30th September, 2014
14th International Workshop on
OCL and Textual Modeling
Applications and Case Studies

IK4 IKERLAN
Research Alliance

40urte 1974 • 2014

MQT, an Approach for Runtime Query Translation: From EOL to SQL

Xabier De Carlos | Goiuria Sagardui | Salvador Trujillo
[xdecarlos@ikerlan.es]

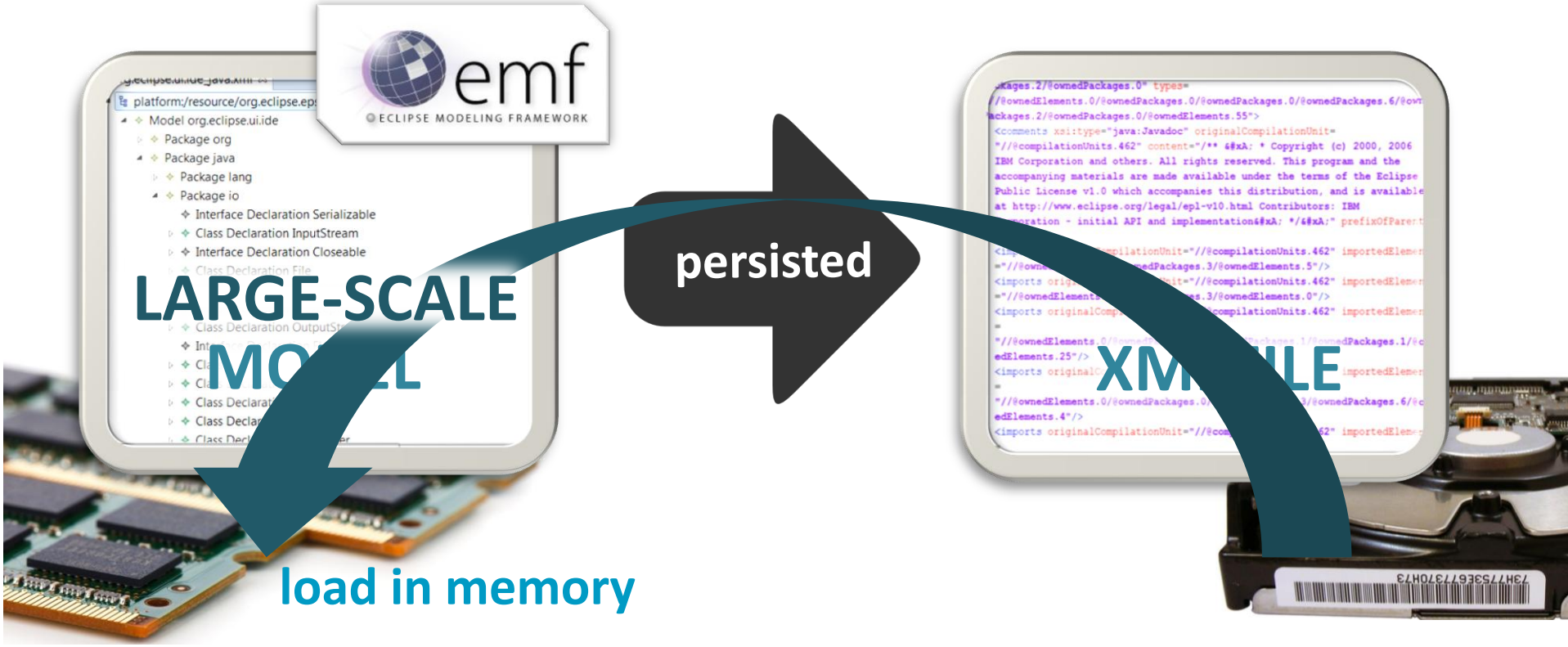
IN THIS PRESENTATION...

