REPUBLIQUE DU CAMEROUN
Paix – Travail - Patrie
REPUBLIC OF CAMEROON
Peace – Work - Fatherland

UNIVERSITE DE DSCHANG
UNIVERSITY OF DSCHANG
Scholae Thesaurus Dschangensis Ibi Cordum

ECOLE DOCTORALE
POST GRADUATE SCHOOL

FACULTE DES SCIENCES
FACULTY OF SCIENCE

Département de Mathématiques et Informatique
Department of Mathematics and Computer Science

Réunion à distance de l'équipe FUCHSIA, Avril 2020

Utilisation de GEMOC Studio pour le développement d'une version simplifiée d'un langage de modélisation

Présenté par

ZEKENG NDADJI Milliam Maxime
Doctorant, Université de Dschang
ndadji.maxime@univ-dschang.org

Inria

FUCHSIA

PLAN DE LA PRÉSENTATION

- **L'ingénierie des langages avec l'approche GEMOC**
- **Travail Pratique: création des syntaxes abstraites et concrètes d'un langage d'arbre**
- **Perspectives et conclusion**

L'INGÉNIERIE DES LANGAGES AVEC L'APPROCHE GEMOC

GEMOC ???

GEMOC est une initiative libre et internationale, qui vise à coordonner et à diffuser les résultats de la recherche concernant le domaine de l'utilisation coordonnée de divers langages de modélisation.

("***coordination*" *(globalization) des langages de modélisation***).

GEMOC ???

GEMOC développe des techniques, des frameworks et des environnements pour faciliter la création, l'intégration et le traitement automatisé de langages de modélisation hétérogènes. Tous les résultats sont intégrés de manière transparente dans le logiciel GEMOC Studio.

- Site officiel de GEMOC → http://gemoc.org
- Télécharger GEMOC Studio → http://gemoc.org/download.html
- Documentation de GEMOC Studio → https://archive.eclipse.org/gemoc/docs/milestones/3.1.0-20190627/
- Trouvez une liste de publications relatives à GEMOC → http://gemoc.org/publications.html

eclipse

http://gemoc.org/studio

Language Workbench

Concevez et composez vos xDSML

Modeling Workbench

Editez et déboguez vos modèles hétérogènes

L'INGENIERIE DES LANGAGES AVEC L'APPROCHE GEMOC

GEMOC STUDIO: LANGUAGE WORKBENCH

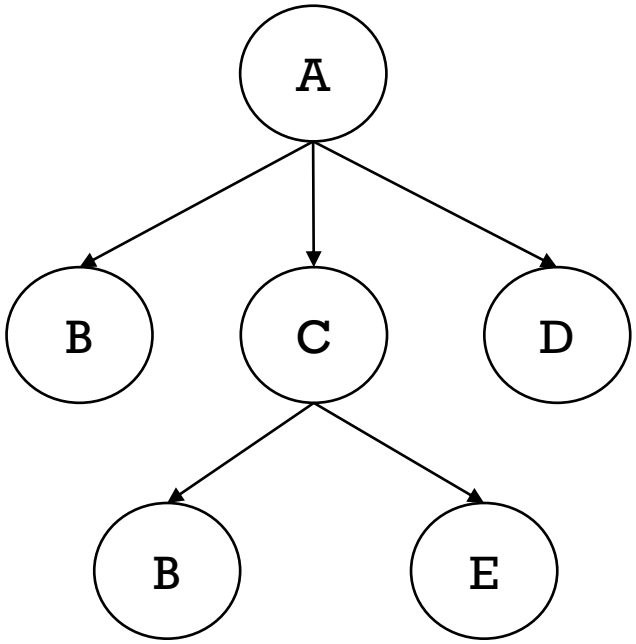
- Création et extension de méta-modèles avec des interpréteurs, des compilateurs et des vérificateurs de modèles (**Xtend/Kermeta, XMOF, Java**)
- Définition de la sémantique opérationnelle des langages exécutables ainsi qu'une couche d'animation au dessus de la syntaxe graphique concrète (**Sirius**)
- Déclaration, extension et assemblage des composants des langages exécutables (**Melange**)
- Génération automatique de méta-modèles de traces d'exécution pour prendre en charge les fonctionnalités avancées du modeling workbench
- Spécification d'un modèle de concurrence et de communication (MoCC) et son mapping à une syntaxe abstraite spécifique et aux fonctions d'exécution associées d'un langage de modélisation (**MoCCML**)
- Spécification des modèles de coordination des langages pour assurer la coordination automatique de l'exécution de modèles hétérogènes (**BCOol**)

GEMOC STUDIO: MODELING WORKBENCH

- Framework d'exécution générique composé de divers moteurs d'exécution et services d'exécution partagés
- Animation graphique et textuelle, définition des points d'arrêt et exécution du modèle étape par étape (simulation des modèles)
- Débogage avancé, navigation en avant et en arrière grâce aux traces d'exécution
- Analyse des contraintes de concurrence basée sur les traces d'exécution des modèles (analyse de la concurrence)
- Exécution simultanée et coordonnée de modèles hétérogènes
- Ajout dynamique de plugins au framework d'exécution

TRAVAIL PRATIQUE: CRÉATION DES SYNTAXES ABSTRAITES ET CONCRÈTES D'UN LANGAGE D'ARBRE

TRAVAIL PRATIQUE
UNE VERSION SIMPLIFIEE D'UN LANGAGE D'ARBRES (TREE_L)



Un type Haskell pour ce langage:

```
data Artifact a = BuildArtifact { nodeLabel :: a, sons :: [Artifact a] }
```

TRAVAIL PRATIQUE
LES DIFFERENTES ETAPES (GEMOC)

1. Création du projet xDSML (executable Domain Specific Modeling Language)

2. Modélisation de la syntaxe abstraite du langage grâce à Ecore/EMF

3. Génération d'un éditeur (tree-based) pour le langage (Ecore/EMF)

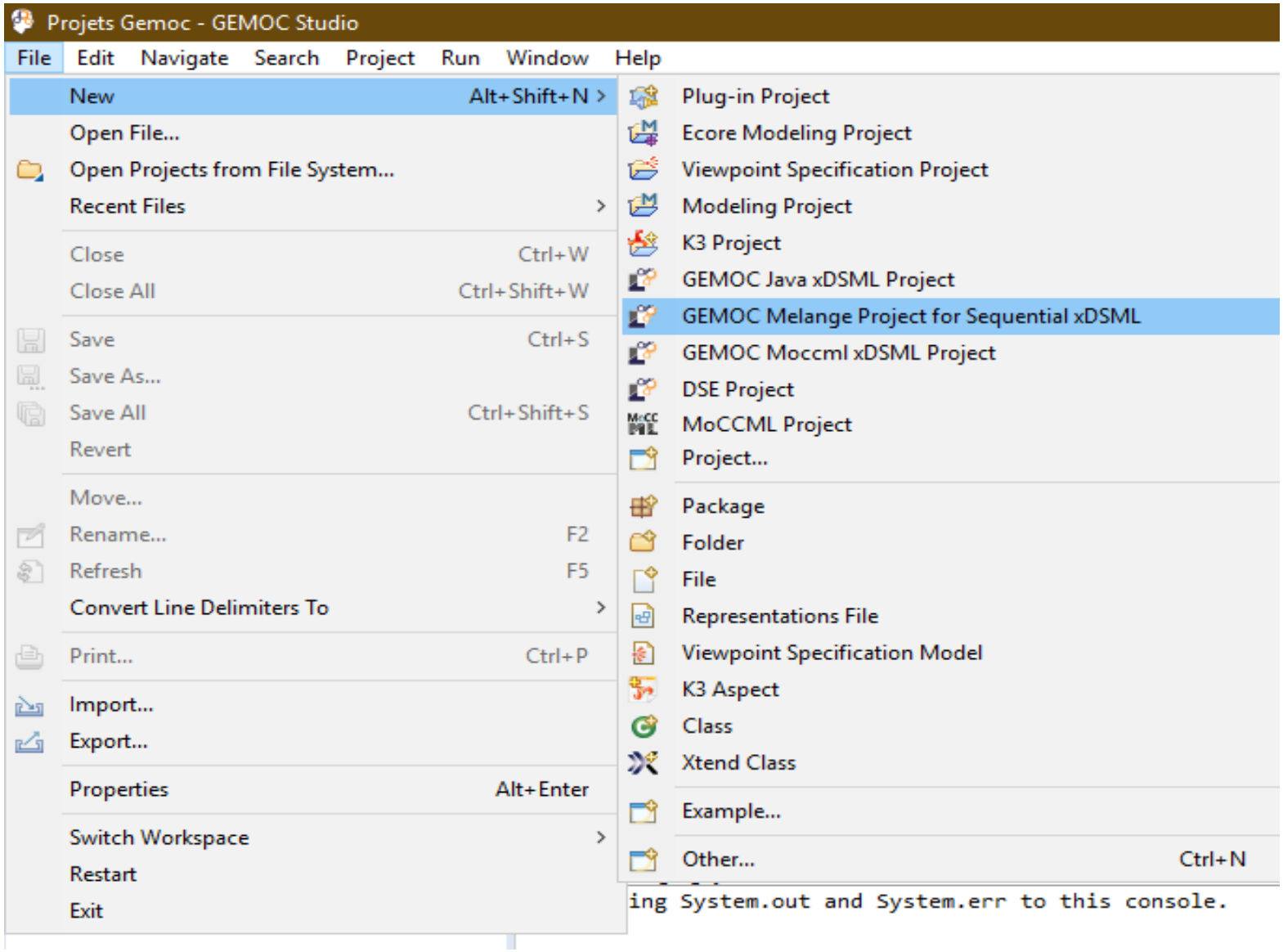
4. Génération des interfaces Java du langage (Ecore/EMF)

5. Génération d'une syntaxe textuelle (xtext) concrète du langage

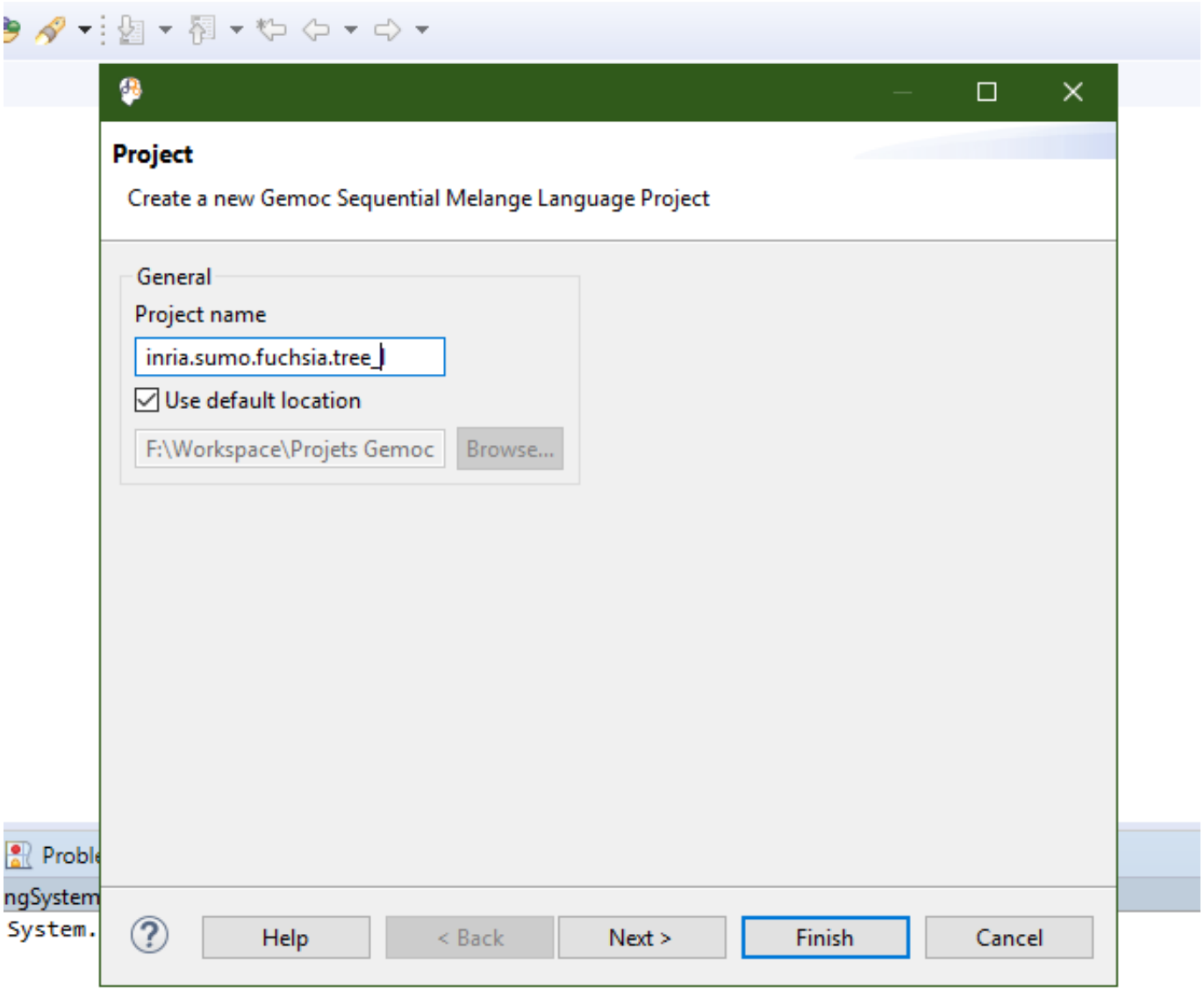
6. Génération d'une syntaxe graphique (sirius) concrète du langage

