Instantiation of meta-models constrained with OCL: A CSP approach

Adel Ferdjoukh, Anne-Elisabeth Baert, Eric Bourreau, Annie Chateau, Rémi Coletta and Clémentine Nebut

Lirmm laboratory, Montpellier, France.

Angers, February 10^{th} 2015

Synopsis

Synopsis

Model Generation

Objectives

- Generating models conform to meta-models.
- Representing them as instance diagrams of a meta-model.
- Fully Automating this process is desirable.

Model Generation ?

Why do we generate models ?

- Validation of meta-models by assisting designers of meta-models.
- Test data for programs handling models (Compiler, Code generator, ...).

Model Generation: characteristics

Characteristics of generation process/generated models

- **Validity.** Producing valid models conform to meta-model and considering OCL.
- **Scalability.** Using *big-sized* meta-models and computing of *big-sized* models.
- **Diversity.** Generating a set of *different* and *distant* models.
- **Relevance.** Generating realistic models.

Model Generation: Related Work

	Approach	Scalability	Validity	Diversity	Flexibility	
	Random graphs	?	no	no	no	
	Grammar graphs	yes	no	yes	no	
	Alloy	no	manual	no	medium	
→	CSP	rather not	yes	possible	yes	←
	SMT	rather not	yes	no	?	

Why do we choose CSP ?

- Other approaches suffer from a lack of flexibility.
- CSP is able to treat OCL constraints.
- The existing CSP approach does not scale due to non-efficient modelling.
- Possibility of generating diverse solutions.

CSP - Constraint Satisfaction Problem

CSP - Constraint Satisfaction Problem

An AI paradigm for solving combinatorial problems.

CSP - Constraint Satisfaction Problem

Formal definition

A triple $\mathcal{P} =< X, D, C >$.

- $X = \{x_1, x_2, \ldots, x_n\}$ a set of n variables.
- $D = \{D(x_1), D(x_2), \ldots, D(x_n)\}$ a set of n domains.
- $C = \{c_1, c_2, \ldots, c_m\}$ a set of m constraints.

A variable x_i takes its values in $D(x_i)$.

CSP - Constraint Satisfaction Problem

A solution

is an affectation of X which does not violate any constraint of C.
It is found by a CSP Solver.

CSP - Constraint Satisfaction Problem

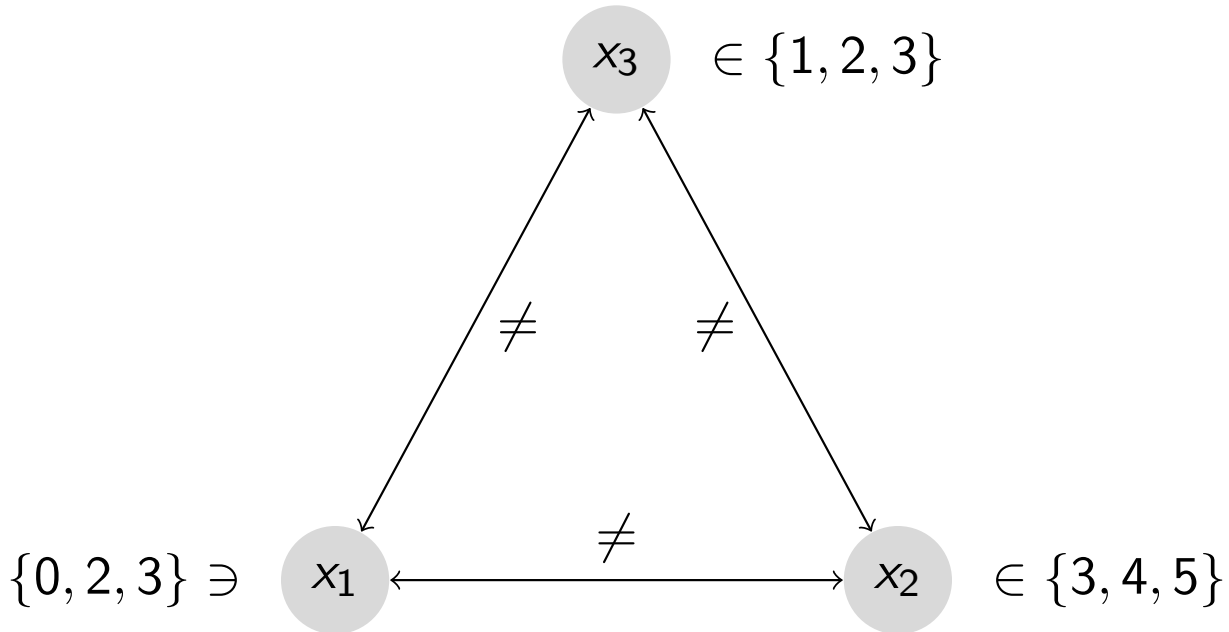
$x_3 \in \{1, 2, 3\}$

$\{0, 2, 3\} \ni x_1$

$x_2 \in \{3, 4, 5\}$

- 3 variables with their domains.

CSP - Constraint Satisfaction Problem



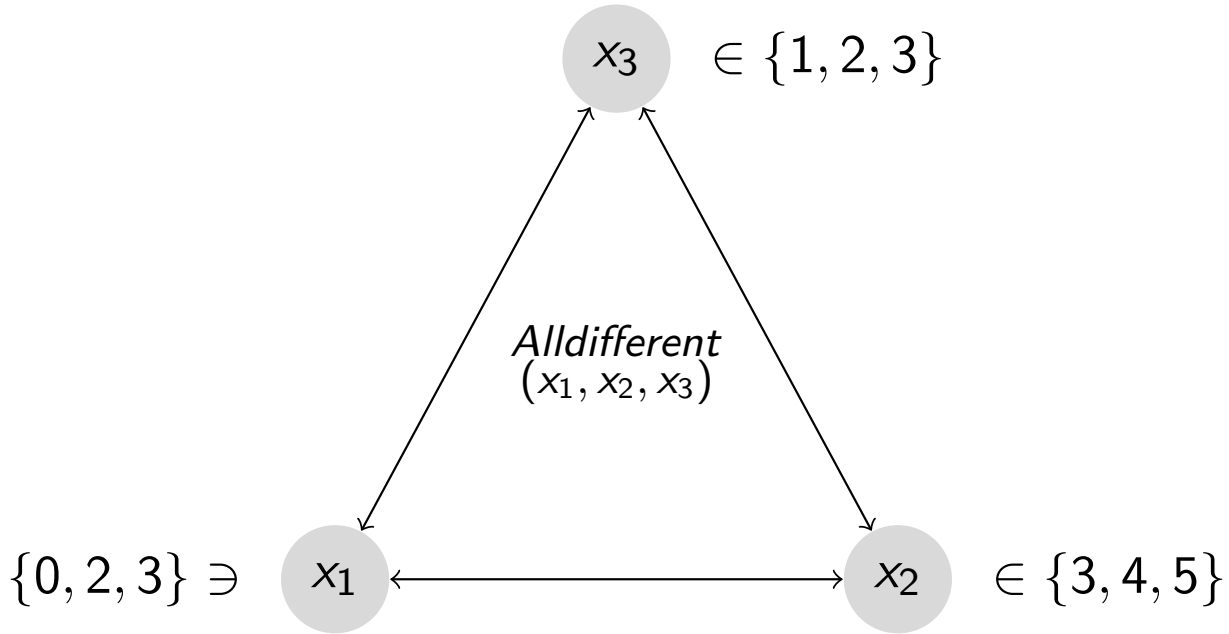
- $(x_1 = 0, x_2 = 4, x_3 = 1)$ is one of the solutions.
- These are binary constraints.

