IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Efficient model similarity estimation with robust hashing

P4S
Seminar
21 Octobre 2021

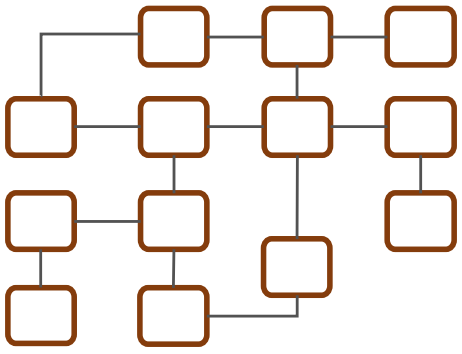
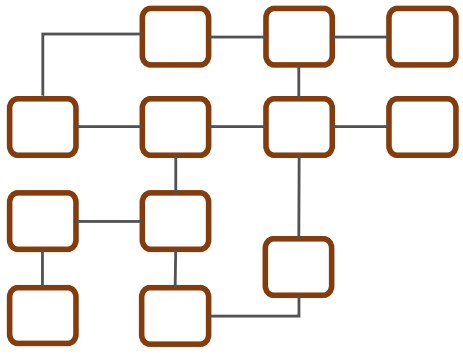
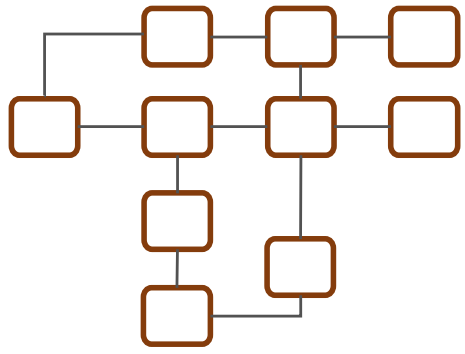
What do we want to do ?

What do we want to do ?

Estimate the similarity between a set of models

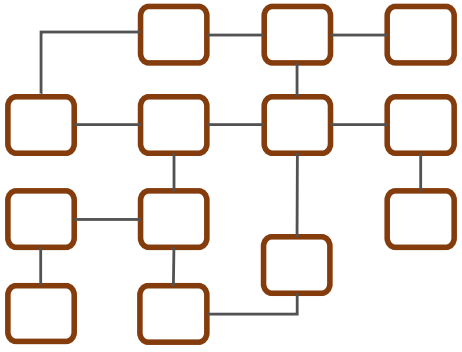
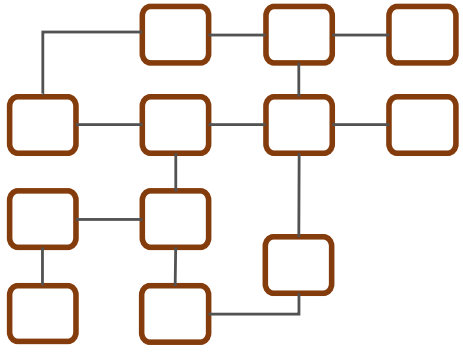
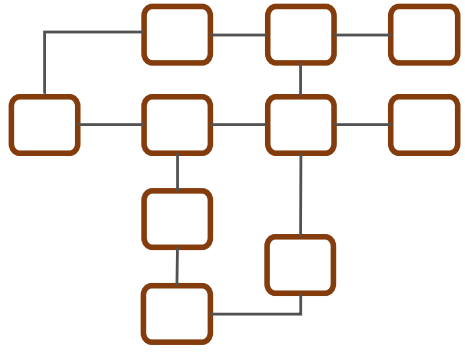
What do we want to do ?

Estimate the similarity between a set of models

What do we want to do ?

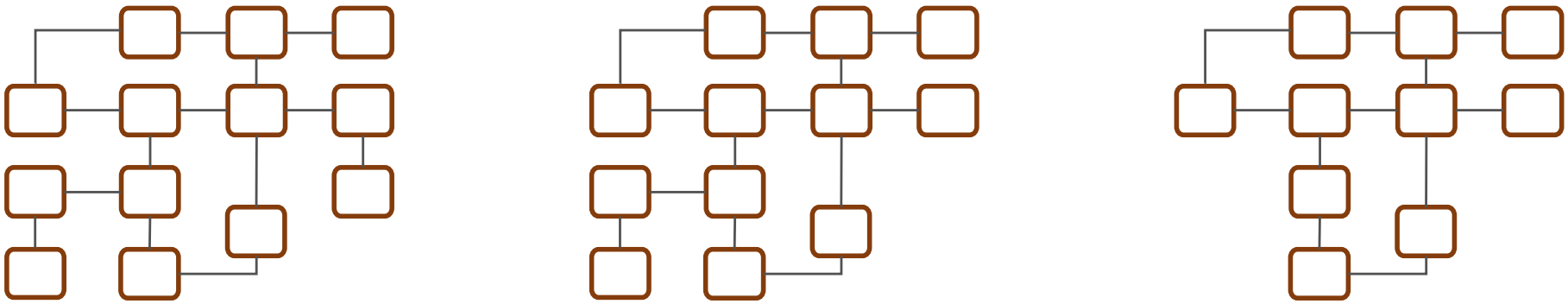
Estimate the similarity between a set of models

- We do not want to compare manually
- We do not want to compare graphs (hard problem)
- We want to compare 100, 1000, 1000000 models.

What do we want to do ?

Estimate the similarity between a set of models



- We do not want to compare manually
- We do not want to compare graphs (hard problem)
- We want to compare 100, 1000, 1000000 models.

Idea : Transform the models to something else.

