Eclipse Modelling Framework : Foundations and Testing

Sagar Sen (2nd Year PhD student, INRIA, Rennes)

Date: 16th July, 2008

Venue: Modelling and Simulation Design Lab, McGill University

Eclipse Modelling Framework

- A set of software tools that come with Eclipse, the very popular code editor, manager, highlighter etc.
- Specifying a Meta-model: *Ecore*
- Code Generation : Java Classes
- Specification of Model Transformation in Java

Core of EMF = ECore

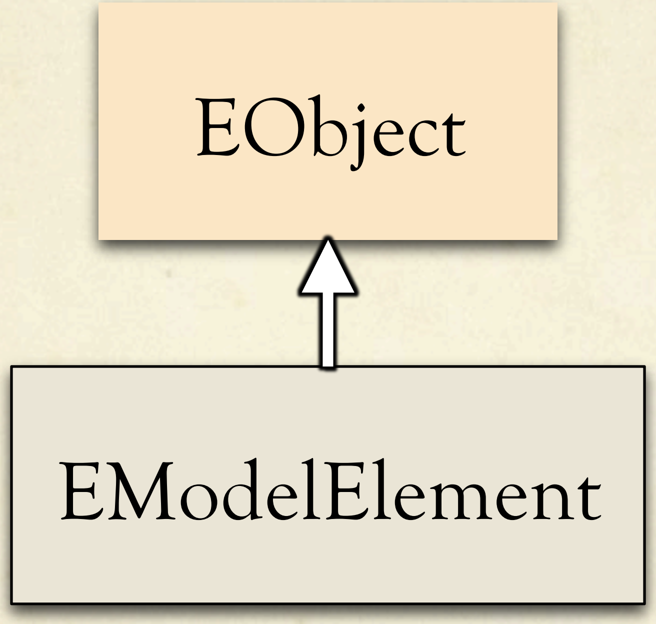
- An IBM Eclipse standard for defining concepts and relationships in a meta-model
- Embodiment: XML file with extension “*.ecore*”
- Visual Syntax: Like UML Class Diagrams
- But, minimal compared to UML
- Adopted by industry and academia extensively in Europe, and some parts of North America

ECore

In Ecore everything is an object.

Actually an EObject.

ECore

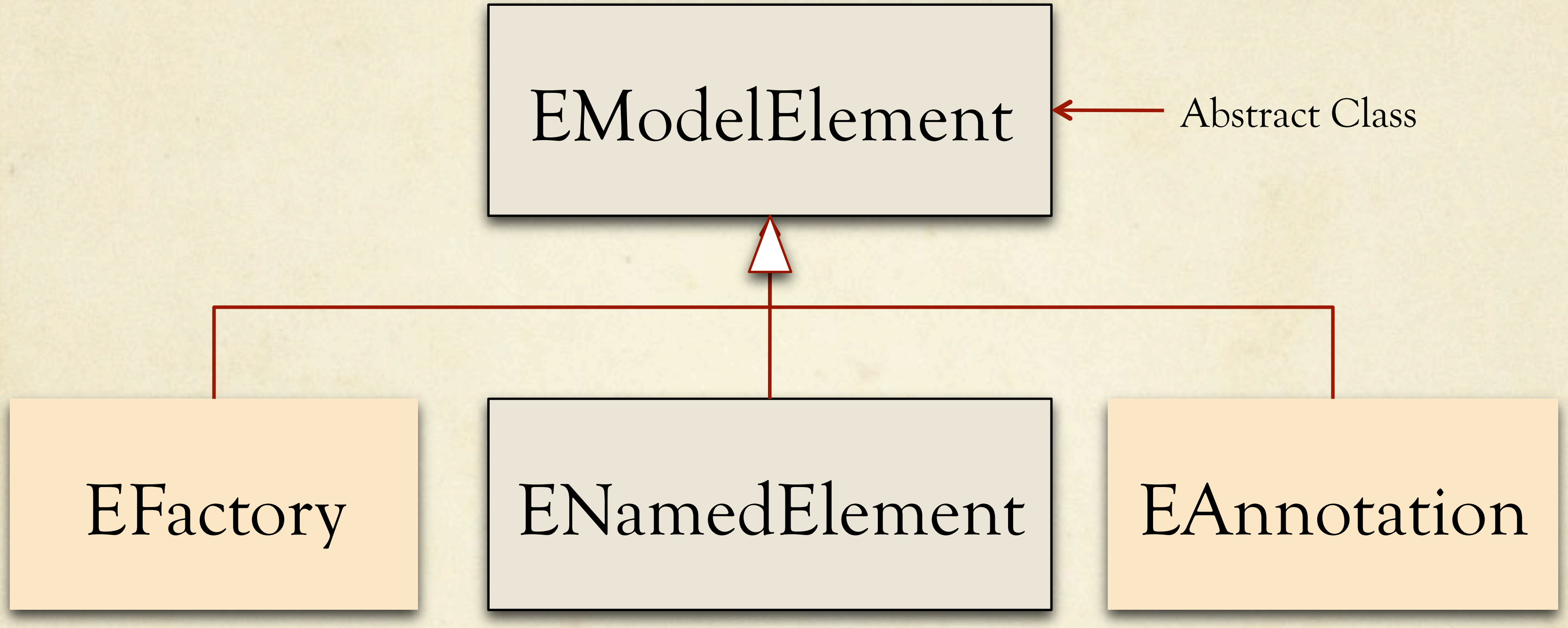


EModelElement extends EObject to create Modelling Elements

But objects are a notion from the programming world!

(Keeps in touch with the programming world! < Key Ecore Idea!)

