

# Domain Specific Languages

*external DSL*

**Julien Deantoni**  
Universite Cote d'Azur,  
CNRS I3S, INRIA KAIROS  
[Julien.deantoni@univ-cotedazur.fr](mailto:Julien.deantoni@univ-cotedazur.fr)

# Outline

*Not specific to external DSLs*

- Definition of the Domain Model, i.e. its concepts and their relations
- Definition of the concrete syntax(es) (i.e., the one manipulated by the end-user(s))
- Populating the Domain Model
- Generating code

# Definition of the Domain Concepts

- There is the need to determine what are the **concepts** of the DSL **domain**. This **can be** done in a metamodel, at least conceptually. This must be explicit in a class diagram, at least in this course :)

# Definition of the Domain Concepts

- There is the need to determine what are the **concepts** of the DSL **domain**. This can be done in a metamodel, at least conceptually. This must be explicit in a class diagram, at least in this course :)
- Example 1 : my colleagues are recurrently defining exams and need to generate html forms for their MOOC. I'd like to help them with a DSL. Let's identify the concepts of the domain

# Definition of the Domain Concepts

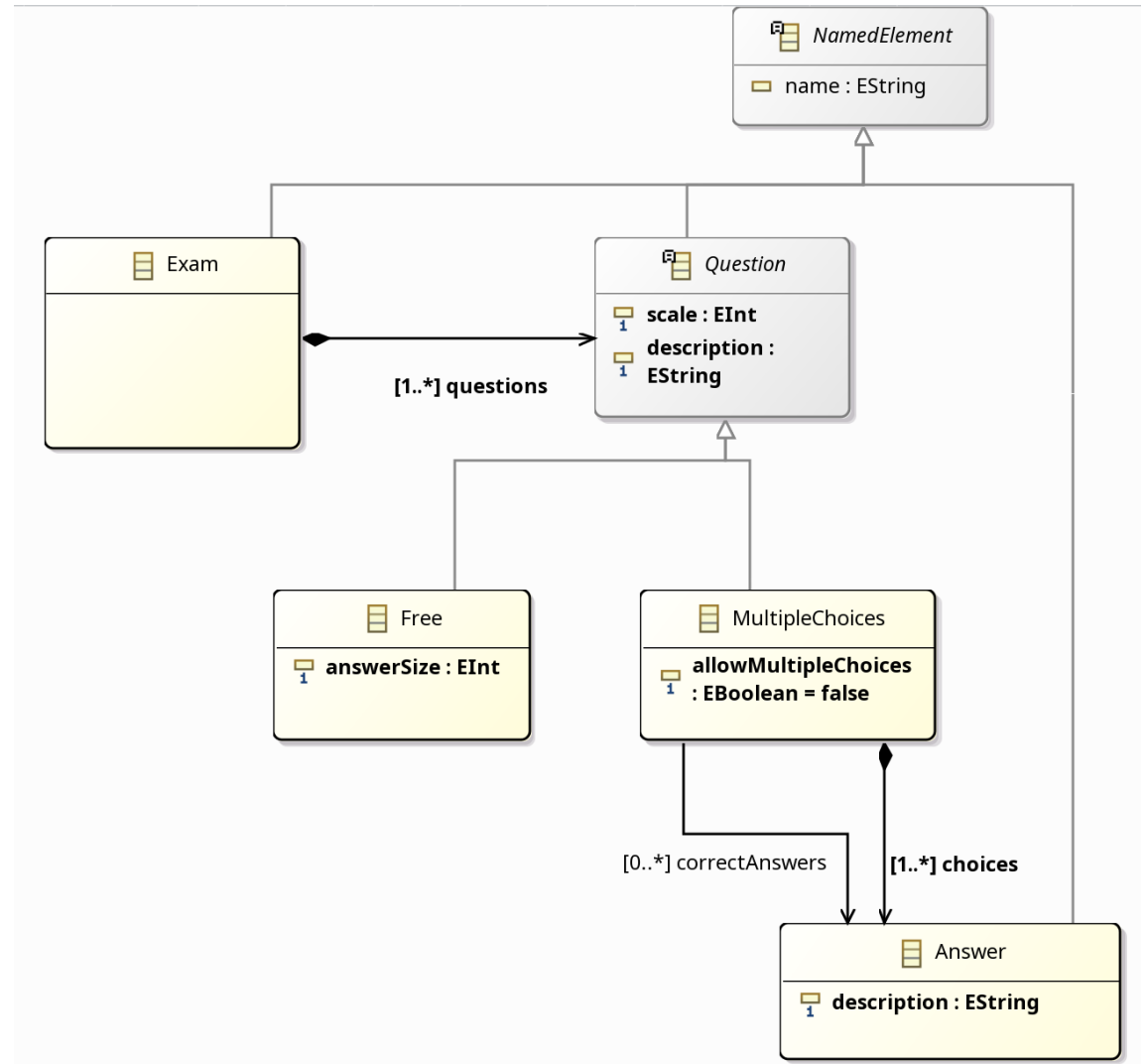
- There is the need to determine what are the **concepts** of the DSL **domain**. This **can be** done in a metamodel, at least conceptually. This must be explicit in a class diagram, at least in this course :)
- Example 1 : my colleagues are recurrently **defining exams** and need to generate html forms for their MOOC. I'd like to help them with a DSL. Let's identify the concepts of the domain

# Definition of the Domain Concepts

- There is the need to determine what are the **concepts** of the DSL **domain**.
- my colleagues are recurrently **defining exams** and need to generate html forms

# Definition of the Domain Concepts

- There is the need to determine what are the **concepts** of the DSL **domain**.



# Definition of the concrete syntax(es)

- Can be textual, graphical, forms, modal gui based, gestual, etc.
- Can be more or less close to the domain model (let's see later)
- Can be projectional or not (let's see later)
- For a textual language, the **grammar** defines the way it is defined.



