# How to modify UML element programmatically

Modifications shall be executed in the context of command to be able to do UNDO/REDO [1]

Papyrus uses also the FrameworkElementType [2] in order to manage the life cycle. This framework allow to ensure the consistency of the UML model and the graphical Model.

Papyrus provides thanks to GMF runtime an API to modify UML element.

1. you need to get the editing domain

2. your need to create a request: there are gmf request (SetRequest, CreateElementRequest, DestroyRequest, ... all these request implements org.eclipse.gmf.runtime.emf.type.core.requests.IEditCommandRequest

3. you need to get the command provider to

IElementEditService provider = ElementEditServiceUtils.getCommandProvider(editedElement);

command = provider.getEditCommand(yourRequest)

//execution

domain.getCommandStack().execute(new GMFtoEMFCommandWrapper(command));

## Example to be able to create, destroy, and rename a class:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 \* Copyright (c) 2020 CEA LIST and others.

 \*

 \* All rights reserved. This program and the accompanying materials

 \* are made available under the terms of the Eclipse Public License 2.0

 \* which accompanies this distribution, and is available at

 \* http://www.eclipse.org/legal/epl-2.0/

 \*

 \* SPDX-License-Identifier: EPL-2.0

 \*

 \* Contributors:

 \* CEA LIST - Initial API and implementation

 \*

 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

package org.eclipse.papyrus.uml.diagram.clazz.custom.command;

import org.eclipse.emf.ecore.EObject;

import org.eclipse.emf.transaction.TransactionalEditingDomain;

import org.eclipse.emf.transaction.util.TransactionUtil;

import org.eclipse.gmf.runtime.common.core.command.ICommand;

import org.eclipse.gmf.runtime.emf.type.core.requests.CreateElementRequest;

import org.eclipse.gmf.runtime.emf.type.core.requests.DestroyElementRequest;

import org.eclipse.gmf.runtime.emf.type.core.requests.SetRequest;

import org.eclipse.papyrus.infra.emf.gmf.command.GMFtoEMFCommandWrapper;

import org.eclipse.papyrus.infra.services.edit.service.ElementEditServiceUtils;

import org.eclipse.papyrus.infra.services.edit.service.IElementEditService;

import org.eclipse.papyrus.uml.service.types.element.UMLElementTypes;

import org.eclipse.uml2.uml.NamedElement;

import org.eclipse.uml2.uml.Package;

import org.eclipse.uml2.uml.UMLPackage;

/\*\*

 \* editing elements example

 \*/

public class EditElementExample {

 /\*\*

 \* Create a Class in a UML Package

 \*

 \* @param pack

 \* the package owner

 \*/

 public static void createAClass(final Package pack) {

 TransactionalEditingDomain domain = TransactionUtil.getEditingDomain(pack);

 // create a Class inside a package (here the containment feature is automatically deduced by the framework, but you can specify it

 CreateElementRequest request = new CreateElementRequest(domain, pack, UMLElementTypes.CLASS);

 IElementEditService provider = ElementEditServiceUtils.getCommandProvider(pack);

 ICommand command = provider.getEditCommand(request);

 domain.getCommandStack().execute(new GMFtoEMFCommandWrapper(command));

 }

 /\*\*

 \* Destroy an element

 \*

 \* @param element

 \* the element to destroy

 \*/

 public static void destroyAnElement(final EObject element) {

 TransactionalEditingDomain domain = TransactionUtil.getEditingDomain(element);

 DestroyElementRequest request = new DestroyElementRequest(element, false);

 IElementEditService provider = ElementEditServiceUtils.getCommandProvider(element);

 ICommand command = provider.getEditCommand(request);

 domain.getCommandStack().execute(new GMFtoEMFCommandWrapper(command));

 }

 /\*\*

 \* How to rename a NamedElement

 \*

 \* @param namedElement

 \* the element to rename

 \* @param newName

 \* the new name of the element

 \*/

 public static void renameANamedElement(final NamedElement namedElement, final String newName) {

 TransactionalEditingDomain domain = TransactionUtil.getEditingDomain(namedElement);

 SetRequest request = new SetRequest(namedElement, UMLPackage.eINSTANCE.getNamedElement\_Name(), newName);

 IElementEditService provider = ElementEditServiceUtils.getCommandProvider(namedElement);

 ICommand command = provider.getEditCommand(request);

 domain.getCommandStack().execute(new GMFtoEMFCommandWrapper(command));

 }

}

[1] <https://download.eclipse.org/modeling/emf/emf/javadoc/2.7.0/org/eclipse/emf/common/command/package-summary.html>

[2] <https://www.ibm.com/support/knowledgecenter/en/SSCLKU_7.5.5/com.ibm.xtools.modeler.doc.isv/prog-guide/creating-types.html>