7. Utilisation du langage modélisé

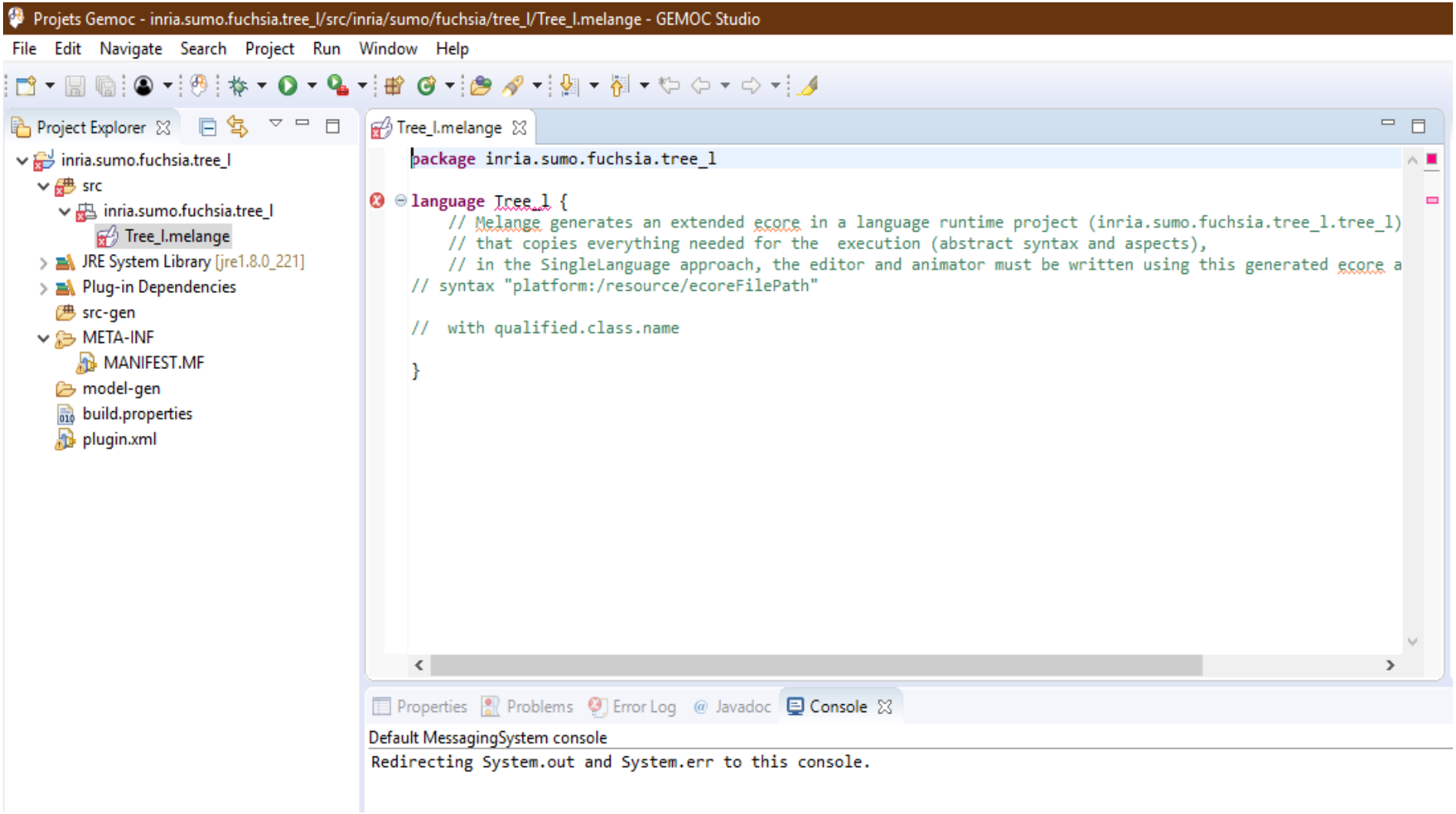
TRAVAIL PRATIQUE
1- *CRÉATION DU PROJET XDSML*



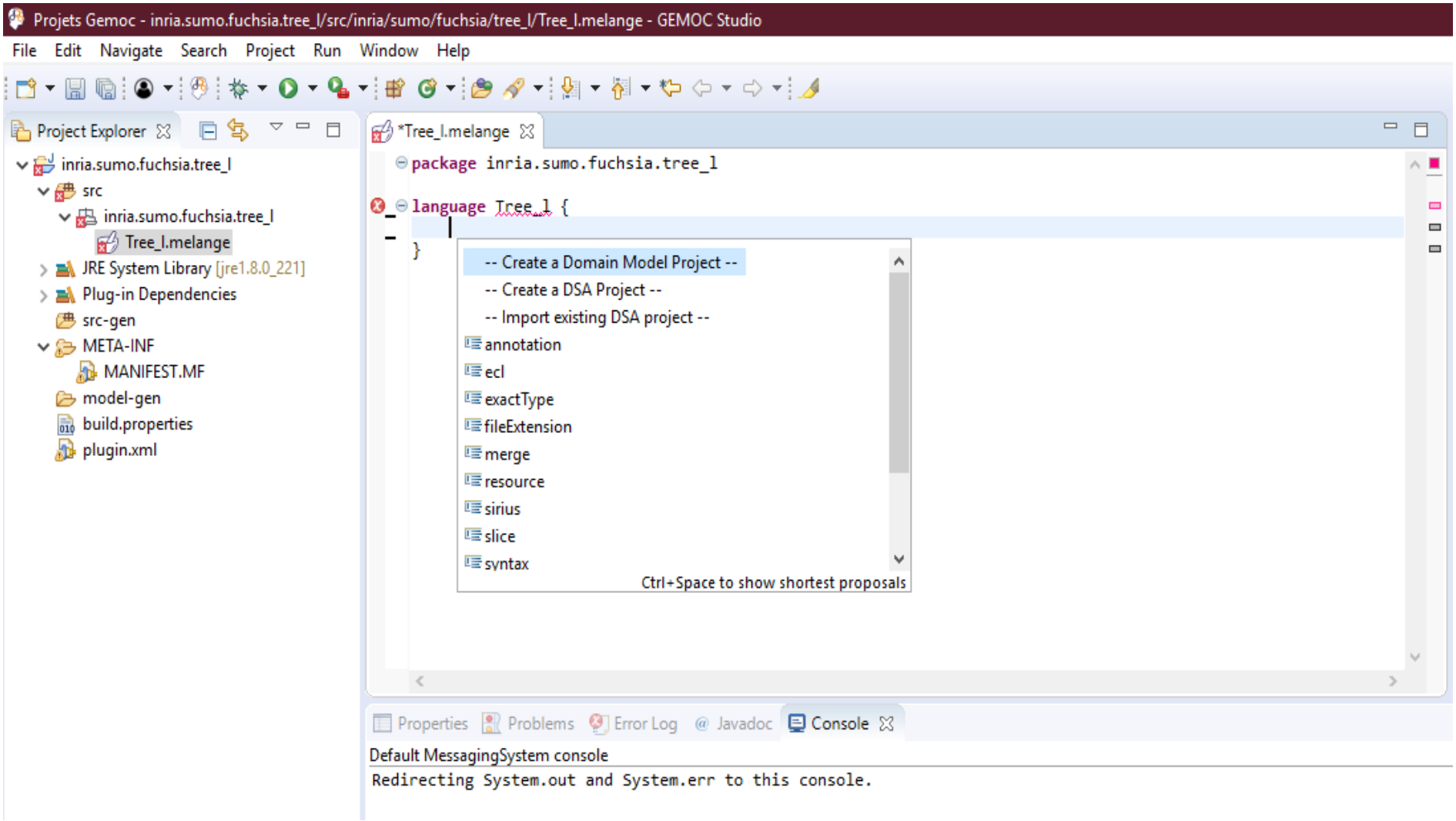
TRAVAIL PRATIQUE
1- *CRÉATION DU PROJET XDSML*



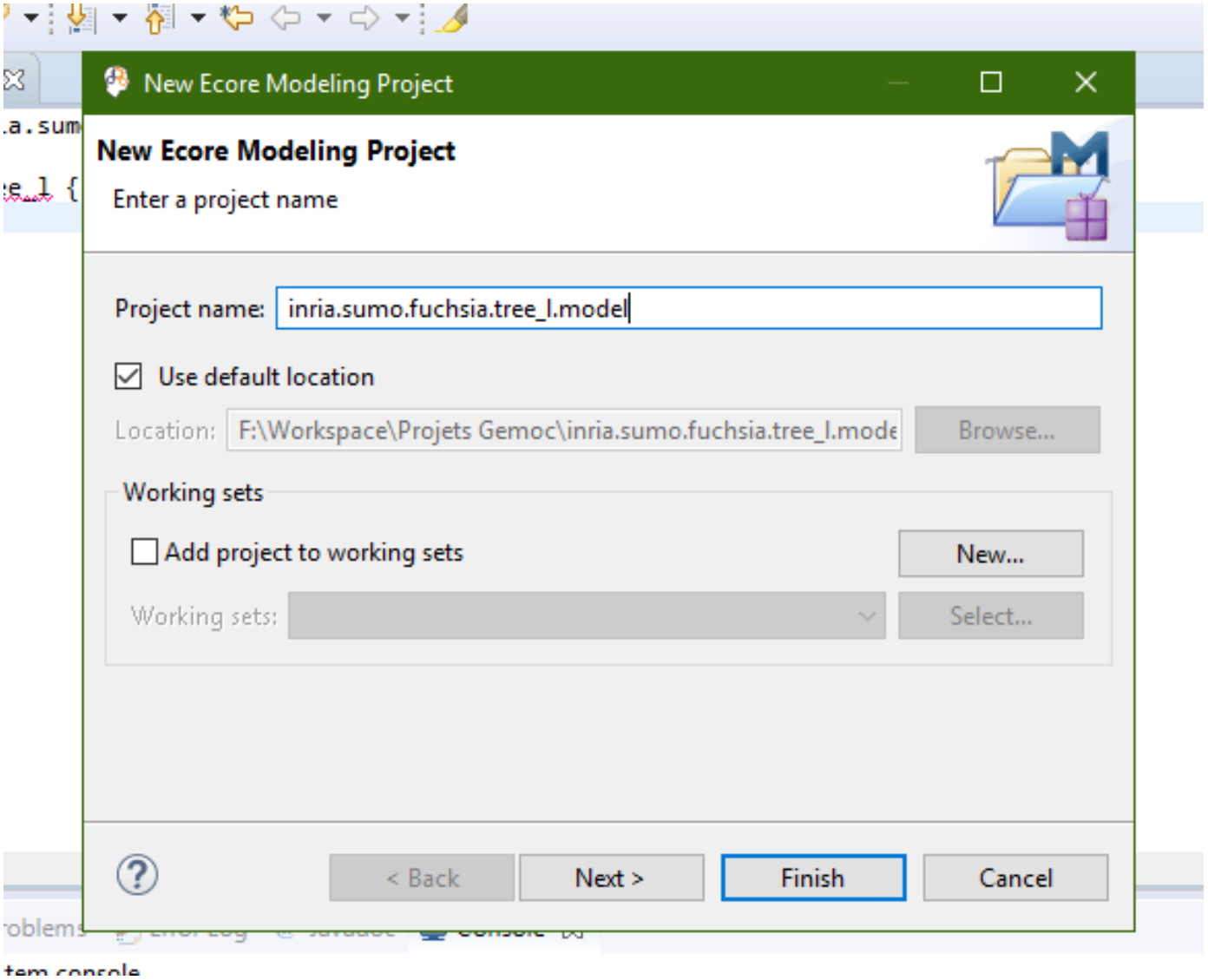
TRAVAIL PRATIQUE
1- *CRÉATION DU PROJET XDSML*



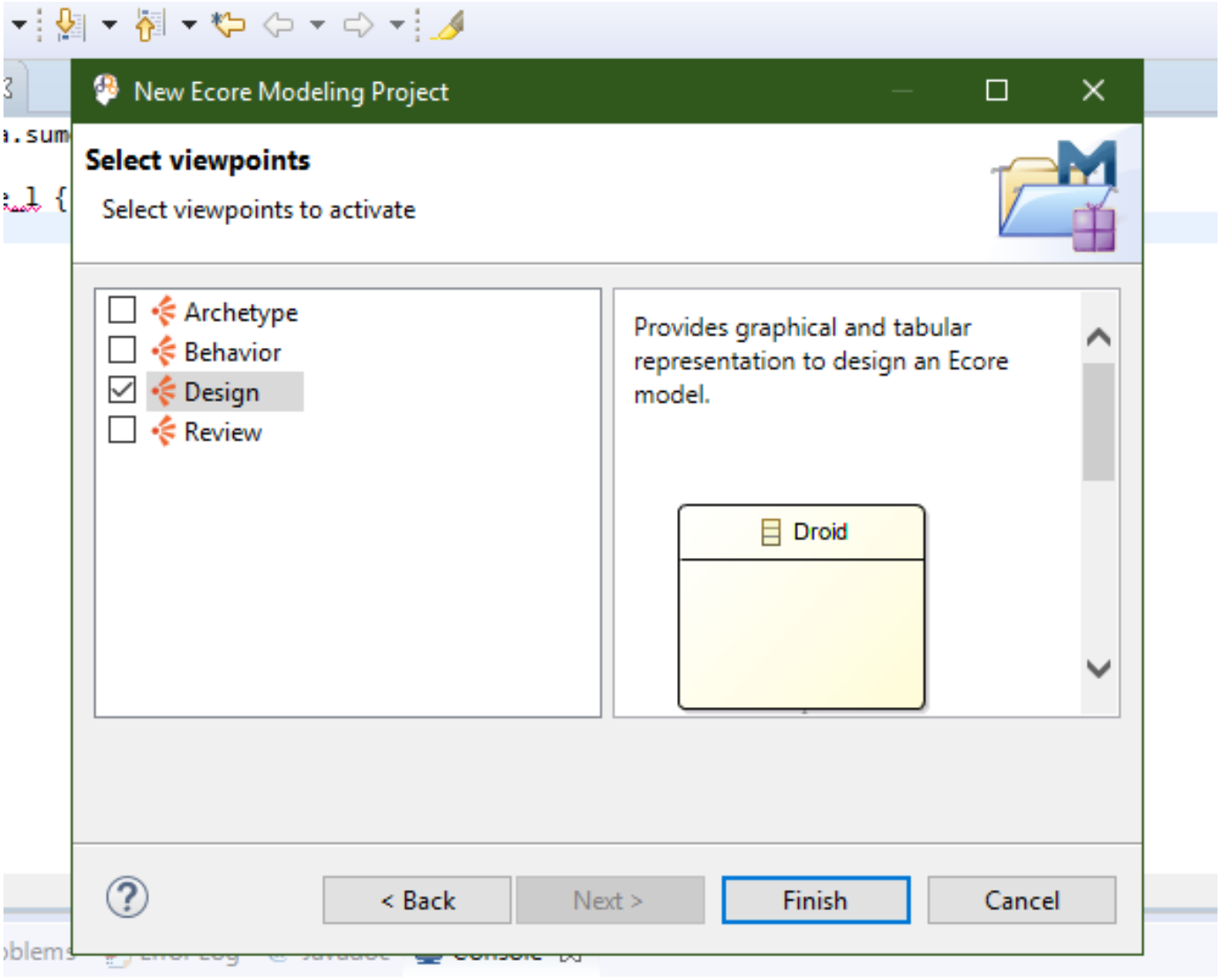
TRAVAIL PRATIQUE
2- *MODÉLISATION DE LA SYNTAXE ABSTRAITE DU LANGAGE GRÂCE À ECORE/EMF*



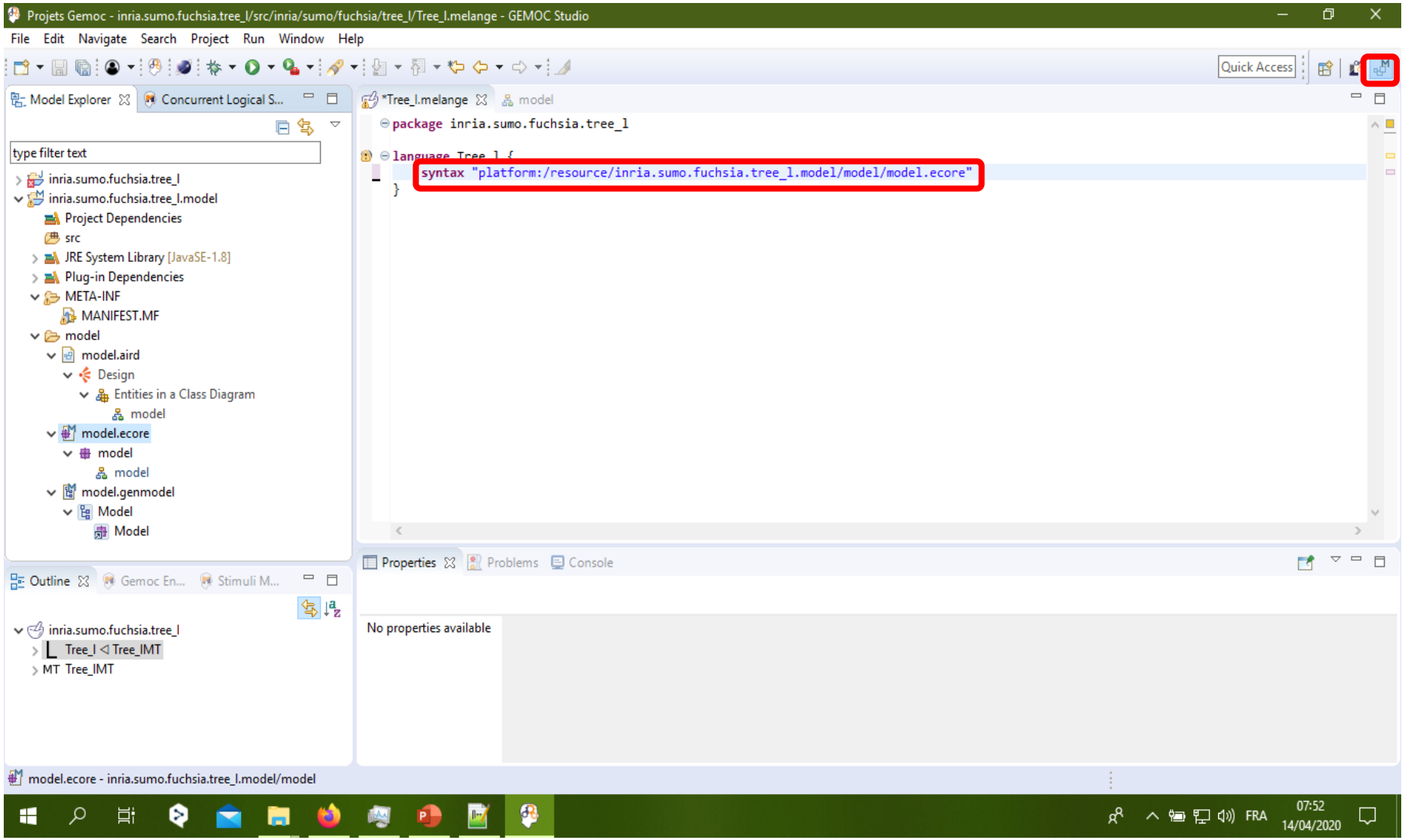
TRAVAIL PRATIQUE
2- *MODÉLISATION DE LA SYNTAXE ABSTRAITE DU LANGAGE GRÂCE À ECORE/EMF*