- Something easier to compare.
- Something that preserves *locality*

We could use (cryptographic) hashing but... small changes lead to very different hashes.

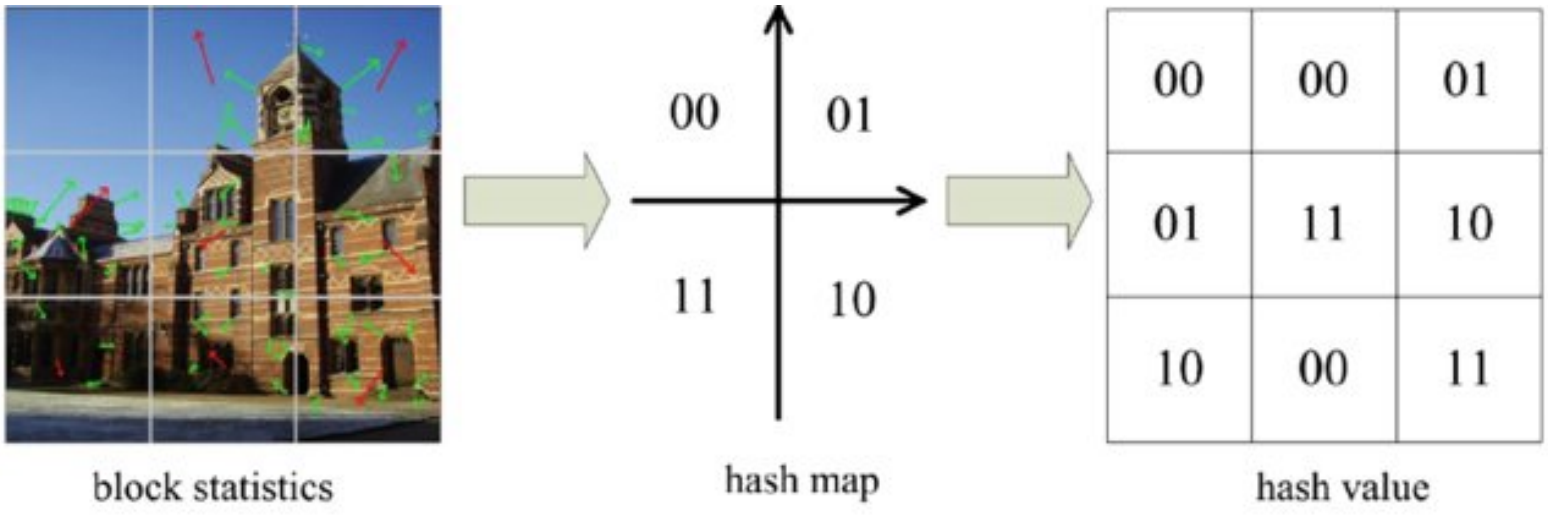
We could use (cryptographic) hashing but... small changes lead to very different hashes.

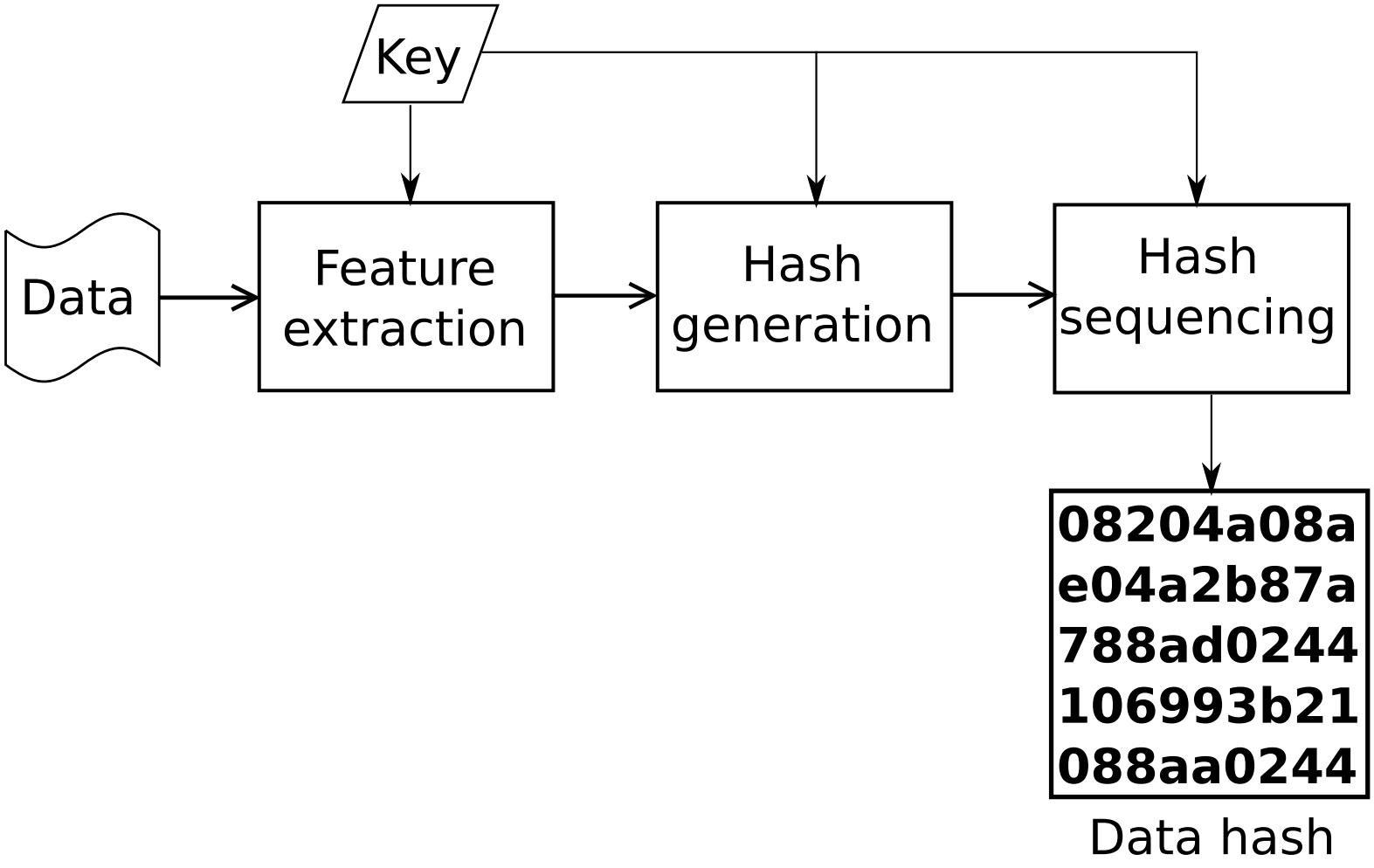
- Locality Sensitive hashing.
- Robust Hashing : locality preserving + resistant to *attacks.*