- Background and Motivation
- MQT
- Preliminary Evaluation
- Related Work
- Conclusions and Future Work

QUERYING MODELS

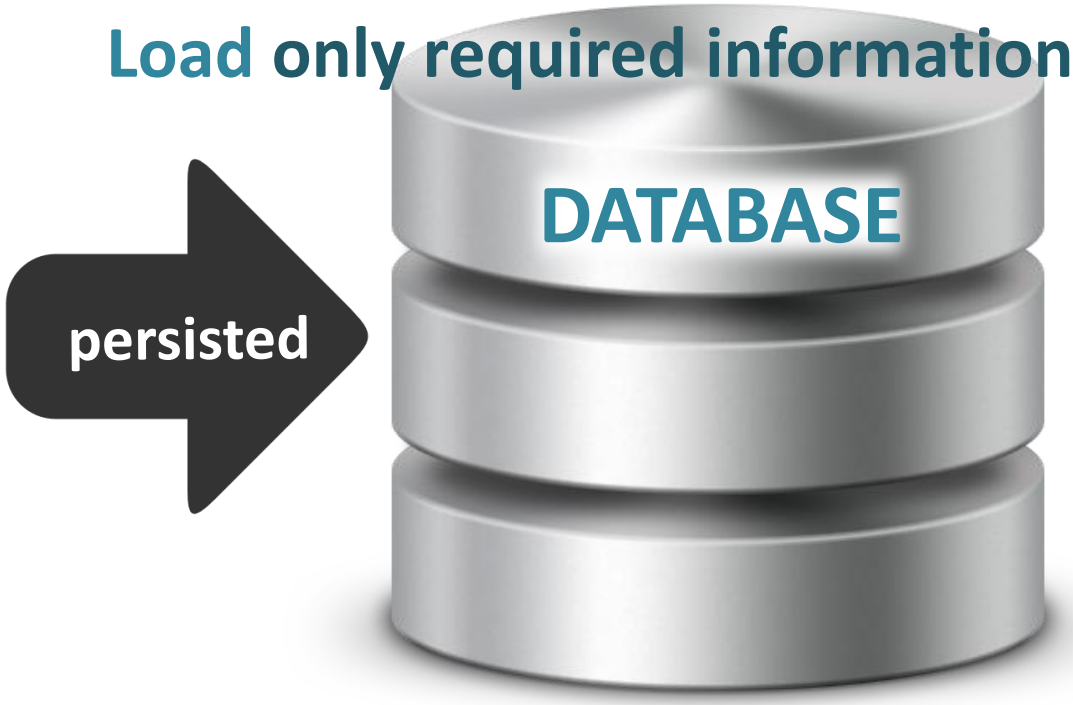
"XMI-based serialization in EMF results to be extremely inefficient"

[Benelallam, A., Gómez, A., Sunyé, G., Tisi, M., & Launay, D. (2014, July). ***Neo4EMF, a Scalable Persistence Layer for EMF Models.*** *In ECMFA-European conference on Modeling Foundations and applications.]*



QUERYING LARGE-SCALE MODELS

Load only required information.



MORSA, EMF Fragments, Neo4EMF, CDO, MongoEMF, etc.

Persistence-level query languages

- Leverage capabilities of persistence.
- Persistence-specific and dependent.
- For example: MorsaQL, SQL, Cypher,etc.

