

In the beginning, please complete the questions of the “Public” section in the questionnaire.

## 1. Introducing the SEET tool

SEET (Symbolic Execution of ETL Transformations) is a tool to help the detection of logical errors in ETL transformations. A logical error is a mistake in a program that results in an incorrect output or unexpected behavior. For example, assigning the wrong value to the property of output class or returning ‘false’ instead of ‘true’ in the return statement may produce unwanted results.

The tool aims to help the user find out whether the ETL transformation is written correctly. The tool does not localize the fault. It uses the symbolic execution technique and generates a Symbolic Metamodel Footprint (SMF). The user by investigating the SMF can ensure the correctness of the transformation. The tool works on both automatic and interactive model. In the automatic mode all the execution paths are explored. In the interactive mode, one execution path is selected by the user consciously.

## 2. Before using the tool

We have selected the Families2Persons case study. For more information about this case, please refer to the **Appendix**.

1. Import the Families2Persons project into eclipse (It has been sent along with the manual).
2. In the Families2Persons project, open the transformation folder. Three Families2Persons transformations have been provided with different logics namely Trans1, Trans2, and Trans3.
3. Without using the testing tool, please check which of these transformations are faulty and complete the “Before using the tool” section of the questionnaire.

## 3. Using the tool

### 3.1. Installation

Requirements:

- Windows OS
- Eclipse
- Epsilon 1.3 (<http://download.eclipse.org/epsilon/1.3/updates/>)
- ATL (version 3 or later) (<http://download.eclipse.org/mmt/atl/updates/releases/>)
- Jre 8
- (optional) graphviz (To show test cases it is required which is not the aim of this evaluation)

SEET Update site: <https://raw.githubusercontent.com/BanafshehAzizi/SEET/master/site/site.xml>

### 3.2. Usage

1. Right click on an etl file and select “SEET” from a popup menu (Figure 1).
2. In the configuration wizard, select the source and target metamodels and click on the “Finish” button (Figure 2).
3. As can be seen in Figure 3, you are asked about selecting the mode. If you select the automatic mode, SEET traverses executable paths of the transformation automatically. Otherwise, in the case that you select the interactive mode, satisfiability of conditions is determined by your control. This mode is applied when the user wants to explore one execution path consciously. Please select the “Automatic” mode.

