Eclipse Modeling Framework Introduction

Niko Stotz

2021-05-13

EMF

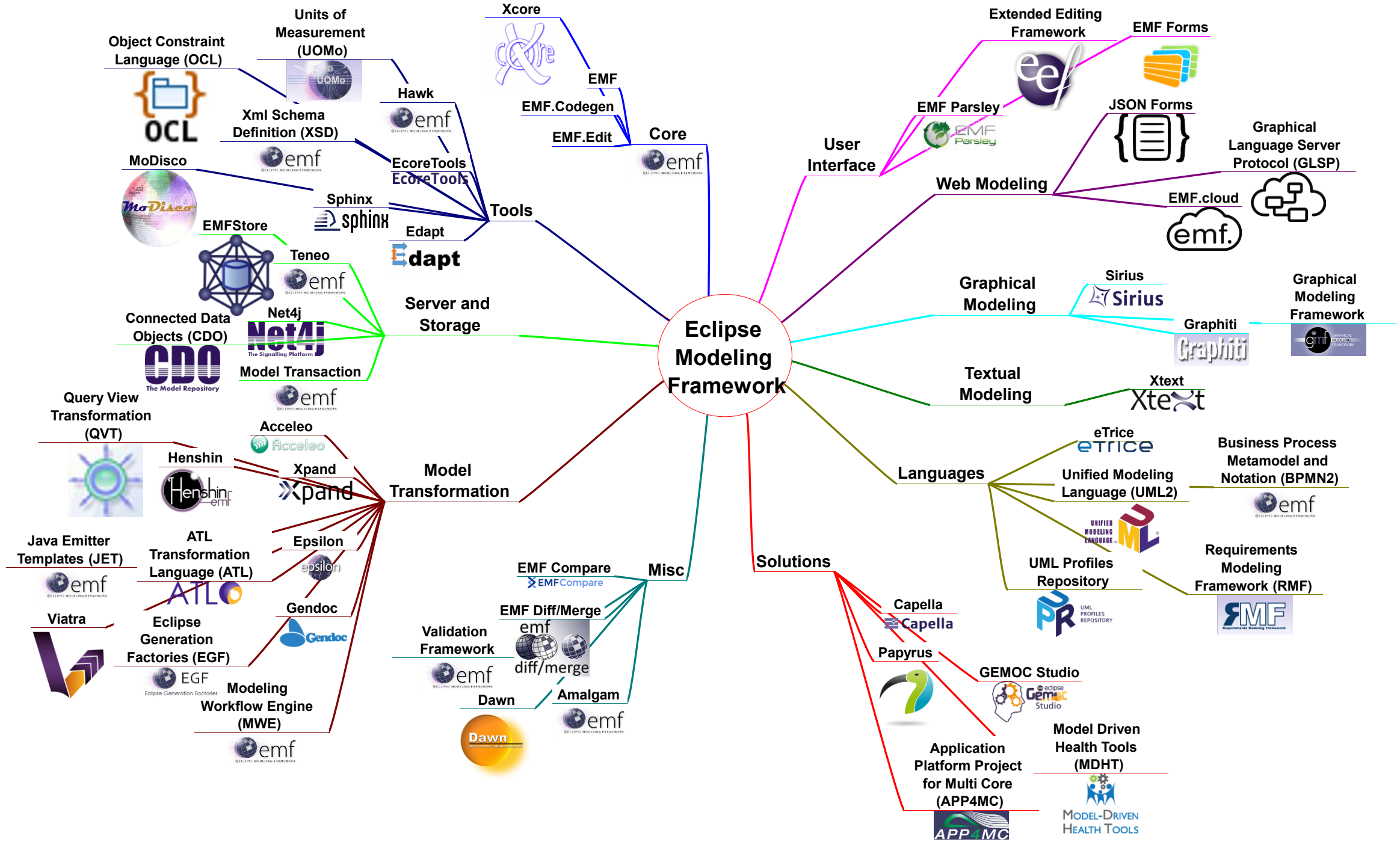
- Eclipse Modeling Framework
- aka Ed Merks Framework
- Created at IBM ~2002
- Hosted by Eclipse Foundation

What do we want to do with Models?