# Definition of the concrete syntax(es)

- For a textual language, the **grammar** defines the way it is defined.  
Here defined in extended Backus Naur Form :

**Exam:**

```
'Exam' name= ID
'{'
    Question ( "," Question)*
'}
```

[https://fr.wikipedia.org/wiki/Forme\\_de\\_Backus-Naur](https://fr.wikipedia.org/wiki/Forme_de_Backus-Naur)  
[https://en.wikipedia.org/wiki/Backus%E2%80%93Naur\\_form](https://en.wikipedia.org/wiki/Backus%E2%80%93Naur_form)  
[https://en.wikipedia.org/wiki/Extended\\_Backus%E2%80%93Naur\\_form](https://en.wikipedia.org/wiki/Extended_Backus%E2%80%93Naur_form)  
[https://www.eclipse.org/Xtext/documentation/301\\_grammarlanguage.html](https://www.eclipse.org/Xtext/documentation/301_grammarlanguage.html)

**Question:**

```
Free | MultipleChoices;
```

**Free:**

```
'question' ID
'on' Int
':' String
(' 'maxSize' '=' Int '')
```

**MultipleChoices:**

```
'MCQ' ID
'on' Int
':' String
'choices' '{' Answer ( "," Answer)* '}'
('correctAnswers' ('allowMultipleChoices')? ':' [Answer] ( "," [Answer])*)?
```

**Answer:**

```
'Answer' String ':' String
```

# Definition of the concrete syntax(es)

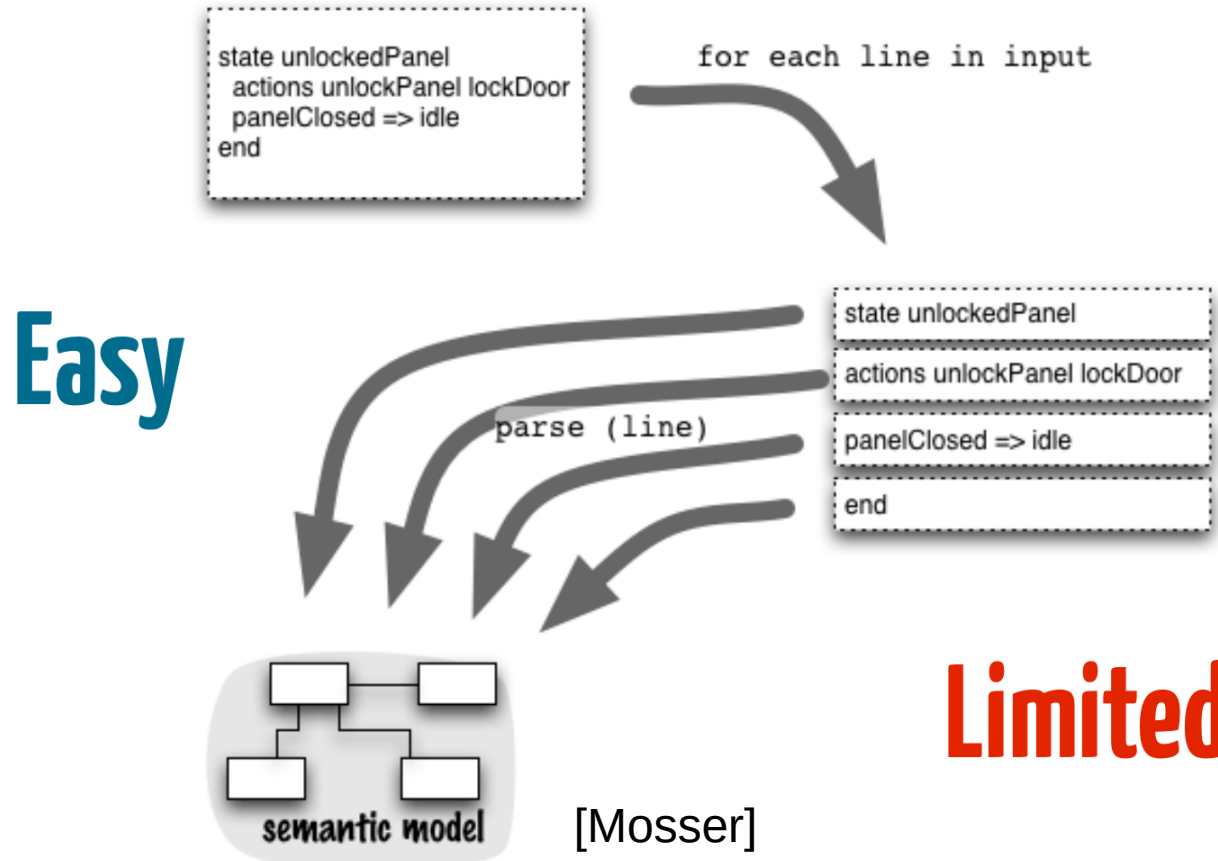
- For a textual language, the **grammar** defines the way it is defined.

```
Exam cc1 {  
  question q1 on 5 : "What is the answer to this question" (maxSize = 1),  
  
  MCQ q2 on 3 : "Do you like course given at Polytech ?"  
  choices {  
    Answer a1 : "Yes",  
    Answer a2 : "No, sorry",  
    Answer a3 : " what ?"  
  }correctAnswers : a1,  
  
  question q3 on 0 : "Is there any remark about the course ?" (maxSize = 10)  
}
```