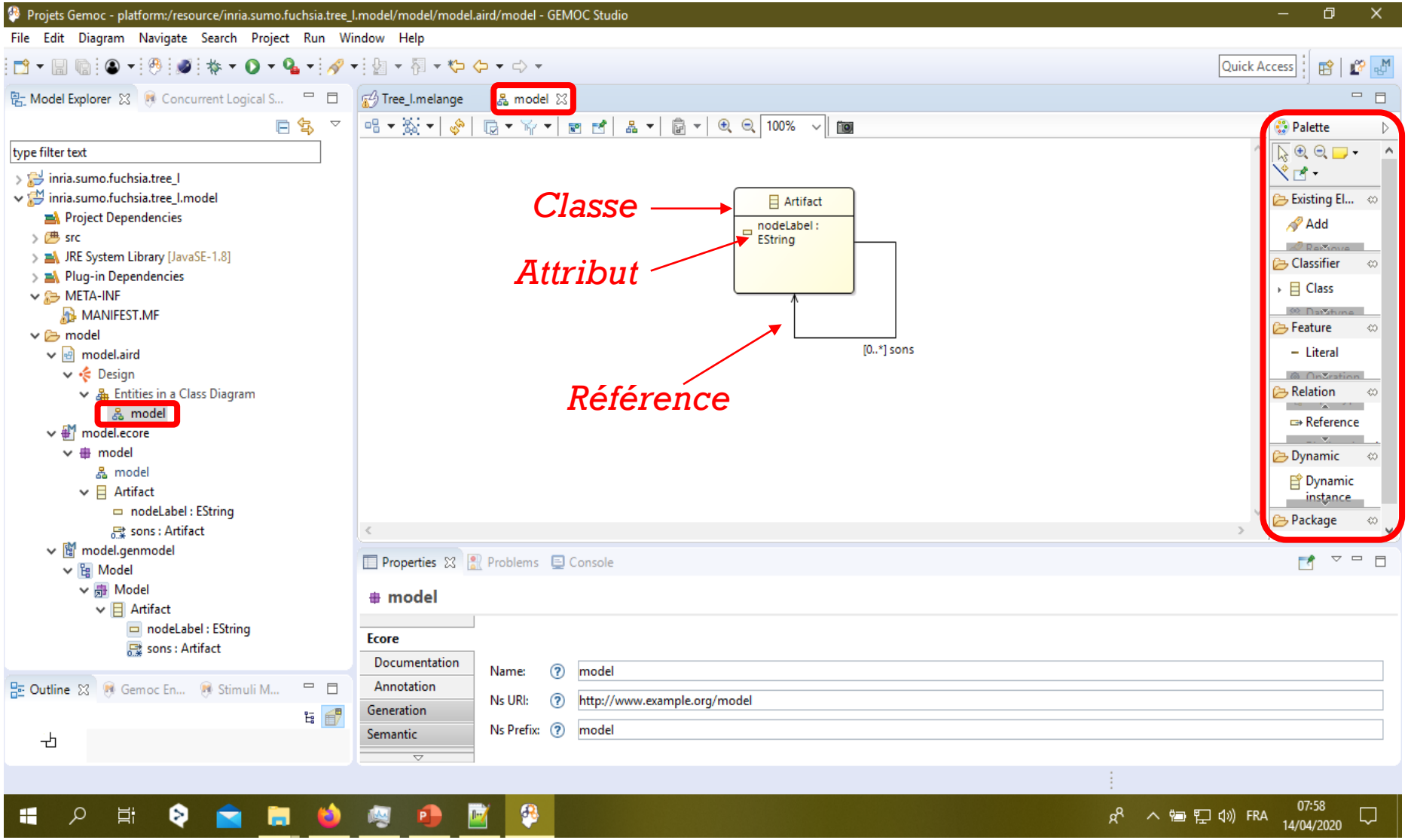
TRAVAIL PRATIQUE
2- *MODÉLISATION DE LA SYNTAXE ABSTRAITE DU LANGAGE GRÂCE À ECORE/EMF*



TRAVAIL PRATIQUE
2- *MODÉLISATION DE LA SYNTAXE ABSTRAITE DU LANGAGE GRÂCE À ECORE/EMF*

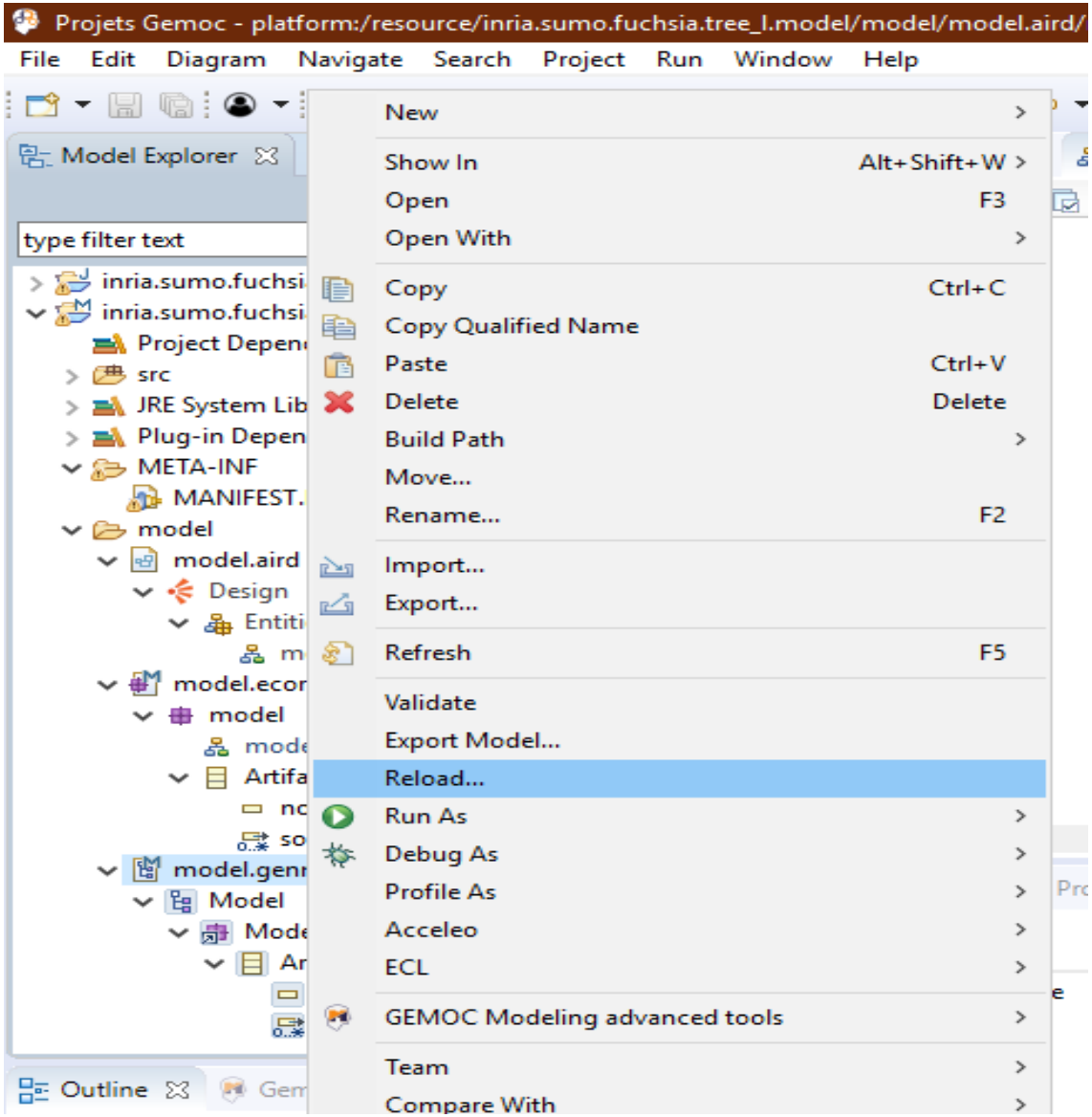


TRAVAIL PRATIQUE
2- MODÉLISATION DE LA SYNTAXE ABSTRAITE DU LANGAGE GRÂCE À ECORE/EMF

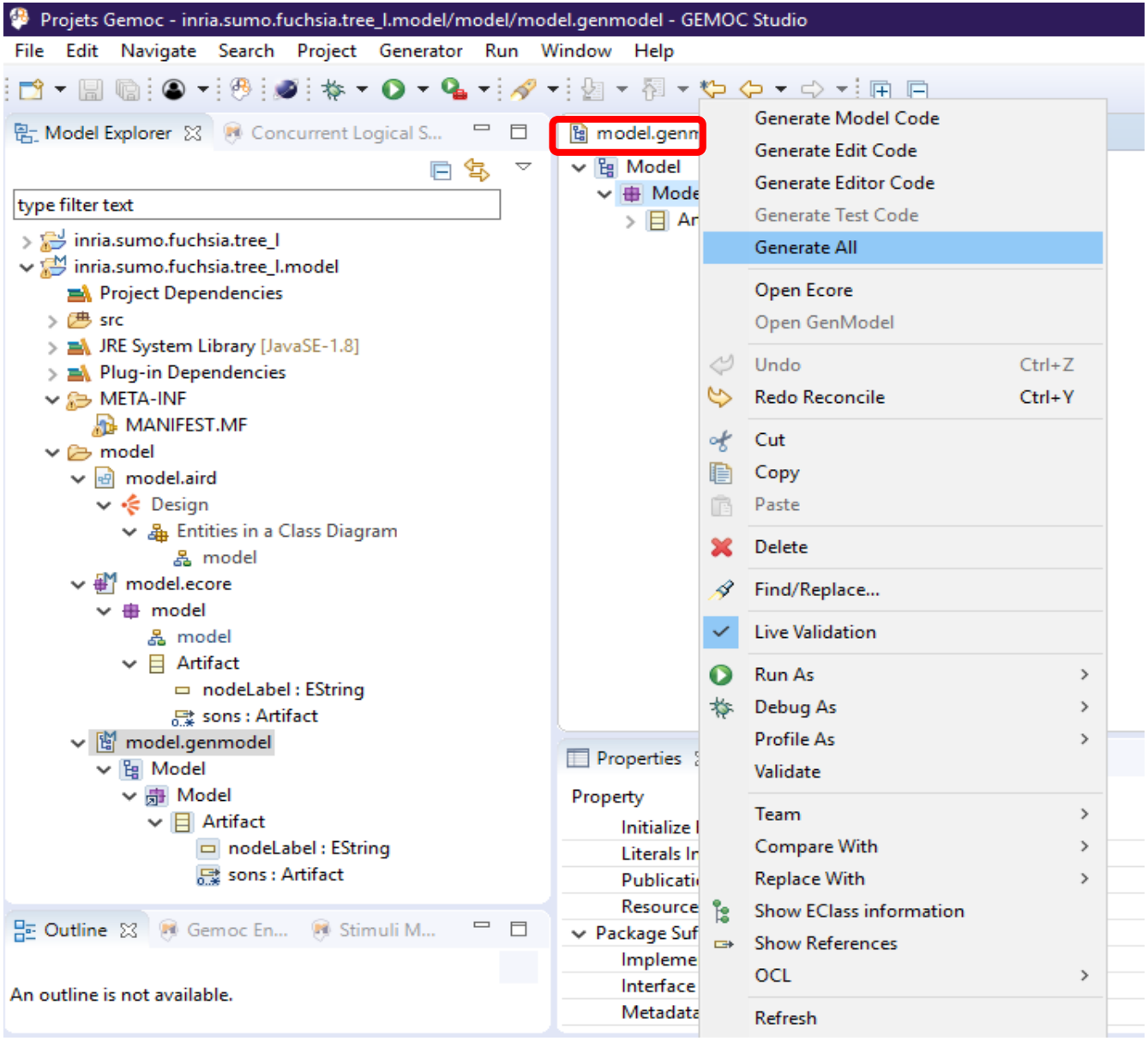


En savoir plus sur EMF: https://www.vogella.com/tutorials/EclipseEMF/article.html
https://www.eclipse.org/sirius/doc/user/general/Modeling%20Project.html

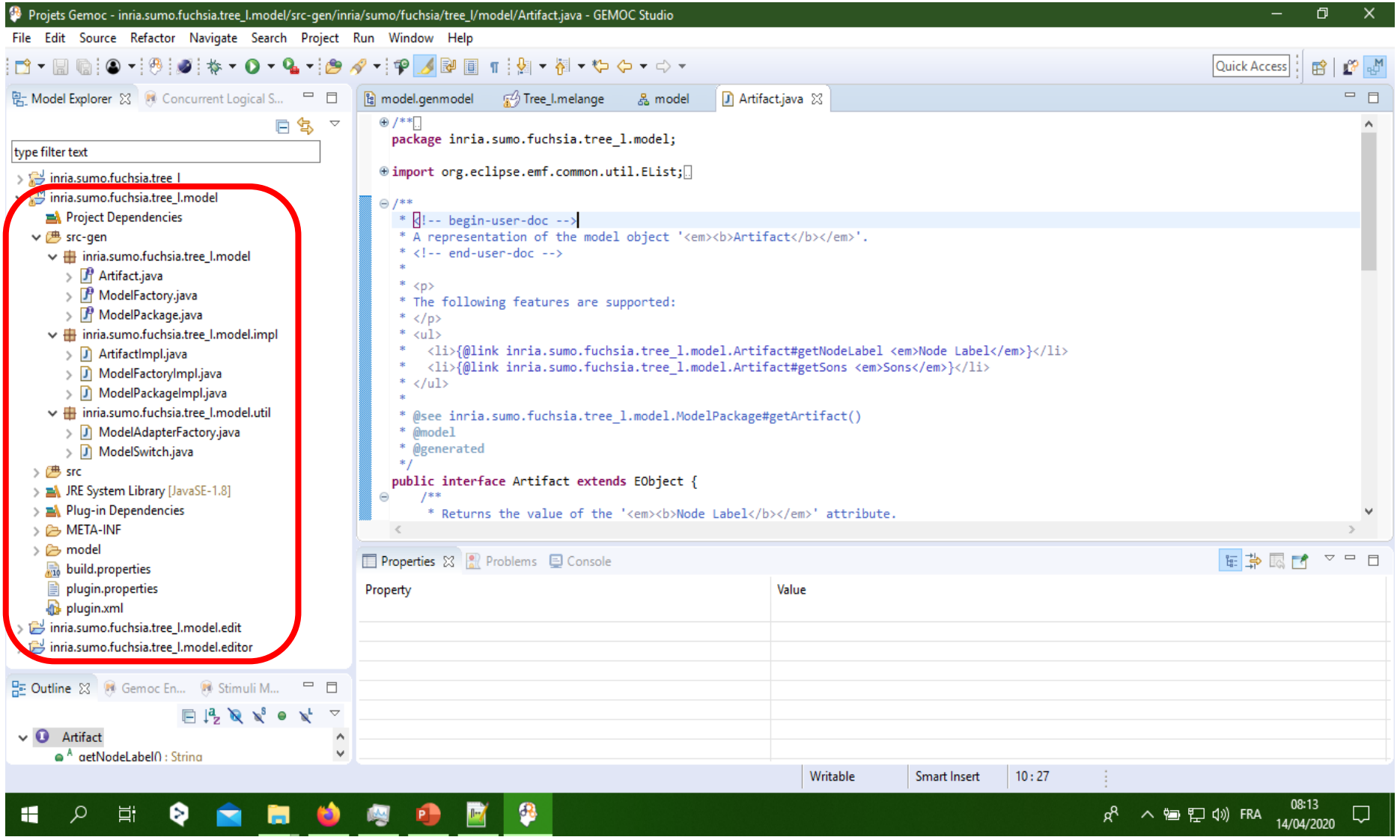
TRAVAIL PRATIQUE
2- *MODÉLISATION DE LA SYNTAXE ABSTRAITE DU LANGAGE GRÂCE À ECORE/EMF*