MOTIVATION

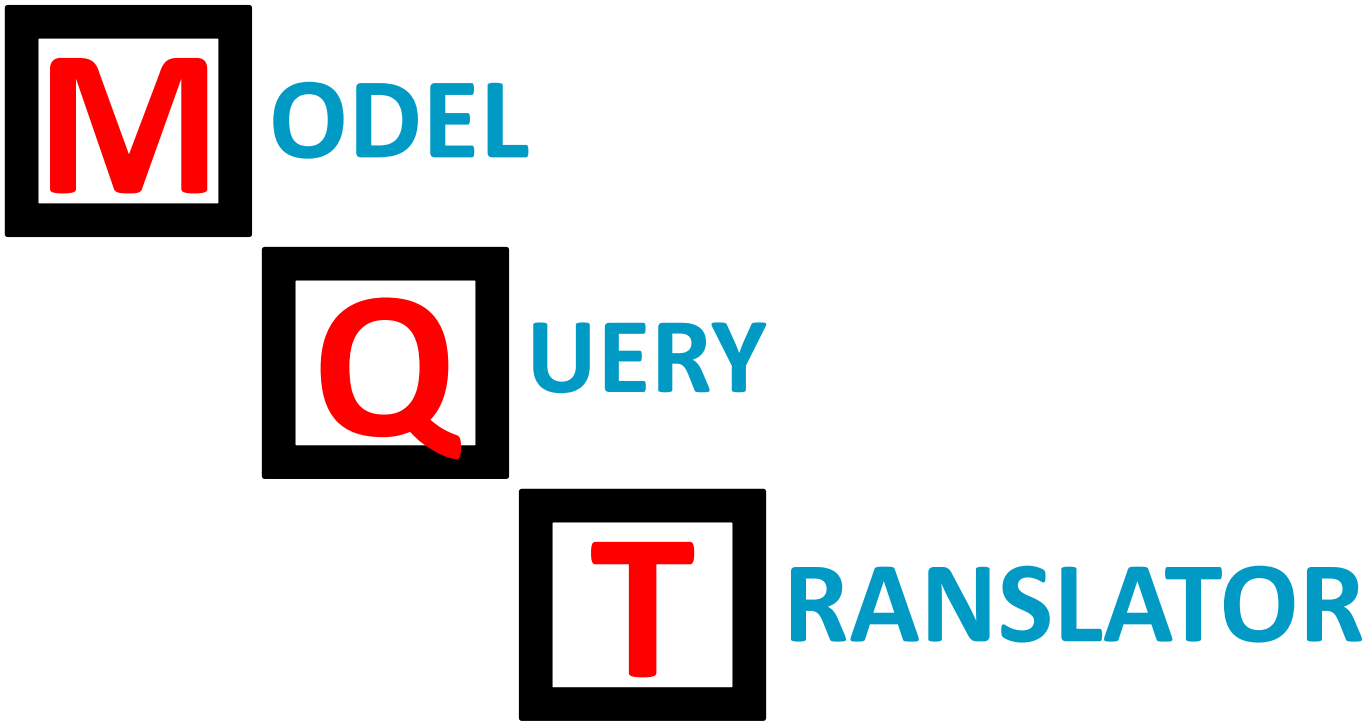
PROBLEM. Model-Level query languages are closer to modelling engineers but they do not have the efficiency of persistence-level query languages to query large models persisted in databases.

MOTIVATION

CHALLENGE. Use a model-level query language with the efficiency of a persistence-level language.

PROPOSED SOLUTION.