# Populating the Domain Model

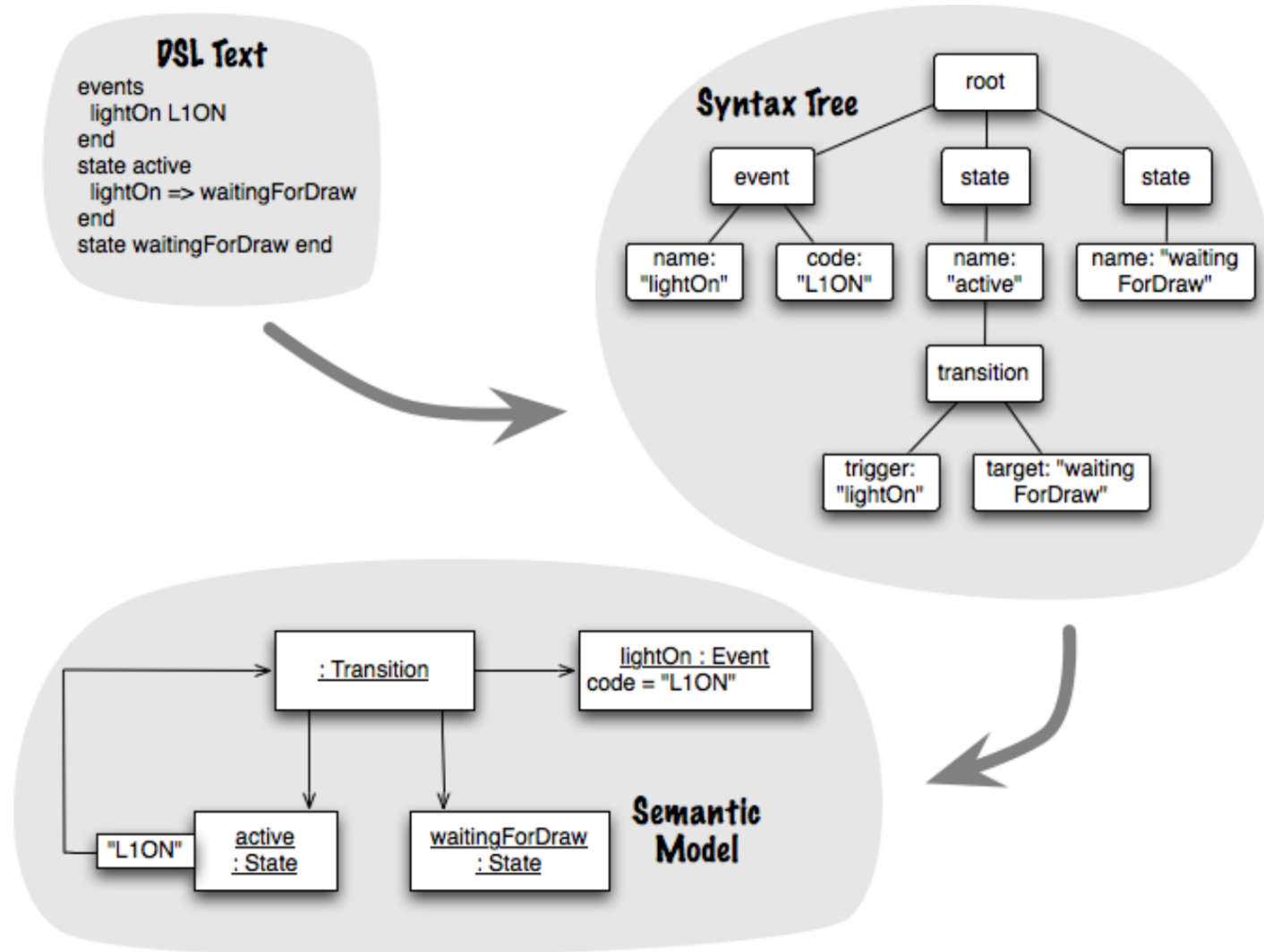
- This is not a compilation course ! Moreover, many frameworks make it easier in most of the cases (ANTLR, Xtext, Langium...)

## Delimiter-directed Translation



# Populating the Domain Model

- This is not a compilation course ! Moreover, many frameworks make it easier in most of the cases (ANTLR, Xtext, ...)



[Domain-Specific Languages] ← [Mosser]