Solution : (robust) hashing

We could use (cryptographic) hashing but... small changes lead to very different hashes.

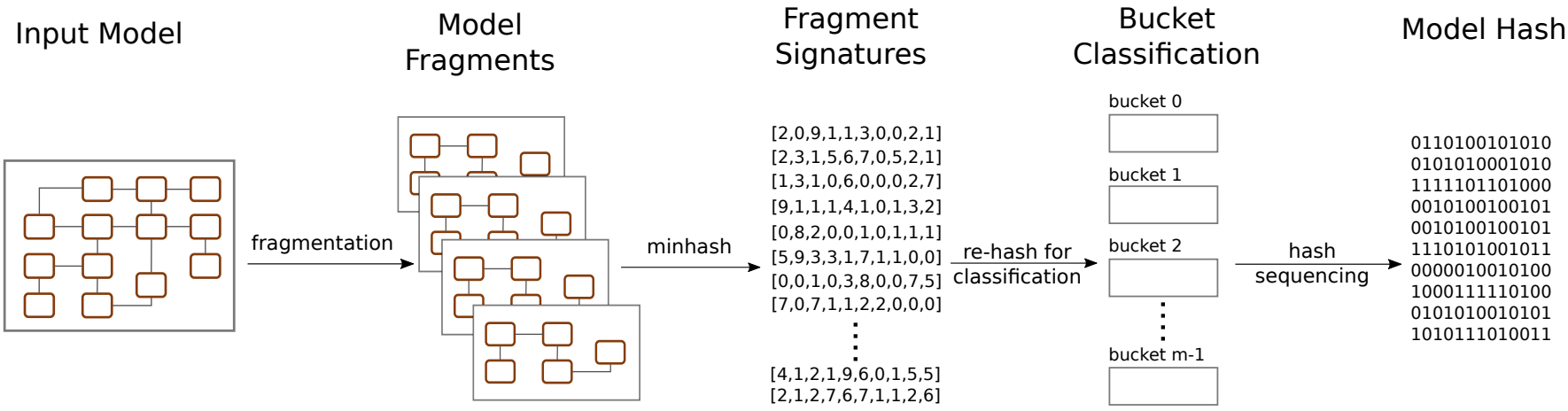
- Locality Sensitive hashing.
- Robust Hashing : locality preserving + resistant to *attacks.*




Key
Data
Feature extraction
Hash generation
Hash sequencing
08204a08a
e04a2b87a
788ad0244
106993b21
088aa0244
Data hash

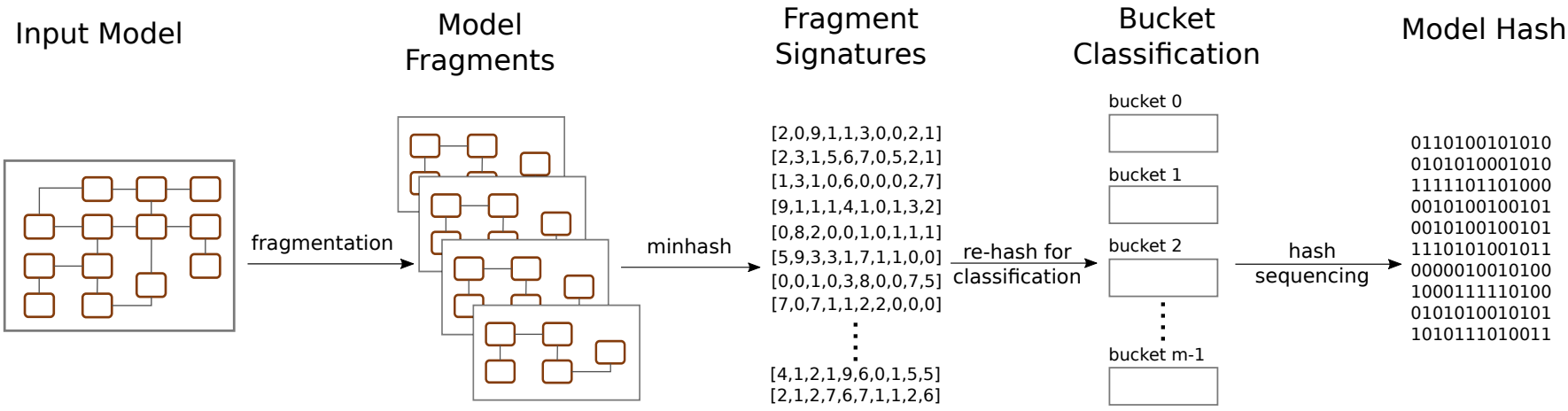
- Independent of the storage format
- Independent of the graphical layout
- Independent of the concrete syntax
- Must take into account not only *content* but also *structure*.

