



HAL
open science

Neo4EMF : when big models are no longer an issue

Amine Benelallam, Hugo Bruneliere

► **To cite this version:**

Amine Benelallam, Hugo Bruneliere. Neo4EMF : when big models are no longer an issue. EclipseCon France 2014, Jun 2014, Toulouse, France. hal-01026219

HAL Id: hal-01026219

<https://inria.hal.science/hal-01026219>

Submitted on 11 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Neo4EMF: When big models are no longer an issue!

Amine Benelallam, Hugo Brunelière
AtlanMod Team (Inria, Mines Nantes, LINA)

Big data, big models...



Time Vs. Memory



Finding the
right
balance?



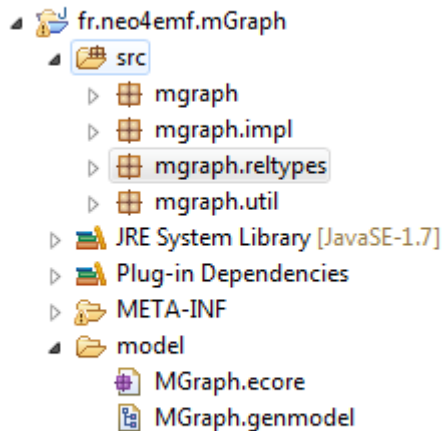
...Neo4EMF!