Automate query translation from model-level to persistence-level


MODEL
QUERY
TRANSLATOR

MQT: MODEL QUERY TRANSLATOR

Runtime Translation

EOL to SQL

Based on a metamodel-agnostic data-schema

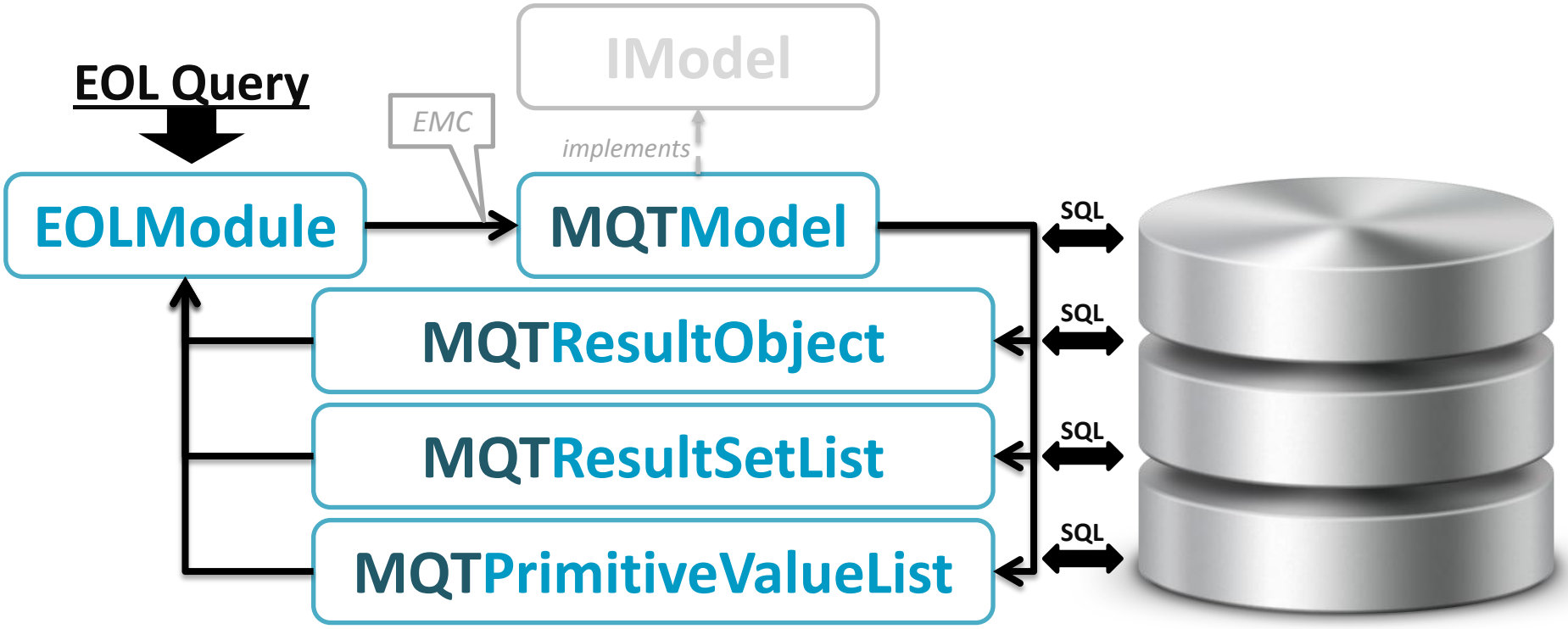
Work based on *"An Approach for Efficient Querying of Large Relational Datasets with OCL-based Languages"* [D.S. Kolovos D.S., R. Wei, K. Barmpis In XM'13]

MQT uses EMC

for EOL Query↔Model interaction

Naive and Custom translation

MQT OVERVIEW



MQT+NAIVE TRANSLATION

- Based on naive translation provided by EMC.
- Each query expression translated and executed one-by-one

EClass.all.select(...)

1) *Parses and translates* **EClass.all:**
   - *New instance of MQTResultSetList*
   - *Executes constructed SQL query*
2) *Parses and translates* **.select(...):**
   - *Executes a SQL query for each result of the list to check the condition.*

MQT+CUSTOM TRANSLATION

- MQTResultSetList implements IAbstractOperationContributor, overriding translation of select, collect, reject, etc.
- Group dependent queries into a single translated SQL query to be executed once.

EClass.all.select(...)

1) *Parses and translates* **EClass.all:**
   - *New instance of MQTResultSetList*
2) *Parses and translates* **.select(...):**
   - *Completes query construction with the select condition*
   - *Executes query and return results.*