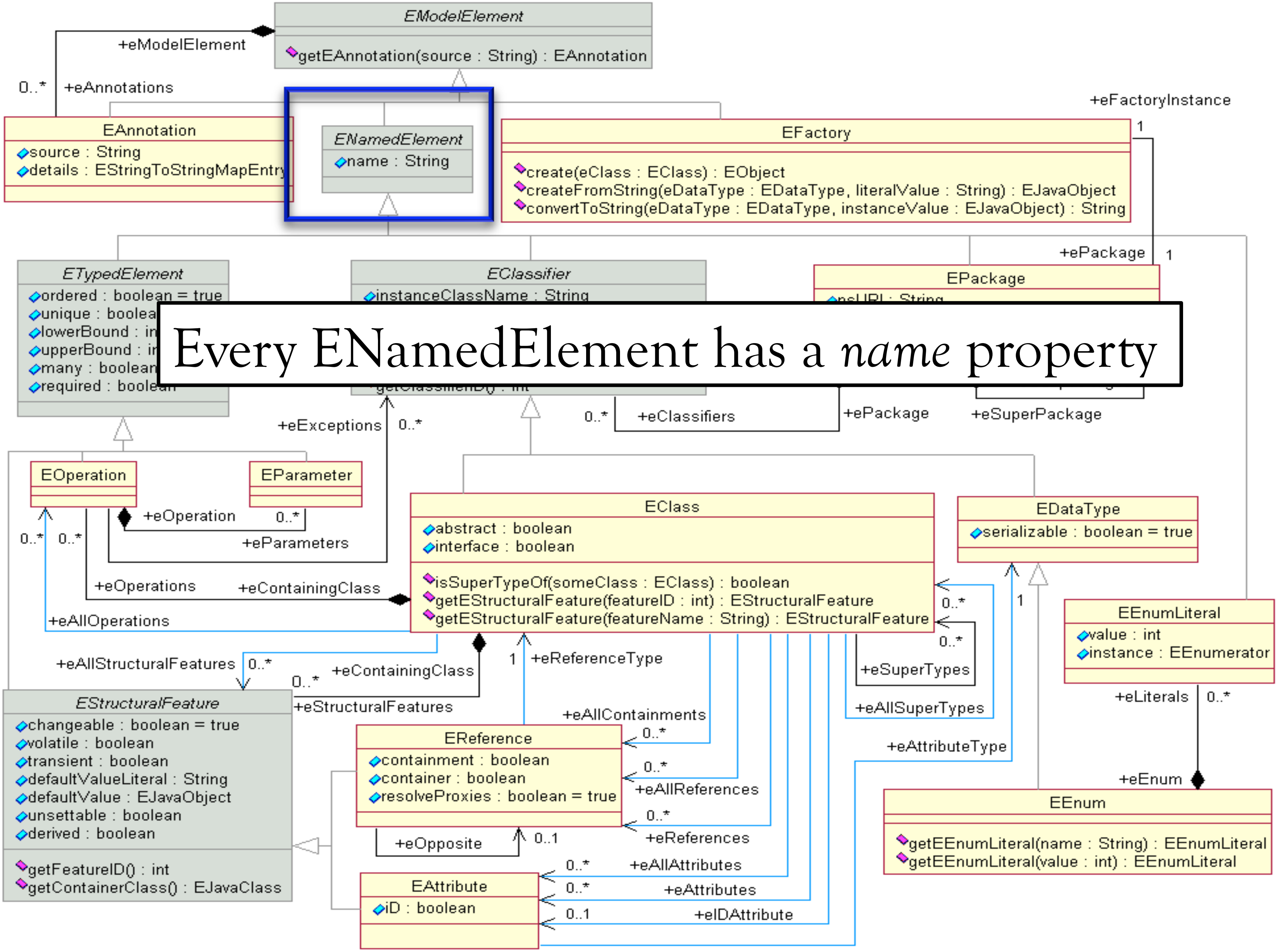
ECore



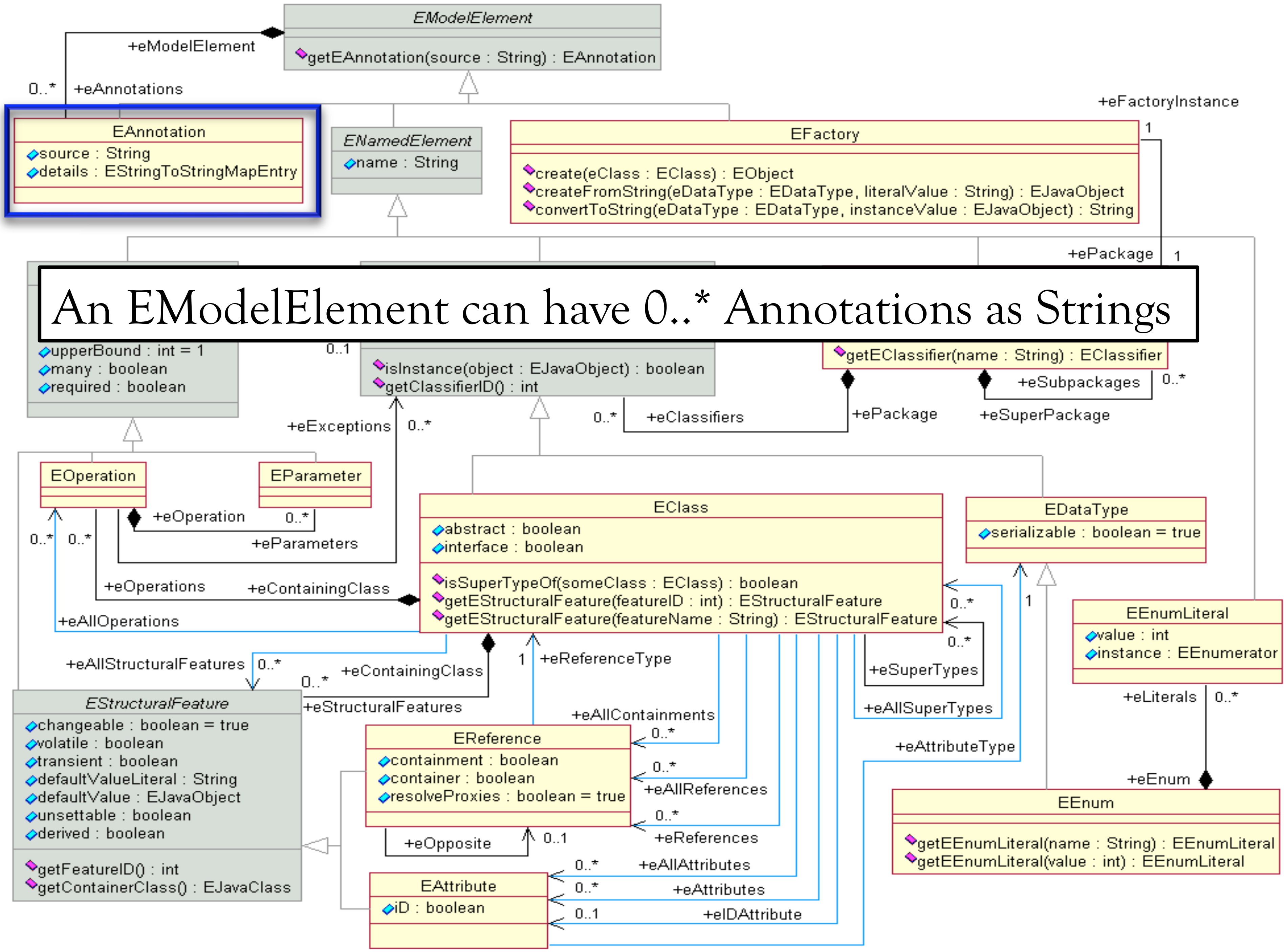
A Model Factory to create Model Elements

