Ecore Mutator

Written by Administrator
Sunday, 09 May 2010 18:29 - Last Updated Thursday, 17 February 2011 20:14

[EMF](#) -based framework to randomly mutate models conforming to a metamodel specified by an
[Ecore](#)
metamodel. This frameword can be used to test, benchmark, and evaluate model comparison related tools and implementations.

Implementation

The source code is published under the [EPL](#) and hosted at [Eclipse Labs](#) . You are kindly invited to use, test, and contribute to this framework. Besides checking out the source code, you can also   [download](#)
 [a compiled jar file](#)
as well as the
 [JavaDocs](#)
.

Usage

To randomly mutate a specified model for, let's say, 50 times with all provided mutations, use the code below. Please note that the mutations can be applied to any ecore-based model (a model conforming to a metamodel specified in Ecore). Currently, it is not checked whether the resulting model is valid (conforming to constraints of the metamodel). This might be enhanced using the EMF Validation framework in future releases.

Instead of the provided mutations you also can specify your own implementations of the Mutatio n
interface or by specializing the
AbstractMutation
class.

```
  // initialize mutator
EcoreMutator mutator = new EcoreMutator(); // configure mutations to apply
mutator.addMutation(new AddAnnotationMutation());
mutator.addMutation(new AddObjectMutation());
mutator.addMutation(new DeleteObjectMutation());
mutator.addMutation(new MoveObjectMutation());
mutator.addMutation(new UnsetFeatureMutation()); mutator.addMutation(new
UpdateFeatureMu                                                                    tati
```

Ecore Mutator

Written by Administrator
Sunday, 09 May 2010 18:29 - Last Updated Thursday, 17 February 2011 20:14

```
on(
));
// configure a change tracker
mutator.setTracker(); // load input model to mutate (any ecore-based model resource)
Resource inputResource = loadResource(inputUri); // initialize model provider
IModelProvider modelProvider = new ModelProvider();
modelProvider.setModelResource(inputResource); // mutate model
mutator.mutate(modelProvider, 50); // save output model
Resource outputResource = createResource(outputUri);
outputResource.getContents().addAll(inputResource.getContents());
outputResource.save(Collections.emptyMap());
```