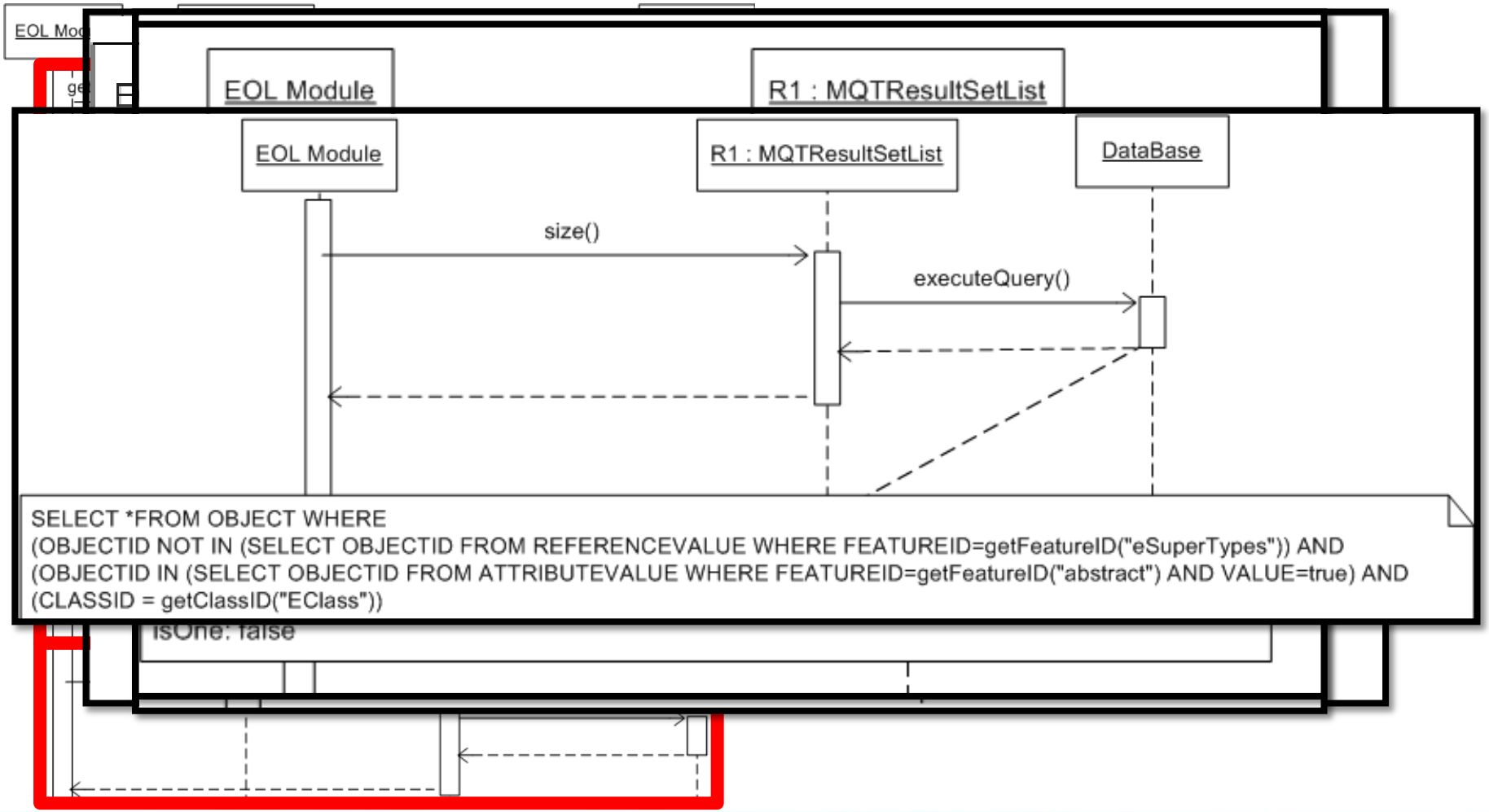
MQT CUSTOM TRANSLATION EXAMPLE

- Translation example:
  - Translation example of custom translation
  - Compare with naive translation
- Query:

```
1 var list = EClass.all.select(c|c.abstract=true);
2 var list2 = list.select(c|c.eSupertypes=null);
3 list2.size().println();
```