$$\text{Neo4EMF} = \text{EMF} + \text{Neo4j}$$



Neo4EMF features



Java code
generation

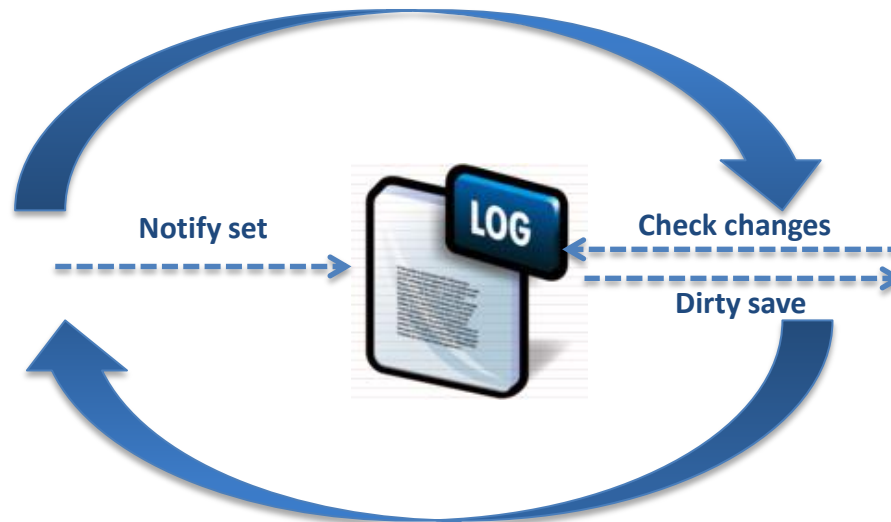
On-demand
loading

Improved
querying

Neo4EMF overview



Create / save data

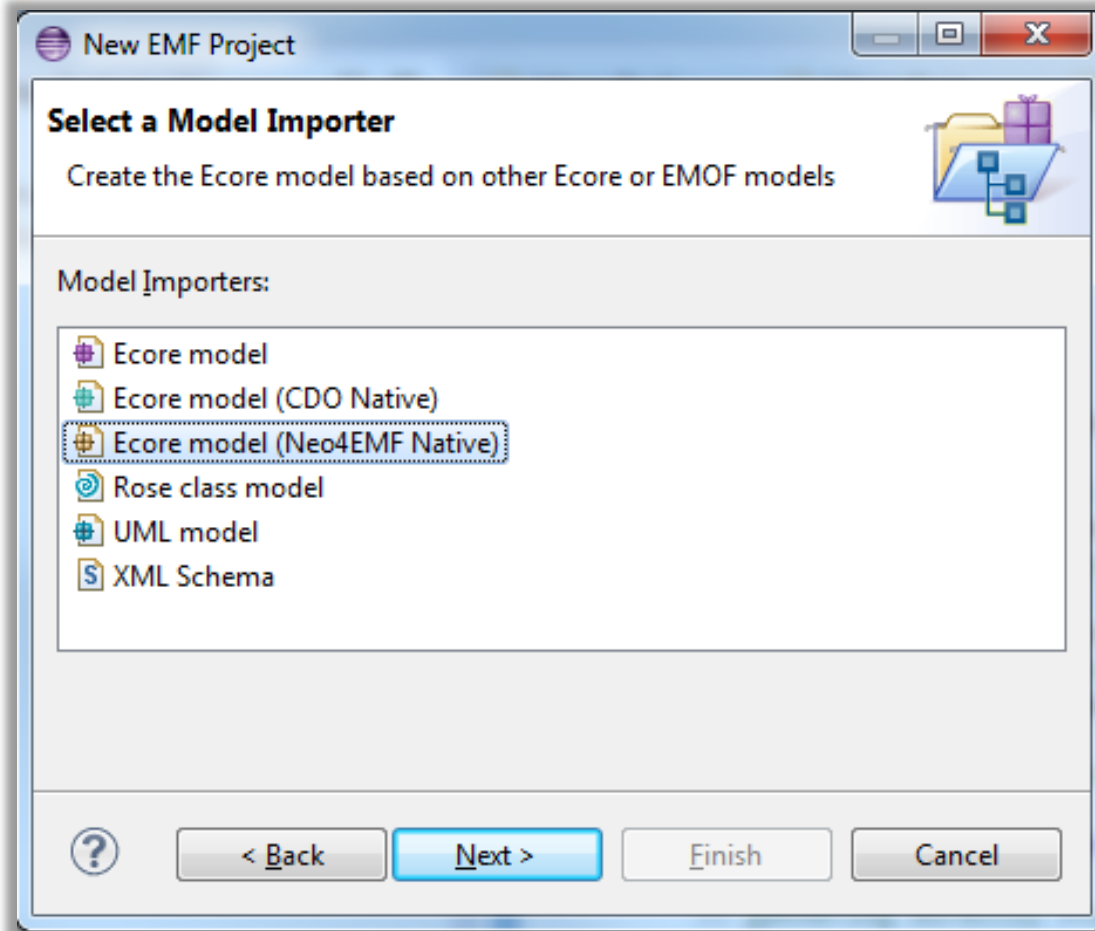


Neo4EMF Resource



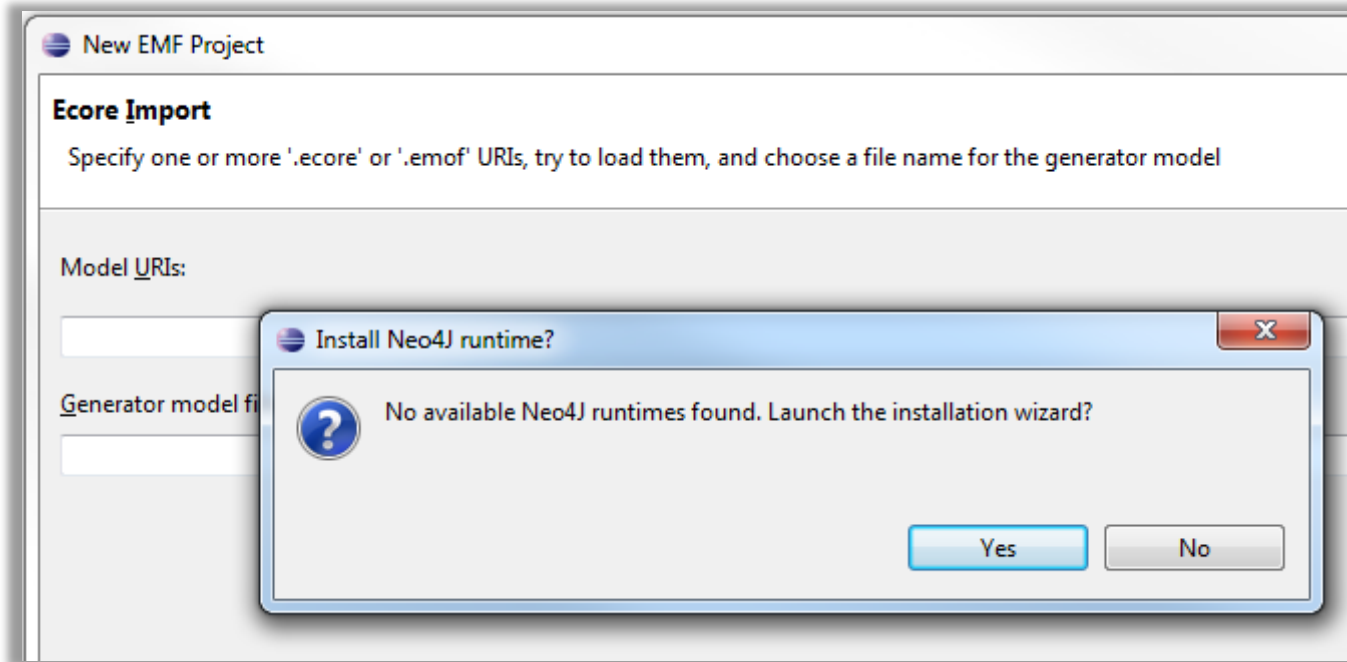
Load on-demand data

Using Neo4EMF...



Dedicated
Ecore model
importer

Using Neo4EMF...



Install Neo4J runtime if not there

Using Neo4EMF...



Property	Value
Minimal Reflective Methods	<input checked="" type="checkbox"/> true
Model Directory	<input type="checkbox"/> /fr.inria.atlanmod.neo4emf.sample/src
Model Plug-in Class	<input type="checkbox"/>
Model Plug-in ID	<input type="checkbox"/> fr.inria.atlanmod.neo4emf.sample
Model Plug-in Variables	<input type="checkbox"/> NEO4J=fr.inria.atlanmod.neo4emf.neo4j-1.9.4, NEO4EMF=fr.inria.atlanmod.neo4emf
Operation Reflection	<input checked="" type="checkbox"/> true
Model Class Defaults	
Public Constructors	<input checked="" type="checkbox"/> false
Root Extends Class	<input type="checkbox"/> fr.inria.atlanmod.neo4emf.impl.Neo4emfObject
Root Extends Interface	<input type="checkbox"/> fr.inria.atlanmod.neo4emf.INeo4emfObject
Root Implements Interface	<input type="checkbox"/>
Redirection Pattern	<input type="checkbox"/>
Template Directory	<input type="checkbox"/> platform:/plugin/fr.inria.atlanmod.neo4emf.codegen/templates
Template Plug-in Variables	<input type="checkbox"/> NEO4EMF_GENERATOR=fr.inria.atlanmod.neo4emf.codegen
Update Classpath	<input checked="" type="checkbox"/> true

Dedicated Genmodel metadata

Using Neo4EMF...



```

<!-- end-user-doc -->
 * @generated
 */
public EList<Book> getBooks() {
    try {
        setLoadingOnDemand();
        if (getData().books == null || getData().books.isEnqueued()){
            EList<Book> newList = new NeoListDecorator<Book>(new
                EObjectContainmentEList<Book>(Book.class, this,
                    LibraryPackage.LIBRARY__BOOKS));
            getData().books = new WeakReference<EList<Book>>(newList, garbagedData);
            if (isLoading()) {
                ((INeo4emfResource) this.eResource()).getOnDemand(this,
                    LibraryPackage.LIBRARY__BOOKS);
            }
            else {
                getData().strongBooks = newList;
            }
        }
        return getData().books.get();
    } finally {
        unsetLoadingOnDemand();
    }
}

