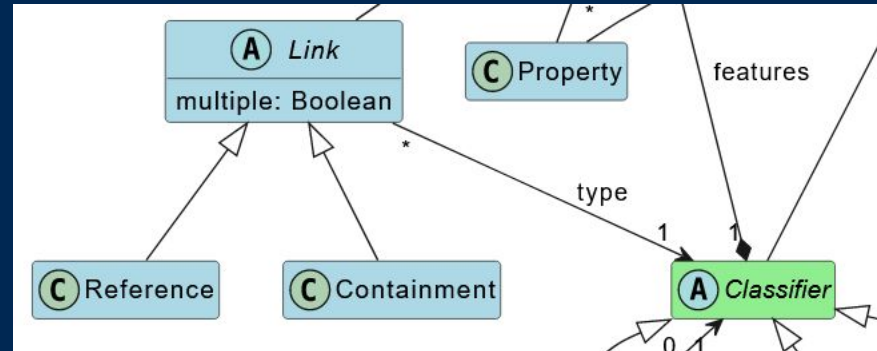
Implementation: top-down in hierarchy, no substitution

- feature / containment:  
subform
- inheritance:  
switch + subform show

LanguageEntity Key	ER1_Entity
Type of Language Entity?	<input checked="" type="radio"/> Classifier <input type="radio"/> DataType Classifier or DataType?
Classifier	Classifier Type <input type="radio"/> Concept <input checked="" type="radio"/> ConceptInterface <input type="radio"/> Annotation A Classifier is either a Concept, a ConceptInterface or an Annotation
	ConceptInterface
	Extends <input type="text"/>
	Features

Implementation: top-down in hierarchy, no substitution

- feature / containment:  
subform
- inheritance:  
switch + subform show
- link / reference