CSP - Constraint Satisfaction Problem

Global Constraints

- Having big variable arity.
- Making the modelling easier and CSP more compact.
- Speeding up the resolution.

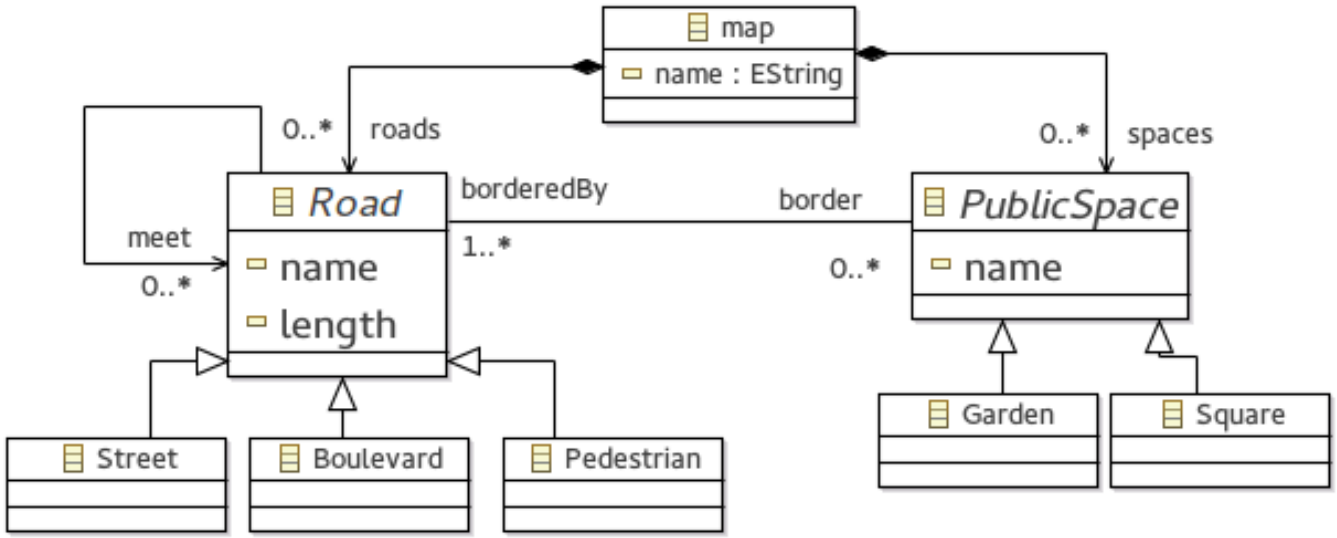
CSP - Constraint Satisfaction Problem



- Replace 3 constraints by one $\mathit{Alldifferent}(x_1, x_2, x_3)$.