```

Neo4EMF generated code

Using Neo4EMF...



```
// URI creation with the neo4emf protocol
URI resourceURI = URI.createURI("neo4emf://" + args[0]);
//create a NEResource responsible of managing and creating resources of a single package
NESession neoSessionLibrary = new NEResource(library.LibraryPackage.eINSTANCE);
// create the neo4emf resource
INeo4emfResource neoResource = neoSessionLibrary.createResource(resourceURI, 100000);

// GetAllthe books of the author "James Patterson"

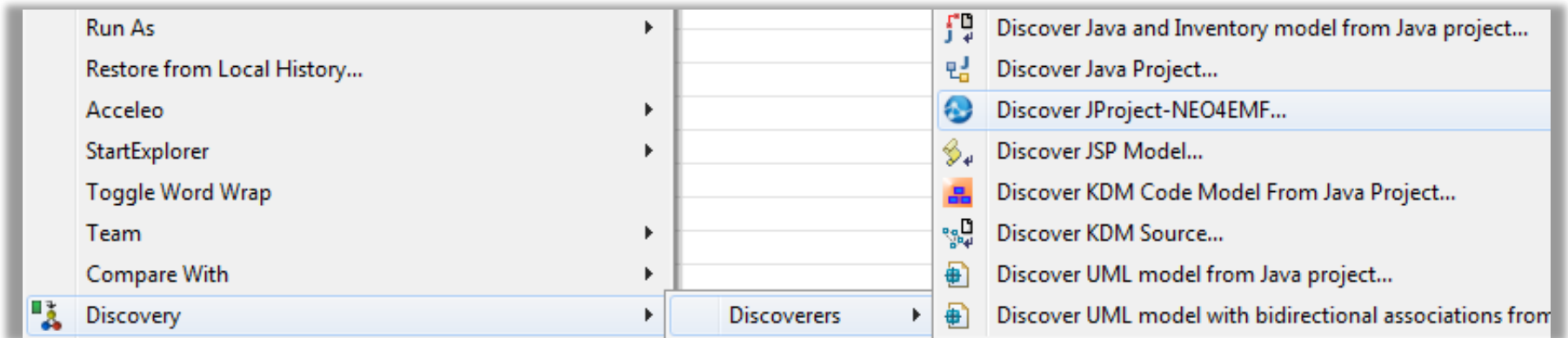
String authorName = "James Patterson";

// gathering directly from the back-end (on-demand) all the instances of type *
// PS : the results of this query are of type INeo4EMF
EList<INeo4emfObject> booksList = neoResource.getAllInstances(library.LibraryPackage.eINSTANCE.getBook());
EList<Book> resultList = new BasicEList<Book>();

for (INeo4emfObject book : booksList) {
    if (((Book)book).getAuthor().equalsIgnoreCase(authorName)) {
        resultList.add((Book) book);
    }
}
```

Handling the Neo4EMF resource

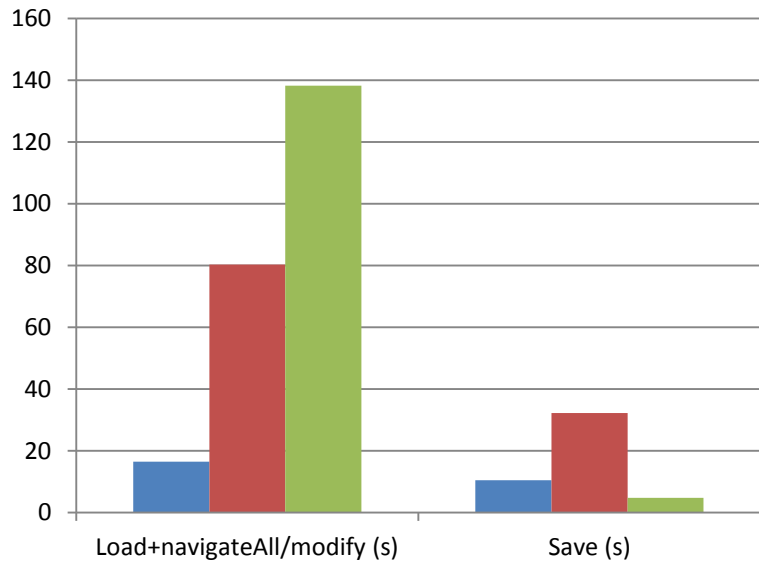
Using Neo4EMF...



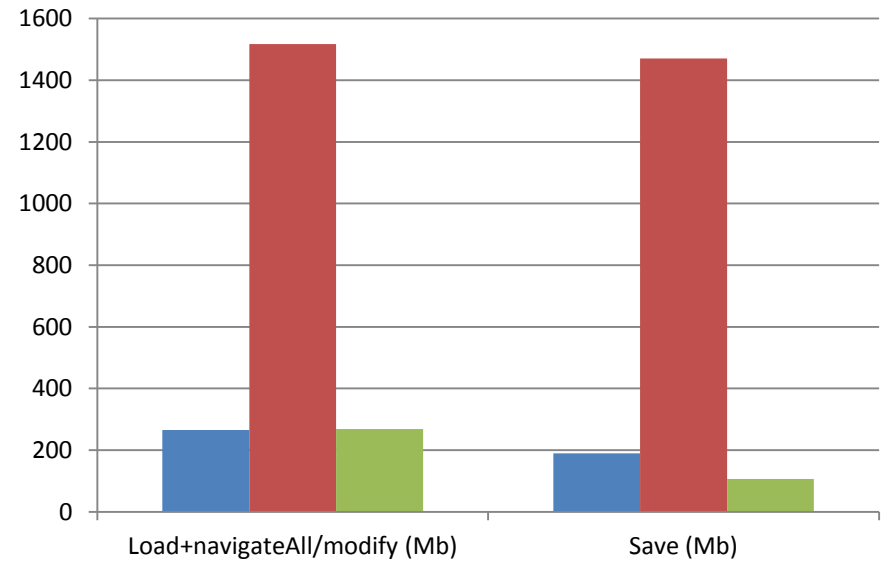
Using Neo4EMF within another tool:
Example of Java model discovery with
Eclipse MoDisco