Some Model Elements are Named

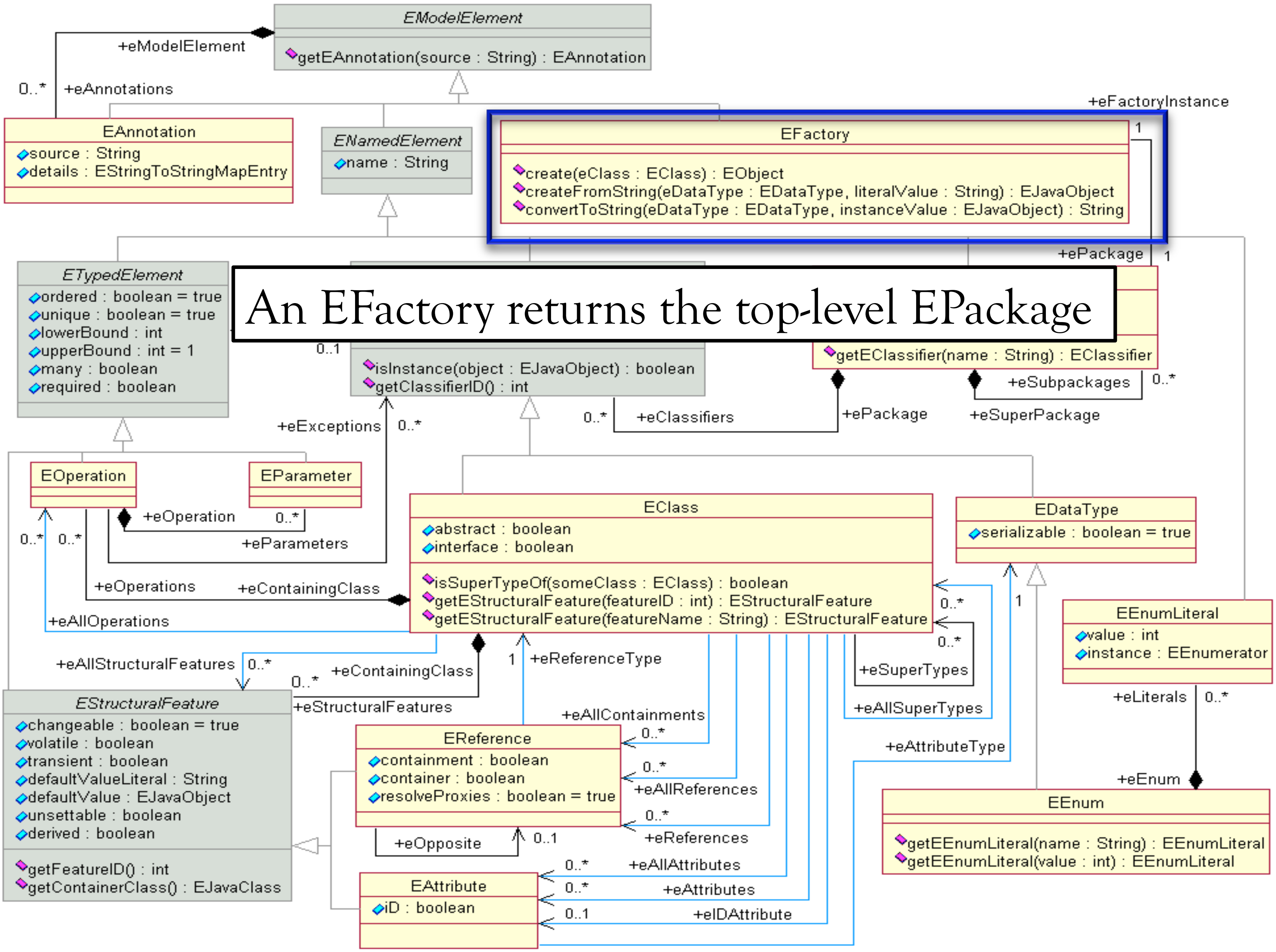
And, Some Model Elements have Annotations



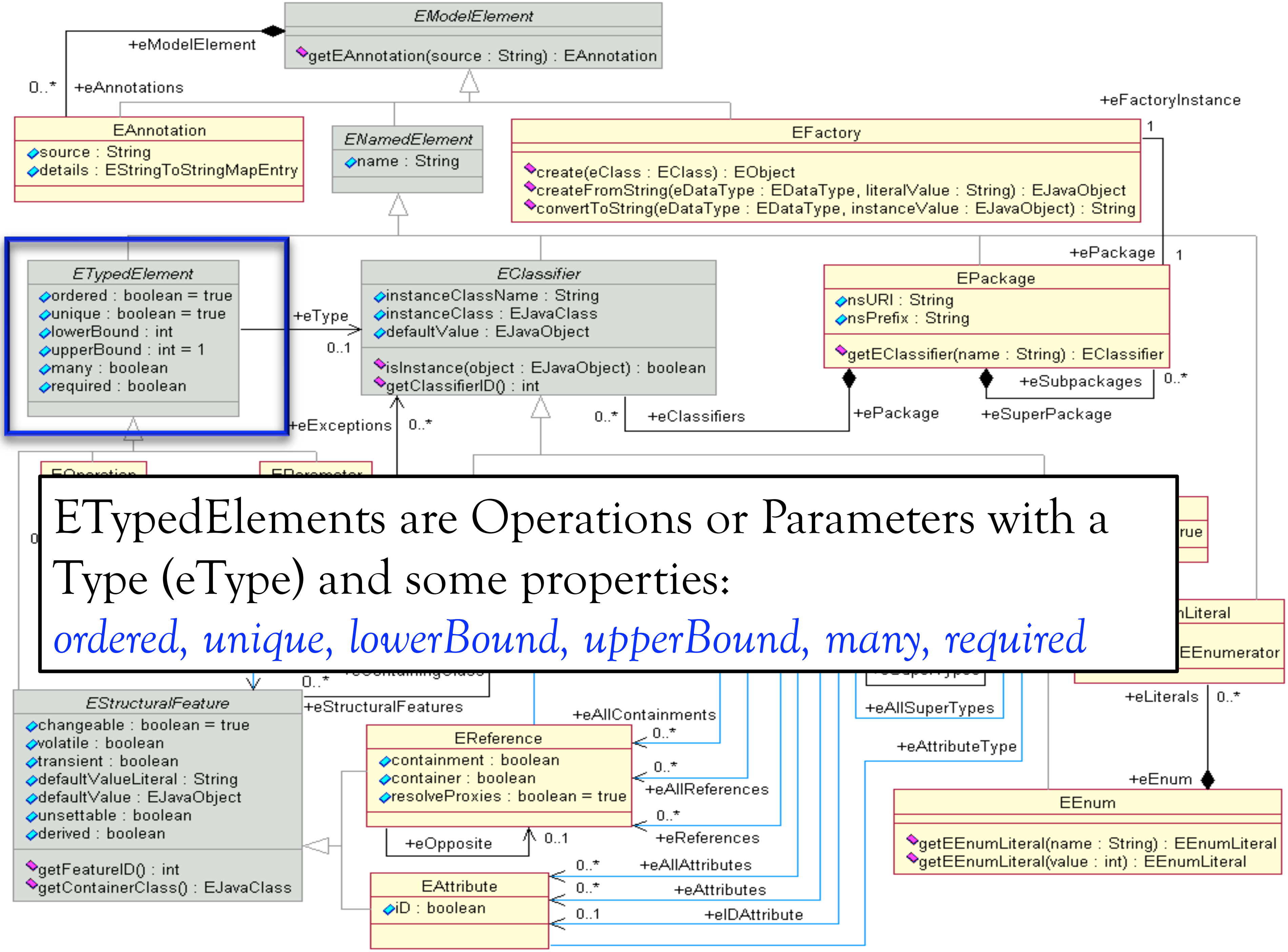
Every ENamedElement has a *name* property