Approach



- Model fragmentation.
  - to record structure and not only content
  - created independently
  - many, and possibly overlapping

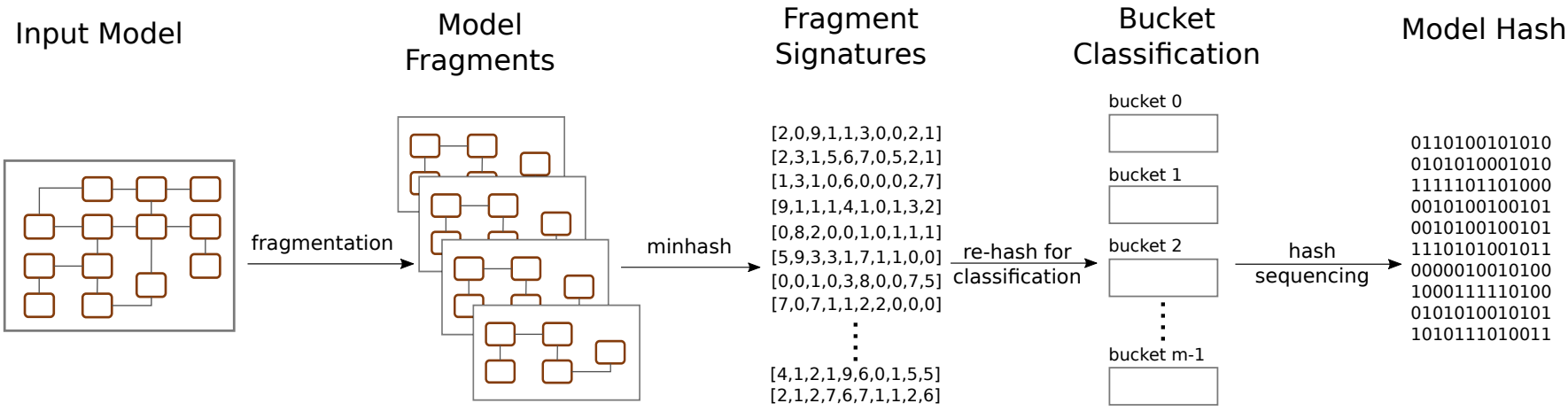
Approach



- Model fragmentation.
- MINHASH Signatures.

- We translate fragments to words
- We use MINHASH to translate fragments to signatures
- We preserve locality

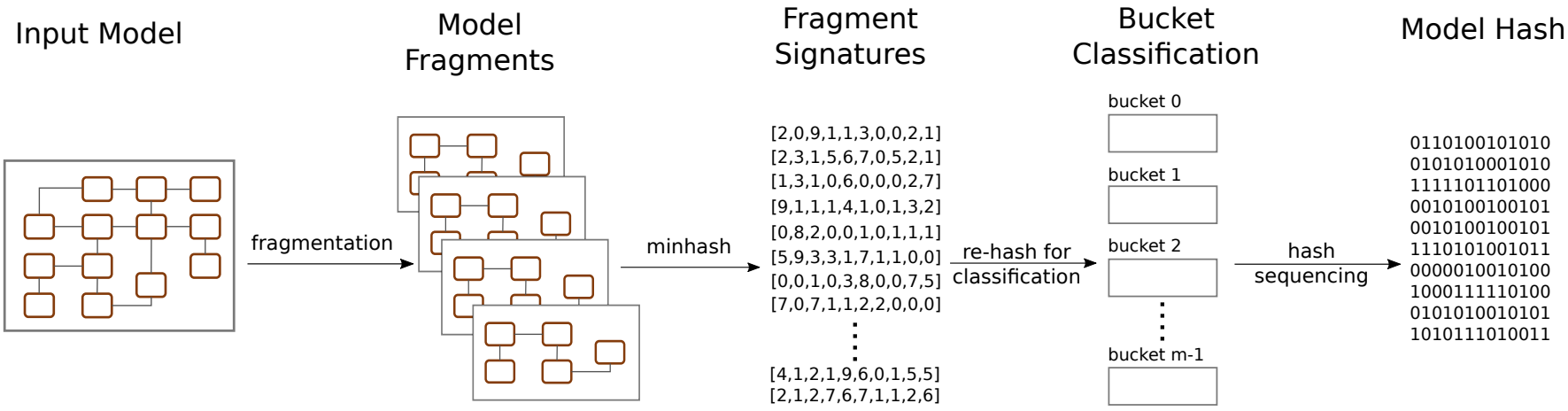
Approach



- Model fragmentation.
- MINHASH Signatures.
- Bucket classification.