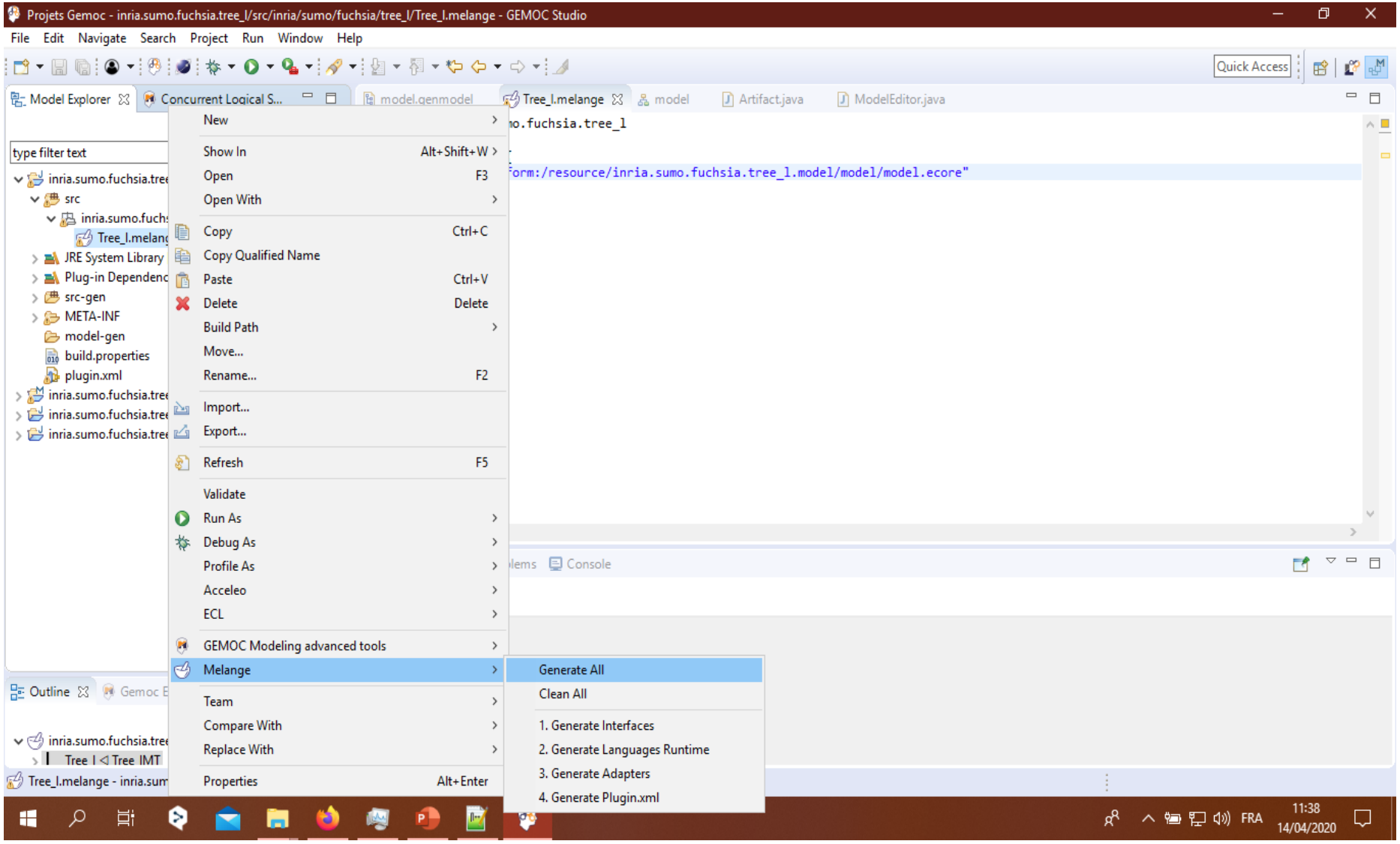
TRAVAIL PRATIQUE
3- *GÉNÉRATION D'UN ÉDITEUR (TREE-BASED) POUR LE LANGAGE (ECORE/EMF)*



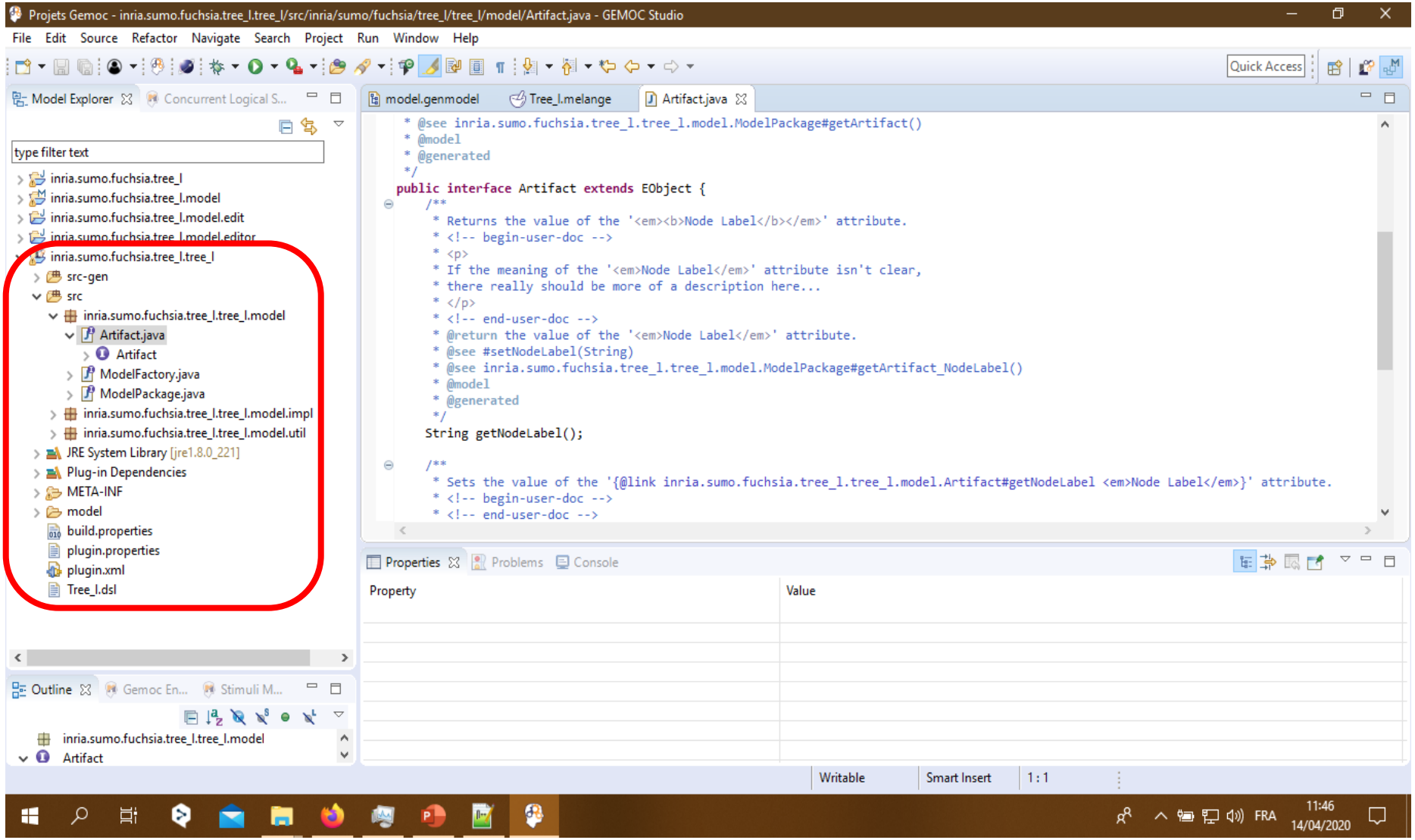
TRAVAIL PRATIQUE
3- *GÉNÉRATION D'UN ÉDITEUR (TREE-BASED) POUR LE LANGAGE (ECORE/EMF)*



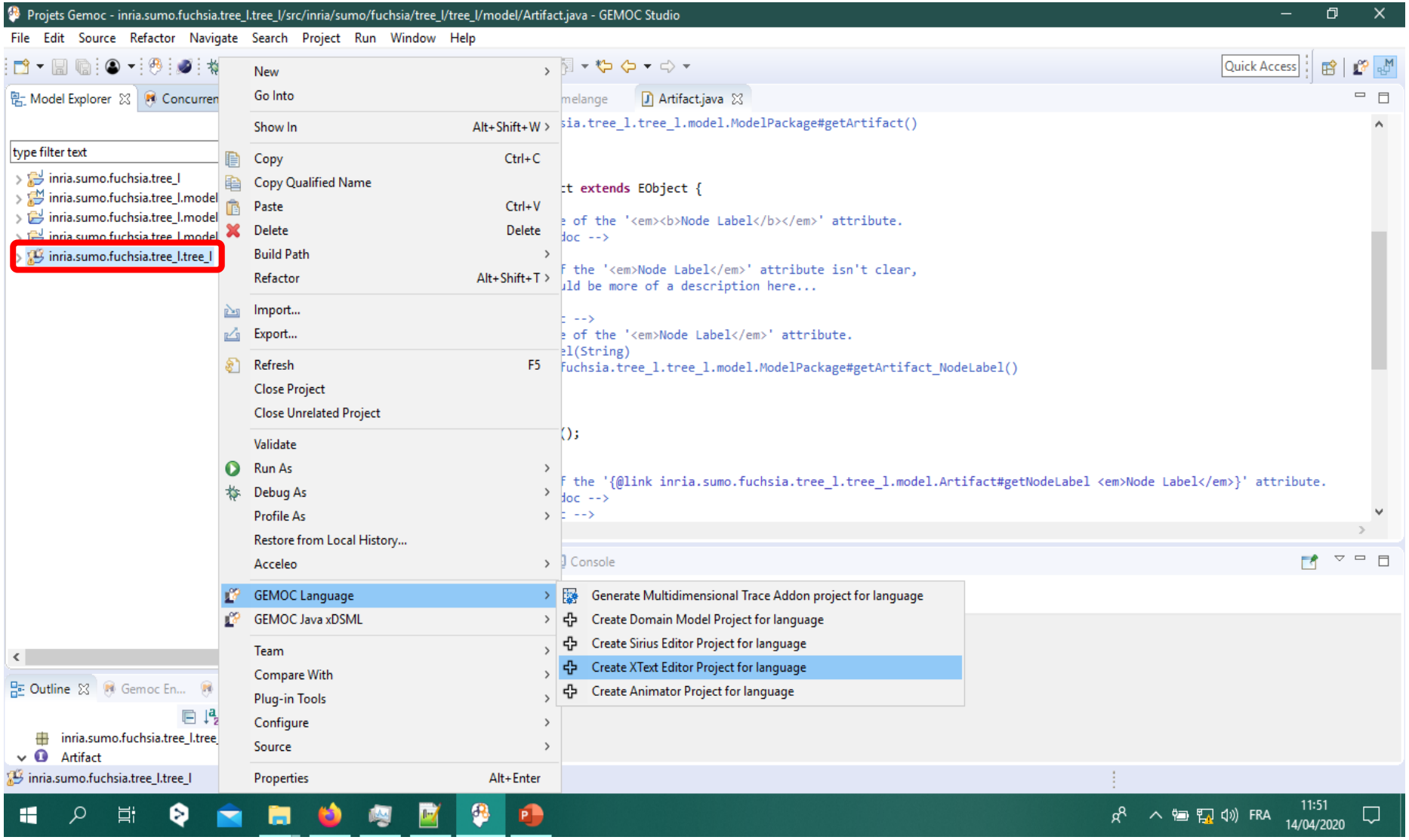
TRAVAIL PRATIQUE
4- GÉNÉRATION DES INTERFACES JAVA DU LANGAGE (ECORE/EMF)



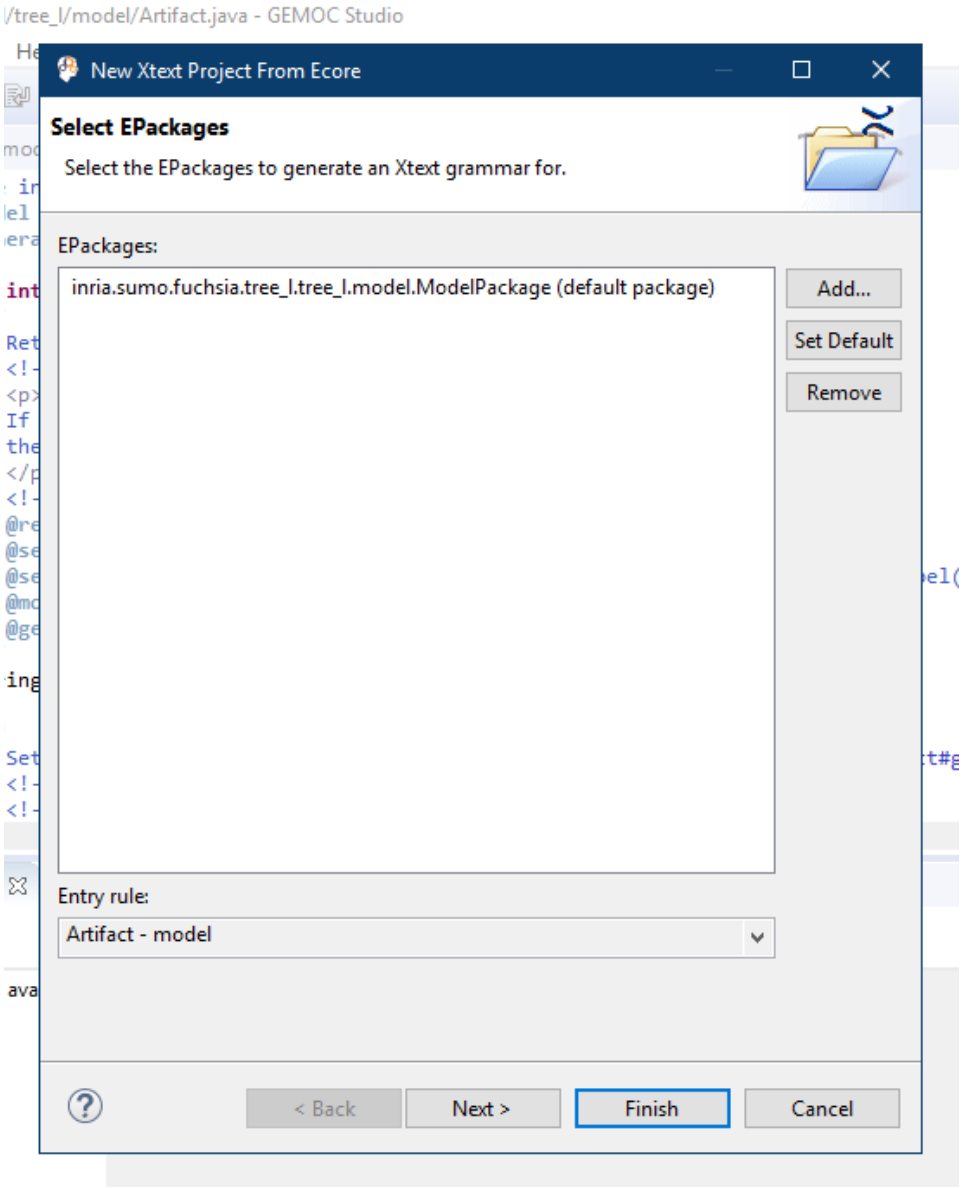
TRAVAIL PRATIQUE
4- GÉNÉRATION DES INTERFACES JAVA DU LANGAGE (ECORE/EMF)