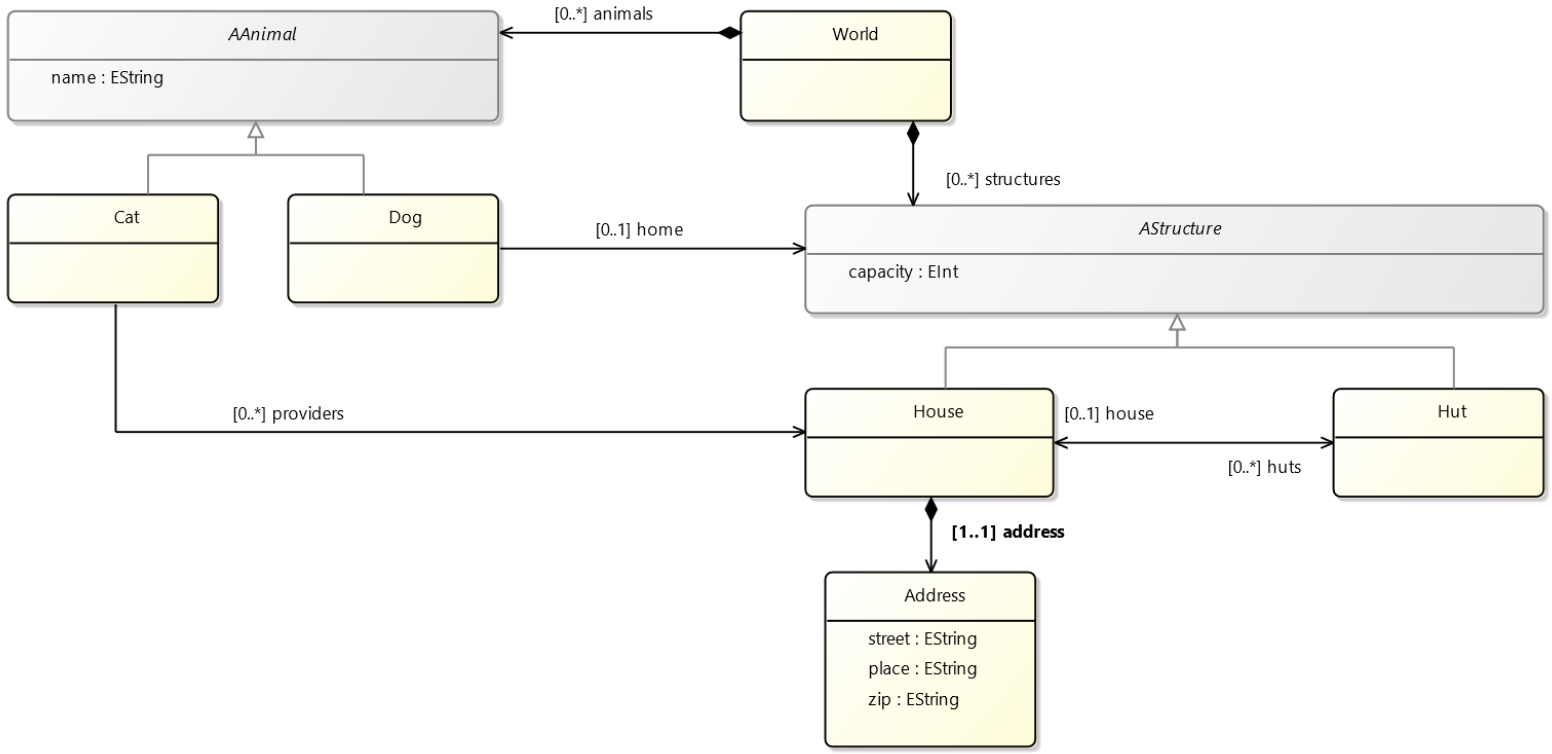
- Describe structure
- Edit
- View
- Store
- Exchange
- Compare
- Check