- Similar fragments go to the same bucket
- We minimize mutation effect
- We prevent mutation propagation

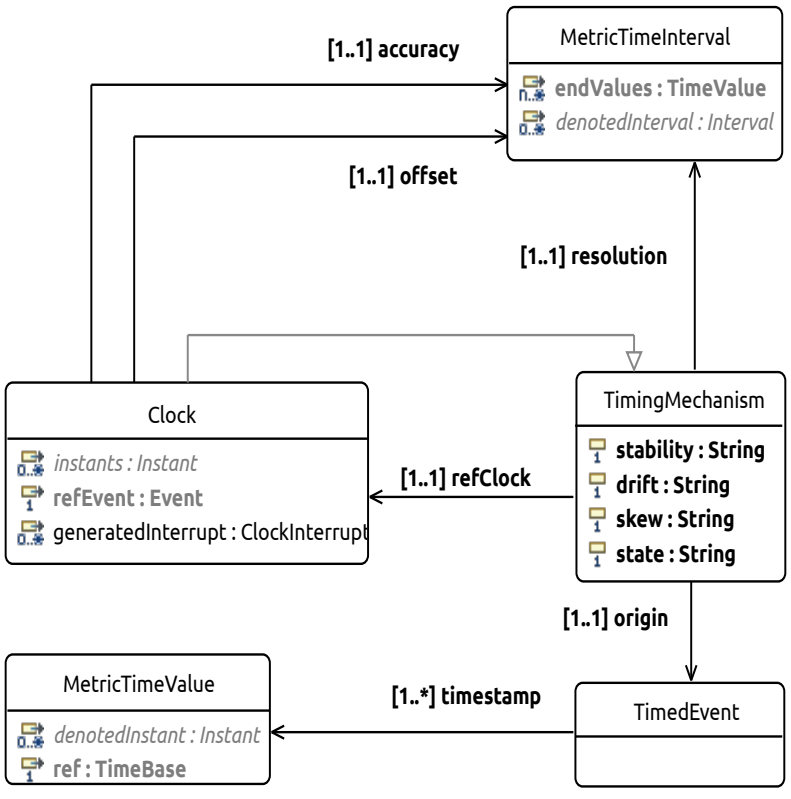
Approach



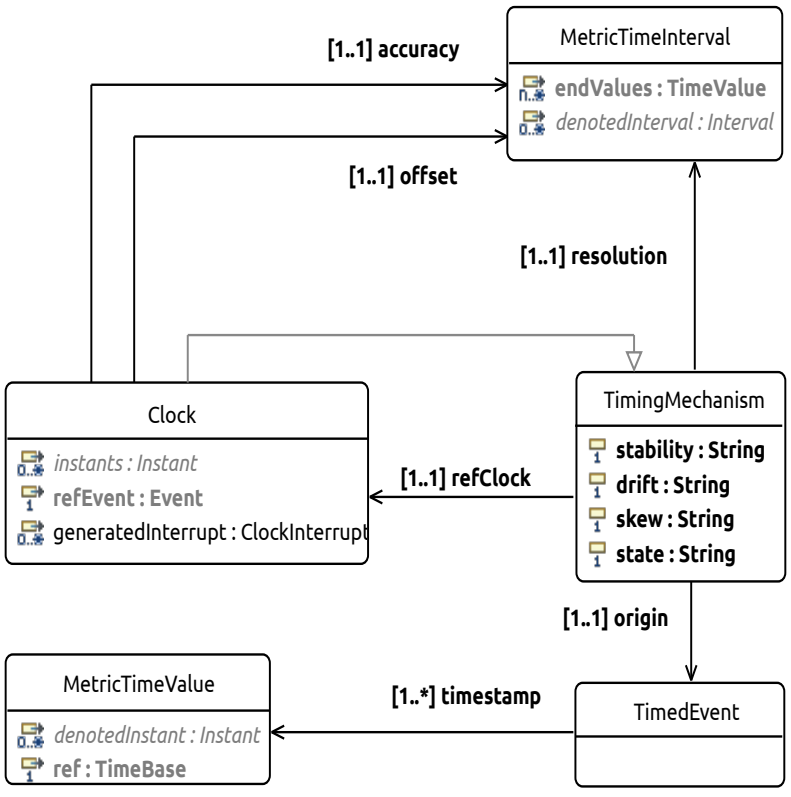
- Model fragmentation.
- MINHASH Signatures.
- Bucket classification.
- Hash assembling.

- We pick a number of signatures
- Different bucket means the fragment is different enough
- We prevent mutation propagation

Approach : fragment example



Approach : fragment example

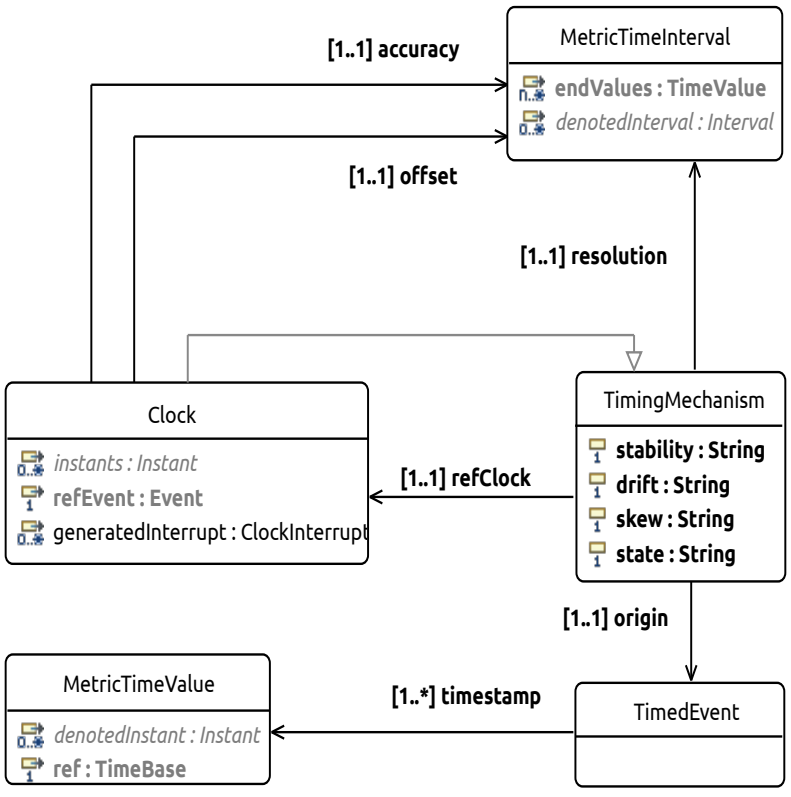


Fragment A Summary : {TimingMechanism, TimedEvent, state, Clock, MetricTimeValue, stability, MetricTimeInterval, drift, skew}