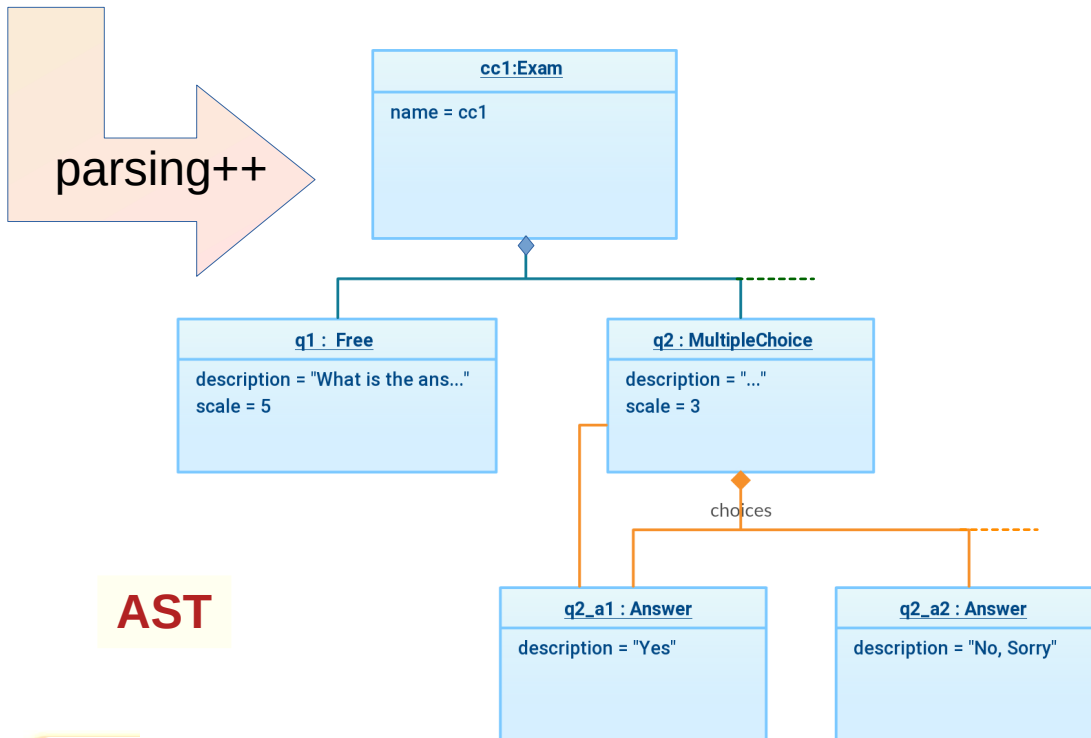
# Populating the Domain Model

- This is not a compilation course ! Moreover, many frameworks make it easier in most of the cases (ANTLR, Xtext, ...)

```
Exam cc1 {
  question q1 on 5 : "What is the answer to this question" (maxSize = 1),

  MCQ q2 on 3 : "Do you like course given at Polytech ?"
  choices {
    Answer a1 : "Yes",
    Answer a2 : "No, sorry",
    Answer a3 : " what ?"
  }correctAnswers : a1,

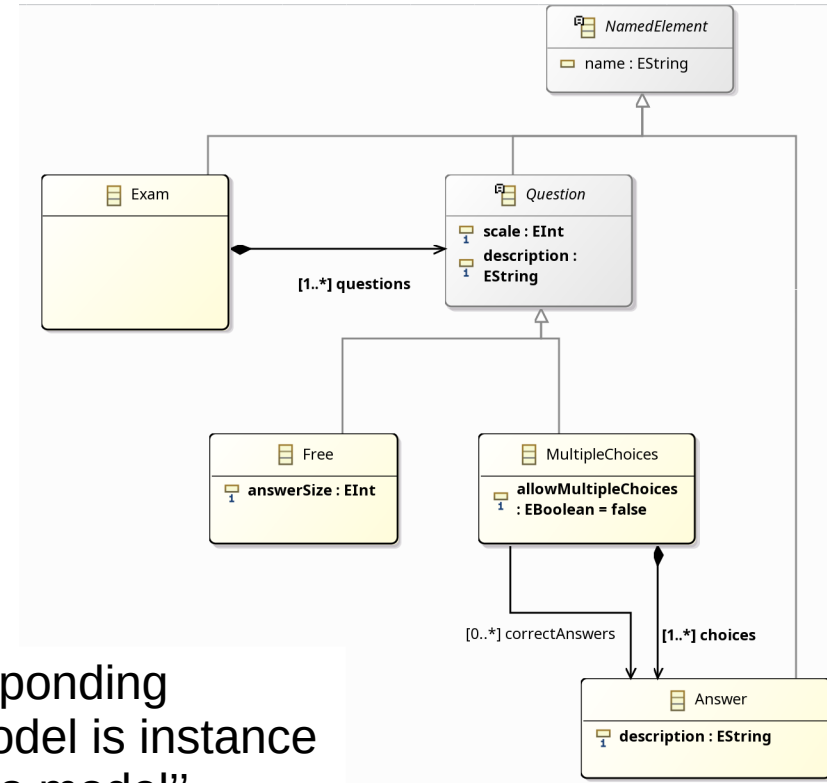
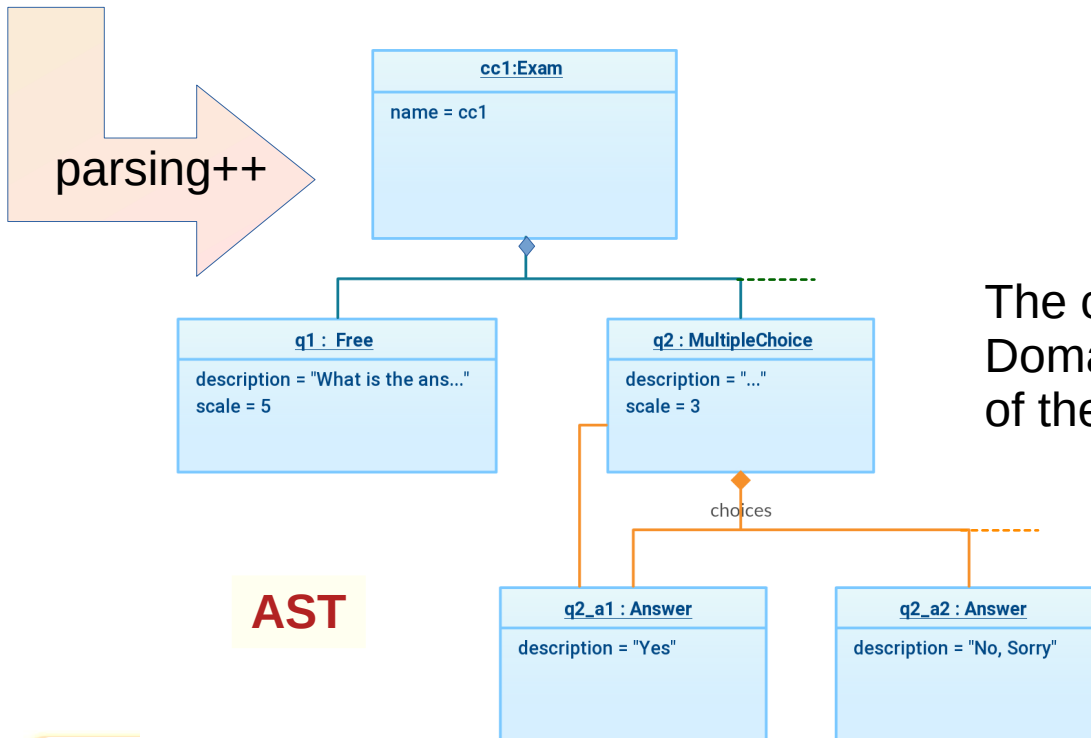
  question q3 on 0 : "Is there any remark about the course ?" (maxSize = 10)
}
```