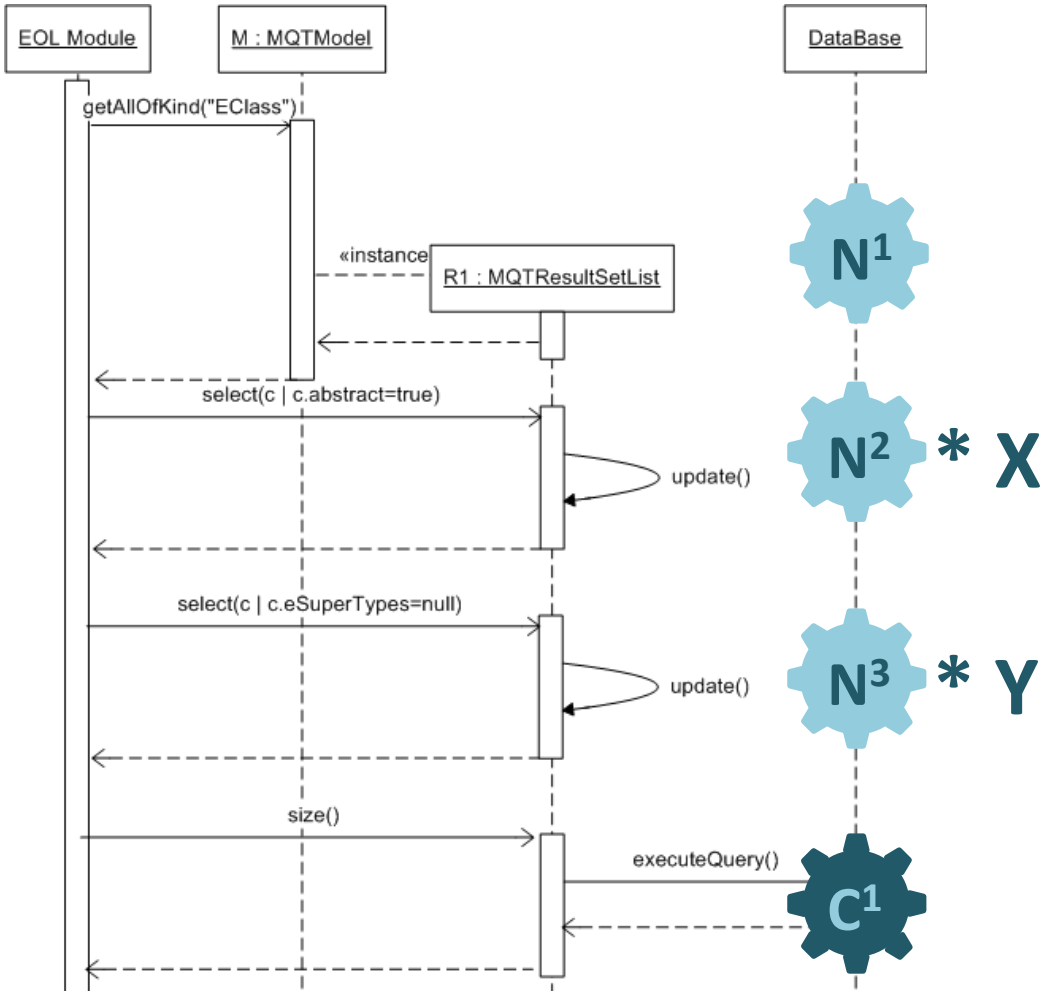
MQT CUSTOM TRANSLATION EXAMPLE

```
1 var list = EClass.all.select(c|c.abstract=true);
2 var list2 = list.select(c|c.eSupertypes=null);
3 list2.size().println()
```



MQT CUSTOM TRANSLATION EXAMPLE

```
1 var list = EClass.all.select(c|c.abstract=true);
2 var list2 = list.select(c|c.eSupertypes=null);
3 list2.size().println();
```



MQT+CUSTOM = C^1

MQT+NAIVE= $N^1 + N^2*X + N^3*Y$

MQT PRELIMINARY EVALUATION

- **Models: five, from 45MB to 403MB**
  - Created using Java Discoverer of MoDISCO
  - Models conform to a JAVA metamodel
  - Persisted in a relational DB with a metamodel-agnostic schema and using H2 database back-end
- **Query: identify singleton classes**
  - Based on the GraBats'09 Reverse Engineering Contest
  - EOL
- **Execution: 100 times**
  - MQT+Naive
  - MQT+Custom