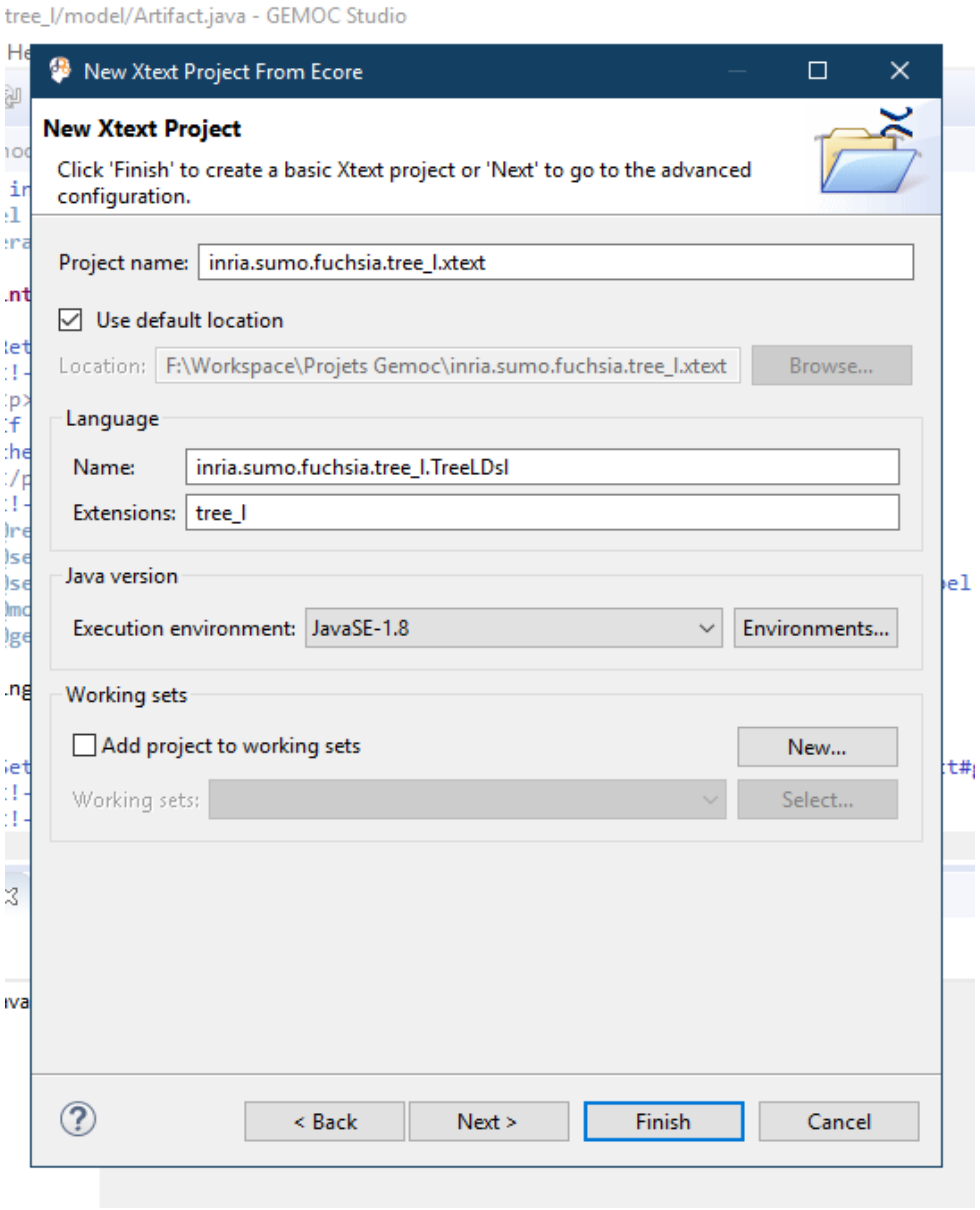
TRAVAIL PRATIQUE
5- *GÉNÉRATION D'UNE SYNTAXE TEXTUELLE (XTEXT) CONCRÈTE DU LANGAGE*



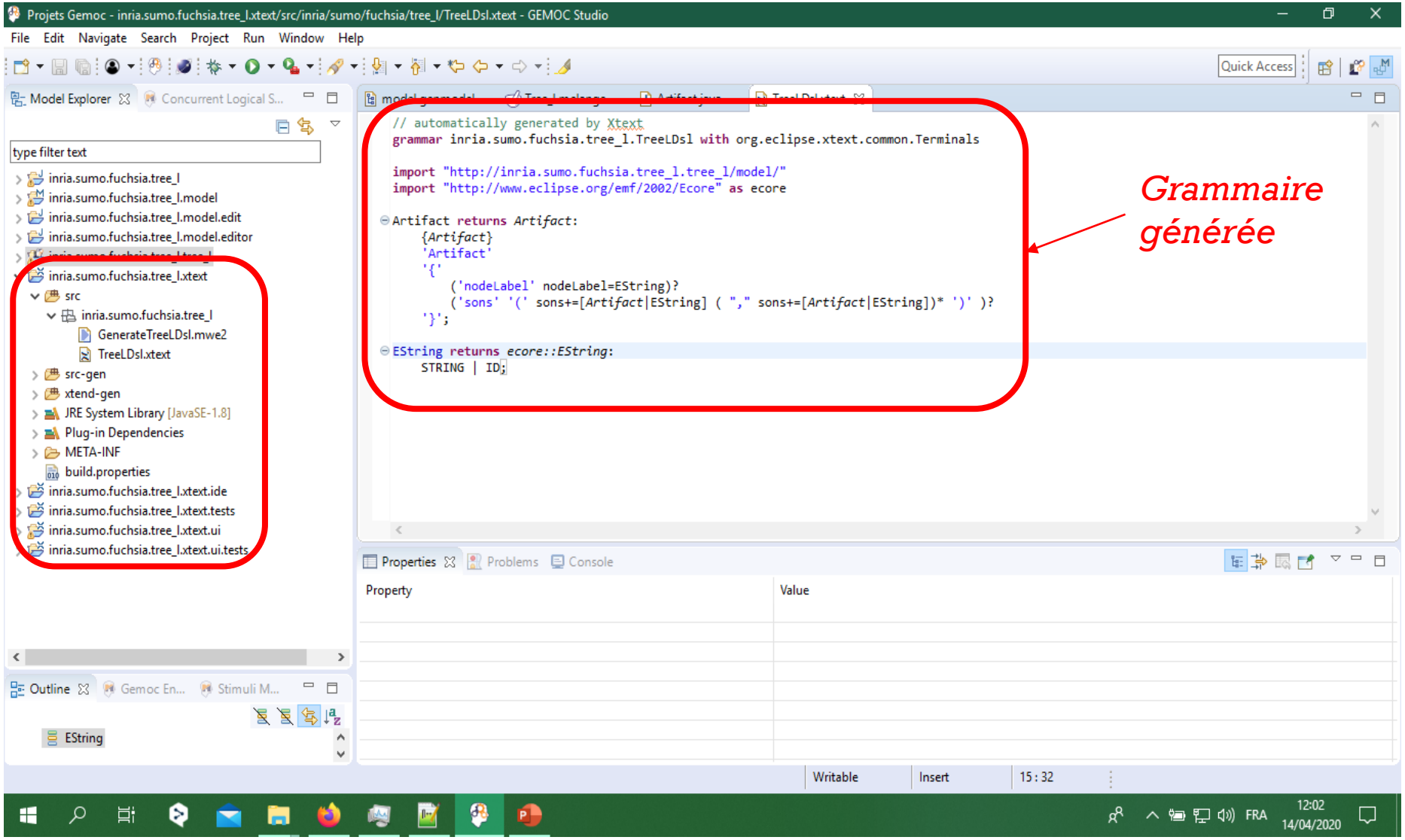
TRAVAIL PRATIQUE
5- *GÉNÉRATION D'UNE SYNTAXE TEXTUELLE (XTEXT) CONCRÈTE DU LANGAGE*



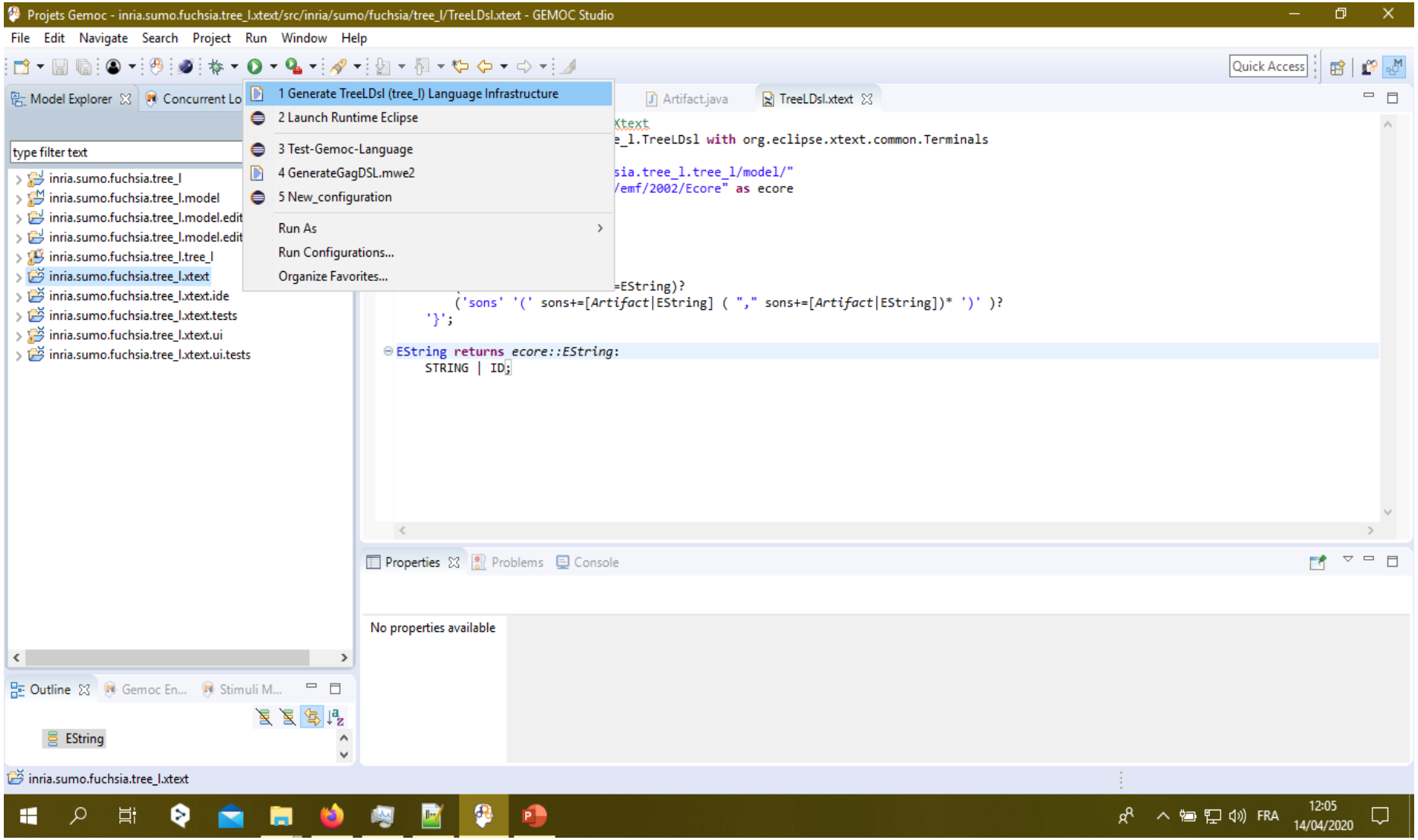
TRAVAIL PRATIQUE
5- *GÉNÉRATION D'UNE SYNTAXE TEXTUELLE (XTEXT) CONCRÈTE DU LANGAGE*



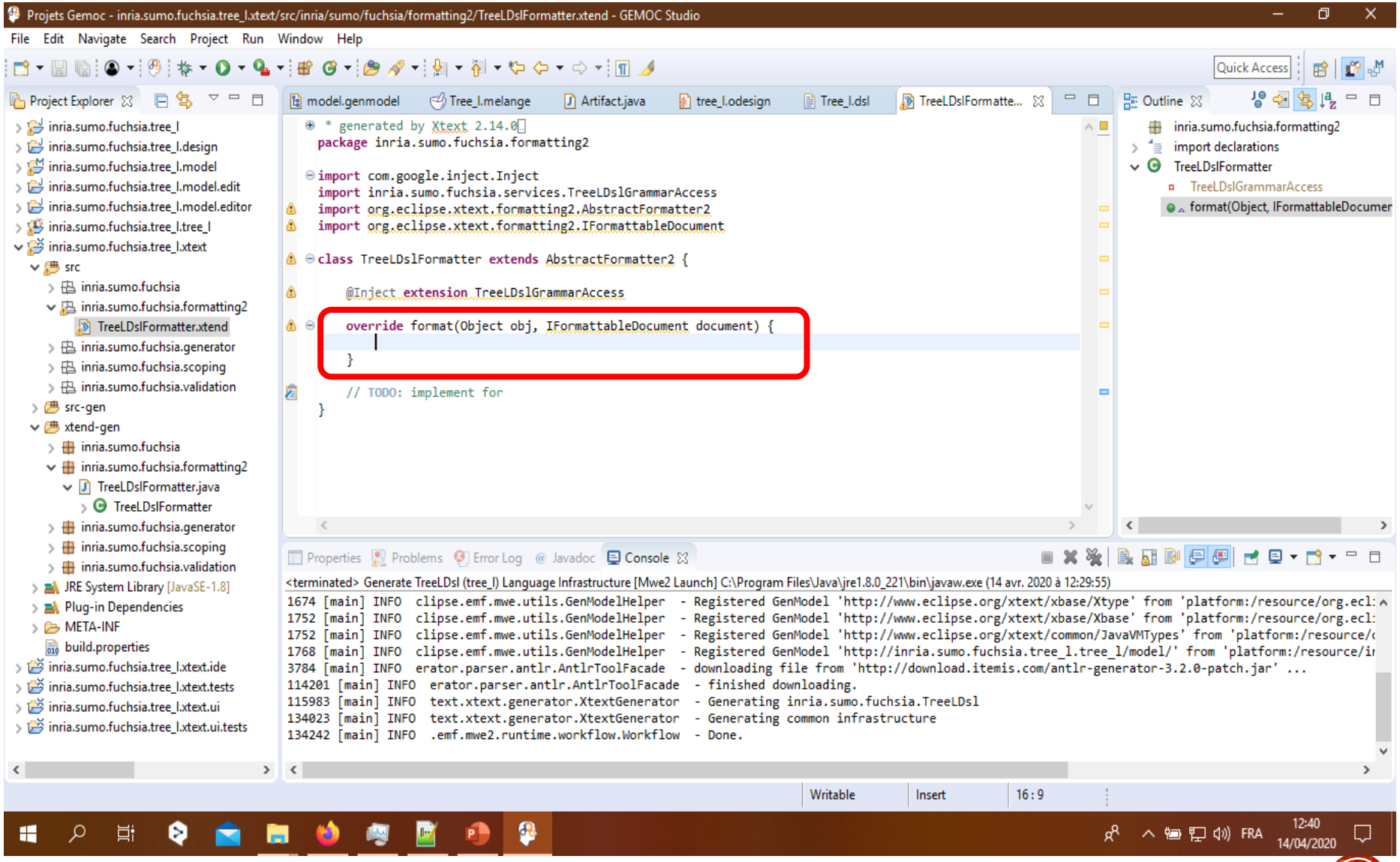
TRAVAIL PRATIQUE
5- *GÉNÉRATION D'UNE SYNTAXE TEXTUELLE (XTEXT) CONCRÈTE DU LANGAGE*