An EModelElement can have 0..* Annotations as Strings

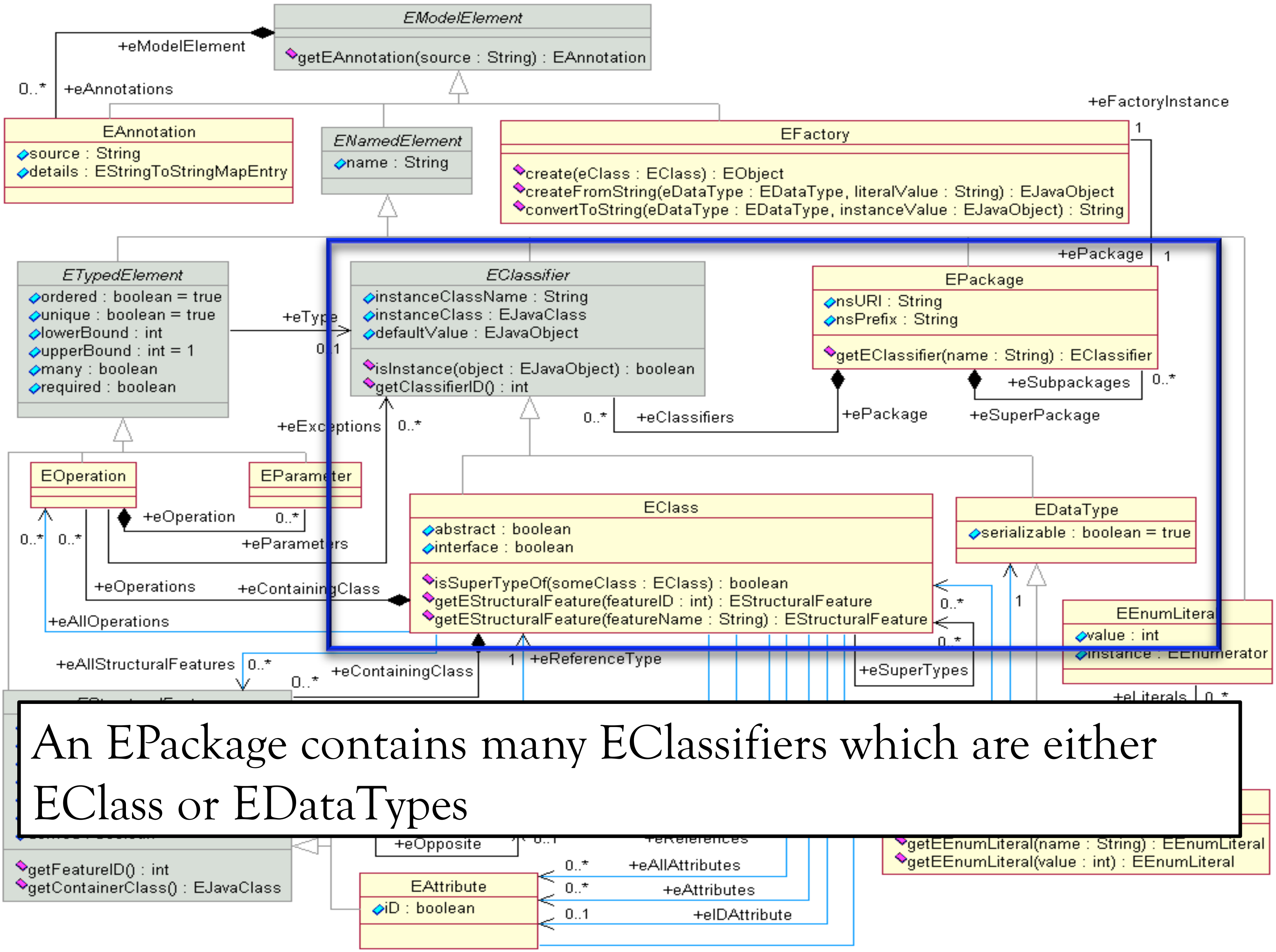


An EFactory returns the top-level EPackage

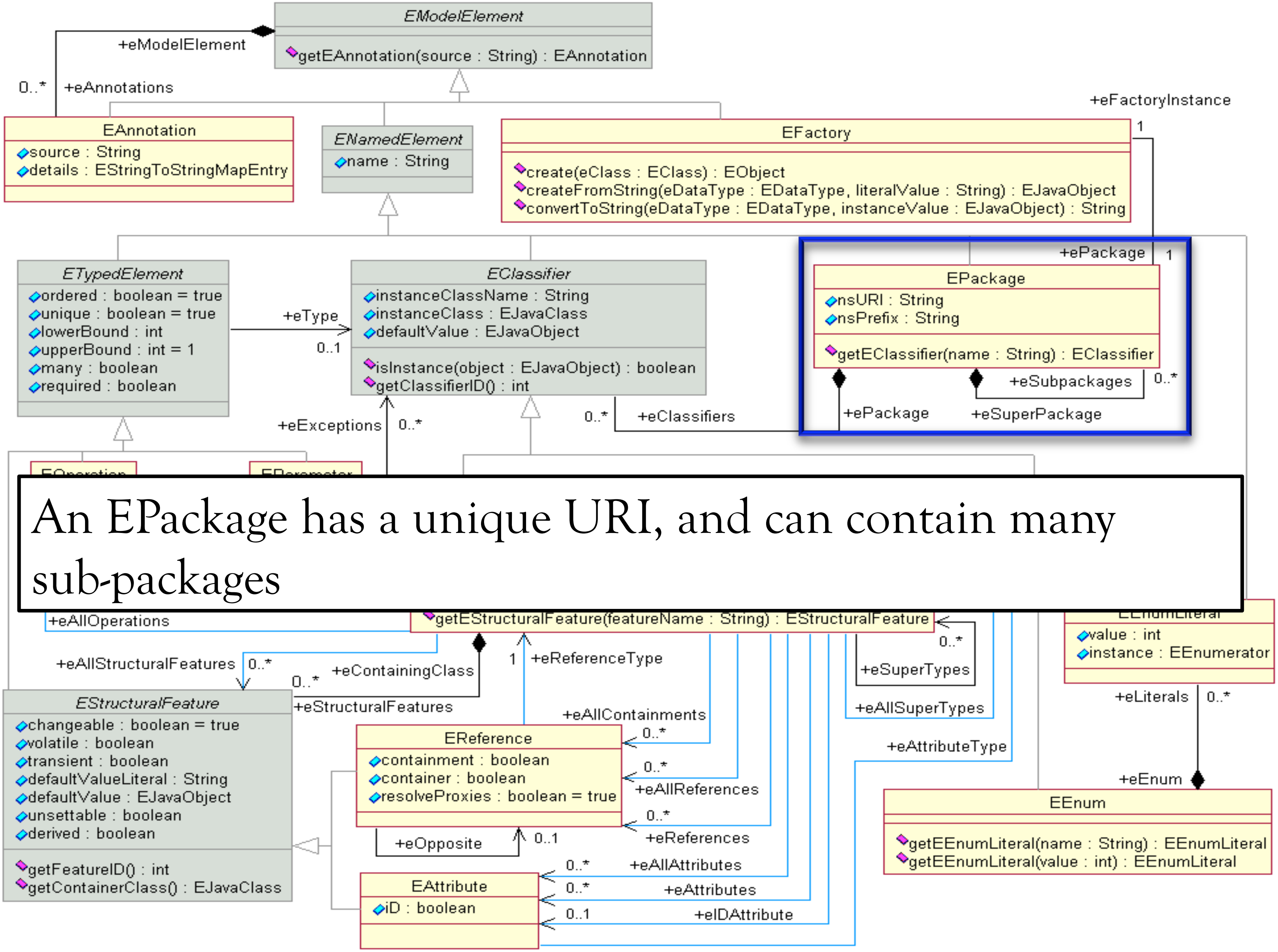


ETypedElements are Operations or Parameters with a Type (eType) and some properties:

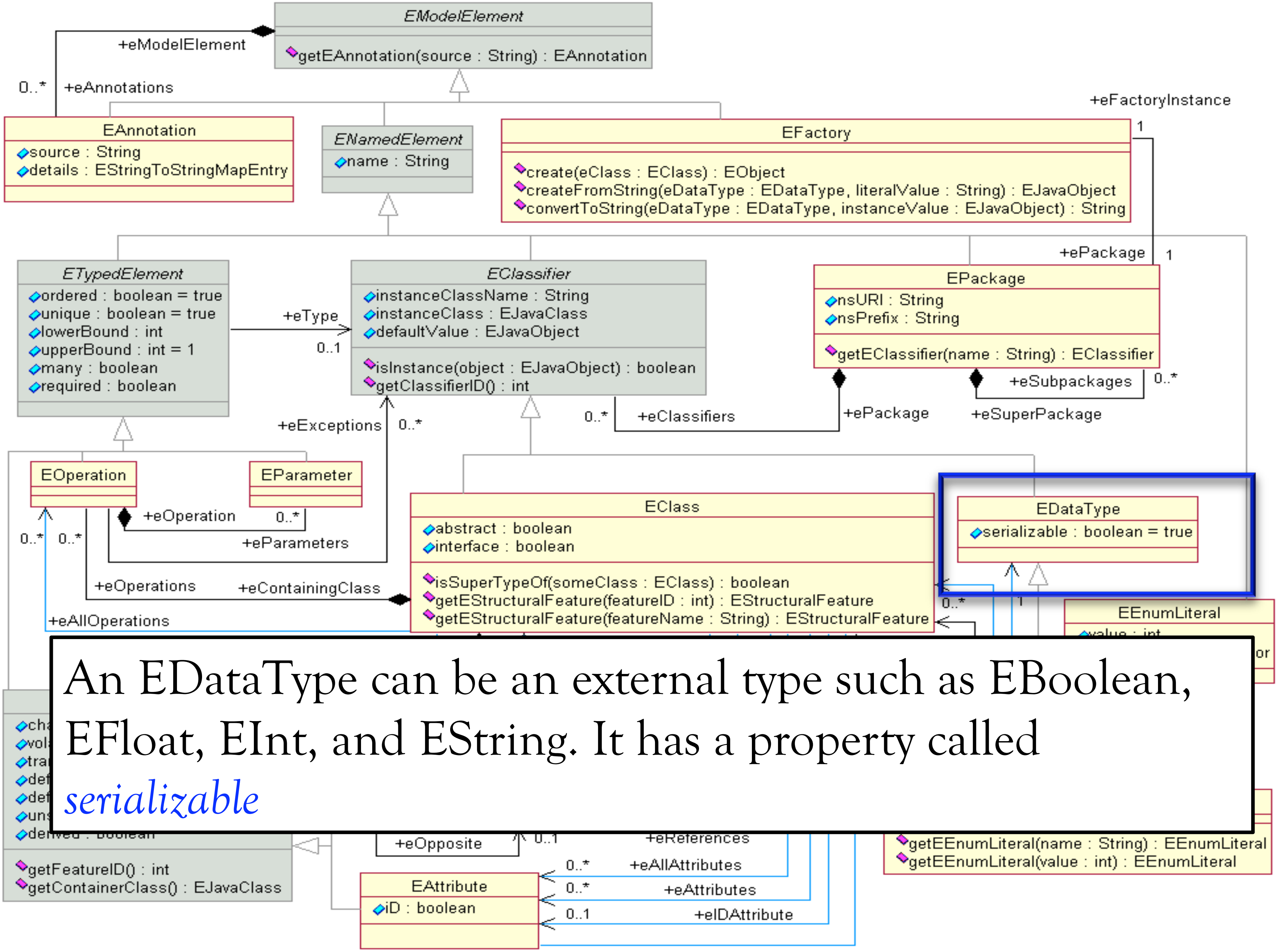
ordered, unique, lowerBound, upperBound, many, required



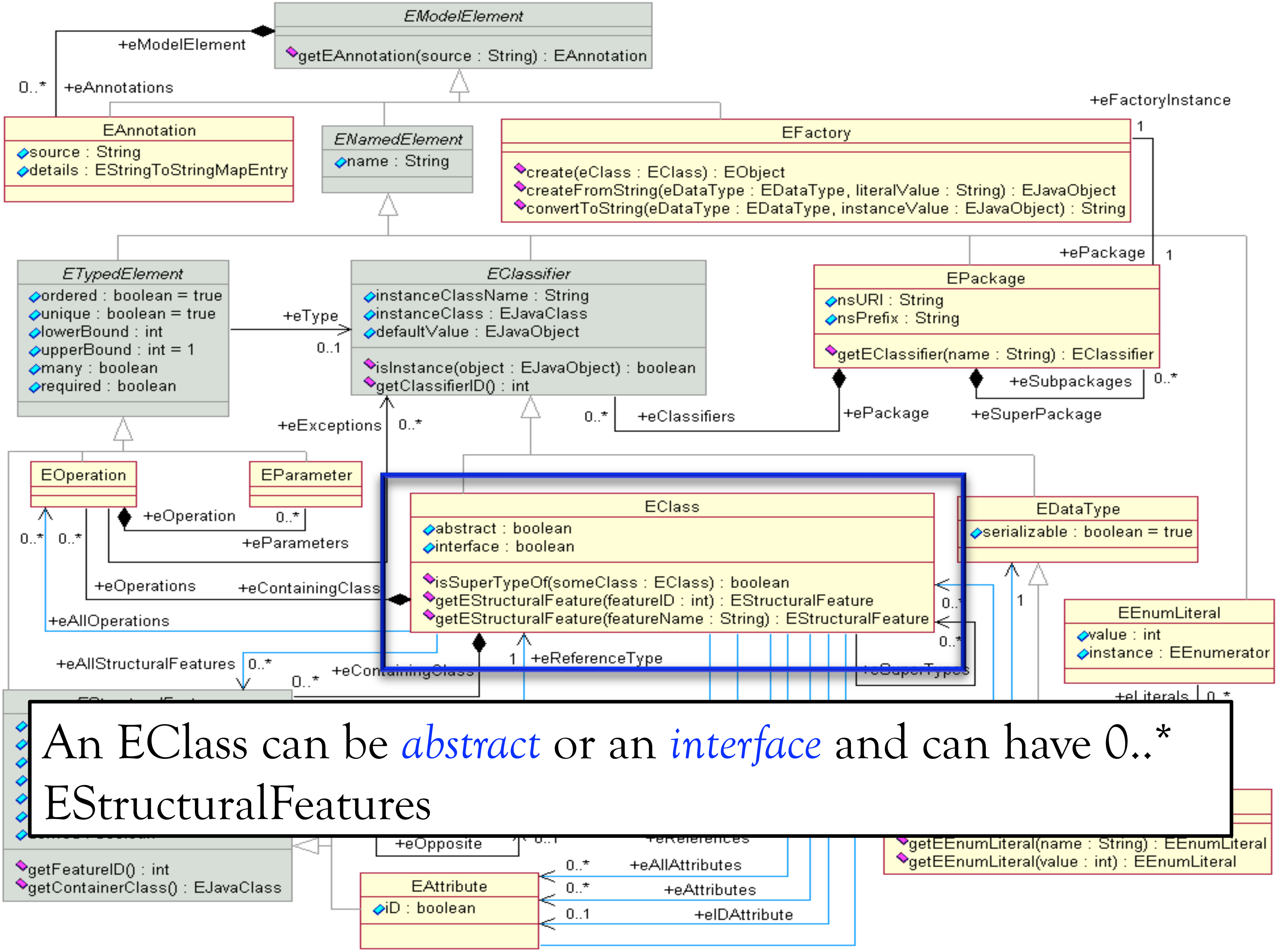
An EPackage contains many EClassifiers which are either EClass or EDataTypes