Fragment A Minhash Signature :

[3, 1, 2, 3, 0, 6, 1, 2, 4, 5, 1, 2, 0, 10, 0, 0, 5, 0, 2, 5]

Approach : fragment example



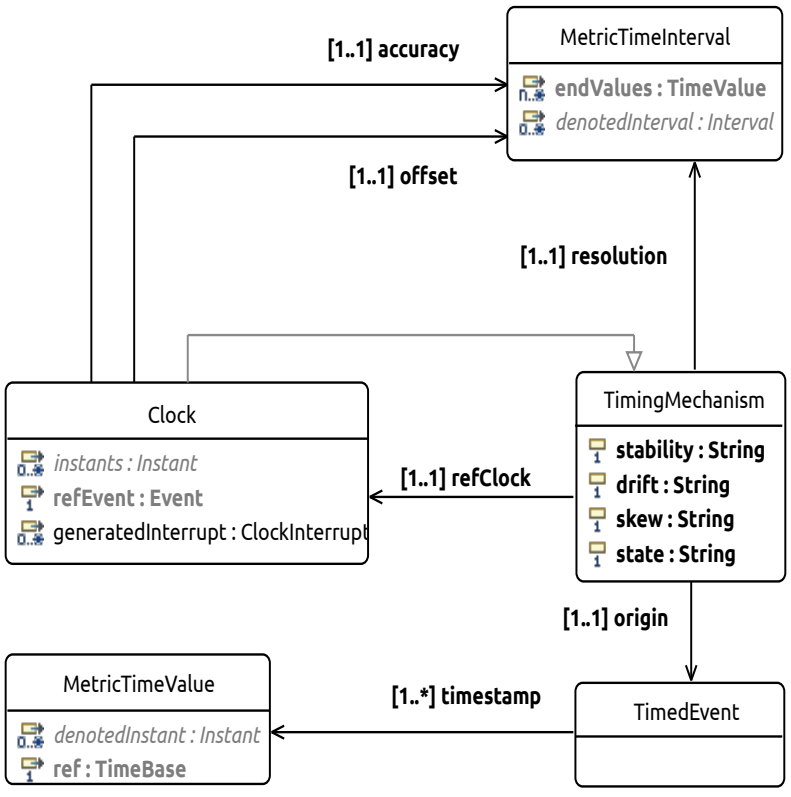
Fragment A Summary : {TimingMechanism, TimedEvent, state, Clock, MetricTimeValue, stability, MetricTimeInterval, drift, skew}

Fragment A Minhash Signature :
[3, 1, 2, 3, 0, 6, 1, 2, 4, 5, 1, 2, 0, 10, 0, 0, 5, 0, 2, 5]

Mutated Fragment A Summary :
{TimingMechanism, TimedEvent, state, MetricTimeValue, stability, MetricTimeInterval, drift, skew}

Mutated Fragment A Minhash Signature :
[3, 1, 2, 3, 0, 9, 1, 2, 4, 5, 1, 2, 0, 13, 0, 0, 5, 0, 2, 5]

Approach : fragment example



Fragment A Summary : {TimingMechanism, TimedEvent, state, Clock, MetricTimeValue, stability, MetricTimeInterval, drift, skew}

Fragment A Minhash Signature :
[3, 1, 2, 3, 0, 6, 1, 2, 4, 5, 1, 2, 0, 10, 0, 0, 5, 0, 2, 5]