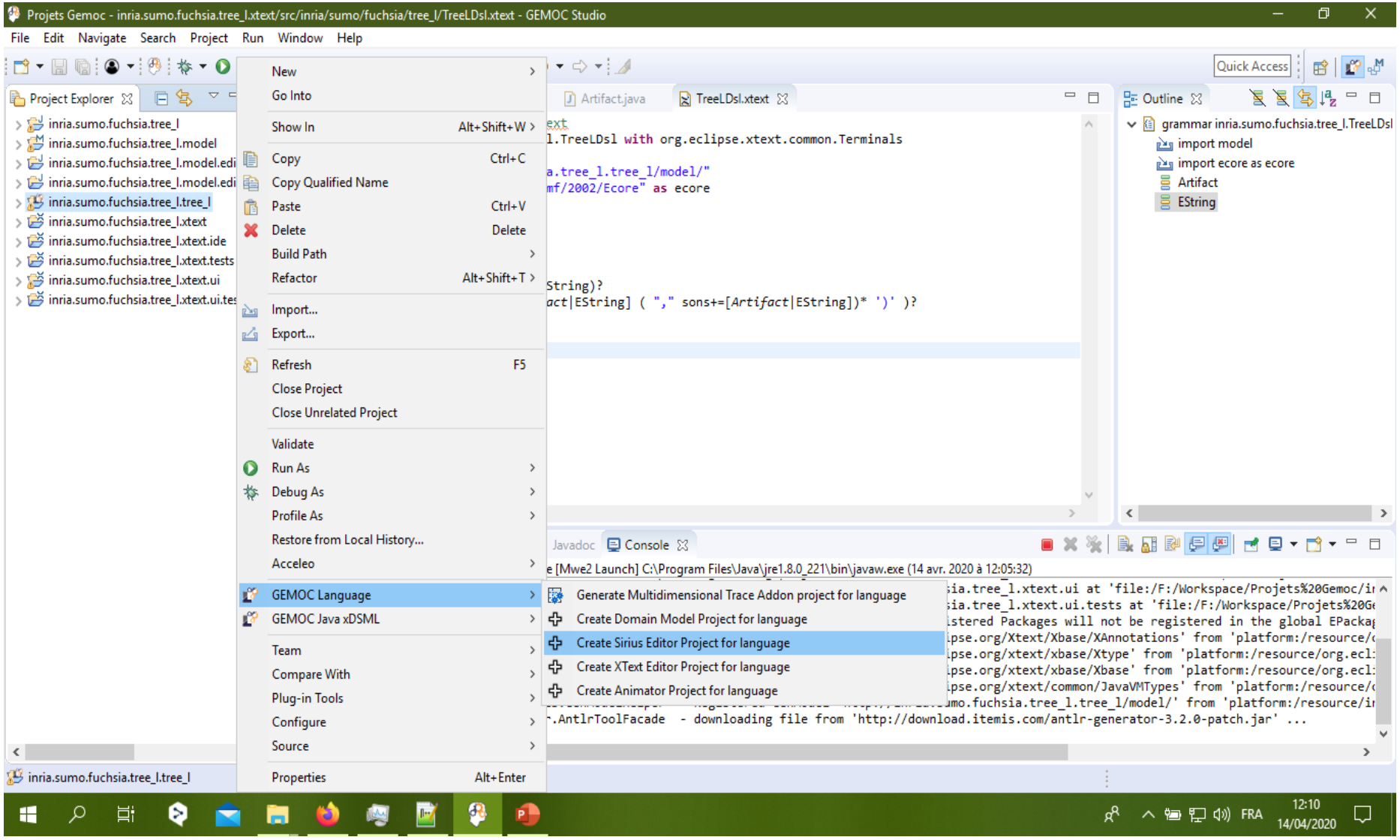
TRAVAIL PRATIQUE
5- GÉNÉRATION D'UNE SYNTAXE TEXTUELLE (XTEXT) CONCRÈTE DU LANGAGE



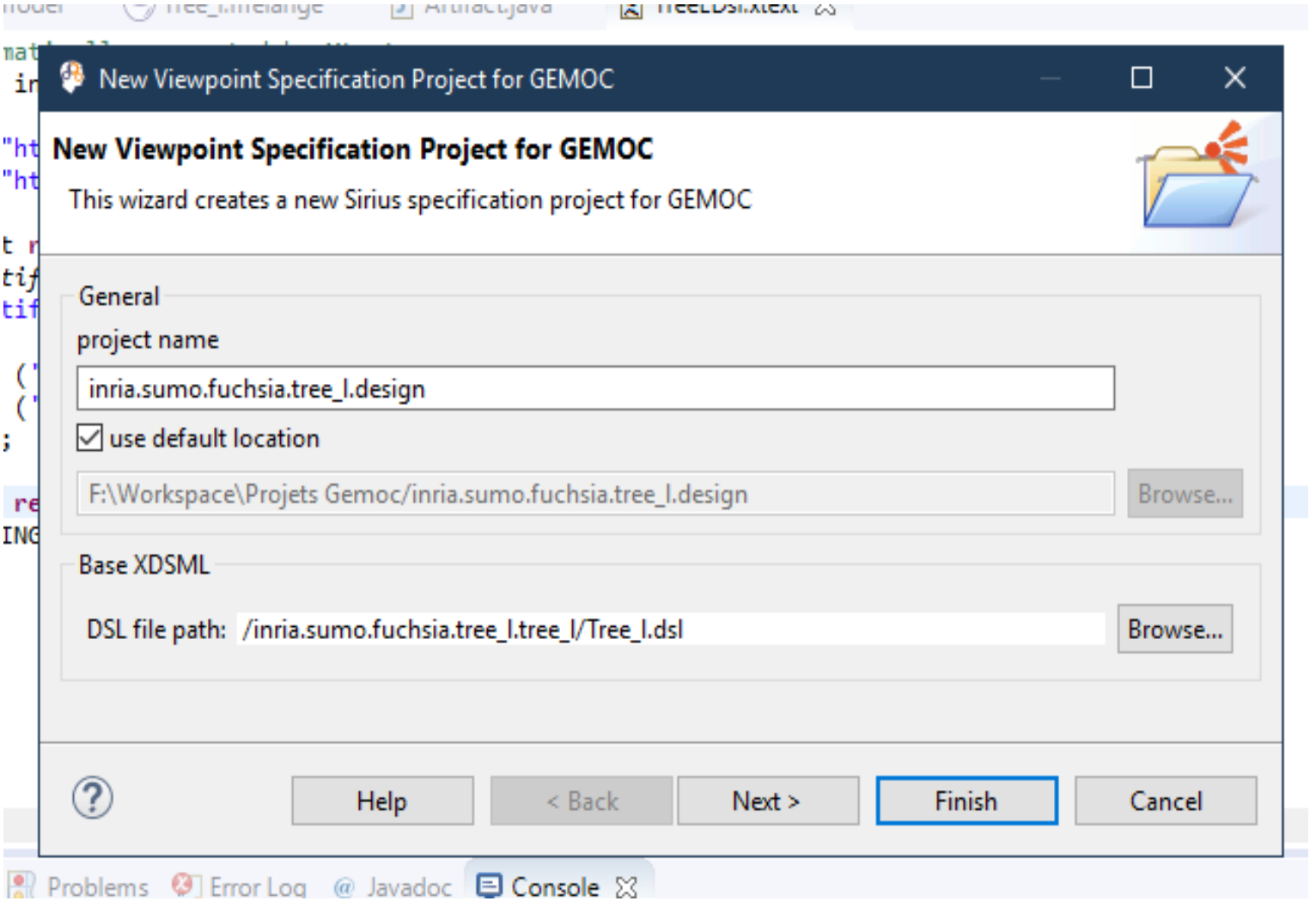
TRAVAIL PRATIQUE
5- *GÉNÉRATION D'UNE SYNTAXE TEXTUELLE (XTEXT) CONCRÈTE DU LANGAGE*



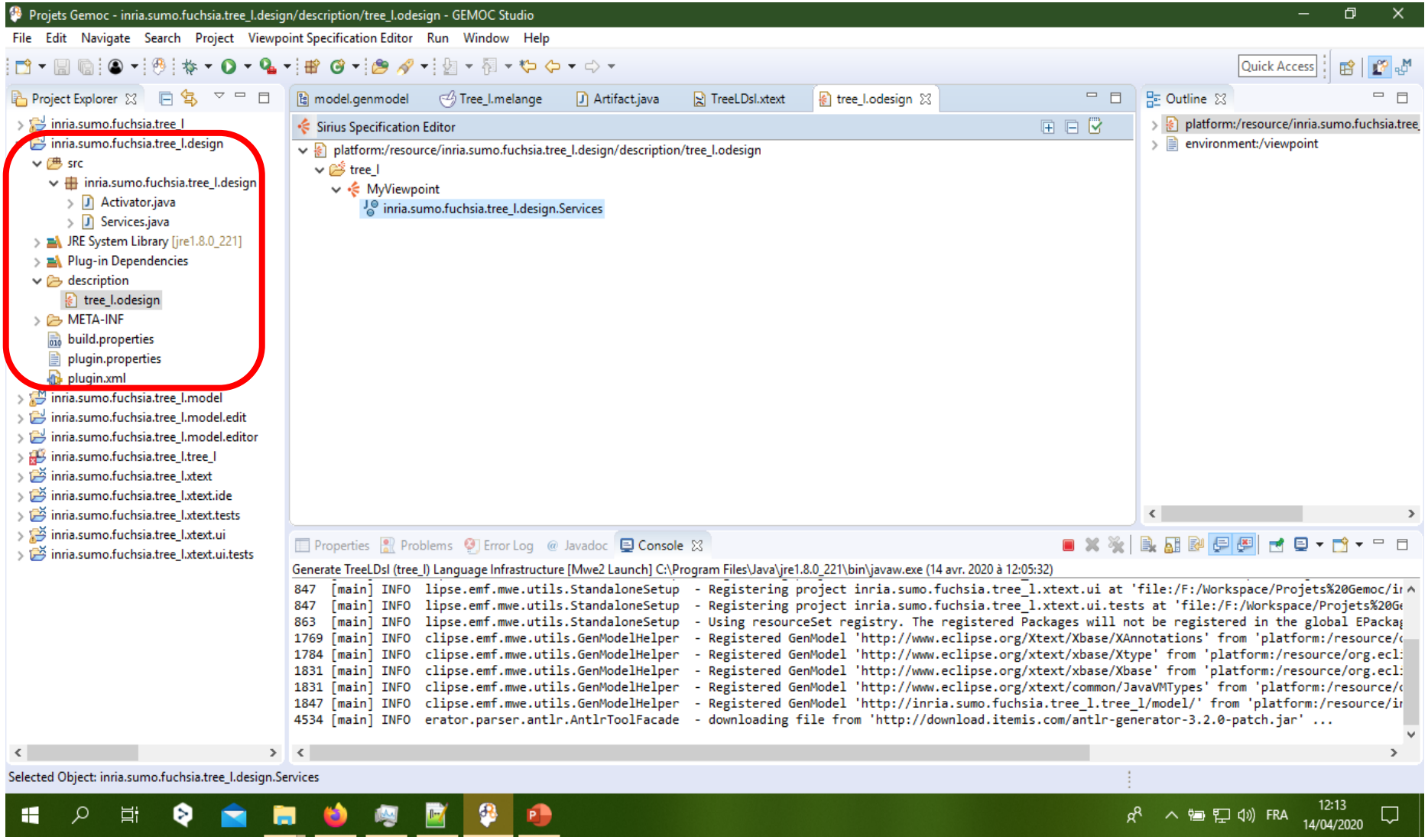
TRAVAIL PRATIQUE
6- *GÉNÉRATION D'UNE SYNTAXE GRAPHIQUE (SIRIUS) CONCRÈTE DU LANGAGE*



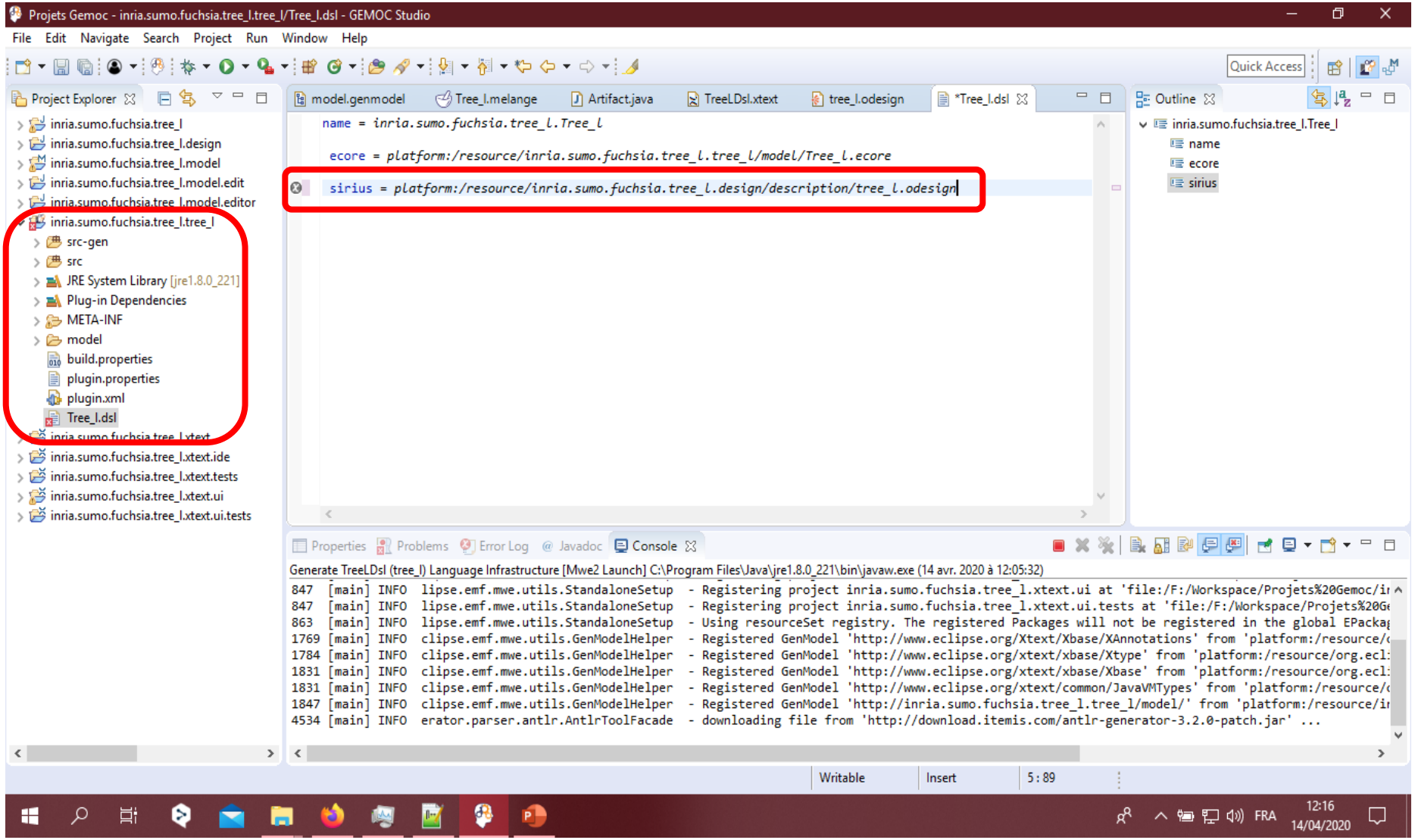
TRAVAIL PRATIQUE
6- *GÉNÉRATION D'UNE SYNTAXE GRAPHIQUE (SIRIUS) CONCRÈTE DU LANGAGE*