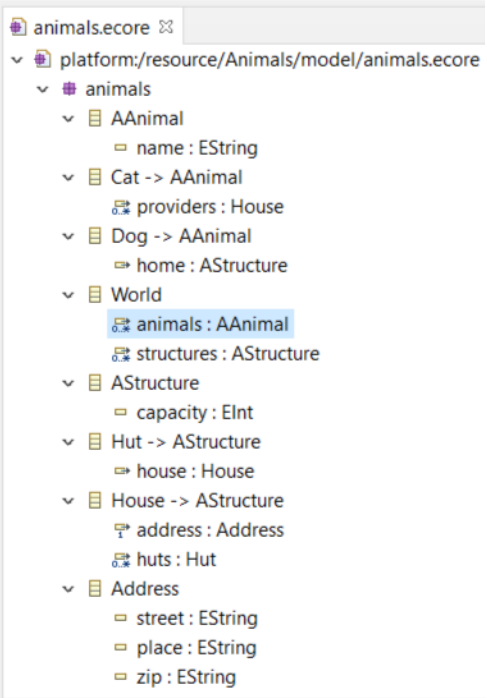
- Convert
- Generate
  - Text
  - Code
  - Graphics
  - Documents

  ➔ Artifacts


Eclipse Modeling Framework
Core
Xcore
EMF
EMF.Codegen
EMF.Edit
Tools
Units of Measurement (UOMo)
Object Constraint Language (OCL)
Hawk
Xml Schema Definition (XSD)
EcoreTools
MoDisco
Sphinx
Edapt
Server and Storage
EMFStore
Teneo
Net4j
Connected Data Objects (CDO)
Model Transaction
Model Transformation
Acceleo
Query View Transformation (QVT)
Henshin
Xpand
Java Emitter Templates (JET)
ATL Transformation Language (ATL)
Epsilon
Viatra
Gendoc
Eclipse Generation Factories (EGF)
Modeling Workflow Engine (MWE)
Misc
EMF Compare
EMF Diff/Merge
Validation Framework
Dawn
Amalgam
User Interface
Extended Editing Framework
EMF Forms
EMF Parsley
Web Modeling
JSON Forms
Graphical Language Server Protocol (GLSP)
EMF.cloud
Graphical Modeling
Sirius
Graphiti
Graphical Modeling Framework
Textual Modeling
Xtext
Languages
eTrice
Unified Modeling Language (UML2)
Business Process Metamodel and Notation (BPMN2)
UML Profiles Repository
Requirements Modeling Framework (RMF)
Solutions
Capella
Papyrus
GEMOC Studio
Application Platform Project for Multi Core (APP4MC)
Model Driven Health Tools (MDHT)