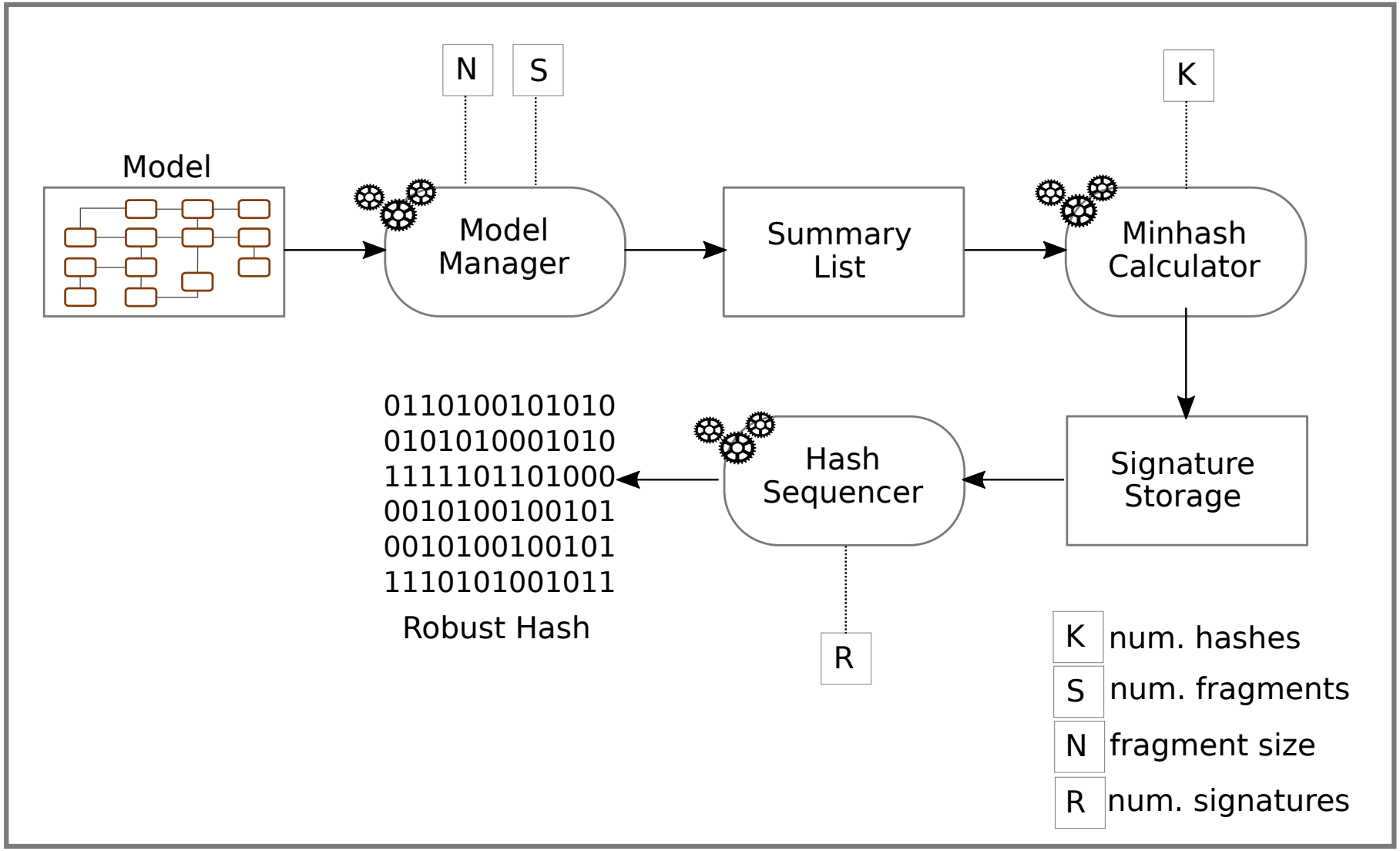
Mutated Fragment A Summary :
{TimingMechanism, TimedEvent, state, MetricTimeValue, stability, MetricTimeInterval, drift, skew}

Mutated Fragment A Minhash Signature :
[3, 1, 2, 3, 0, 9, 1, 2, 4, 5, 1, 2, 0, 13, 0, 0, 5, 0, 2, 5]

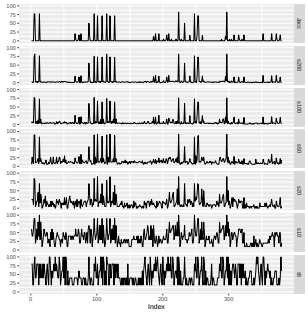
Fragment B Summary :
{NFPCategory, QualitativeNFP, NFPLibrary, AnnotatedModelElement, Quantity, NFP}

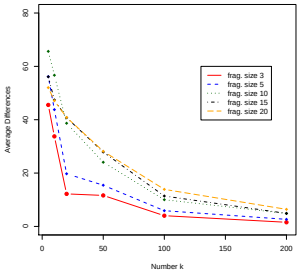
Fragment B Minhash Signature :
[5, 8, 5, 1, 17, 4, 11, 12, 1, 2, 15, 3, 1, 3, 1, 4, 0, 2, 2, 4]

