Using ATL transformation services in the MDEForge collaborative modeling platform ⋆

(Accompanying paper)

Juri Di Rocco[1], Davide Di Ruscio[1], Alfonso Pierantonio[13], Jesús Sánchez Cuadrado[2], Juan de Lara[2], Esther Guerra[2]

[1] University of L'Aquila (Italy) - name.surname@univaq.it
[2] Universidad Autónoma de Madrid (Spain) - name.surname@uam.es
[3] Mälardalen University, Västerås (Sweden) - name.surname@mdh.se

Abstract. This is an accompanying paper of the tool demo paper *"Using ATL transformation services in the MDEForge collaborative modeling platform"* presented at ICMT 2016, Vienna (Austria). The demo was about the integration of ATL services in the MDEForge modeling platform enabling the remote execution, automated testing, and static analysis of ATL transformations.

1 Demonstration walk through

The actual presentation the ICMT 2016 conference of the ATL services added to MDEForge was performed according to the following schedule:

1. Overview of MDEForge (management of workspaces, projects, artifacts, and users);
2. Overview of the extensions that have been required to add ATL services in MDEForge
3. Use of the Web client and REST APIs to upload new ATL transformations;
4. Use of the Web client and REST APIs to execute ATL transformations on simple source models;
5. Use of the Web client and REST APIs to apply the ATL analysis services.

As explanatory example, the *Families2Persons* ATL transformation[4] was used during the demonstration.

1.1 Quick overview of MDEForge

A quick presentation of MDEForge was done and an overview of its main functionalities was also given (Fig. 1).

The concepts of workspaces, projects and how modeling artifacts are organized was briefly presented by showing a logged user area like the one in Fig. 2.

⋆ Work supported by the Spanish MINECO (TIN2014-52129-R), the Madrid Region (S2013/ICE-3006), and the EU commission (#611125).

[4] http://www.eclipse.org/atl/atlTransformations/#Families2Persons

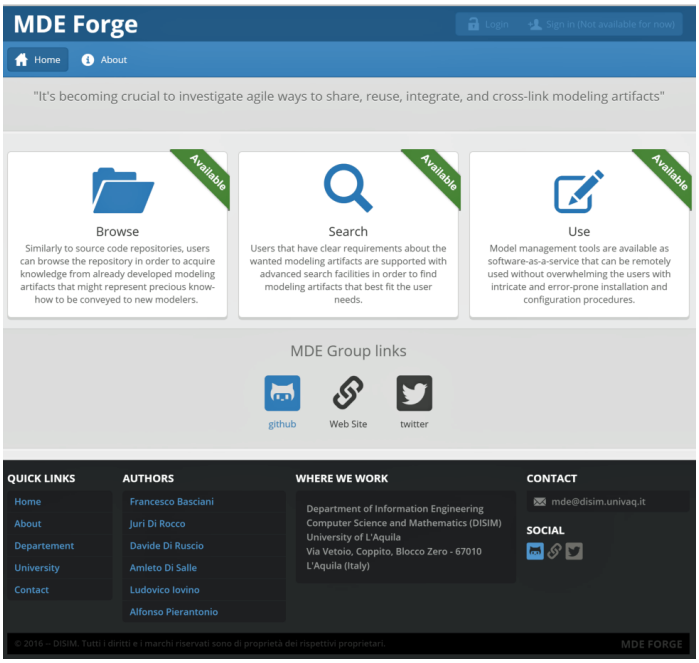

Fig. 1. The MDEForge home page

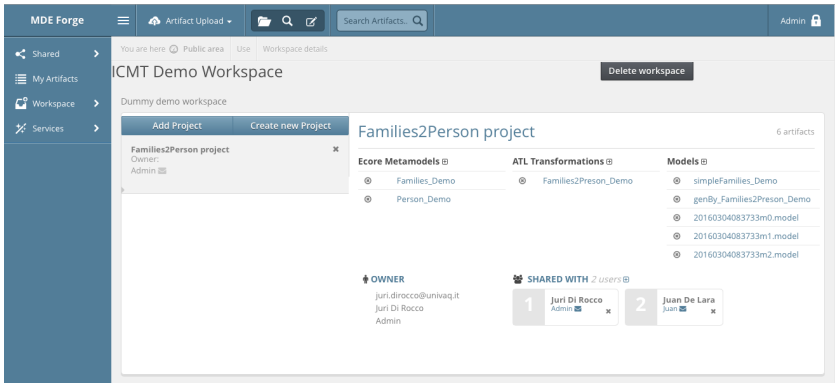

Fig. 2. The MDEForge logged user area

1.2 Overview of the extensions that have been required to add ATL services in MDEForge

By referring to the diagrams in Fig. 3 and Fig. 4, and to Table 1 an overview of the MDEForge extensions that have been required to add ATL services was presented.