# Populating the Domain Model

- This is not a compilation course ! Moreover, many frameworks make it easier in most of the cases (ANTLR, Xtext, ...)

```
Exam cc1 {
  question q1 on 5 : "What is the answer to this question" (maxSize = 1),
  MCQ q2 on 3 : "Do you like course given at Polytech ?"
  choices {
    Answer a1 : "Yes",
    Answer a2 : "No, sorry",
    Answer a3 : " what ?"
  }correctAnswers : a1,
  question q3 on 0 : "Is there any remark about the course ?" (maxSize = 10)
}
```

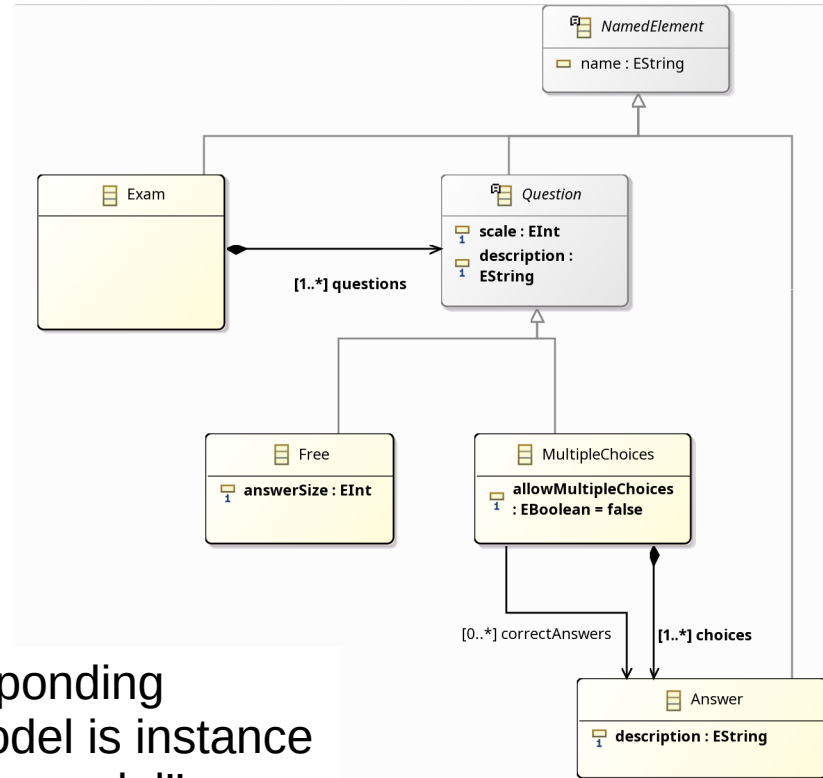
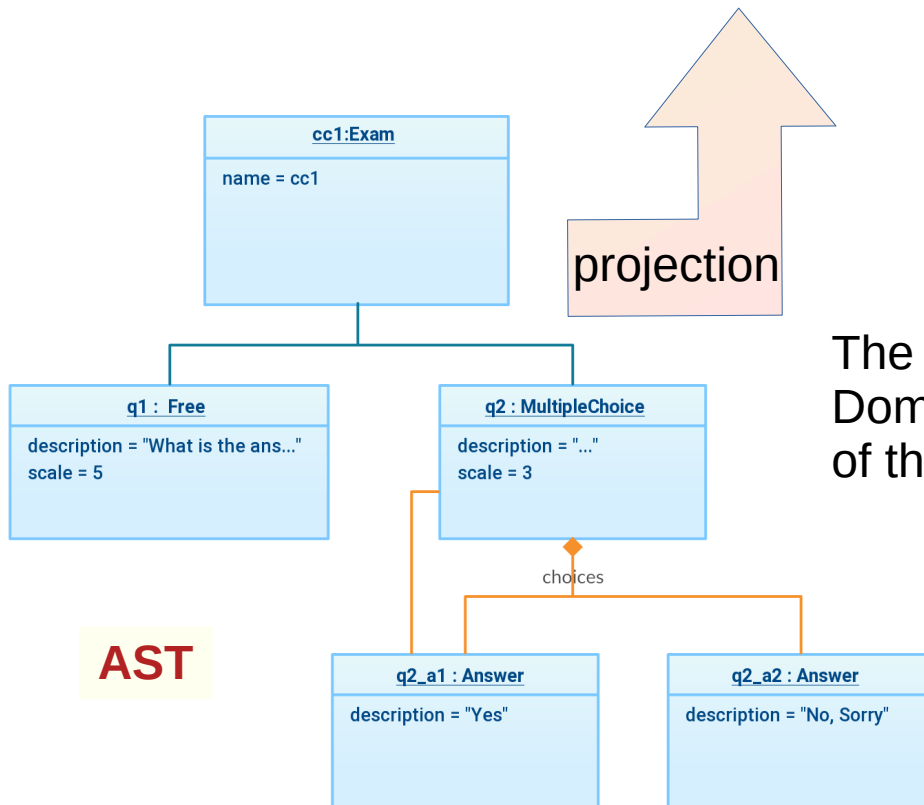


The corresponding Domain Model is instance of the "meta model"

# Populating the Domain Model

- Projectional concrete syntax (like in MPS or Sirius), the AST is directly presented to the user, in an editable way, by a projection

```
Exam cc1 {
  question q1 on 5 : "What is the answer to this question" (maxSize = 1),
  MCQ q2 on 3 : "Do you like course given at Polytech ?"
  choices {
    Answer a1 : "Yes",
    Answer a2 : "No, sorry",
    Answer a3 : " what ?"
  }correctAnswers : a1,
  question q3 on 0 : "Is there any remark about the course ?" (maxSize = 10)
}
```



The corresponding Domain Model is instance of the "meta model"

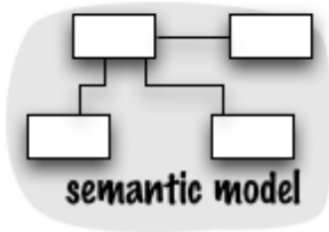
# Generating Code

- Several ways to do it, like the
  - Visitor pattern
  - PrettyPrint weaving
  - Template based
  - ...