Example: Animals.ecore

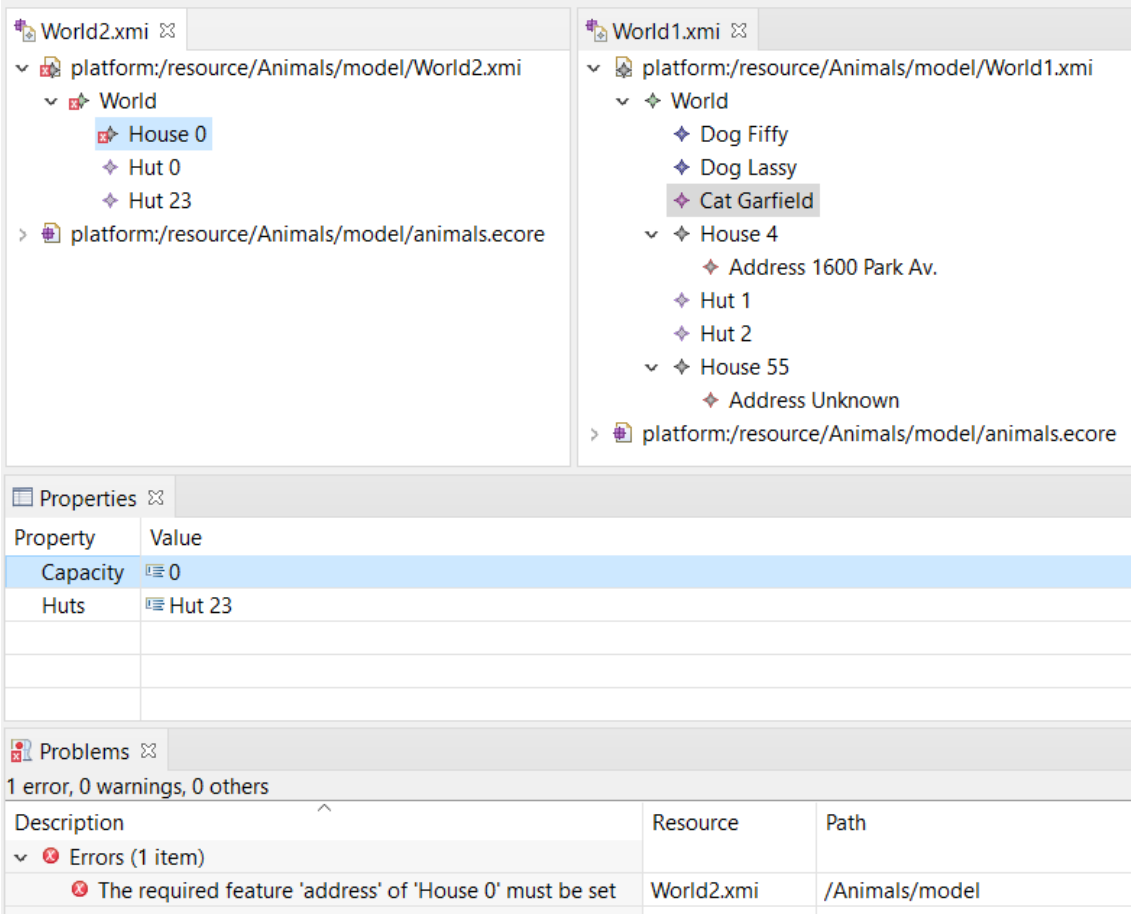




Properties

Property	Value
Changeable	true
Container	false
Containment	true
Default Value Literal	
Derived	false
EKeys	
EOpposite	
EType	AAnimal
Lower Bound	0
Name	animals
Ordered	true
Resolve Proxies	true
Transient	false
Unique	true
Unsettable	false
Upper Bound	-1
Volatile	false

Example: Two Animal Worlds



Why can all these parts work together?

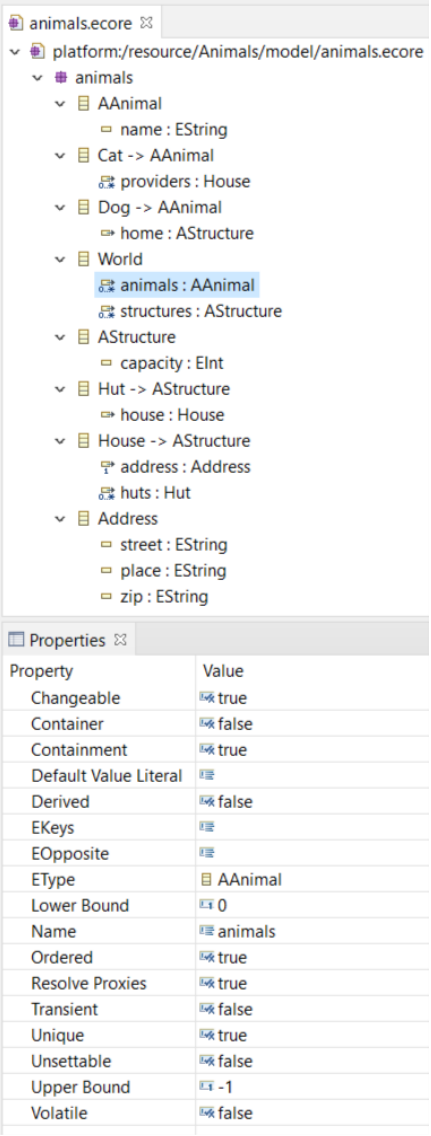
• **Shared structure description**	.ecore
• Generic concepts	Can describe everything
• Reflection	“Tell me about your own structure”
• Common runtime	Small + Fast
• Extensibility	Plug in + improve other parts