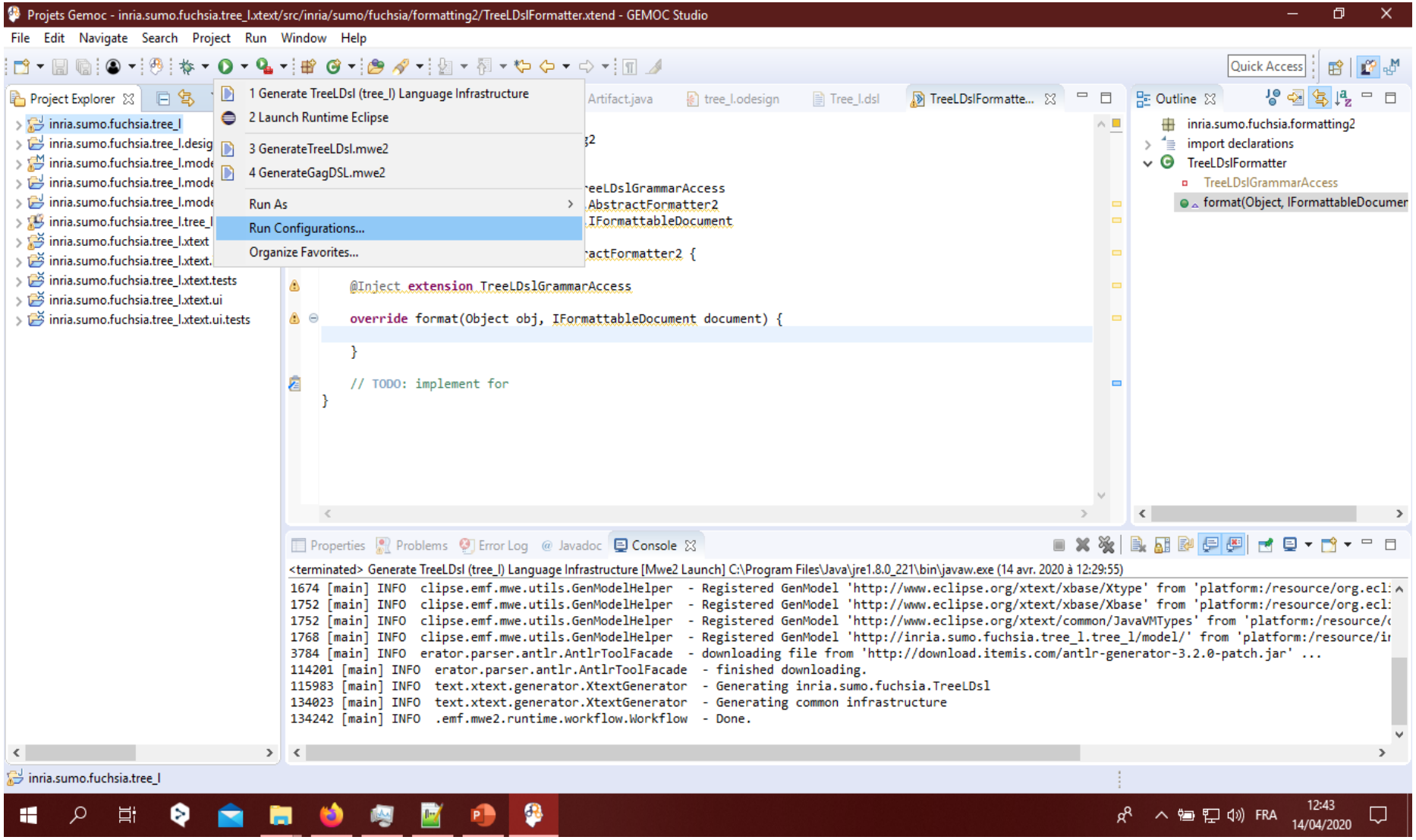
TRAVAIL PRATIQUE
6- *GÉNÉRATION D'UNE SYNTAXE GRAPHIQUE (SIRIUS) CONCRÈTE DU LANGAGE*



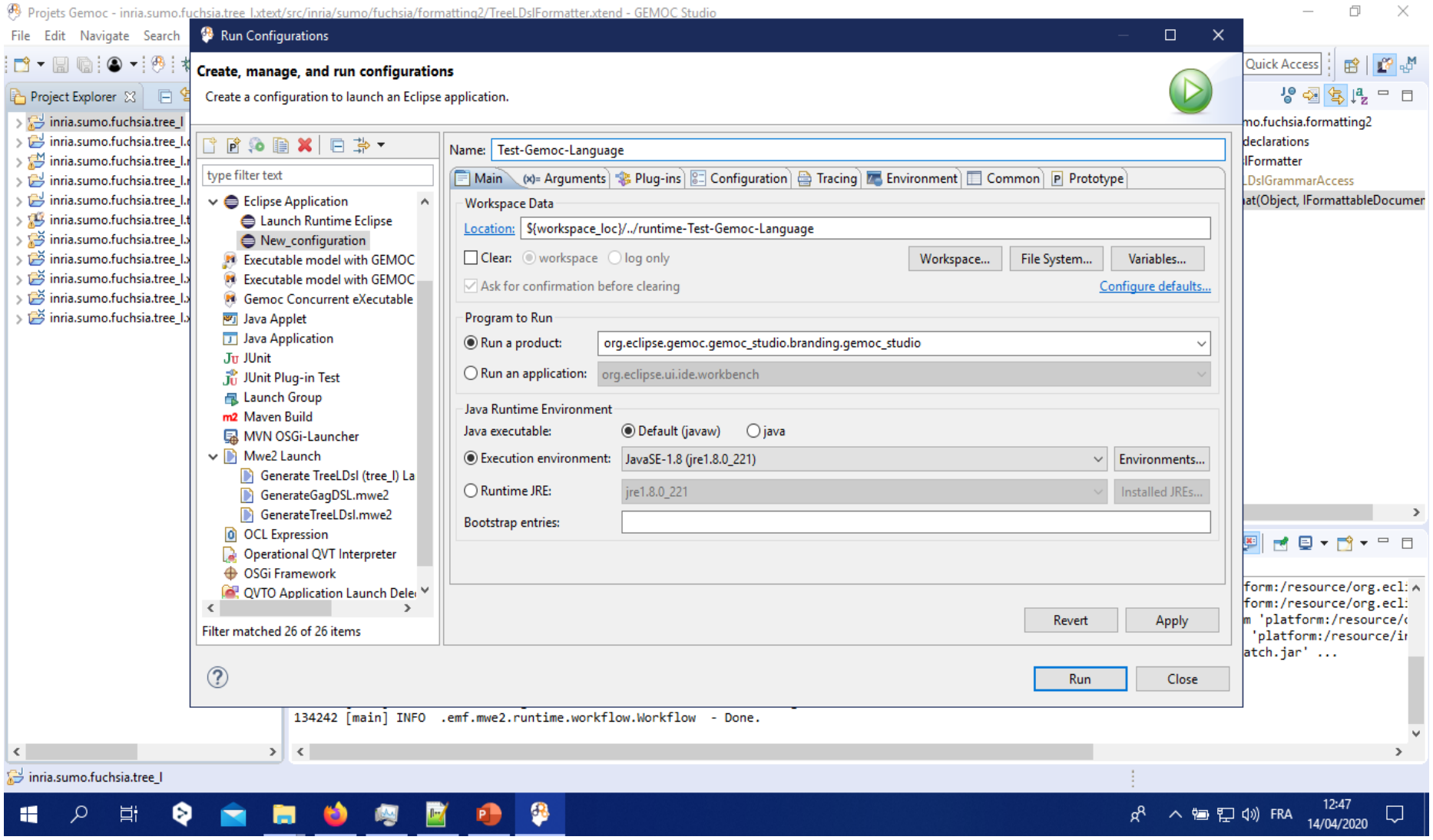
TRAVAIL PRATIQUE
6- *GÉNÉRATION D'UNE SYNTAXE GRAPHIQUE (SIRIUS) CONCRÈTE DU LANGAGE*