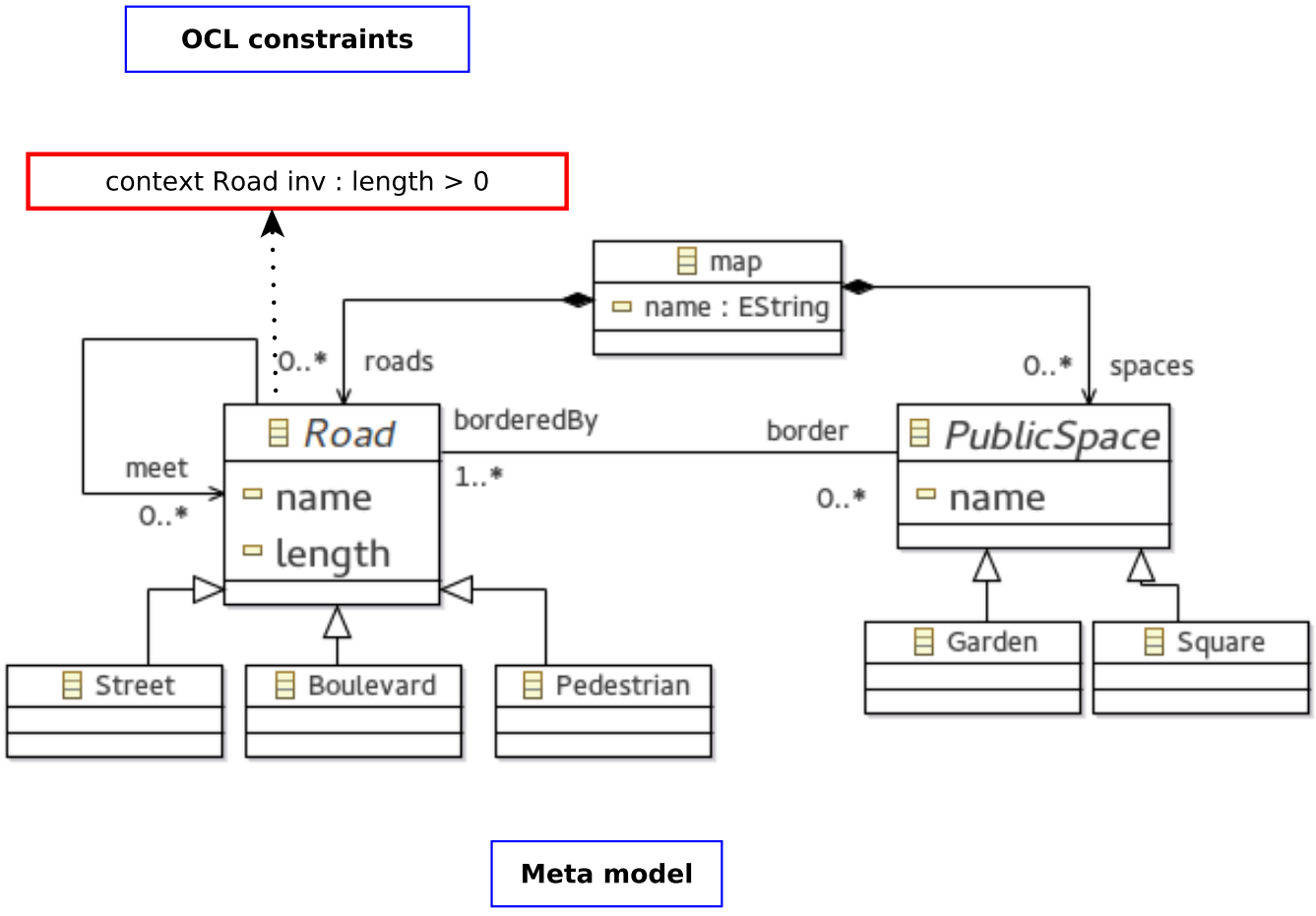
Synopsis

Running example

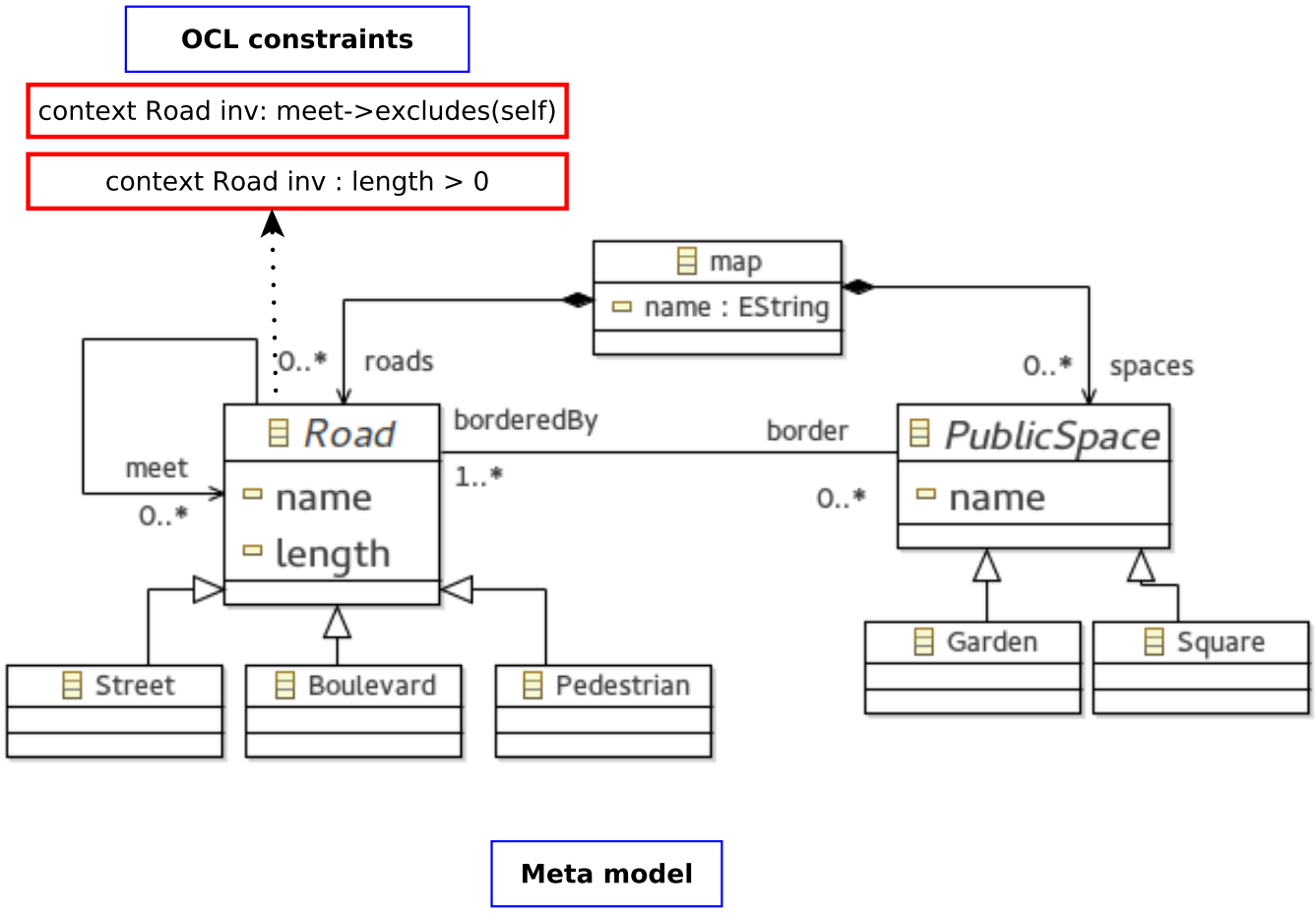


Meta model

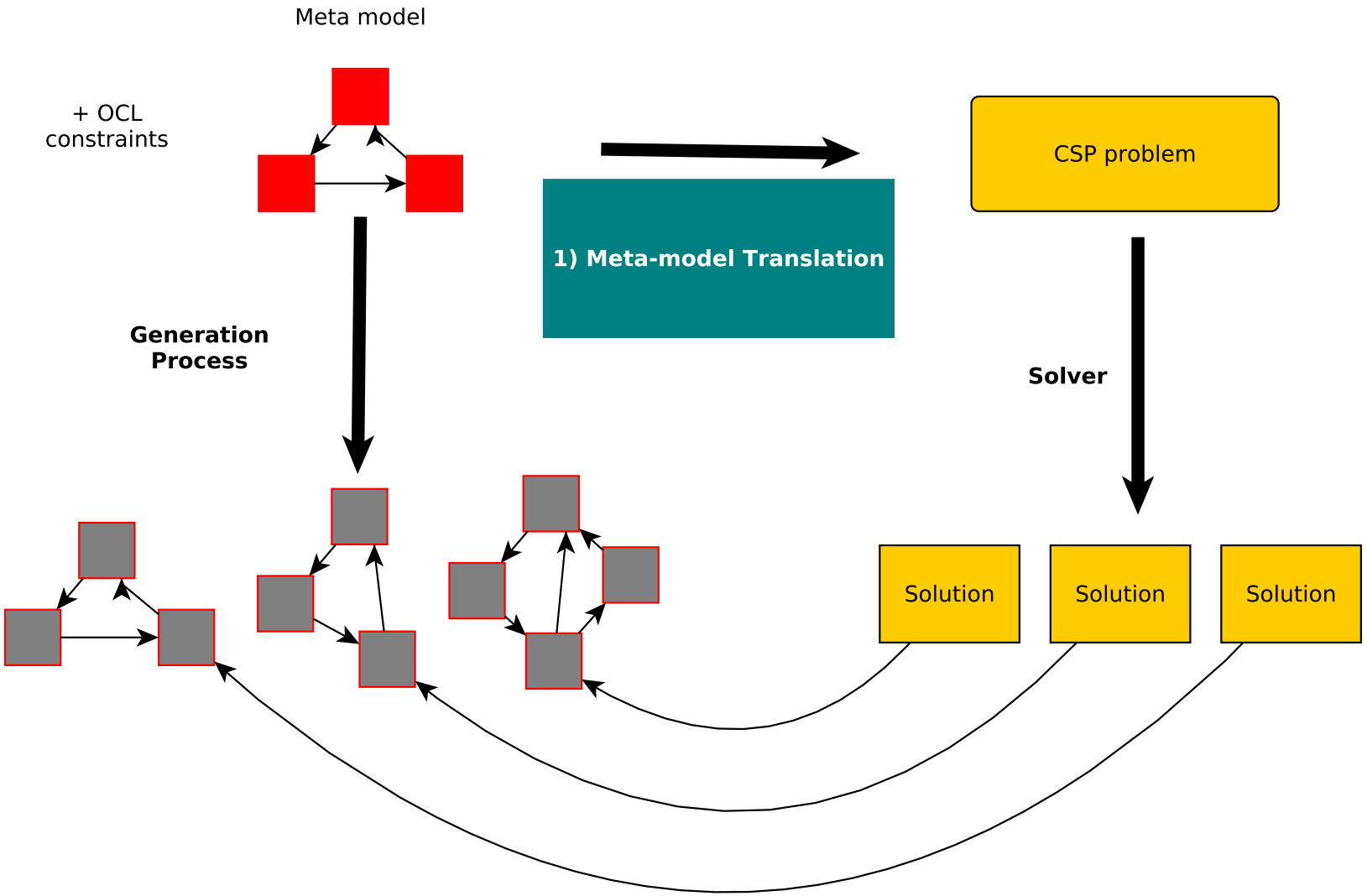
Running example



Running example



Meta-model Translation



Meta-model Translation

Related work

- A Lack of efficiency in the existing approaches.

Improvement

- Put as few variables as possible.