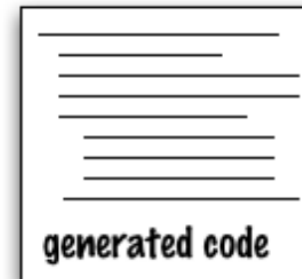
# Generating Code

## Transformer Generation



**ToArduinoCode.java**

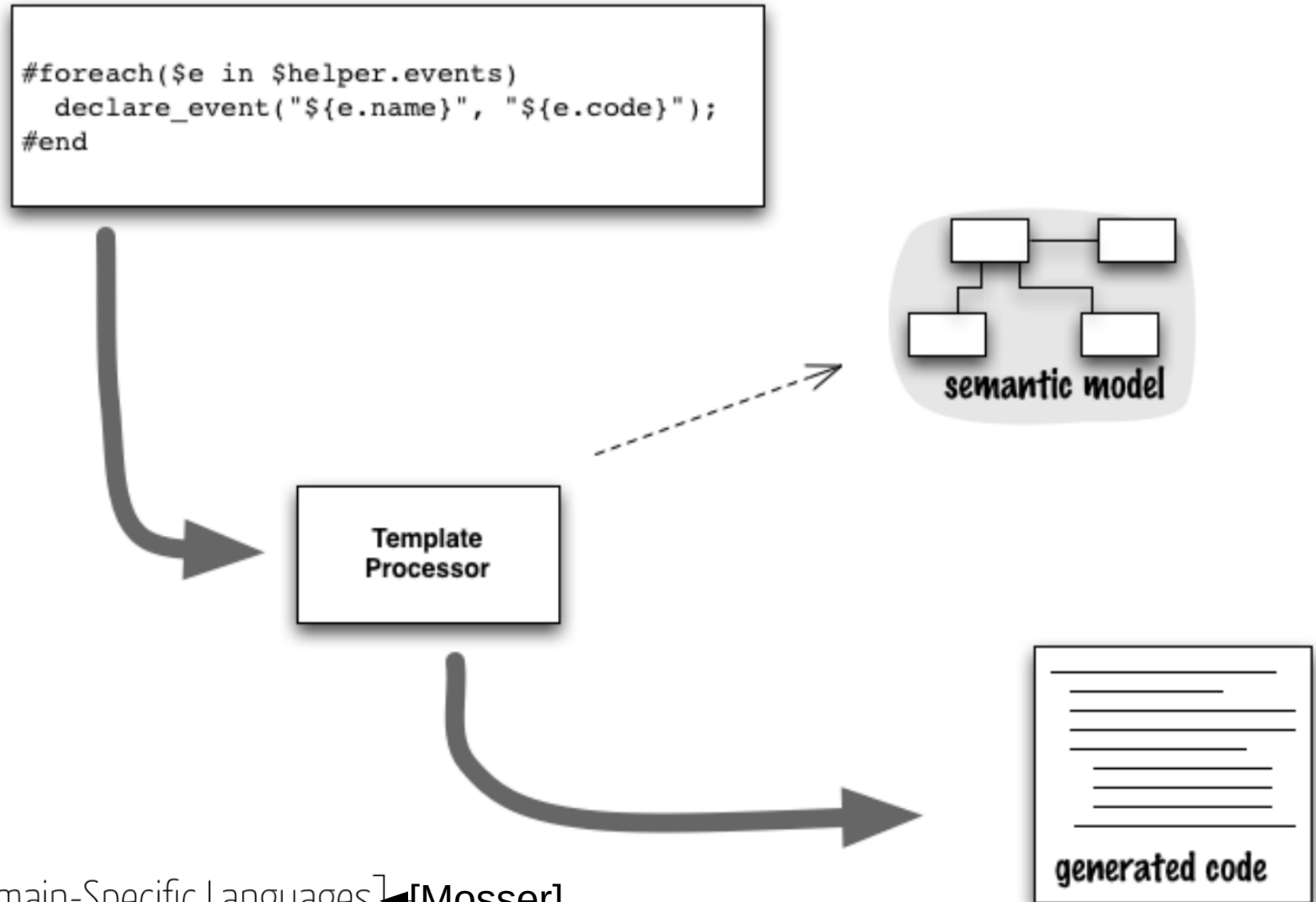
```
private void generateEvents(Writer output) throws IOException {  
    for (Event e : machine.getEvents())  
        output.write(String.format(" declare_event(\"%s\", \"%s\");\n",  
                                   e.getName(), e.getCode()));  
    output.write("\n");  
}
```



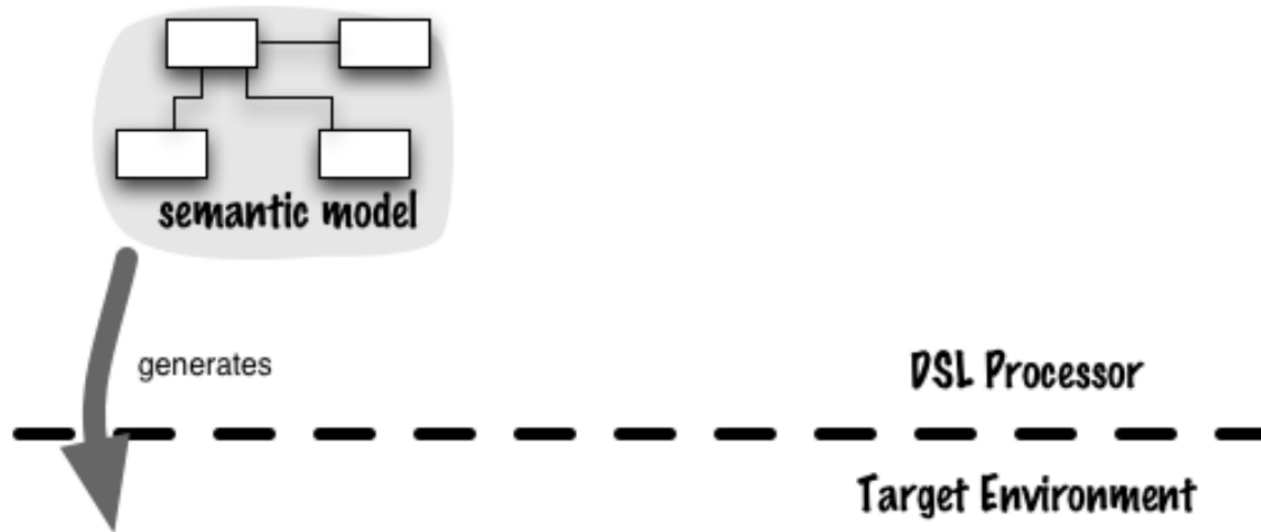
[Domain-Specific Languages] ← [Mosser]