Approach : parameters

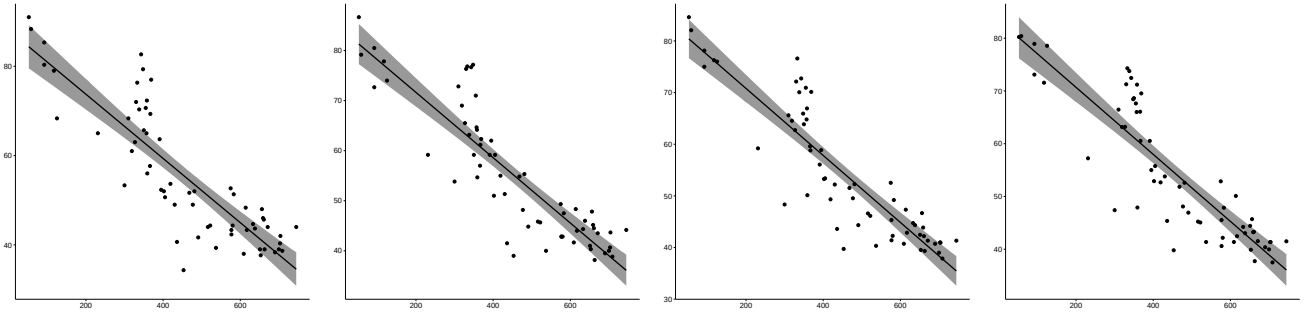


Evaluation of parameters... booring

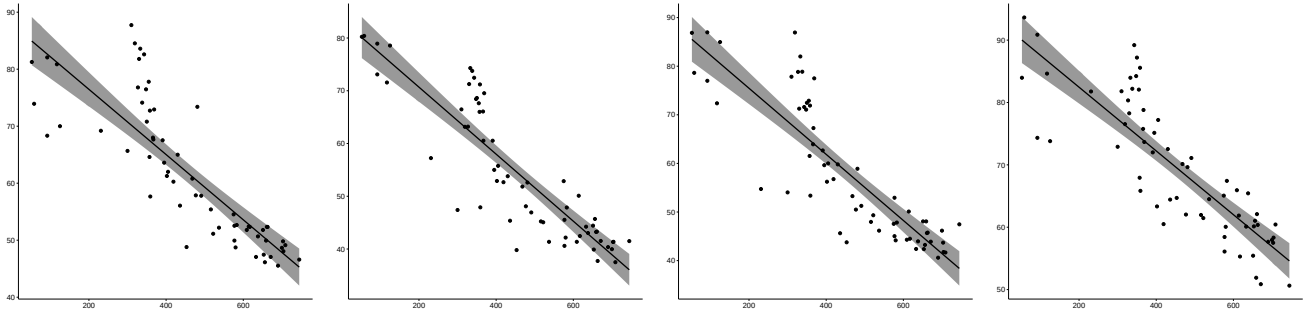




K > 15 and bigger as the fragment size (N) grows.



N (fragment size) < 10 (or raise K)



For models up to 1000 elements S > R and R > 20

- Discrimination : tested by calculating the similarity of *known* different models.
- Robustness : tested by automatic mutation of two (random) models.

TABLE – Pairwise Similarity for Petri Nets

	a	b	c	d	e	f	g	h	i	j	k	l
a	100	40	42	41	39	41	40	40	43	40	40	38
b	40	100	42	43	45	40	40	40	41	43	41	39
c	42	42	100	44	42	45	45	44	43	42	40	39
d	41	43	44	100	42	44	42	42	43	42	41	39
e	39	45	42	42	100	44	41	45	42	42	41	42
f	41	40	45	44	44	100	44	44	42	41	42	40
g	40	40	45	42	41	44	100	43	44	44	42	39
h	40	40	44	42	45	44	43	100	44	44	42	40
i	43	41	43	43	42	42	44	44	100	71	69	41
j	40	43	42	42	42	41	44	44	71	100	66	40
k	40	41	40	41	41	42	42	42	69	66	100	40
l	38	39	39	39	42	40	39	40	41	40	40	100

*a :01_petri.xmi ; b :02_petri.xmi ; c :03_petri.xmi ; d :04_petri.xmi ; e :05_petri.xmi ; f :06_petri.xmi ; g :07_petri.xmi ; h :08_petri.xmi ; i :09_petri_00.xmi ; j :09_petri_05.xmi ; k :09_petri_30.xmi ; l :09_petri_800.xmi ;

Does it work ?

- Discrimination : tested by calculating the similarity of *known* different models.
- Robustness : tested by automatic mutation of two (random) models.

TABLE – Pairwise Similarity for Metamodels

	a	b	c	d	e	f	g	h	i	j	k	l
a	100	11	14	12	8	16	9	10	7	12	11	11
b	11	100	12	10	10	12	12	9	5	16	9	10
c	14	12	100	11	9	13	11	10	6	15	11	10
d	12	10	11	100	9	11	9	9	7	11	82	62
e	8	10	9	9	100	13	10	8	6	13	8	9
f	16	12	13	11	13	100	11	10	7	16	10	10
g	9	12	11	9	10	11	100	9	6	13	8	10
h	10	9	10	9	8	10	9	100	7	11	8	8
i	7	5	6	7	6	7	6	7	100	6	7	7
j	12	16	15	11	13	16	13	11	6	100	10	11
k	11	9	11	82	8	10	8	8	7	10	100	66
l	11	10	10	62	9	10	10	8	7	11	66	100

*a :architecture.ecore ; b :BusinessDomainDsl.ecore ; c : CASL.ecore ; d :JKind.ecore ; e :svg.ecore ; f :jobSearch.ecore ; g : Language.ecore ; h :Rell.ecore ; i :restrain.ecore ; j :xtce.ecore ; k :JKind.ecore ; l :JKind.ecore ;

- Discrimination : tested by calculating the similarity of *known* different models.
- Robustness : tested by automatic mutation of two (random) models.

TABLE – Pairwise Similarity for Model Transformations

	a	b	c	d	e	f	g	h	i	j	k	l
a	100	95	54	45	40	45	44	43	45	44	41	47
b	95	100	54	45	40	45	45	43	45	43	41	47
c	54	54	100	47	40	44	46	43	44	44	43	47
d	45	45	47	100	42	44	44	42	44	43	42	45
e	40	40	40	42	100	43	43	41	43	42	41	42
f	45	45	44	44	43	100	46	42	47	45	44	45
g	44	45	46	44	43	46	100	42	47	46	46	46
h	43	43	43	42	41	42	42	100	43	42	41	43
i	45	45	44	44	43	47	47	43	100	48	46	46
j	44	43	44	43	42	45	46	42	48	100	44	47
k	41	41	43	42	41	44	46	41	46	44	100	42
l	47	47	47	45	42	45	46	43	46	47	42	100

*a :tr01_m0.xmi ; b :tr01_m1.xmi ; c :tr01_m2.xmi ; d :tr02.xmi ; e :tr03.xmi ; f :tr04.xmi ; g :tr05.xmi ; h :tr06.xmi ; i :tr07.xmi ; j :tr08.xmi ; k :tr09.xmi ; l :tr10.xmi ;

Application Scenarios [1]

1. Plagiarism detection. ✓
2. Model diversity.
3. Model Indexing ?
4. IP protection ? (by registering hashes in a ledger...)

1. now that we have a hammer...