- Avoid the creation of constraints when not necessary.
- Minimize their arity anyway.

Meta-model Translation

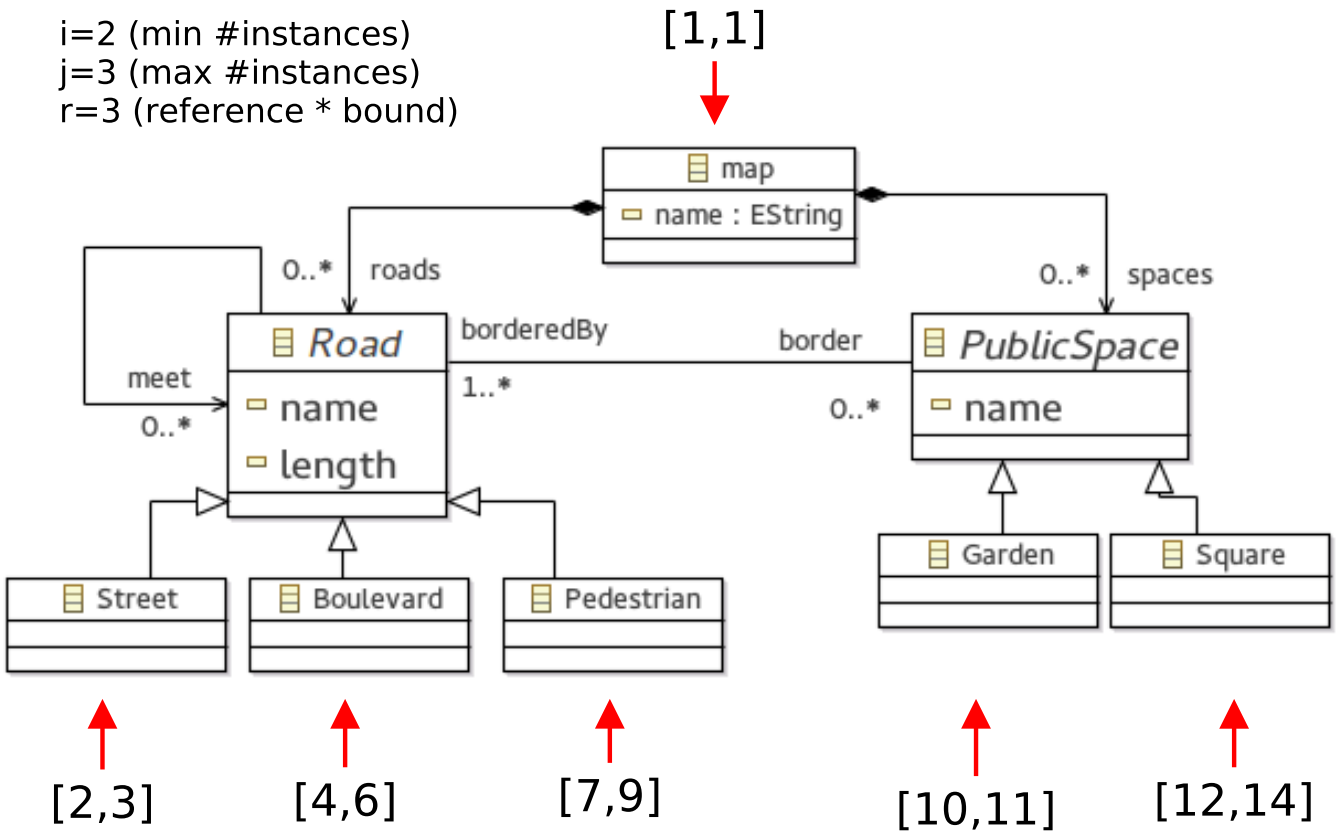
Required inputs

- i: Minimum number of class instances.
- j: Maximum number of class instances).
- r: Reference bound (when bounded with $*$).

Class instances

- An interval to represent the instances of a class.
- The intervals are contiguous.