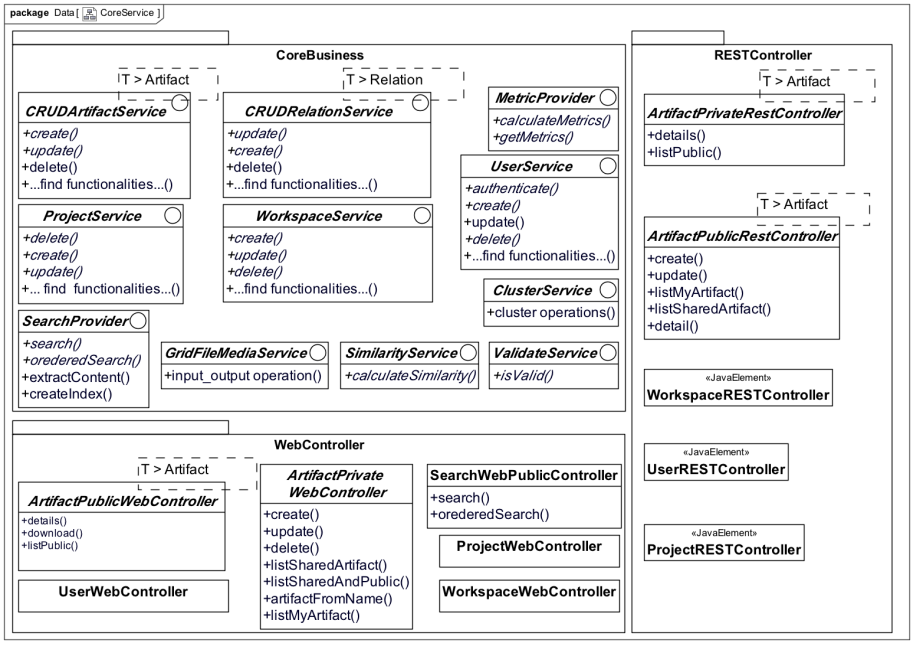

Fig. 3. MDEForge core services

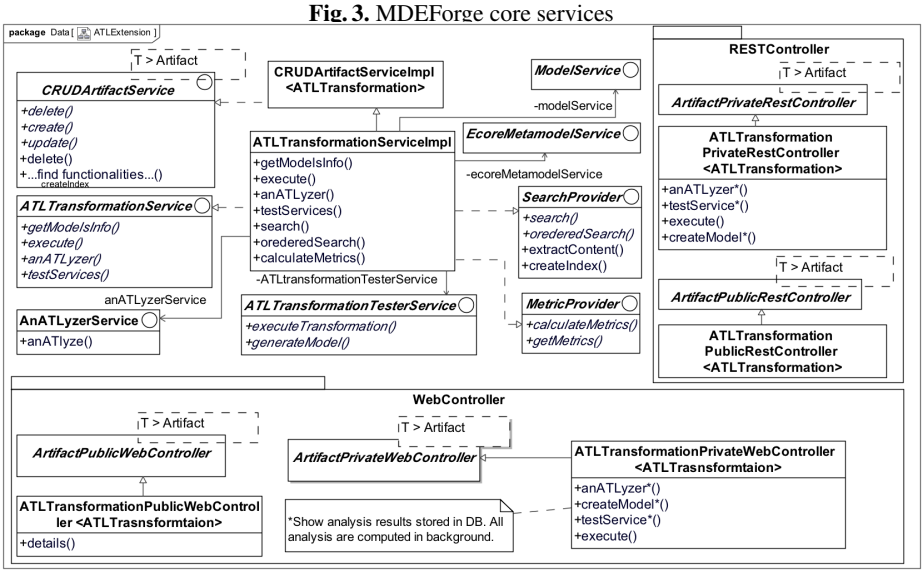

Fig. 4. MDEForge extensions for ATL

1.3 Upload of a new ATL transformation

The explanatory *FamiliesToPersons* ATL transformation was uploaded to the repository by means of the Web page in Fig. 5. The sharing functionalities available during the transformation upload was also discussed.