Time Vs. Memory (2)

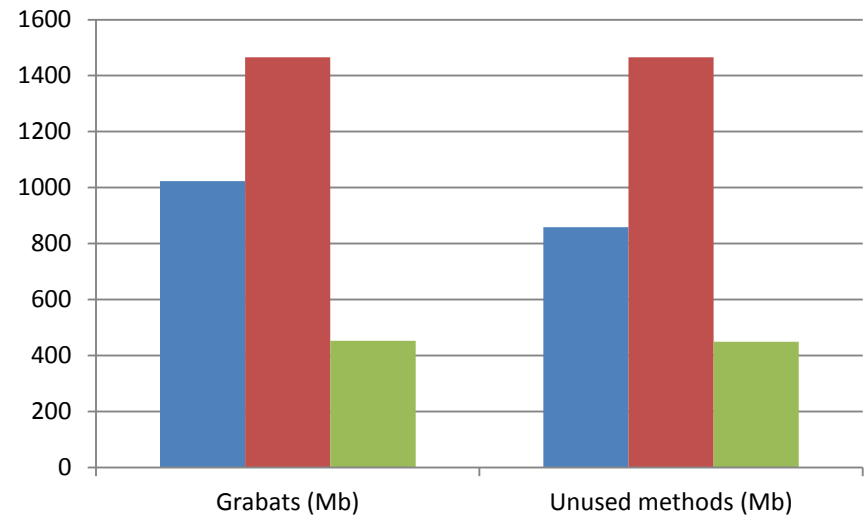
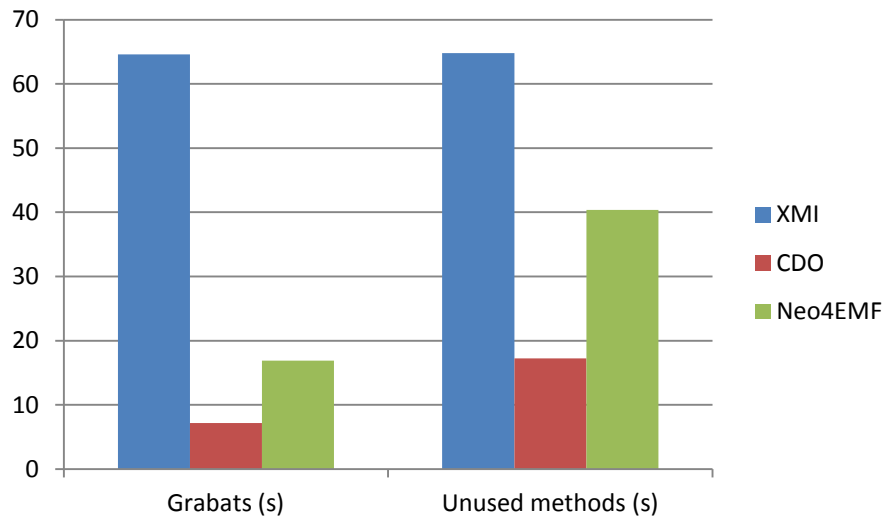


■ XMI
■ CDO
■ Neo4EMF



Load + full navigation & Save
(performance for a 400Mb model)

Time Vs. Memory (3)



Advanced navigation/querying
(performance for a 1Gb model)

Don't forget to...



- Check us out :
www.neo4emf.com
- Contact us :
neo4emf@gmail.com
- Fork us at :
<https://github.com/neo4emf/Neo4EMF>