Meta-model Translation

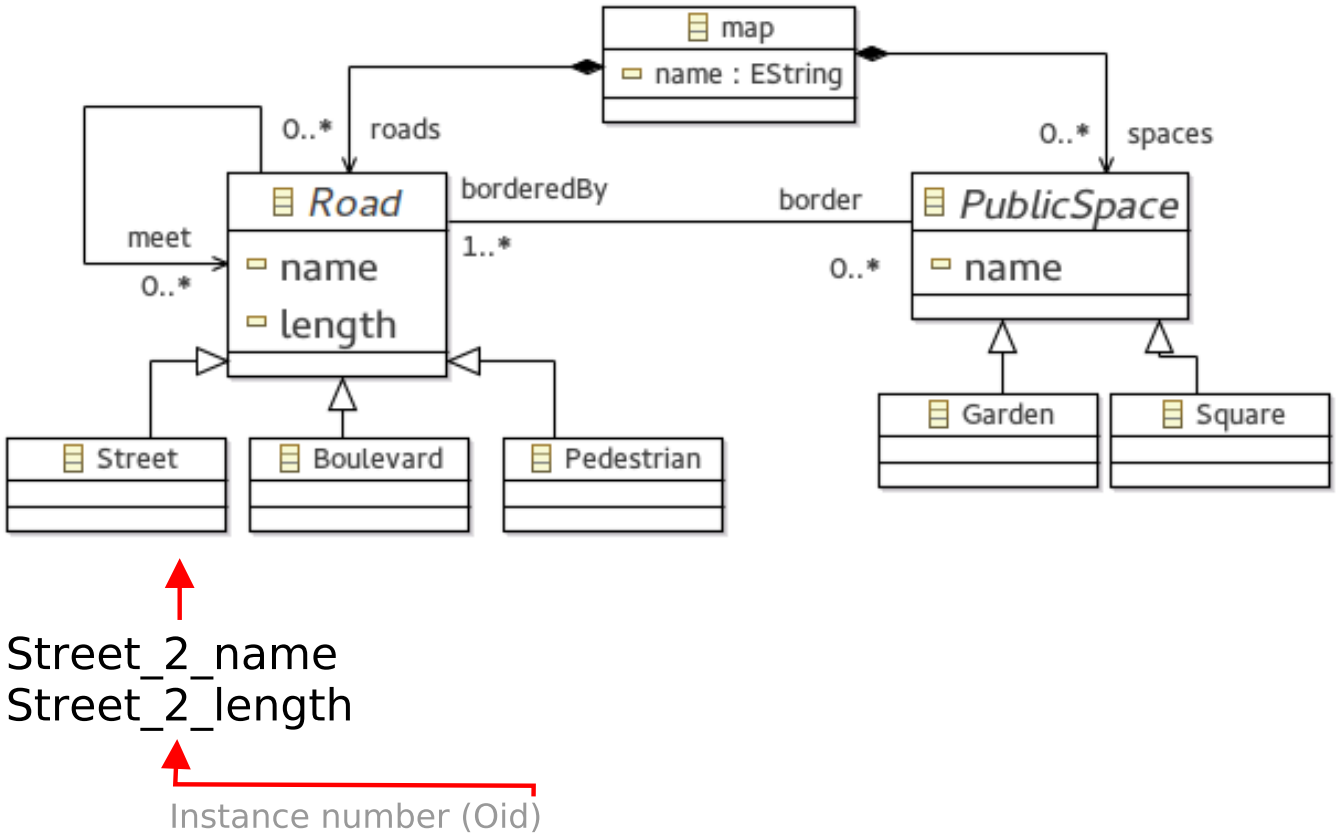


Meta-model Translation

Attributes of class c

- One variable per each attribute and per each instance c.
- Import all inherited attributes.
- Domains are arbitrary intervals.

Meta-model Translation

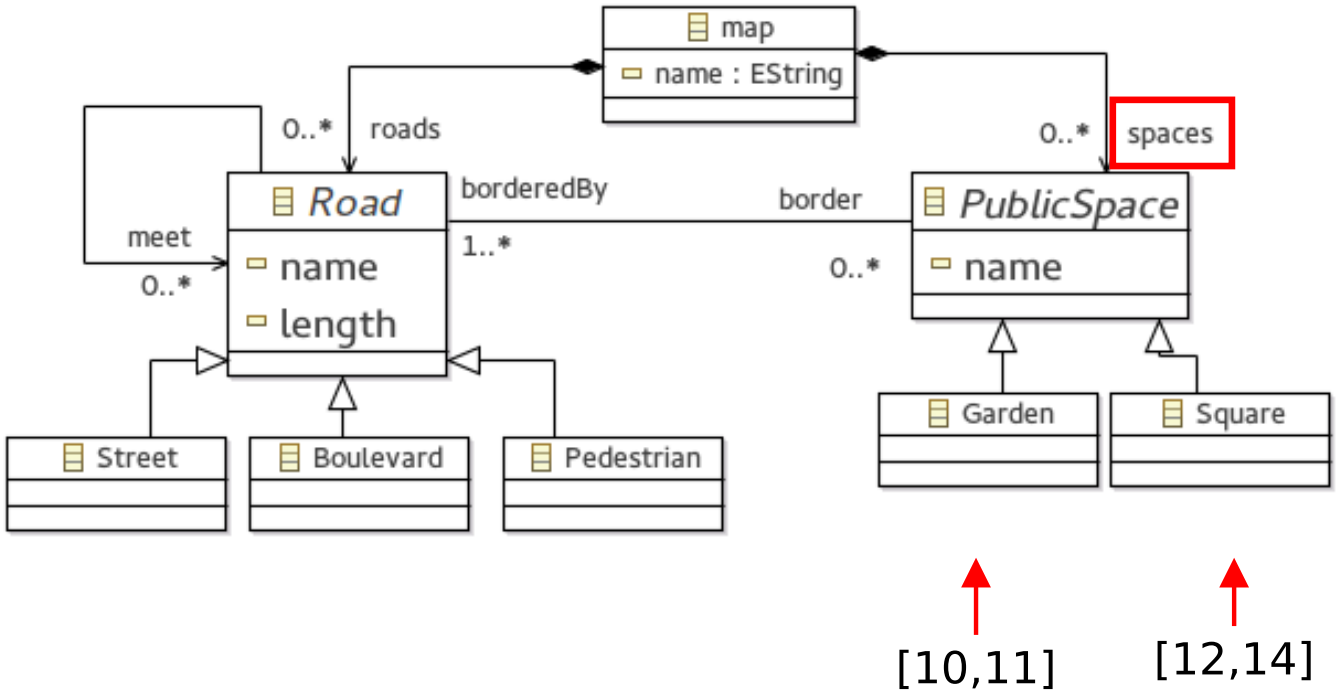


Meta-model Translation

References

- A set of variables for each reference according to its cardinality (or to input parameter r).
- These variables are created on all instances of source class.
- They can be seen as pointers from source to target.
- Their domain is equals to target class interval.