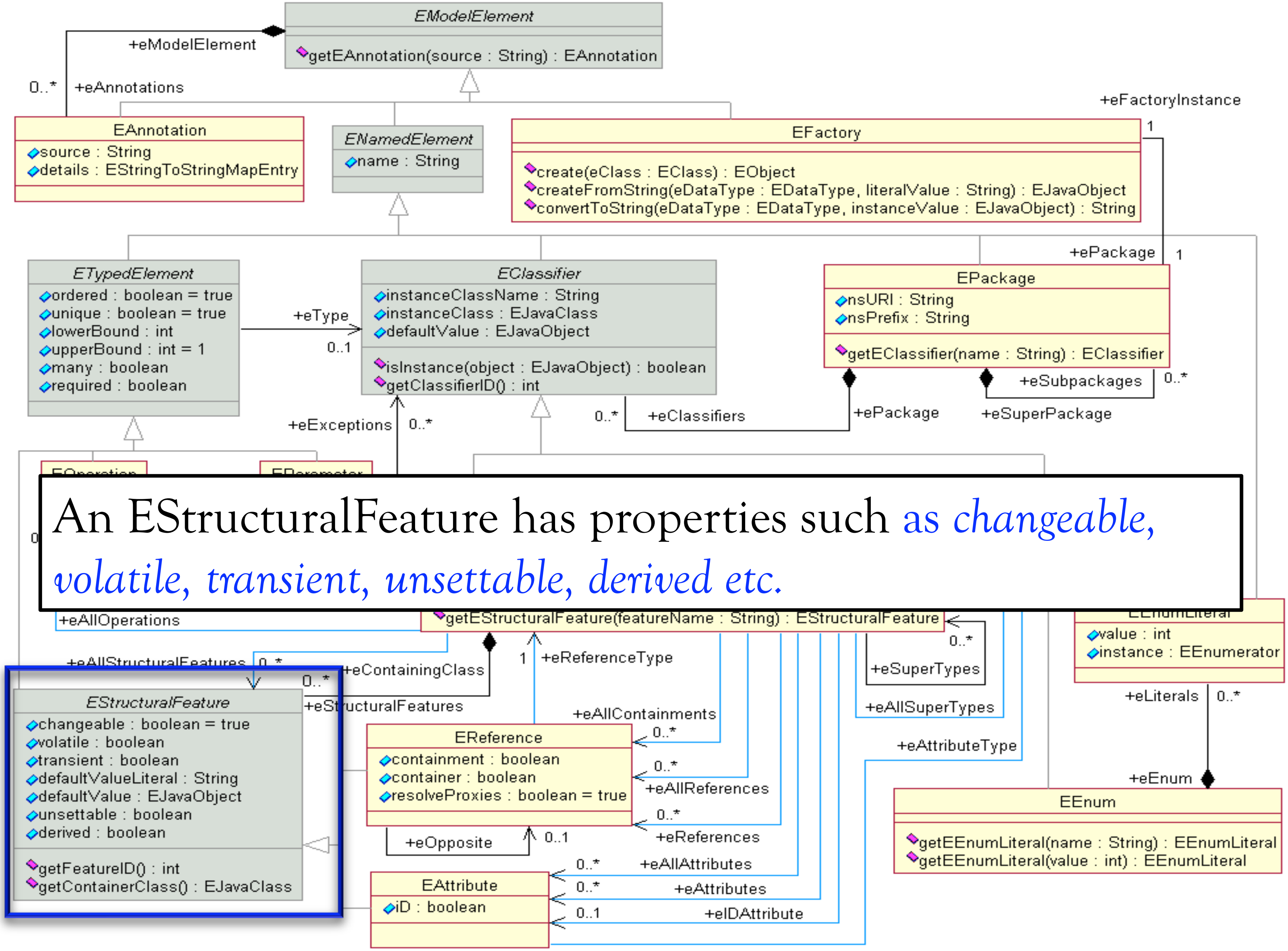
An EPackage has a unique URI, and can contain many sub-packages



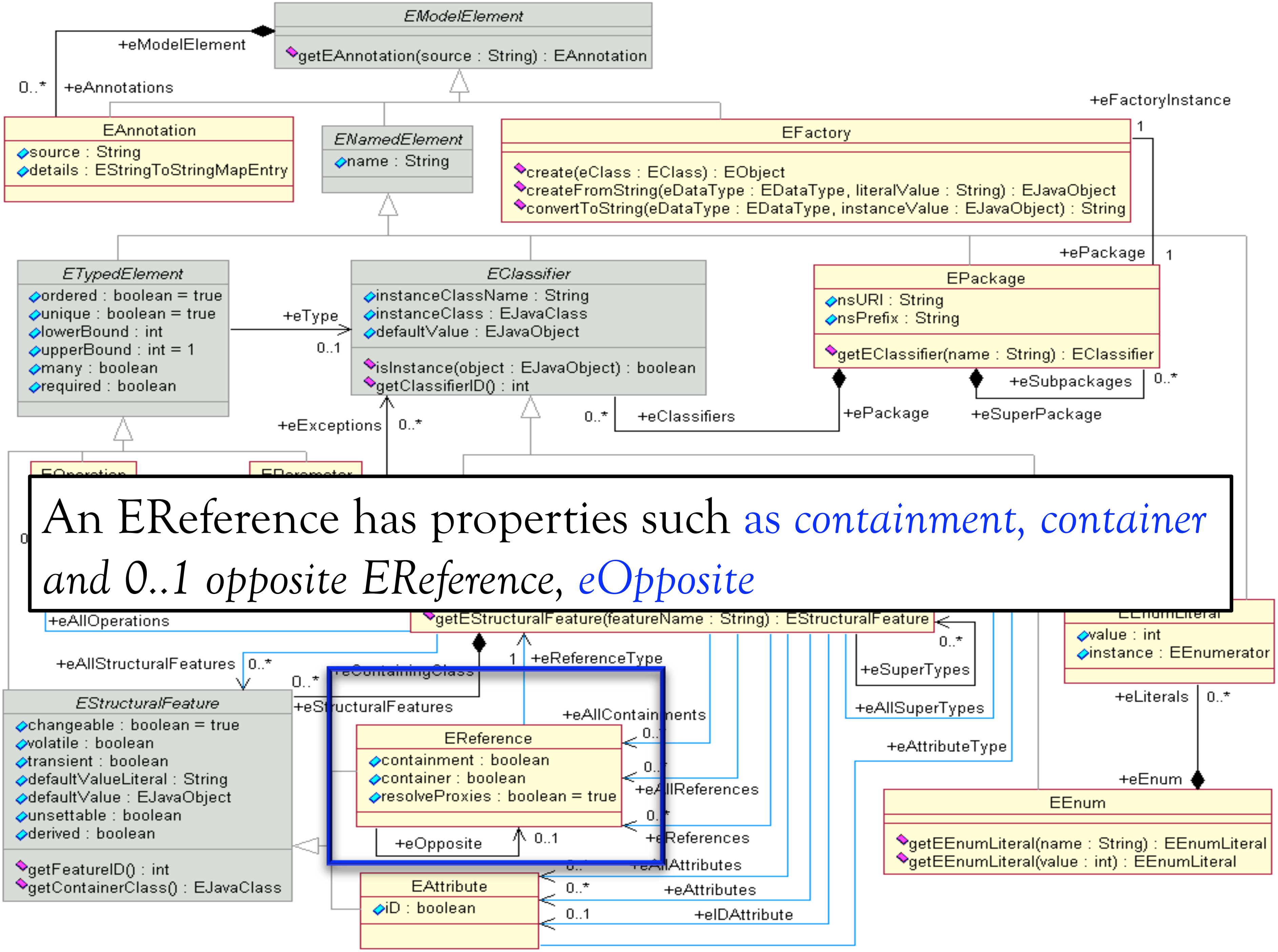
An EDataType can be an external type such as EBoolean, EFloat, EInt, and EString. It has a property called *serializable*