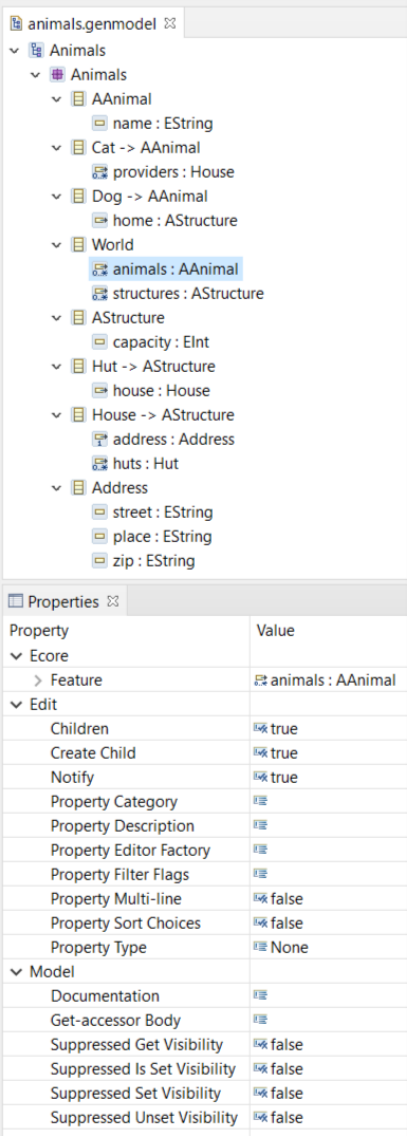
Ecore structural concepts

- EClass: Container *like object-orientation class*
- EAttribute: Primitive value String, int, …
- EReference: Link to other EClass A *contains* B; C *points to* D
- EEnum: List of values (EEnumLiteral) yes | no | maybe
- EOperation: Behavior *like object-orientation method*

Additional: EAnnotation, EPackage, EParameter, EGenericType, ETypeParameter

Model vs. GenModel





Generated Java code

- Animals
  - src-gen
    - animals
      - AAnimal.java
        - AAnimal
          - getName() : String
          - setName(String) : void
      - Address.java
        - Address
          - getPlace() : String
          - getStreet() : String
          - getZip() : String
          - setPlace(String) : void
          - setStreet(String) : void
          - setZip(String) : void
      - AnimalsFactory.java
      - AnimalsPackage.java
      - AStructure.java
        - AStructure
          - getCapacity() : int
          - setCapacity(int) : void
      - Cat.java
        - Cat
          - getProviders() : EList<House>
      - Dog.java
        - Dog
          - getHome() : AStructure
          - setHome(AStructure) : void
      - House.java
        - House
          - getAddress() : Address
          - getHuts() : EList<Hut>
          - setAddress(Address) : void
      - Hut.java
        - Hut
          - getHouse() : House
          - setHouse(House) : void
      - World.java
        - World
          - getAnimals() : EList<AAnimal>
          - getStructures() : EList<AStructure>

Don’t look at this

- animals.impl
  - AAnimalImpl.java
  - AddressImpl.java
  - AnimalsFactoryImpl.java
  - AnimalsPackageImpl.java
  - AStructureImpl.java
  - CatImpl.java
  - DogImpl.java
  - HouseImpl.java
  - HutImpl.java
  - WorldImpl.java
- animals.util
  - AnimalsAdapterFactory.java
  - AnimalsSwitch.java