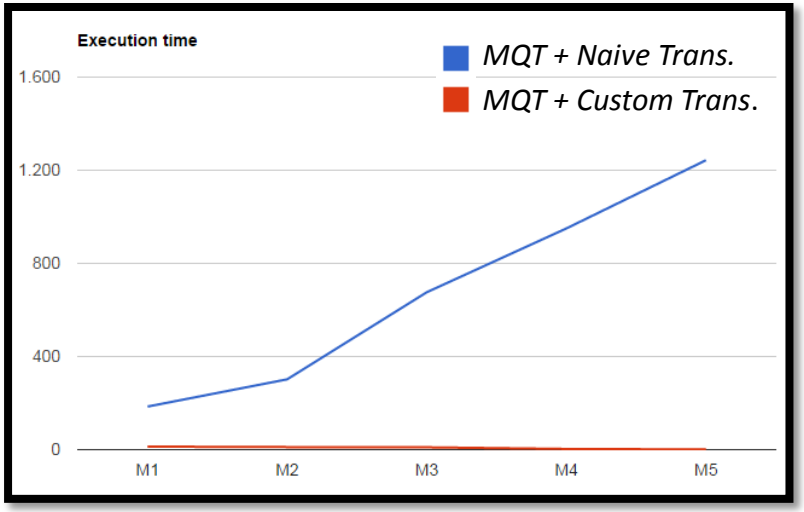
```
1  "Grabats query executed: singleton extraction from a modisco model".println();
2  var method = MethodDeclaration.all.select(m | m.name="getInstance");
3  method = method.select(m | m.modifier<>null);
4
5  var mod;
6  var singletons = new Sequence;
7  for(m in method){
8      mod = m.modifier;
9      if(mod.static = true and mod.visibility.literal =="public"){
10         singletons.add(m.abstractTypeDeclaration);
11     }
12 }
13
14 "Singleton classes:".println();
15 for(c in singletons){
16     c.name.println();
17 }
```

MQT PRELIMINARY EVALUATION

Results

Custom translation more scalable than naive.

	M1	M2	M3	M4	M5
size (MB)	45	72	212	327	403
# objects	165741	330761	875988	1343207	1566890
# methods	5366	8129	11393	15386	19366
# singleton classes	9	8	6	0	0
MQT+naive	185ms	302ms	676ms	950ms	1243ms
MQT+custom	13ms	11ms	10ms	3ms	1ms



Correctness of the query results:

- Execute query against models persisted in XMI

Query translation time:

- M1 3.35ms | M2 4,51ms | M3 0.77ms | M4 0.8ms | M5 0.64ms

RELATED WORK

- ***A Framework for Generating Query Language Code from OCL Invariants*** *[by F. Heidenreich, C. Wende, and B. Demuth]*
  - Generate SQL queries from OCL invariants.
- ***OCL as a Specification Language for Business Rules in Database Applications*** *[by B. Demuth, H. Hussmann and S. Loecher]*
  - Generate views from OCL constraints, and use views to check integrity of persisted data.
- ***A DBMS-Based Approach for Automatic Checking of OCL Constraints*** *[by U. Marder, N. Ritter, H. Steiert]*
  - A similar approach for integrity checking.

- **While these approaches translate queries at compilation-time, our approach performs translation at runtime.**

CONCLUSIONS

- **MQT: approach for runtime translation of EOL queries to SQL.**
- **MQT prototype:**
  - Supports read-only EOL expressions.
  - Modification expressions are not supported.
- **MQT preliminary evaluation:**
  - MQT+Custom translation more scalable than MQT+naive translation
  - Need to perform a more complete evaluation

FUTURE WORK

- Extend MQT with support for:
  - Modification expressions.
  - Additional model-level query languages (e.g. OCL)
  - Additional persistence-level query languages (e.g. Cypher)
- Evaluation:
  - Compare with XMI
  - More complex queries
- Open issues: how to provide extensibility to facilitate the integration of new query languages.

40urte
1974 • 2014

THANK YOU.
QUESTIONS?

30th September, 2014
14th International Workshop on
OCL and Textual Modeling
Applications and Case Studies

IK4 IKERLAN
Research Alliance

40urte 1974 • 2014

MQT, an Approach for Runtime Query Translation: From EOL to SQL

Xabier De Carlos | Goiuria Sagardui | Salvador Trujillo
[xdecarlos@ikerlan.es]