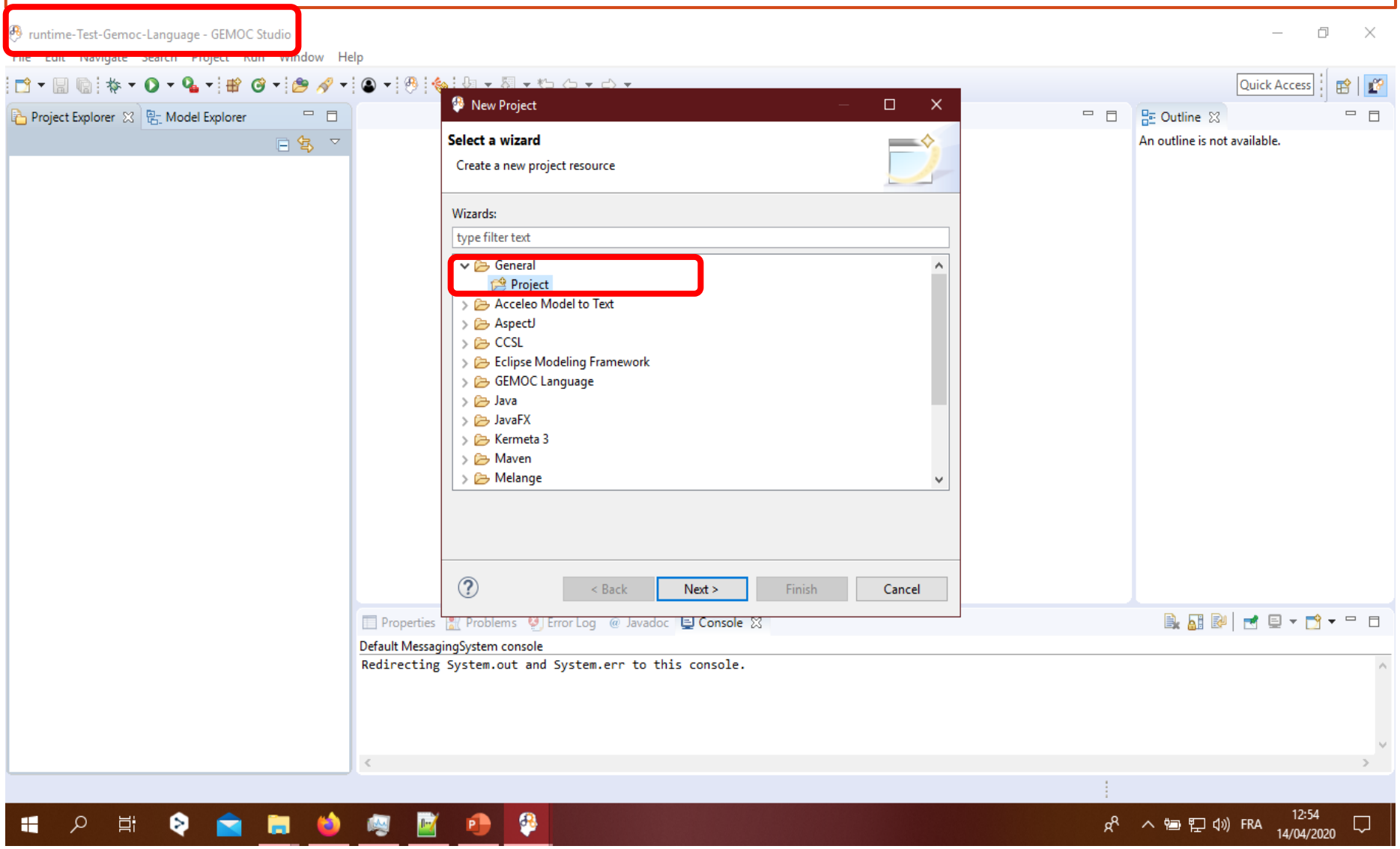
TRAVAIL PRATIQUE
7- *UTILISATION DU LANGAGE MODÉLISÉ*



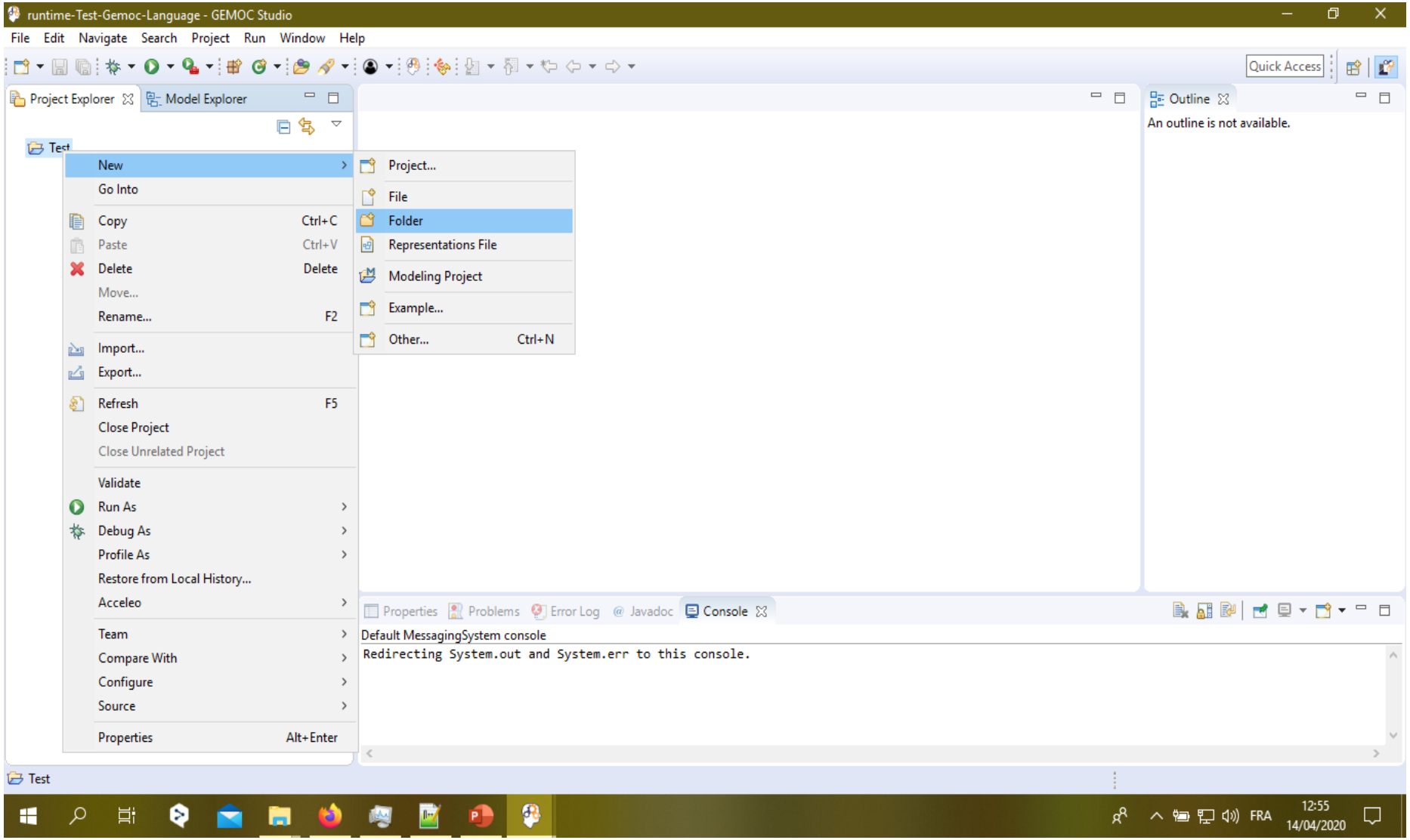
TRAVAIL PRATIQUE
7- *UTILISATION DU LANGAGE MODÉLISÉ*



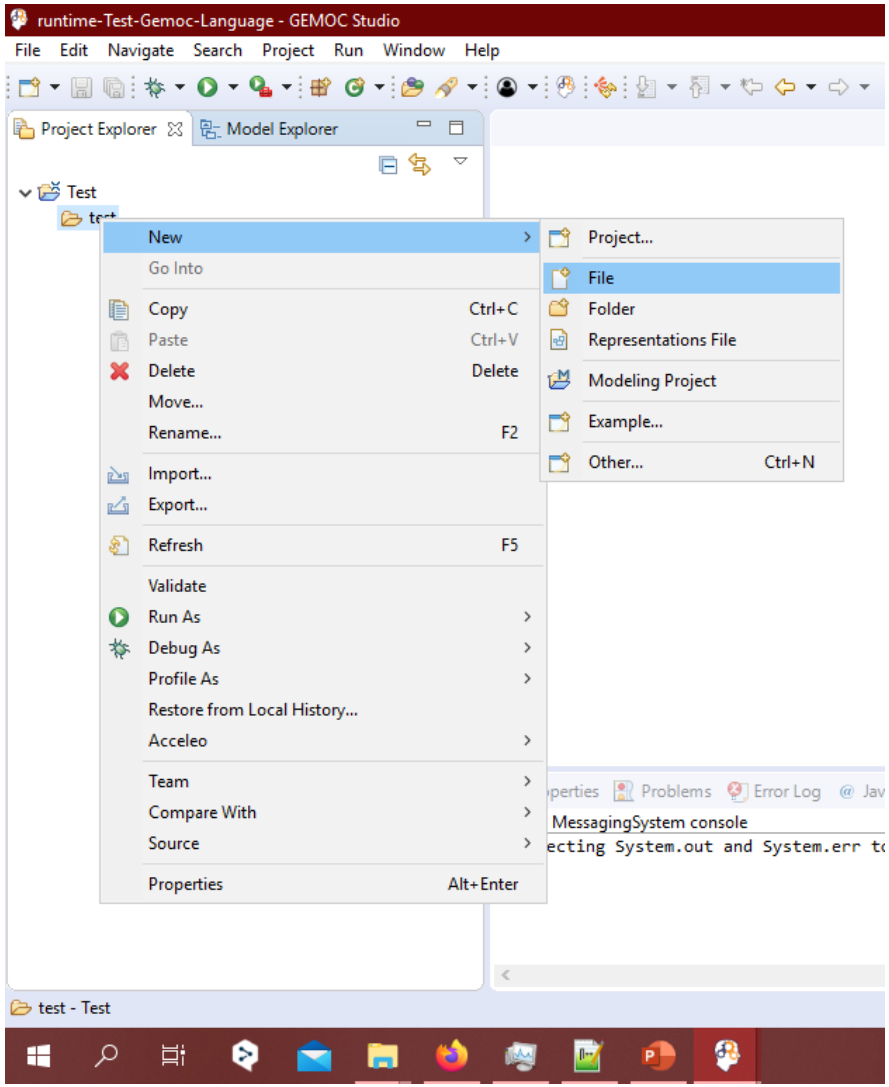
TRAVAIL PRATIQUE
7- *UTILISATION DU LANGAGE MODÉLISÉ*

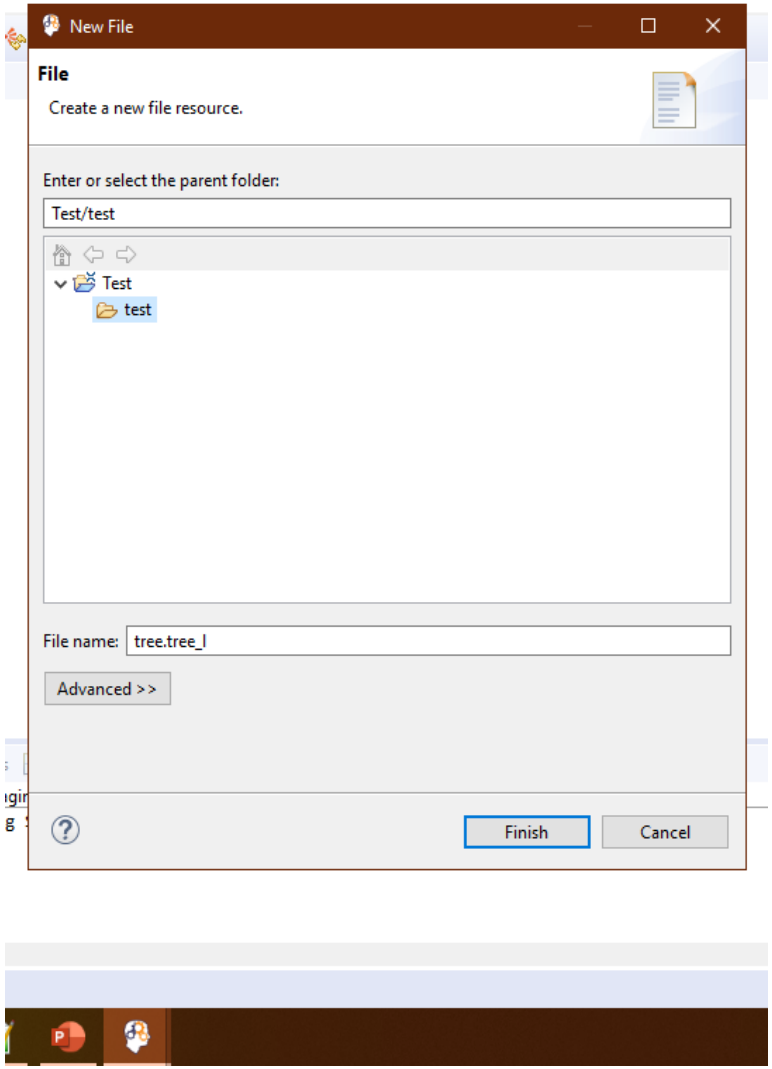


TRAVAIL PRATIQUE
7- *UTILISATION DU LANGAGE MODÉLISÉ*

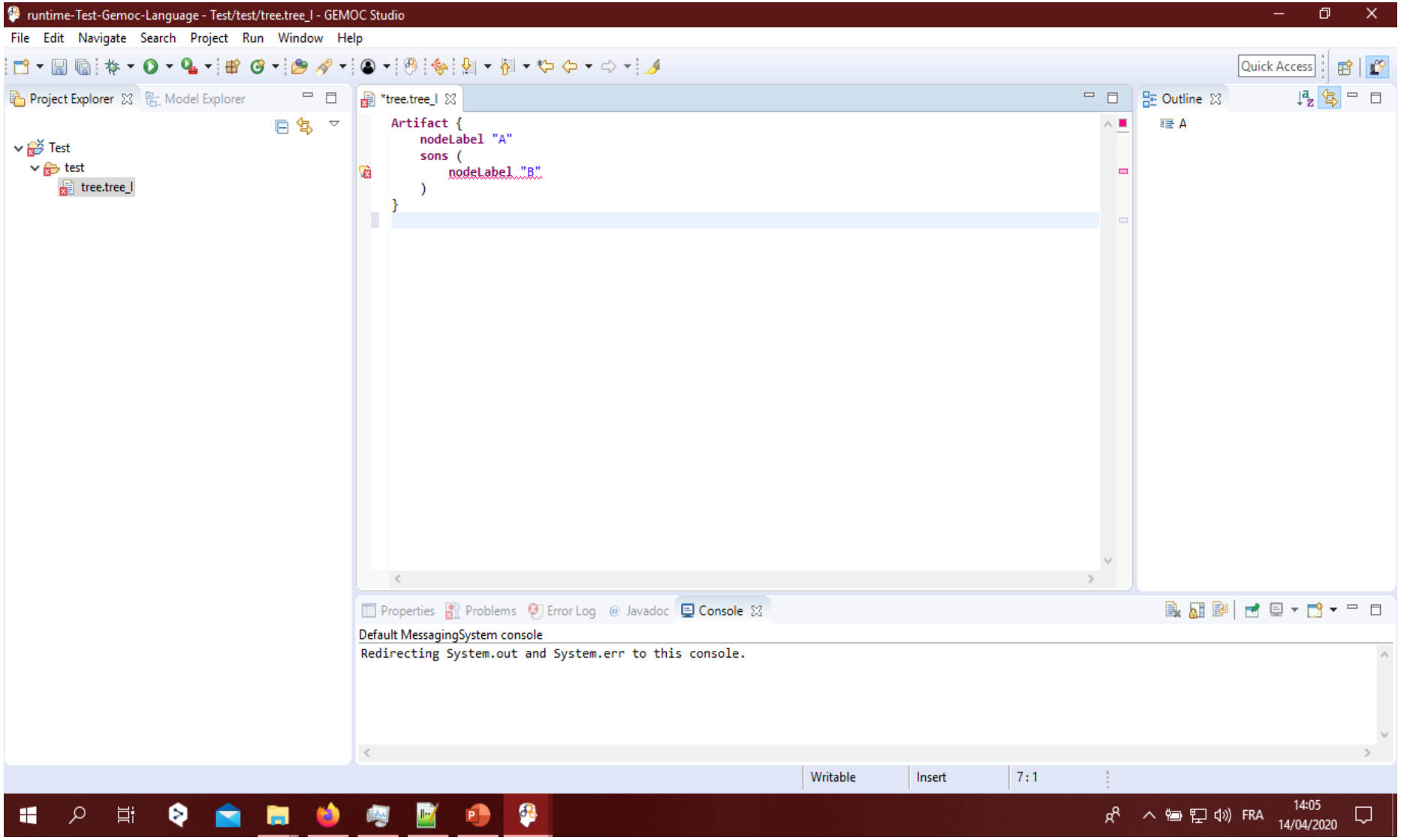


TRAVAIL PRATIQUE
7- *UTILISATION DU LANGAGE MODÉLISÉ*

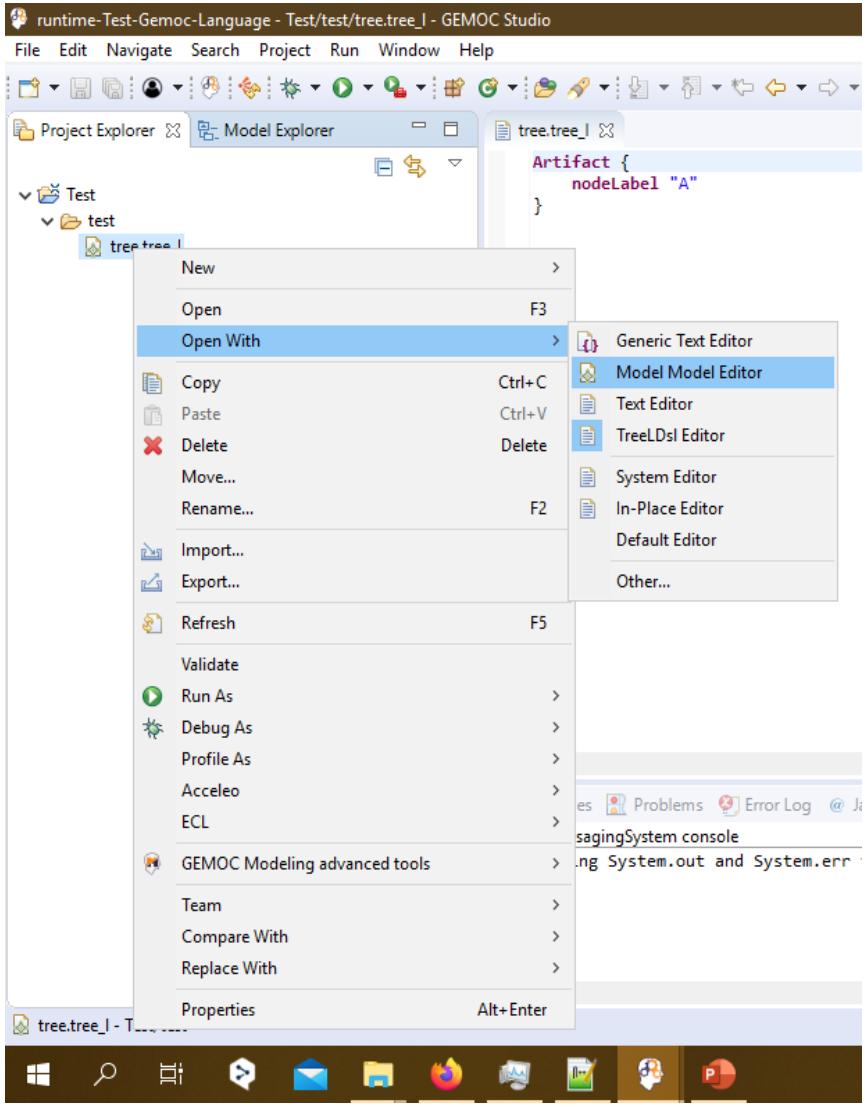


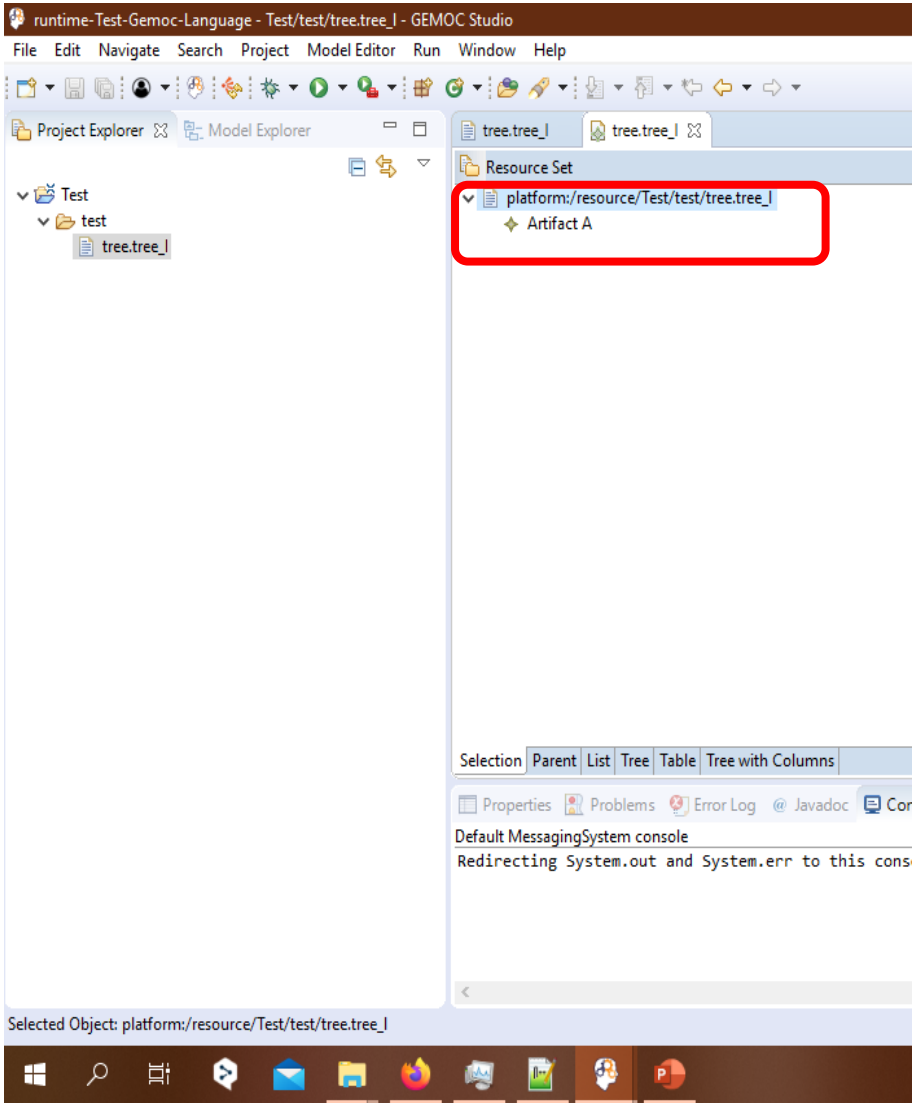


TRAVAIL PRATIQUE
7- *UTILISATION DU LANGAGE MODÉLISÉ*



TRAVAIL PRATIQUE
7- *UTILISATION DU LANGAGE MODÉLISÉ*





PETIT EXTRA

Retrouvez une présentation de l'exemple déroulé dans ce document, sous forme de tutoriel vidéo ici

https://drive.google.com/open?id=1OdYBvDFHPG2q6J4Pyhkt_-8_CAR2vd9E

PERSPECTIVES ET CONCLUSION

- Dérouler un use case plus élaboré. Celui-ci pourrait être intéressant → http://gemoc.org/gemoc-studio-old/publish/tutorial_markedgraph/html_single/GuideTutorialMarkedGraph.html

- Apprendre à créer un syntaxe graphique avec Sirius

- Comprendre la notion de « sémantique opérationnelle » des langages, d'aspects, de coordination et d'exécution de ceux-ci

- GEMOC et GAG ????

- Etc.