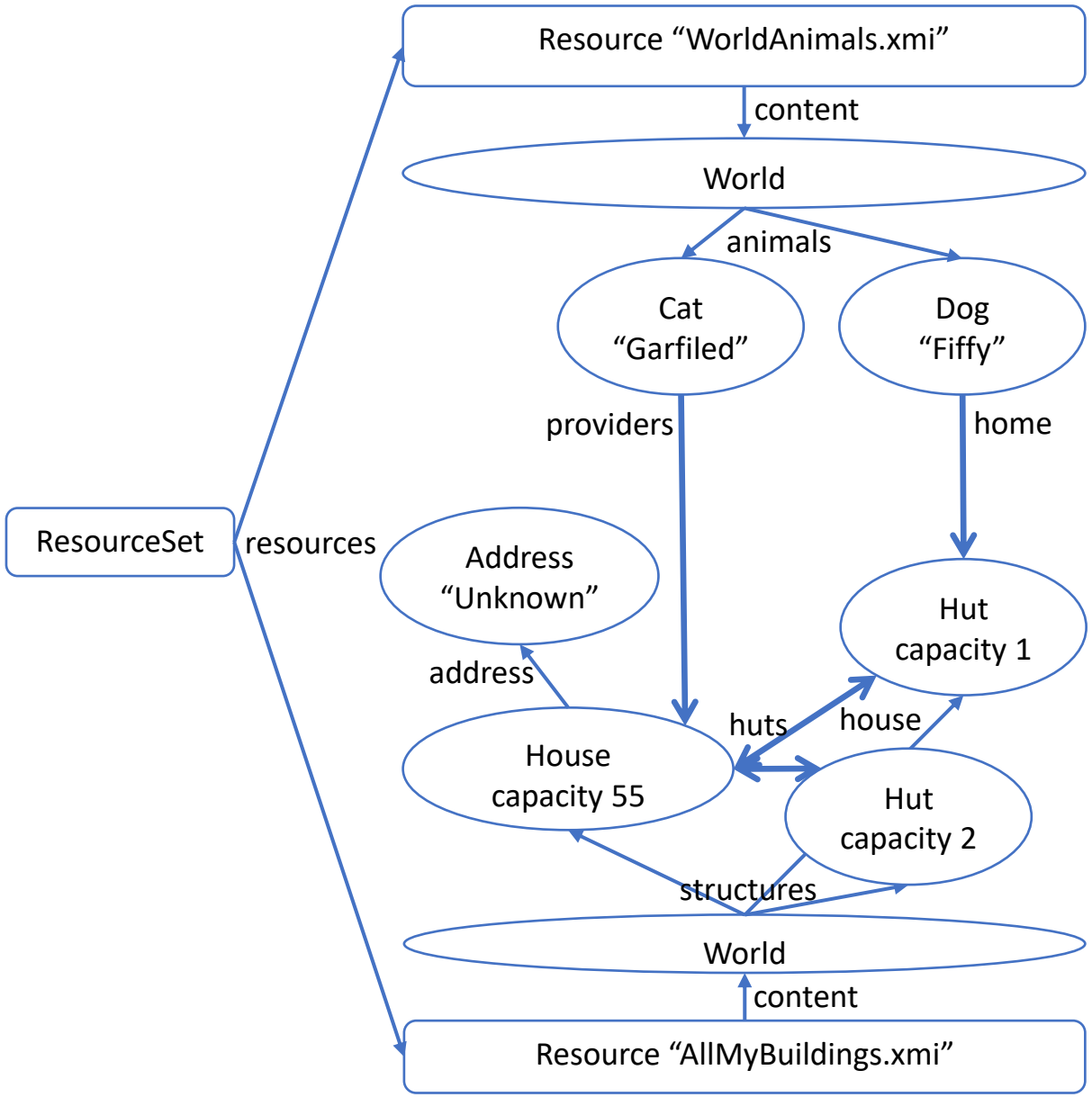
EMF runtime classes

- EObject: Supertype of all model elements — *like Java Object*
- EList: List with powers — e.g. bi-directional references
- Resource: “File” — world1.xmi
- ResourceSet: Connected “files” — *can cross-reference*

Spanning Tree + **Crosslinks** = **Graph**

EMF model instance

- ResourceSet
  - Resource "WorldAnimals.xmi"
    - World
      - Cat "Garfield"
        - providers
      - Dog "Fiffy"
        - home
  - Resource "AllMyBuildings.xmi"
    - World
      - Hut capacity 1
        - house
      - Hut capacity 2
        - house
      - House capacity 55
        - Address "Unknown"
        - huts



EMF API

- Utilities
- Notification
- Proxies
- Reflection

Not covered:

- Transaction
- Commands (UI integration incl. undo/redo)
- Validation
- UI editors
- Maps (Dictionaries)
- Derived properties