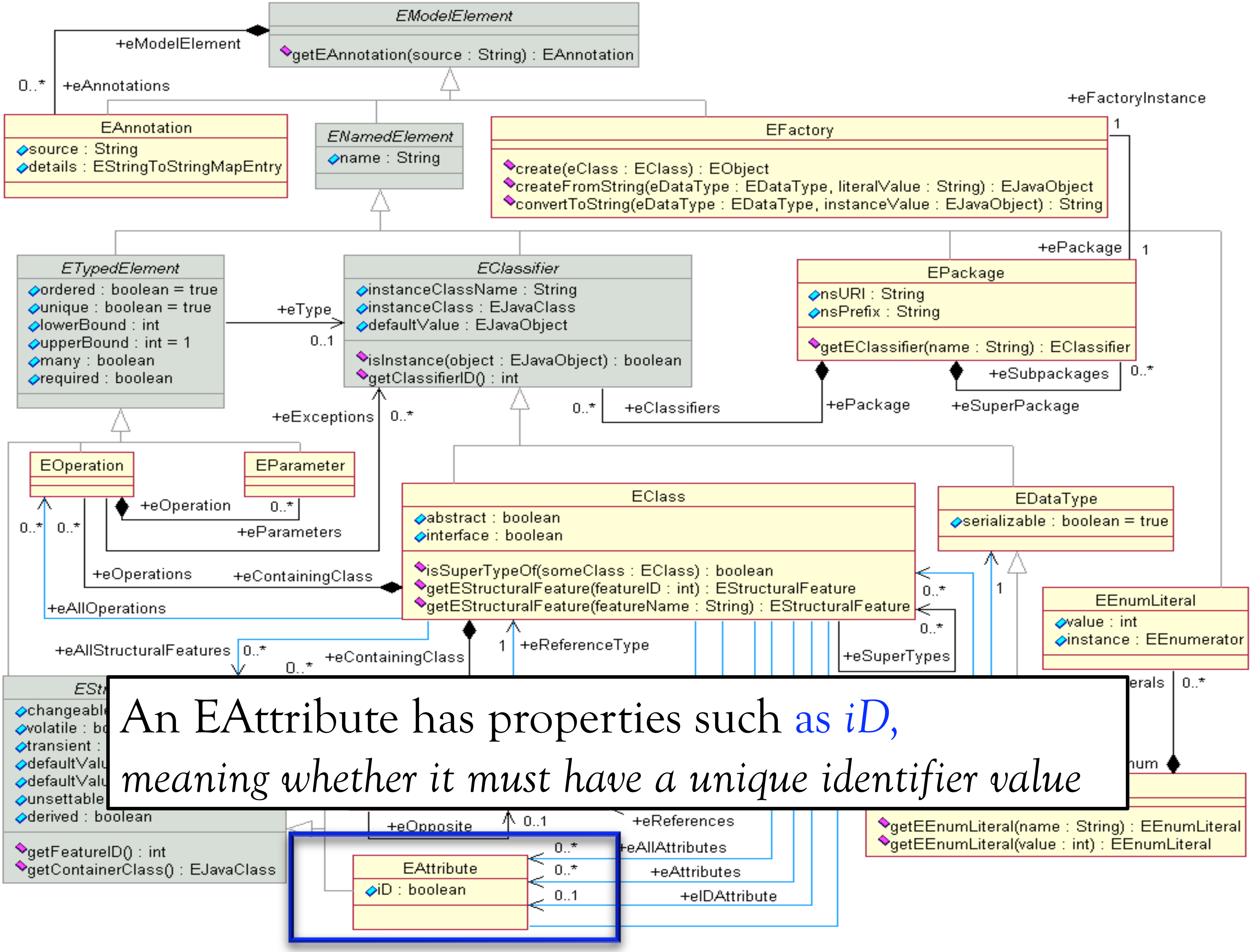
An EClass can be *abstract* or an *interface* and can have 0..* EStructuralFeatures



An EStructuralFeature has properties such *as changeable, volatile, transient, unsettable, derived etc.*