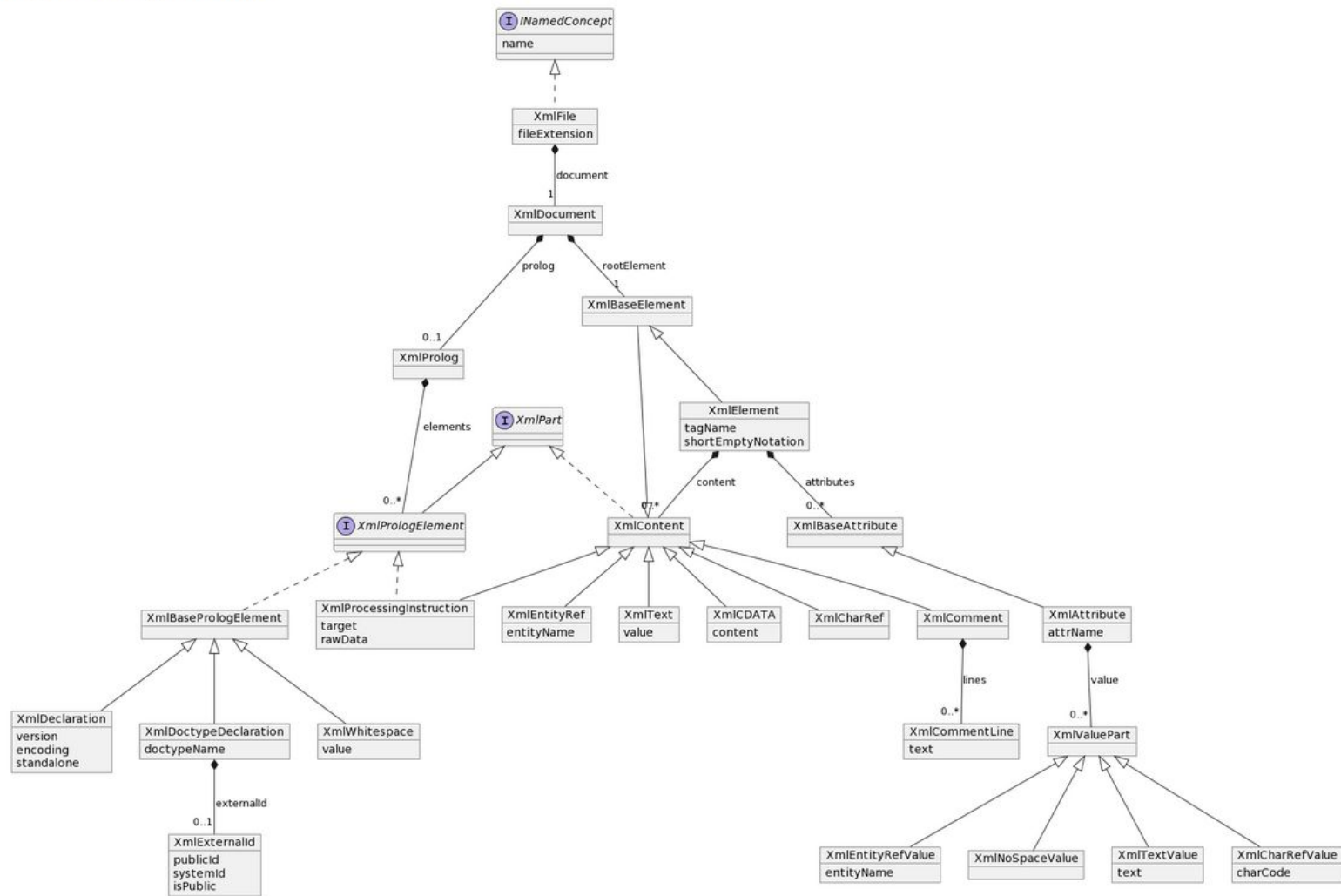


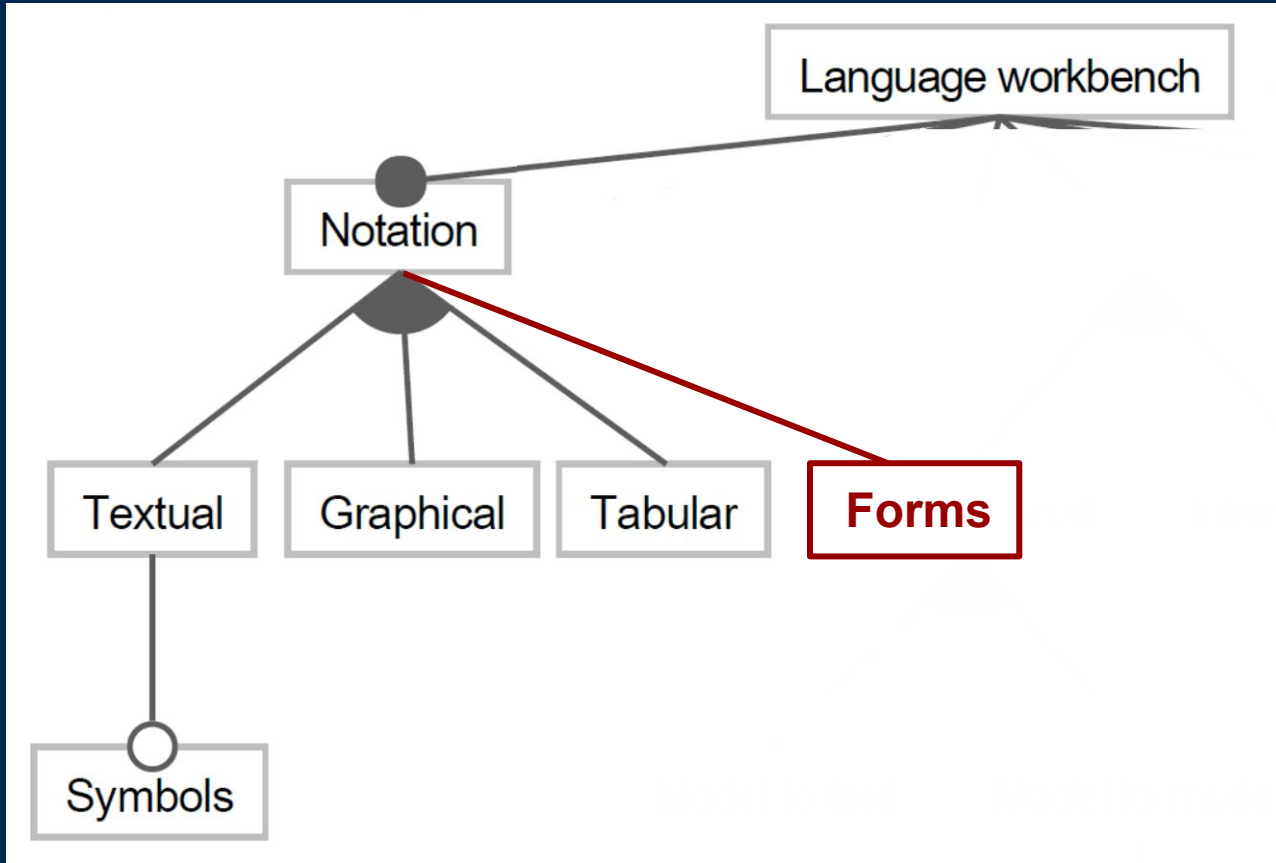
- feature / containment:  
subform
- inheritance:  
switch + subform show
- link / reference:  
dropdown