EMF API: Utilities

- *MyPackage*Factory: Create new EObject instances
- EcoreUtil: Copy, name, resolve, navigate, … EObjects
- Diagnostician: Generic entry point to validation

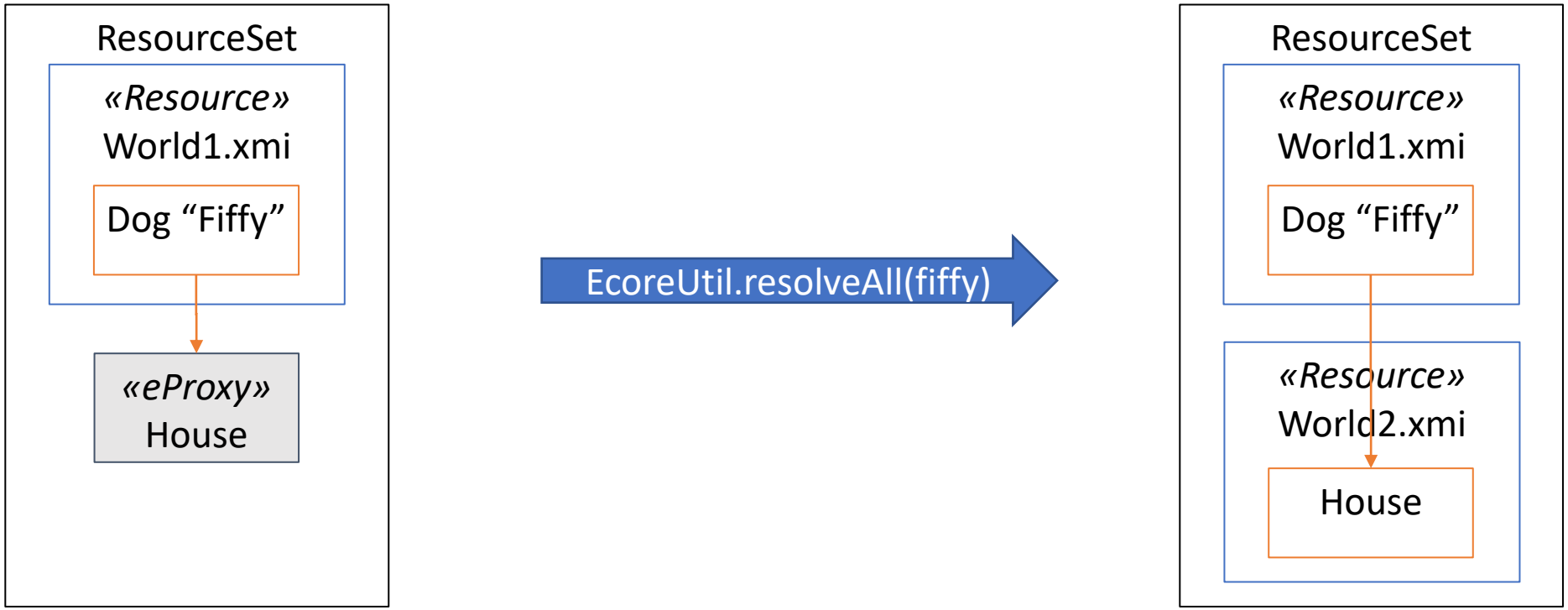
EMF API: Notification

Get informed about any change in your model

- Can react by:
  - Updating UI
  - Updating other parts of model
  - …
- Internally used for bi-directional references

EMF API: Proxies

Stand-ins for References to non-loaded target EObjects



EMF API: Reflection

Process and navigate EMF model without knowing its specific structure

- eContainer(): Parent in spanning tree
- eContents(): Children in spanning tree
- .eGet(): Get Attribute or Reference
- .eSet(): Set Attribute or Reference
- eResource(): Containing Resource
- eClass(): EClass of current EObject

EMF Links

- FAQ: http://wiki.eclipse.org/EMF/FAQ
- Documentation: http://www.eclipse.org/modeling/emf/docs/
- Overview: http://eclipsesource.com/blogs/tutorials/emf-tutorial/
- Forums: https://www.eclipse.org/forums/index.php/i/5/