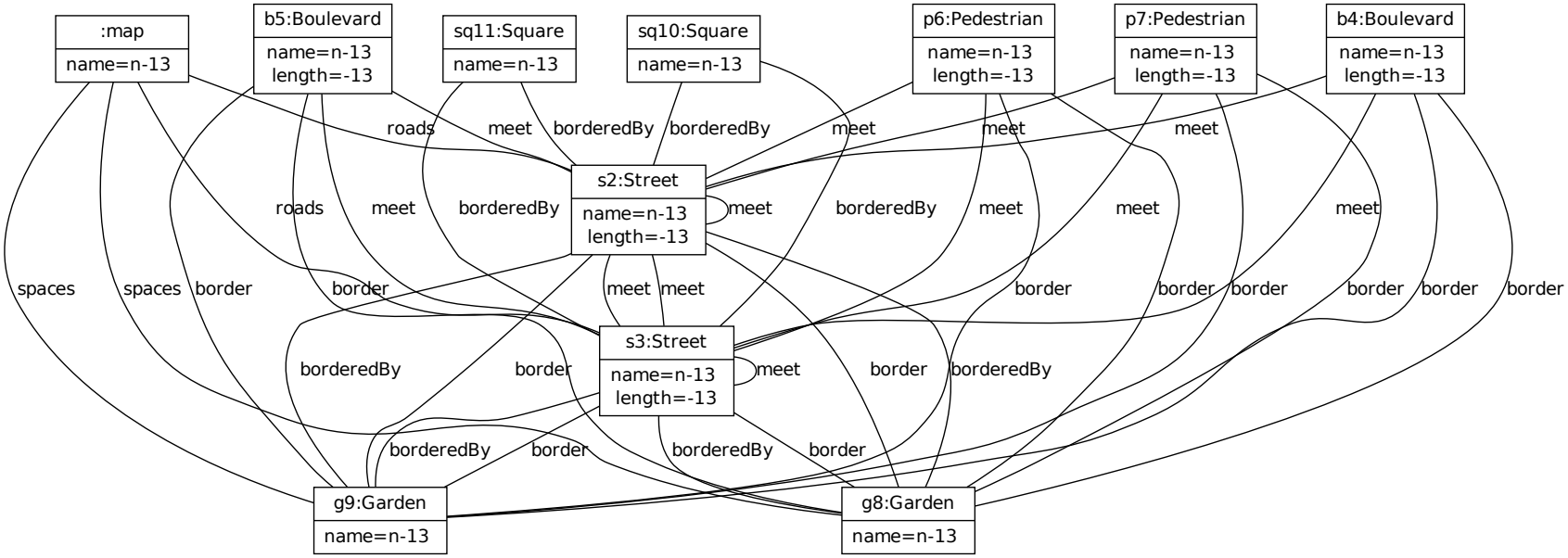
Meta-model Translation



CSP

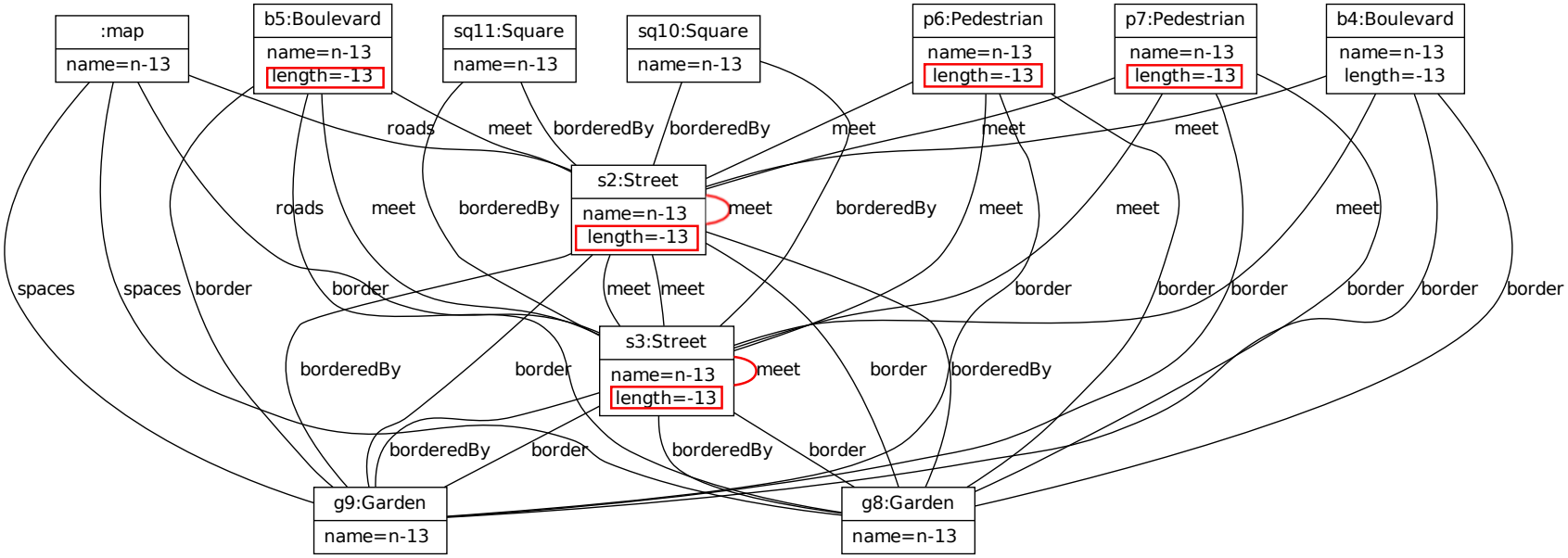
$map\_1\_spaces \in [10, 11] \cup [12, 14]$.