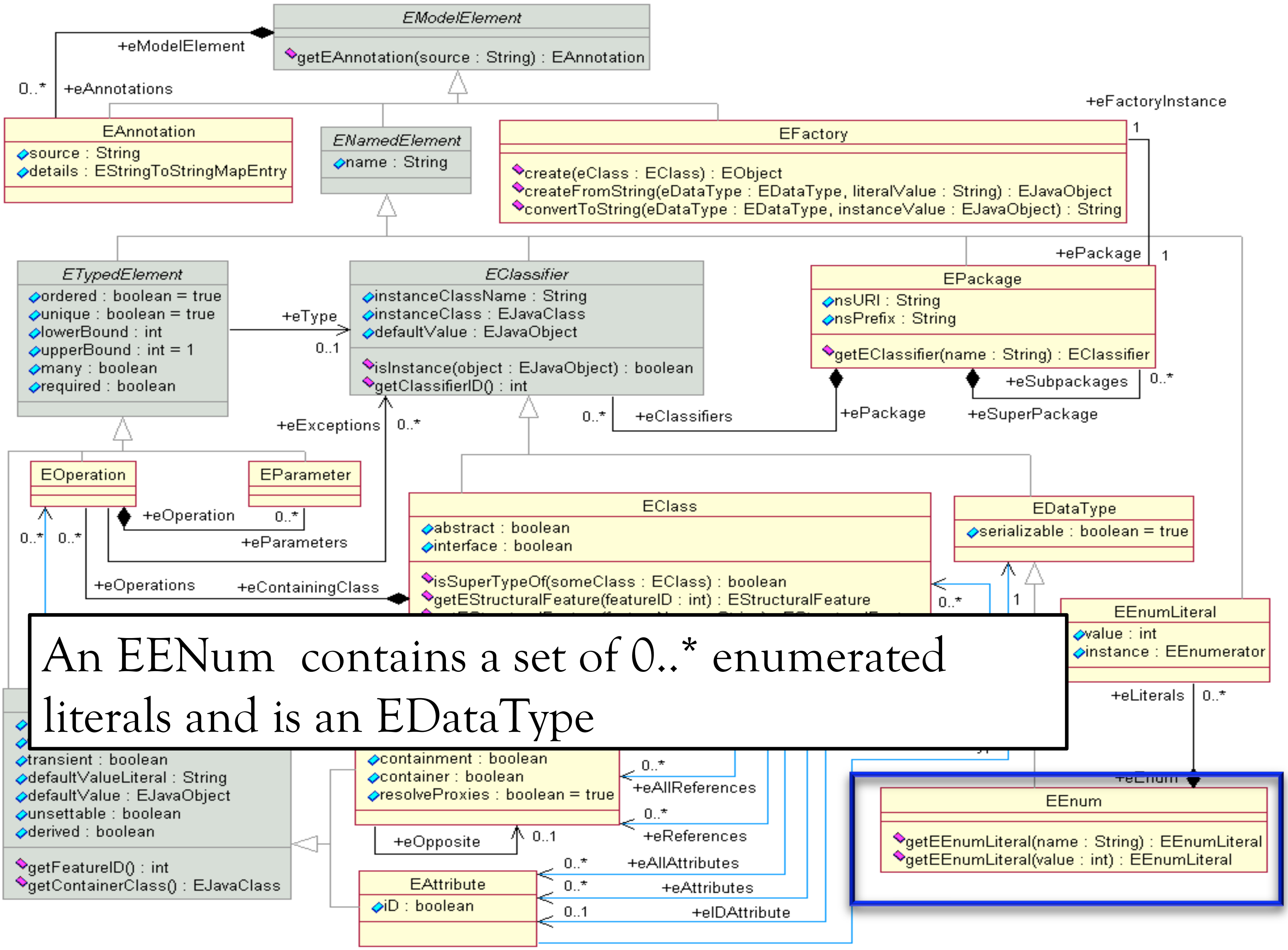


An EReference has properties such *as containment, container and 0..1 opposite EReference, eOpposite*



An EAttribute has properties such as *iD,*

meaning whether it must have a unique identifier value



An EENum contains a set of 0..* enumerated literals and is an EDataType

ECore Transformations

- How is an .ecore file containing the meta-model serialized or stored ?

  Lets dissect an example ECore File for

  Simple UML Class Diagrams!

ECore Model

- **The Top-level EPackage**

```
<?xml version="1.0" encoding="UTF-8"?>
<ecore:EPackage xmi:version="2.0"
    xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance"
    xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore"
name="simpleUML_MM"
    nsURI="http://simpleUML_MM.ecore"
nsPrefix="simpleUML_MM">
```

An ECore model always starts with an EPackage

ECore Model

- **EClassifiers enclosed in EPackages**

```
<eClassifiers xsi:type="ecore:EClass" name="Classifier"
abstract="true">

 </eClassifiers>
```

An ECore model contains eClassifier tags with the specification Of ecore:EClass or ecore:EDataType as Types and associated Properties such as abstract, interface etc.

ECore Serialization Summary

What is in them, basically?

- EPackages with properties
- EClassifiers with properties
- EStructuralFeatures with properties
- We also have EAnnotations but we ignore them for the ECore2Alloy Transformation

ECore Transformation : Testing

- Deals with EPackages, EClassifiers, EStructuralFeatures in an *.ecore* file
- We need to find partitions of the input domain to test a transformation for ECore Models.
- Lets summarize these partitions
- We consider a subset of ECore for the *ECore2Alloy* Transformation

EPackage Partitions

1 , > 1 EPackage objects ePackage with
ePackage.name="", ePackage.name!=""
ePackage.eSubPackeges=0,1,>1
ePackage.eSuperPackages=0,1,>1
ePackage.eClassifiers=0,1,>1

For example, we don't consider the properties nsPrefix and nsURI as we don't use it for the Ecore 2 Alloy Transformation

Total Models = 22

EDataType Partitions

1,>1 EDataType objects eDataType with

eDataType.name="", eDataType.name!="", eDataType.name!="EBool", eDataType.name!="EFloat", eDataType.name!="EInt", eDataType.name!="EBoolean",

eDataType.instanceClassName="", eDataType.instanceClassName!="" eDataType.instanceClassName= boolean, eDataType.instanceClassName= int, eDataType.instanceClassName= float, eDataType.instanceClassName= java.lang.String,

#eDataType.ePackage =0, #eDataType.ePackage=1,#eDataType.ePackage>1,

We consider special strings for the Ecore 2 Alloy Transformation

Total Models = 30

EClass Partitions

1,>1 EClass objects eClass with

eClass.name="", eClass.name!=""

eClass.abstract=True , eClass.abstract=False

#eClass.ePackage =0, #eClass.ePackage=1,#eClass.ePackage>1,
#eClass.eSuperTypes =0, #eClass.eSuperTypes=1,#eClass.eSuperTypes>1,
#eClass.eStructuralFeatures =0, #eClass.eStructuralFeatures =1,
#eClass.eStructuralFeatures >1,

We ignore eOperations and interface for the Ecore 2 Alloy Transformation

Total Models = 26

EAttribute Partitions

1,>1 EAttribute objects eReference with

eAttribute .name="", eAttribute .name!=""

eAttribute .iD=True, eAttribute .iD=False

eAttribute.changeable=True, eAttribute.changeable=False

eAttribute.defaultValue=boolean, eAttribute.defaultValue=int,

eAttribute.defaultValue=float, eAttribute.defaultValue=java.lang.String

eAttribute.defaultValueLiteral="", eAttribute.defaultValueLiteral!=""

#eAttribute.defaultValue.eContainingClass =0,

#eAttribute.defaultValue.eContainingClass =1,

#eAttribute.defaultValue.eContainingClass >1

eAttribute.ordered=True, eAttribute.ordered=False,

eAttribute.many=True, eAttribute.many=False,

eAttribute.lowerBound=0,1,>1,

eAttribute.upperBound=0,1,>1,eAttribute.required=True,

eAttribute.required=False

We ignore unsettable, derived for the Ecore 2 Alloy Transformation

Total Models = 52

EReference Partitions

1,>1 EReference objects eReference with
eReference .name="", eReference .name!=""
eReference.changeable=True, eReference.changeable=False
eReference.ordered=True, eReference.ordered=False,
eReference.many=True, eReference.many=False,
eReference.lowerBound=0,1,>1,
eReference.upperBound=0,1,>1,eReference.required=True,
eReference.required=False
eReference.containment=True, eReference.containment=False,
eReference.container=True, eReference.container=False
#eReference.containingClass=0,1,>1
#eReference.eOpposite=0,1

We ignore unsettable, derived, defaultValue, defaultValueLiteral, resolveProxies ,volatile, transient, unsettable, derived for the Ecore 2 Alloy Transformation

Total Models = 50

Conclusion

- Total number of test models to be synthesized for individual checks= 170 models
- The test models are mainly for ECore2Alloy
- Although they can be used to test other transformations

Thanks!