The image shows a software interface for defining a 'Link'. At the top, the label 'Link' is displayed. Below it is a section titled 'Type (Classifier)' containing a dropdown menu. The dropdown is currently open, showing a list of options: 'Field', 'Package', 'Entity', 'Field' (highlighted), 'Property', 'Reference', 'Page', 'IndexPage', 'DetailsPage', 'Extension', 'Component', 'Module', 'Plugin', and 'Manifest'. A green '+' button is visible on the right side of the dropdown list. The background of the interface is dark blue.

Implementation: select from possibilities

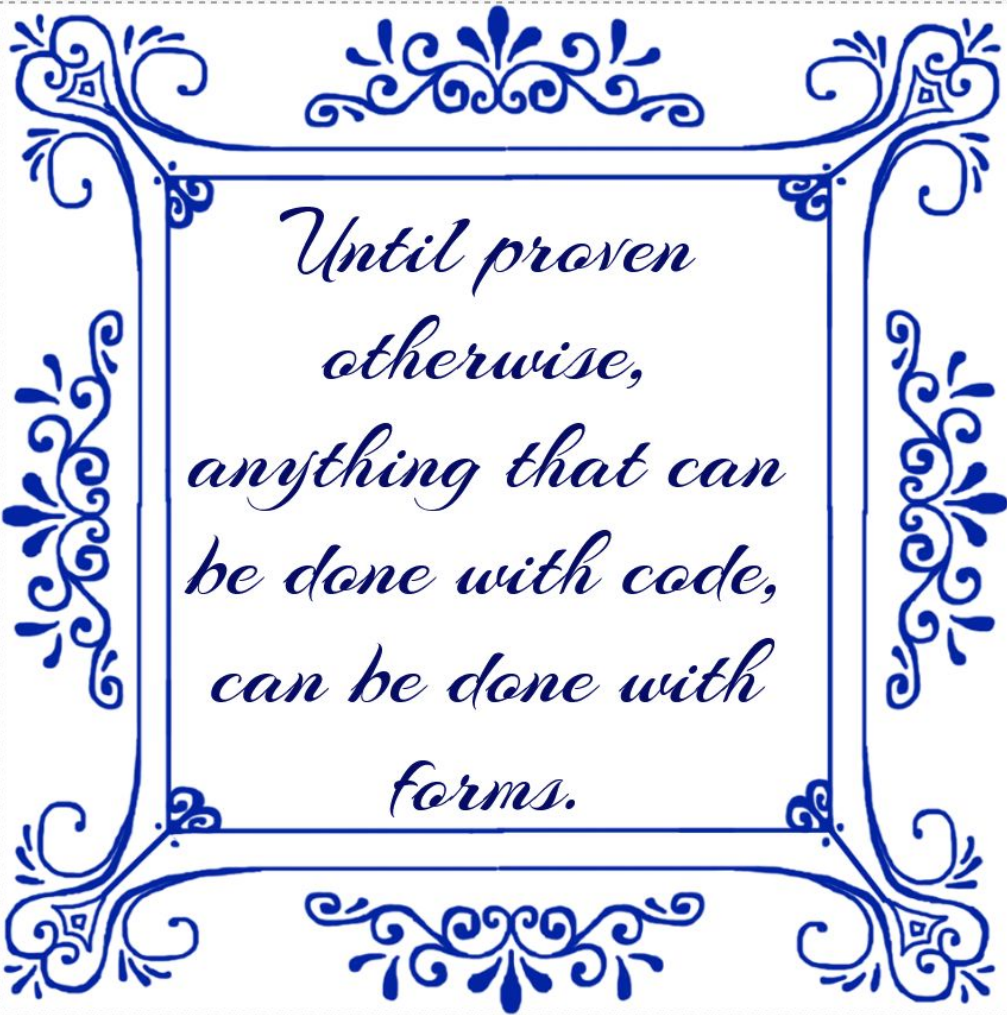
# Project Form Diagram for XML





Towards a forms based language workbench





*Until proven  
otherwise,  
anything that can  
be done with code,  
can be done with  
forms.*