Results: a generated model



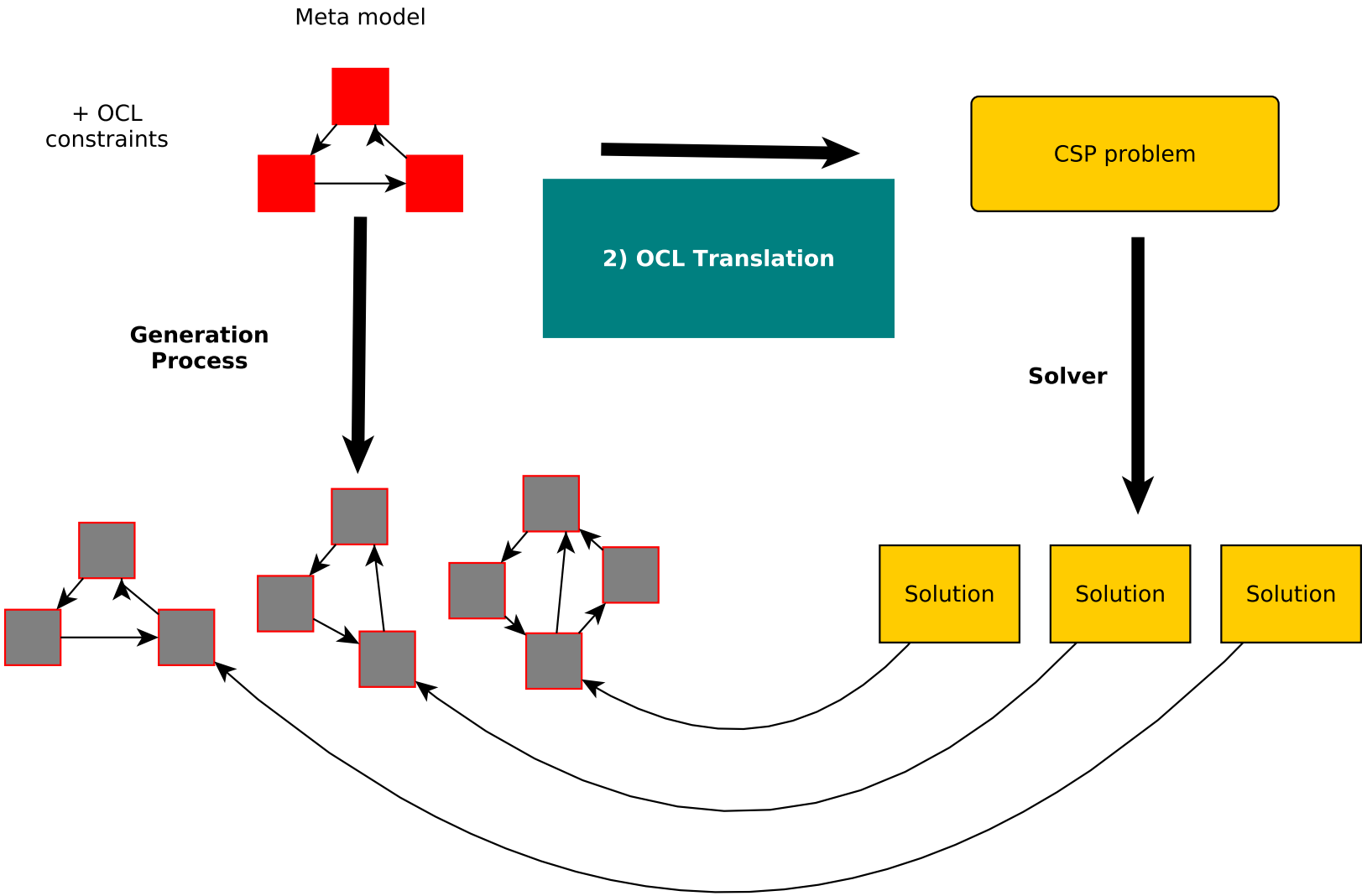
- A produced instance diagram.

Results: a generated model



- Correct it by translating OCL into CSP.

OCL Translation



OCL Translation

Remarks about Related work

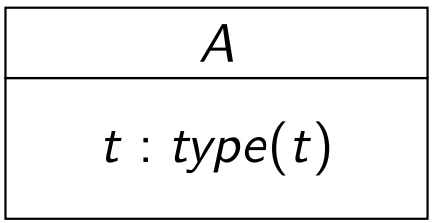
- Not many approaches are dealing with OCL.
- OCL is treated but not as efficiently as it could be.

Solution

- Find a translation corresponding to each kind of OCL construct.
- Use global constraints as frequently as possible.
- Pre-process every time it is possible.
- Avoid the use of constraints when it is not necessary.

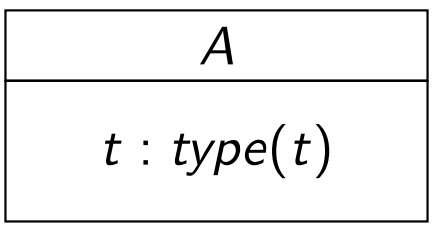
OCL Translation

$$Context\ A\ inv:\ expr(A.t)$$



OCL Translation

$$Context\ A\ inv:\ expr(A.t)$$

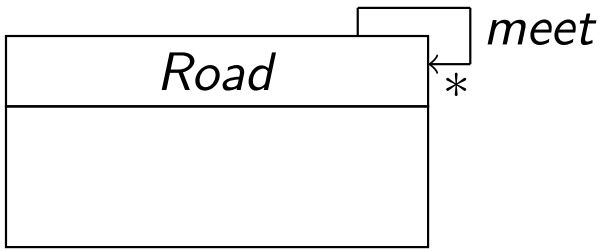


Pre-process

$$D(t) = \{e \in type(t) | expr(e)\}$$

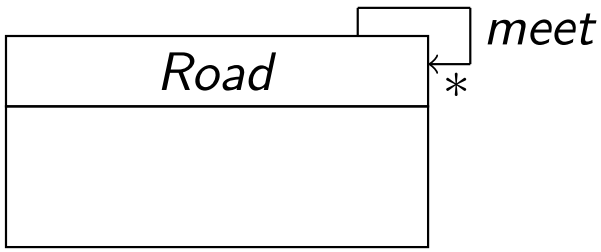
OCL Translation: Operations

$$Context\ Road\ inv:\ meet \rightarrow excludes(self)$$



OCL Translation: Operations

$$Context\ Road\ inv:\ meet \rightarrow excludes(self)$$



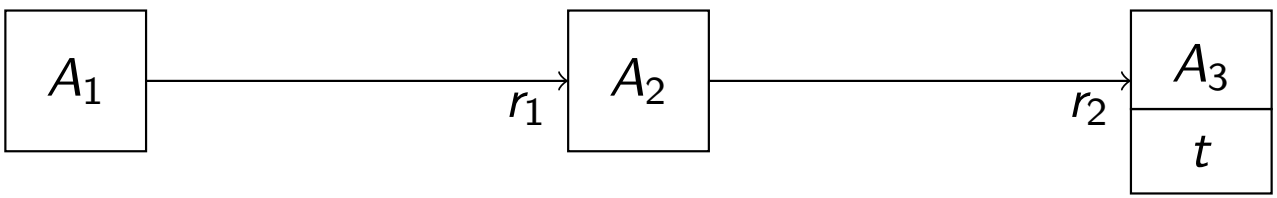
Global Constraint

$$AtMost(0, \{meet\}, self)$$

0 is the number of occurrences of value *self* in the set of variables $\{meet\}$.