Requirements	Extended core classes	Implemented core interfaces	New artifacts
RQ1	`CRUDArtifactServiceImpl`	none	none
RQ2	`CRUDArtifactServiceImpl`	none	none
RQ3	`CRUDArtifactServiceImpl` `CRUDRelationServiceImpl`	none	none
RQ4	none	*`SearchProvider`*	none
RQ5a	none	none	none
RQ5b	none	none	`ATLTransformationService`
RQ5c	`CRUDArtifactServiceImpl`	none	none
RQ6a	none	*`MetricProvider`*	none
RQ6b	none	none	*`ATLTransformationService`* *`AnATLyzerService`*
RQ6c	none	none	*`ATLTransformationService`* *`ATLTransformationTesterService`*

RQ7	`ArtifactPublicWebController`
	`ArtifactPrivateWebController`
	`ArtifactPublicRestController`
	`ArtifactPrivateRestController`
		none	`ATLTransformationPublicWebController`
			`ATLTransformationPrivateWebController`
			`ATLTransformationPublicRestController`
			`ATLTransformationPrivateRestController`

Table 1. MDEForge extensions for ATL

When uploading transformations, it is necessary to specify the source and target metamodels. It is possible to parse the transformation being uploaded to automatically search for candidate metamodels (see the *Parse ATL* button in Fig 5). If they are not available in the repository, users have to upload them before finalizing the addition of the new transformation. Once the upload of the new transformation has been completed, the transformation details page in Fig. 6 is shown. Such a page was considered as starting point for executing the transformation and to show the application of the analysis services.

1.4 Execution of an ATL transformation

The execution of the *Families2Persons* transformation was shown by considering both models already available in the system and models that was contextually uploaded. Specific attention was devoted on how the system is able to automatically retrieve source models owned by the logged user and that conform to the source metamodels of the transformation being executed. The result of the transformation execution was also shown (see Fig. 8).

1.5 Application of the ATL analysis services

The ATL analysis services was applied on the explanatory transformation and the corresponding results was shown and discussed (see the *anATLyzer Transformation errors* and *Test service report* sections in Fig. 6). For this particular transformation, the static analyser[5] finds an error due to a possible navigation through an undefined feature (`self.familyDaughter.lastName`, where `self.familyDaughter` may be undefined). The testing service also detects this error, where three automatically generated input models hit the problematic line. Such models can be downloaded and analysed by the developer to understand the source of the problem.

In the demo, we discussed the strengths and limitations of the static analysis and random testing techniques.

[5] The analysis service uses anATlyzer, which is available as Eclipse plugin at `http://miso.es/tools/anATLyzer.html`