- forms have a lot of **possibilities** as **input interface**



- forms have a lot of possibilities as input interface
- forms give structure and limitation of input



- forms have a lot of possibilities as input interface
- forms give structure and limitation of input
- forms are widely used and familiar for online input

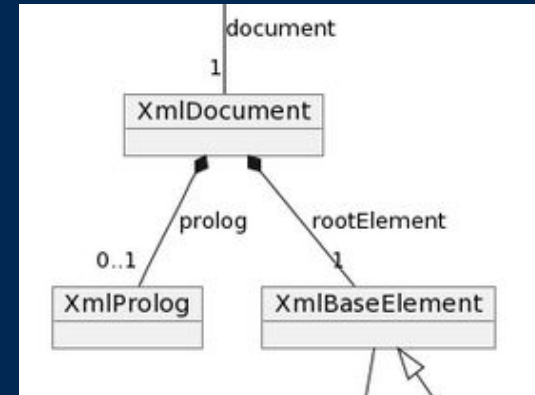
There's no place like

**127.0.0.1**

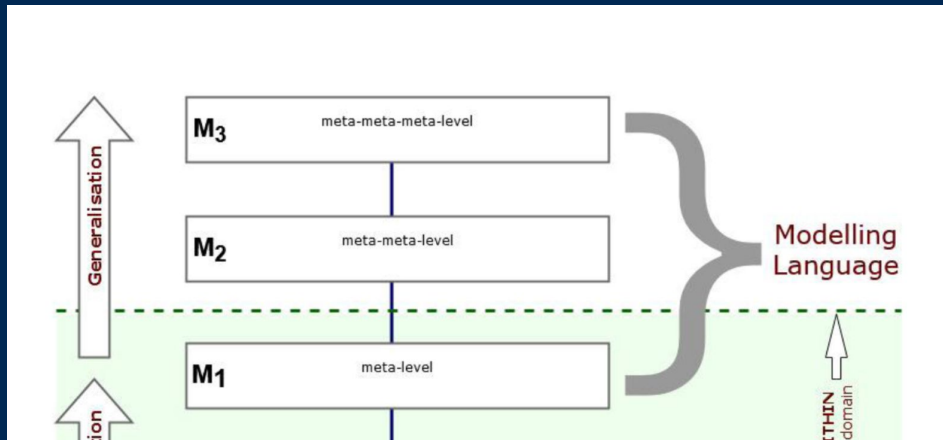
- forms have a lot of possibilities as input interface
- forms give structure and limitation of input
- forms are widely used and familiar for online input
- forms can be used to input a model (AST)

The screenshot shows a web form for defining a LanguageEntity. The form has the following sections:

- LanguageEntity Name:** XmlDocument
- LanguageEntity Key:** XmlDocument
- Type of Language Entity?** Classifier (selected), DataType
- Classifier Type:** Concept (selected), ConceptInterface, Annotation
- Concept:** Concept is abstract? (checkbox), Concept can be root? (checkbox), Extends (dropdown), Implements (button with plus icon)
- Features:** (button with plus icon)
- Feature Details:**
  - Feature Name:** prolog
  - Feature Key:** XmlDocument:prolog
  - Optional?** (checkbox, checked)
  - Can we have zero of this?** (checkbox, unchecked)
  - Property or link?** Property (checkbox, unchecked)



- forms have a lot of **possibilities** as **input interface**
- forms give **structure** and **limitation** of input
- forms are widely used and **familiar** for online input
- forms can be used to **input** a **model** (AST)
- forms can be used on higher **meta-levels**



- forms have a lot of possibilities as input interface
- forms give structure and limitation of input
- forms are widely used and familiar for online input
- forms can be used to input a model (AST)
- forms can be used on higher meta-levels

Forms can be a good addition to (online) input interfaces

[herman@yepr.nl](mailto:herman@yepr.nl)

sites:

Extension Generator [extengen.yepr.nl](http://extengen.yepr.nl)

Projectional Forms Playground [pro-forms.yepr.nl](http://pro-forms.yepr.nl)

repo:

[github.com/HermanPeeren/Extengen](https://github.com/HermanPeeren/Extengen)