# Generating Code

## Template-based generation



[Domain-Specific Languages] ← [Mosser]

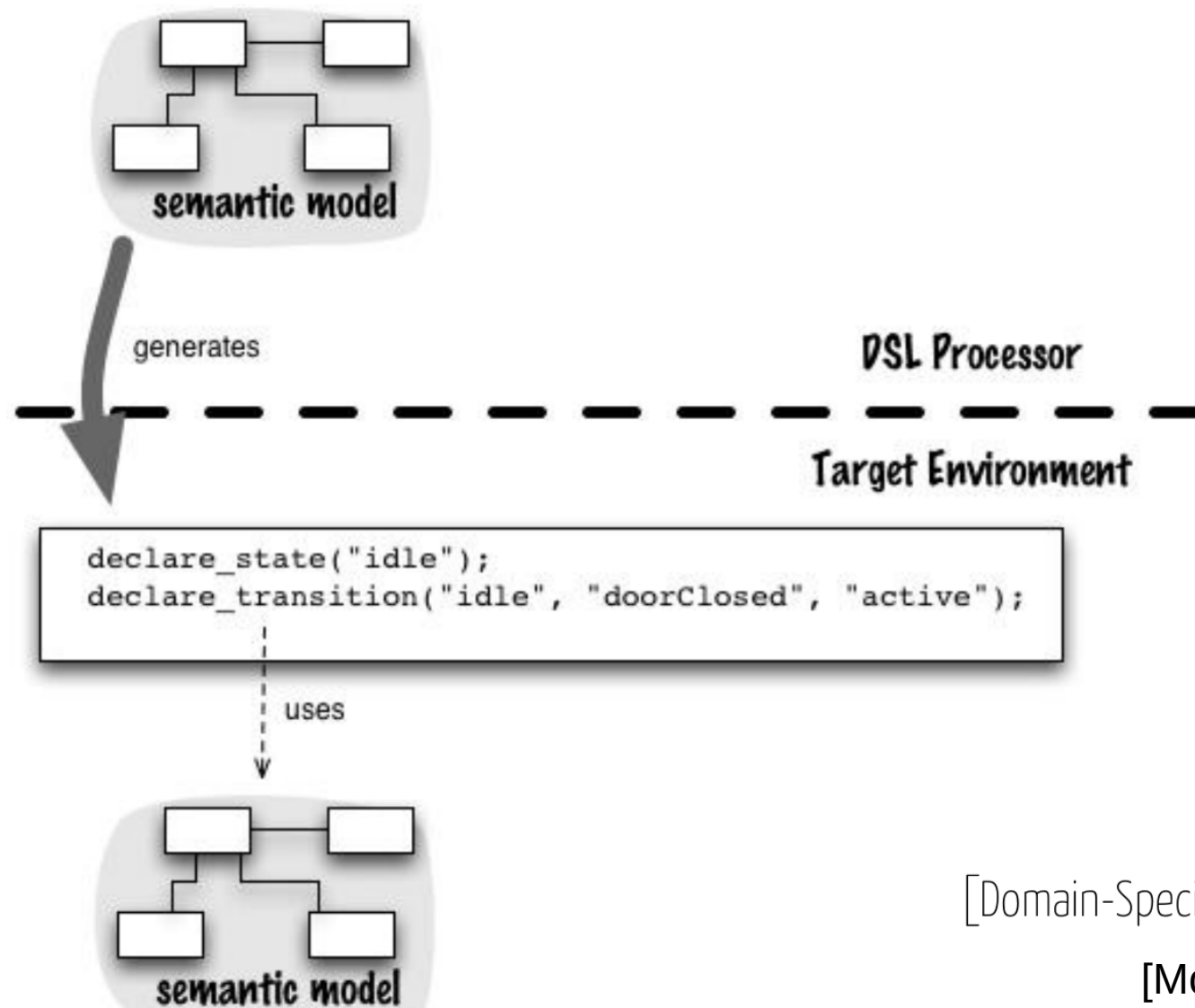


```
void handle_event(char *code) {  
  switch(current_state_id) {  
    case STATE_idle: {  
      if (0 == strcmp(code, EVENT_doorClosed)) {  
        current_state_id = STATE_active;  
      }  
      return;  
    }  
    case STATE_active: {  
      ...  
    }  
  }  
}
```

[Domain-Specific Languages]

▲  
[Mosser]

*Hardcode all logic into the generated code so that there's no explicit representation of the Semantic Model.*



Generate code with an explicit simulacrum of the semantic model of the DSL, so that the generated code has generic-specific separation.