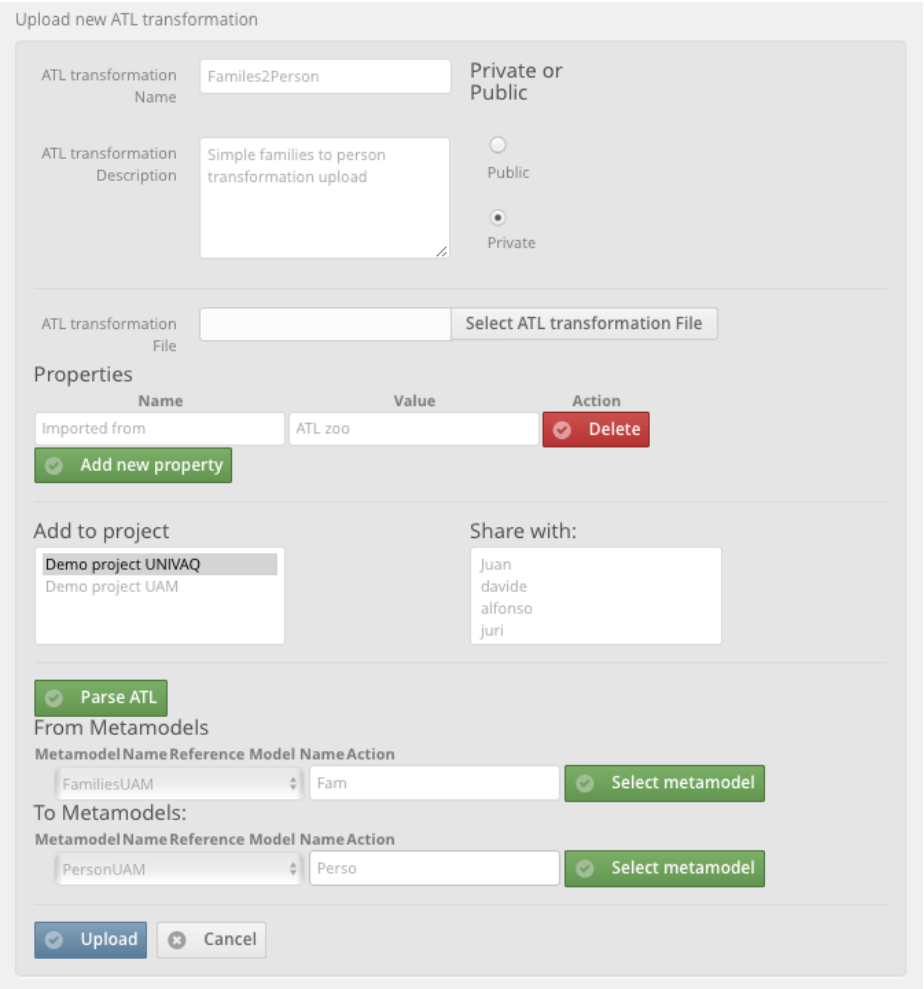

Fig. 5. ATL transformation upload page

2 Additional information

- MDEForge website and source code: `http://www.mdeforge.org`
- Tool demonstration video: the link to a tool demonstration video is available at the end of `https://github.com/MDEGroup/MDEForge`

Families2Preson_Demo *Private Transformation*

Info
Artifact

Used in 1 projects

Importer
Admin
Juri Di Rocco
juri.dirocco@univaq.it

General

Creation Data	04/03/16 8.35
Last Modified	04/03/16 8.35

Description

Transformation File
Visualize Transformation
Download Transformation

SHARED USERS *team (1 people)*
1 **Admin**
Juri Di Rocco
juri.dirocco@univaq.it

anATLyzer Transformation errors

Error 1: Possible access to undefined feature

Local problem: true
Element: anatlyzer.atlext.OCL.impl.NavigationOrAttributeCallExpImpl@638966ea (location: 17:5-17:33, commentsBefore: null, commentsAfter: null, fileLocation: new-model, fileObject: null) (implicitlyCasted: false) (isStaticCall: false) (name: lastName)
File location: new-model
Location: 17:5-17:33
Status: ERROR_CONFIRMED
ProblemId: 21
Description: Possible access to undefined feature
Severity: runtime-error

Test service report

Test 1: Unable to access lastName on OclUndefined

Test 2: Unable to access lastName on OclUndefined

executionRaisesException: true
executionYieldsIllTarget: false
errorKind: EXECUTION_RAISES_EXCEPTION
errorMessage: Unable to access lastName on OclUndefined
model: 20160304083733m1.model

Test 3: Unable to access lastName on OclUndefined

Execute the Transformation

Input Metamodels		Output Metamodels
Families_Demo	Families2Preson_Demo Execute Transformation	Person_Demo

Metrics

Name	Description	Value				
		Max	Min	Avg	Median	Standard Deviation
Number of Units		1				
Number of bindings		2				

Fig. 6. ATL transformation details page

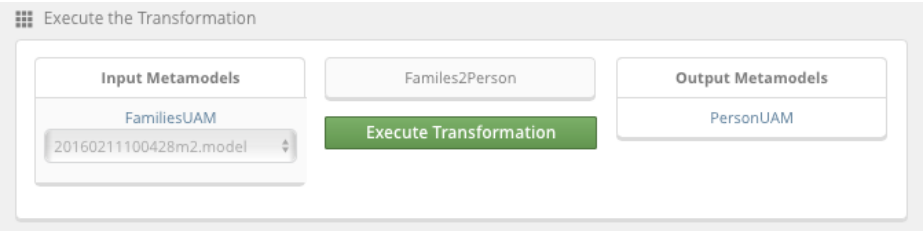

Fig. 7. ATL transformation execution page

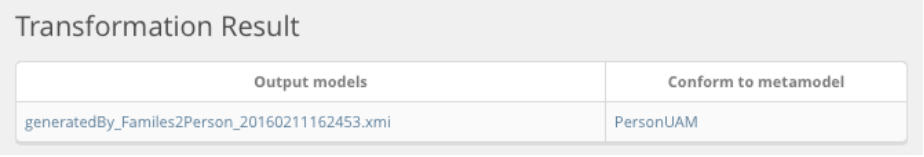

Fig. 8. ATL transformation execution result page