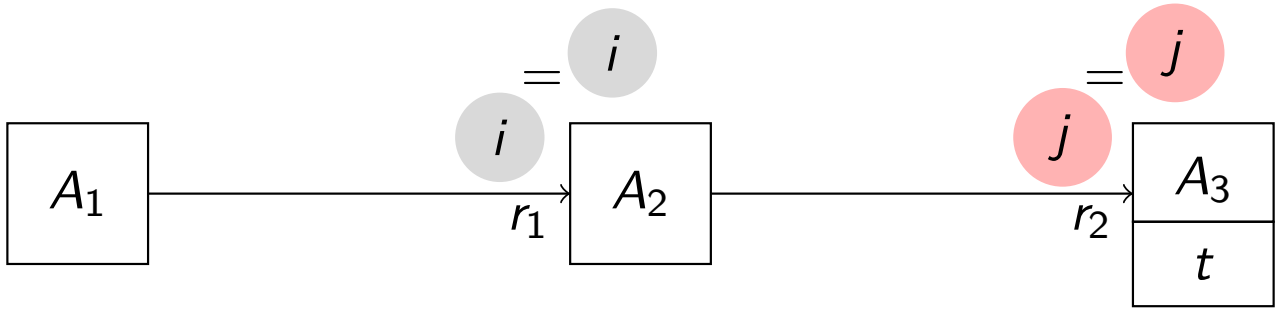
OCL Translation: Reference Navigation

$$Context\ A_1\ inv:\ r_1.r_2.expr(t)$$



OCL Translation: Reference Navigation

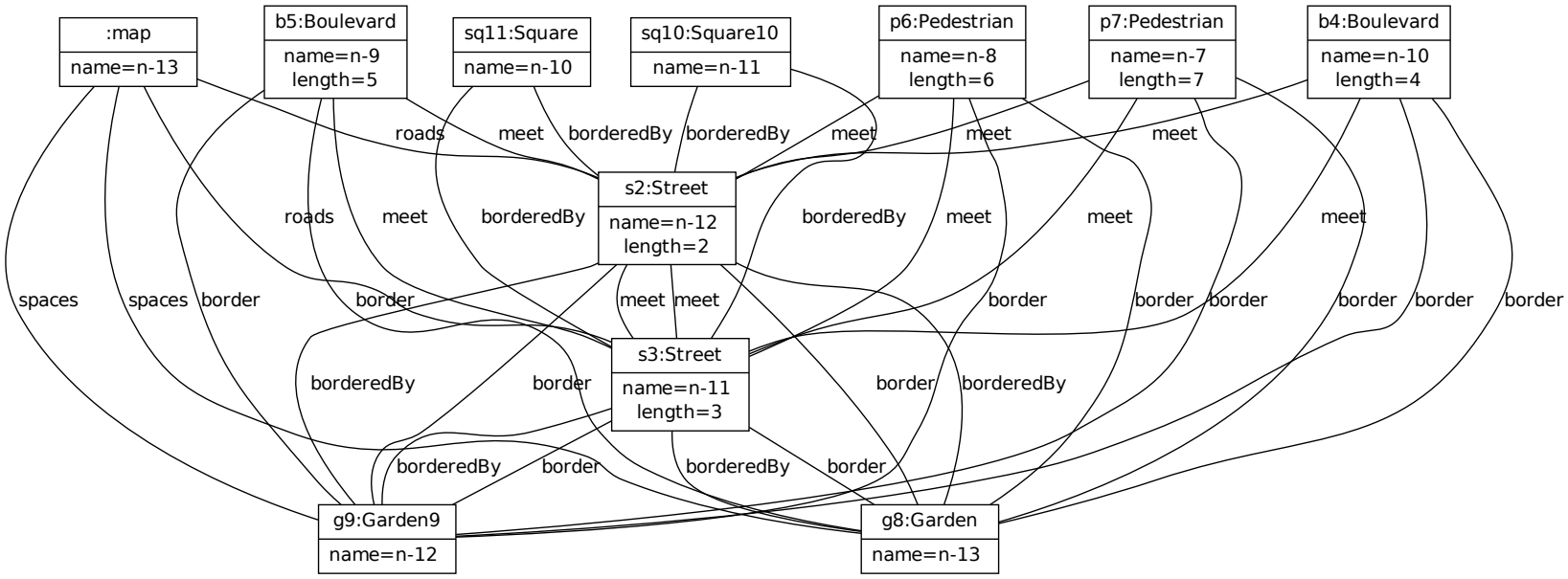
$$Context\ A_1\ inv:\ r_1.r_2.expr(t)$$



Constraint

$$(r_1 = A_2) \wedge (r_2 = A_3) \rightarrow expr(t_{A_3})$$

Results: a generated model



- A valid model conform to meta-model and OCL.

Results: experiments

Meta-model	Petri nets		ER		Feature		Graph Colour.	
#Inst.	OCL ?		OCL ?		OCL ?		OCL ?	
	Yes	No	Yes	No	Yes	No	Yes	No
50	0.39	0.38	0.38	0.36	0.38	0.38	0.36	0.33
100	0.55	0.54	0.54	0.47	0.55	0.54	0.43	0.39
300	1.65	1.64	1.42	1.26	1.65	1.64	0.62	0.47
600	14.3	14.4	7.79	7.22	14.3	14.5	0.85	0.60
Diff-erence	0.02		0.2		0.02		0.12	

Table : Comparing with and without OCL versions

Synopsis

Conclusion

Characteristics of "our" generated models

- Fully automated process.
- **Validity.** Considering the meta-model and OCL.
- **Scalability.** Tests on "big-sized" meta-models.
  - 13 meta-models of different sizes.
  - Models up to 1000 elements.
- **Relevance.** More relevance with OCL.

"Future" work

Next step

- **Relevance.** More and more relevance.
  - Comparison with real models.
- **Diversity.** Ability to produce set of *different* (*distant*) models.
  - A first functional version implemented.

The end

Results: Experiments

Meta-model	#Class	Models #class instances 50	100	200	300	600	750	1000
Graph Col.	2	0.33	0.39	0.41	0.47	0.6	0.67	0.81
Petri nets	5	0.473	0.697	1.25	2.25	38.4	115.73	192.82
ER	5	0.567	0.608	1.04	1.72	19.2	61.94	100.25
DiaGraph	5	0.492	0.551	0.71	0.96	3.1	4.28	7.71
Maps	6	0.41	0.60	1.59	4.42	36.97	112.81	400.14
Jess	7	0.357	0.633	0.88	1.55	14.1	33.4	108.53
UML class	7	0.399	0.699	1.671	9.59	99.3	260.83	599.48
Feature	8	0.38	0.54	0.57	1.64	7.11	14.5	21.75
Ecore	17	–	0.663	0.831	1.84	28.9	28.97	87.138
Business process	25	–	0.45	0.955	1.55	28.8	28.86	109.36
Sad3	**39**	–	–	–	**1.22**	**5.3**	**5.3**	**17.97**
Bmethod	**33**	–	–	–	**0.93**	**2.9**	**2.9**	**18.77**
Royal&Loyal	**34**	–	–	–	**0.75**	**1.6**	**1.62**	**3.95**
BiBtex	**28**	–	–	–	**0.78**	**1.6**	**3.21**	**7.11**

Table : Resolution time (seconds)