# Generating Code, template based

```
[module generateHtml('http://polytech.unice.fr/dsl/exams')] // Based on the metamodel
```

```
[template public generateElement(anExam : Exam)]
[comment @main /]
[file (anExam.name+'.html', false, 'UTF-8')]
<!DOCTYPE html>
<html>
<body>
```

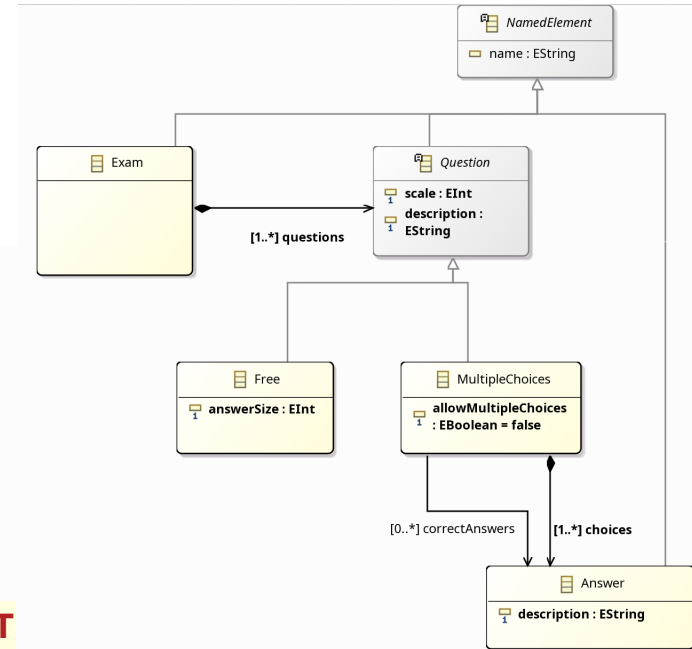
**// Some "static" text**

```
<h2>Examen [anExam.name/] ([anExam.questions.scale->sum()]) points</h2>
```

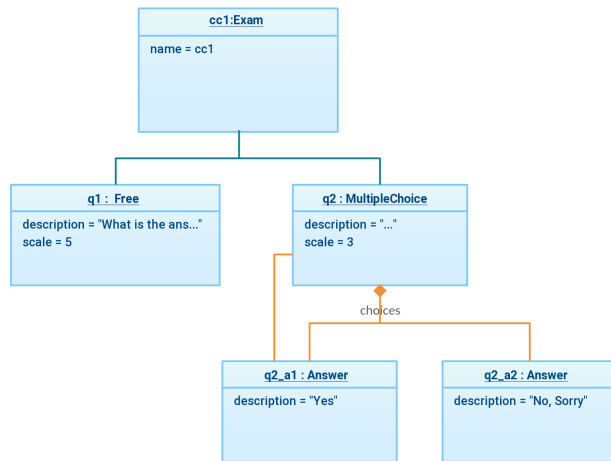
```
<form action="/action_page.php">
[for (question : Question | anExam.questions)] // templated code navigate the AST
```

```
<STRONG>Question [i/] ([question.scale/] points)</STRONG>:
[if (question.ocIsKindOf(Free))]
[question.description/]<br>
<textarea cols="50" "[question.name/]" value="" size="52" rows="[question.ocAsType(Free).answerSize/]">
</textarea>
<br>
[else]
[question.description/]
<br>
[for (ans : Answer | question.ocAsType(MultipleChoices).choices)]
<INPUT TYPE="radio" NAME="[ans.name/]" VALUE="[ans.name/]">[ans.description/] <br>
[/for]
[/if]
```

```
[/for]
<br>
<input type="submit" value="Submit">
</form>
</body>
</html>
```



# Generating Code, template based



```

[module generateHtml('http://polytech.unice.fr/dsl/exams')]

[template public generateElement(anExam : Exam)]
[comment @main /]
[file (anExam.name+'.html', false, 'UTF-8')]
<!DOCTYPE html>
<html>
<body>

<h2>Examen [anExam.name/] ([anExam.questions.scale->sum()/] points)</h2>

<form action="/action_page.php">
[for (question : Question | anExam.questions)]
<STRONG>Question [i/] ([question.scale/] points)</STRONG>:
[if (question.oclIsKindOf(Free))]
[question.description/]<br>
<textarea cols="50" "[question.name/]" value="" size="52" rows="[question.oclAsType(Free).answerSize/]">
</textarea>
<br>
[else]
[question.description/]
<br>
[for (ans : Answer | question.oclAsType(MultipleChoices).choices)]
<INPUT TYPE="radio" NAME="[ans.name/]" VALUE="[ans.name/]">[ans.description/] <br>
[/for]
[/if]

[/for]
<br>
<input type="submit" value="Submit">
</form>

</body>
</html>
  
```

```

<!DOCTYPE html>
<html>
<body>

<h2>Examen cc1 (8 points)</h2>

<form action="/action_page.php">
<STRONG>Question 1 (5 points)</STRONG>:
What is the answer to this question<br>
<textarea cols="50" "q1" value="" size="52" rows="1">
</textarea>
<br>

<STRONG>Question 2 (3 points)</STRONG>:
Do you like course given at Polytech ?
<br>
<INPUT TYPE="radio" NAME="a1" VALUE="a1">Yes <br>
<INPUT TYPE="radio" NAME="a2" VALUE="a2">No, sorry <br>
<INPUT TYPE="radio" NAME="a3" VALUE="a3"> what ? <br>

<STRONG>Question 3 (0 points)</STRONG>:
Is there any remark about the course ?<br>
<textarea cols="50" "q3" value="" size="52" rows="10">
</textarea>
<br>

<br>
<input type="submit" value="Submit">
</form>

</body>
</html>
  
```



# Generating Code, template based

```
Exam cc1 {
  question q1 on 5 : "What is the answer to this question" (maxSize = 1),

  MCQ q2 on 3 : "Do you like course given at Polytech ?"
  choices {
    Answer a1 : "Yes",
    Answer a2 : "No, sorry",
    Answer a3 : " what ?"
  }correctAnswers : a1,

  question q3 on 0 : "Is there any remark about the course ?" (maxSize = 10)
}
```

## Examen cc1 (8 points)

Question 1 (5 points): What is the answer to this question

Question 2 (3 points): Do you like course given at Polytech ?

- Yes
- No, sorry
- what ?

Question 3 (0 points): Is there any remark about the course ?

Submit