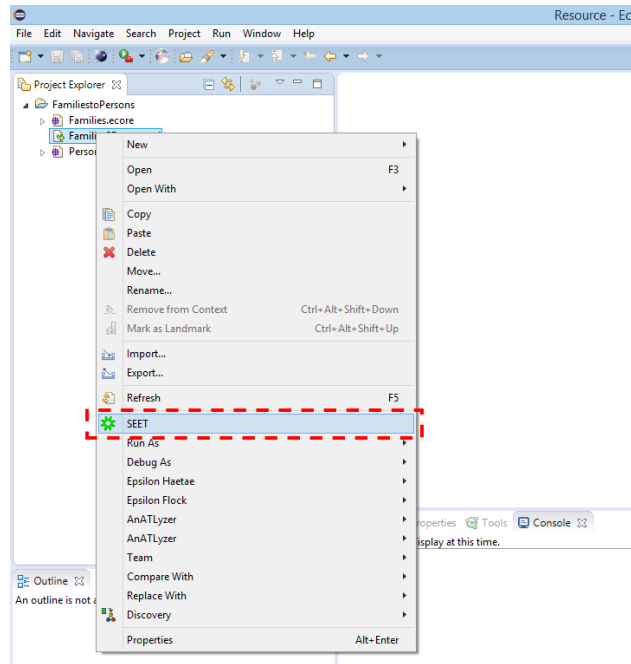


Figure 1- The execution of the SEET tool

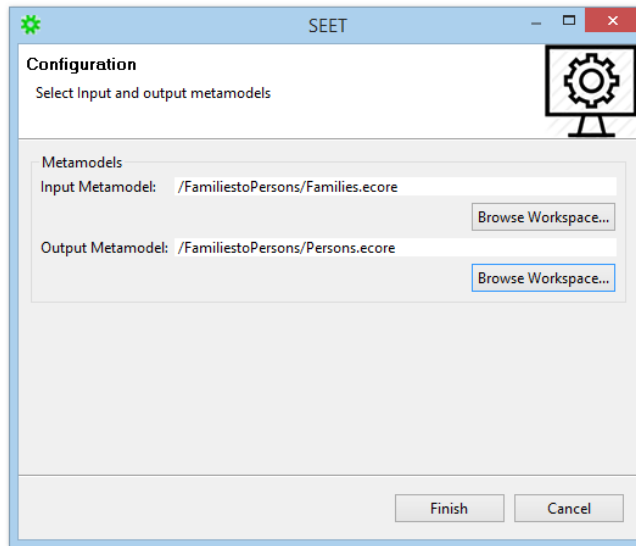


Figure 2- The configuration

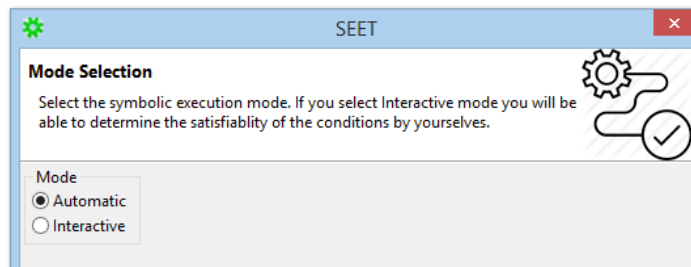


Figure 3- Selecting the execution mode

4. The result is shown (Figure 4). It involves satisfiable path conditions and their corresponding test models. Click on the “Finish” button.

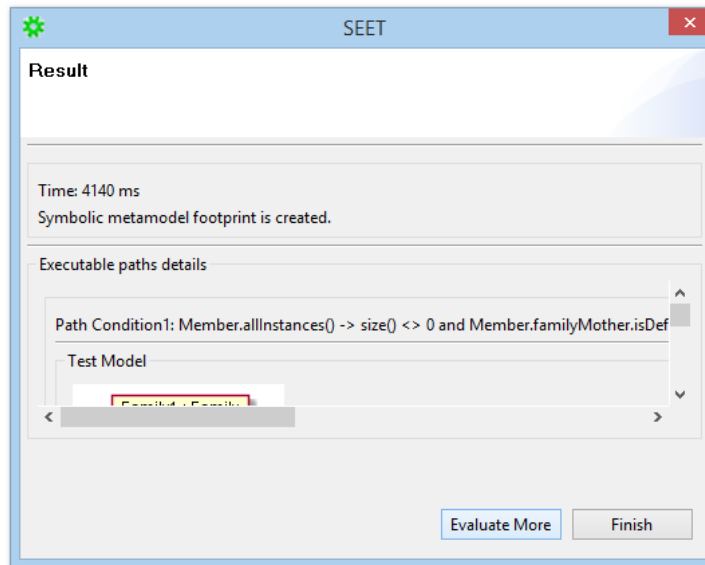


Figure 4- The result containing the satisfiable path conditions and test models

5. Right click on the transformation folder and click “Refresh.”
6. Open SMF.model and PC.txt files. PC.txt contains satisfiable path conditions and SMF.model contains their corresponding symbolic output. By checking these two files in the following manner, you can determine whether or not the transformation is written correctly.

Assuming in the PC.txt file, we have the below path condition.

Path Condition1: Member.allInstances() -> size() <> 0 and Member.familyMother.isDefined() and not Member.familyFather.isDefined()

The “Member.allInstances() -> size() <> 0” condition means the family has a member. The “Member.familyMother.isDefined()” condition means the member is the mother of the family. The “not Member.familyFather.isDefined()” condition means the member is not father.

In the SMF.model file, find the package with the corresponding number. Since this is “Path Condition1”, open “Target1” package (Figure 5).

According to the path condition and the transformation aim, we expect to have the following in the Target1 package.

1. Since the family has a member, Target1 package must not be empty.
2. Since the member is a mother, we expect to have Female.
3. The Female must have a “fullName” attribute.
4. The “fullName” attribute must have a “Member.firstName Member.familyMother.lastName” value.

If at least one of the above conditions is not satisfied, stop checking other path conditions and packages and specify the transformation as faulty in the questionnaire.

If all of the four conditions are satisfied, check the following path conditions and their corresponding packages. If all of them are correct, the transformation is written correctly.

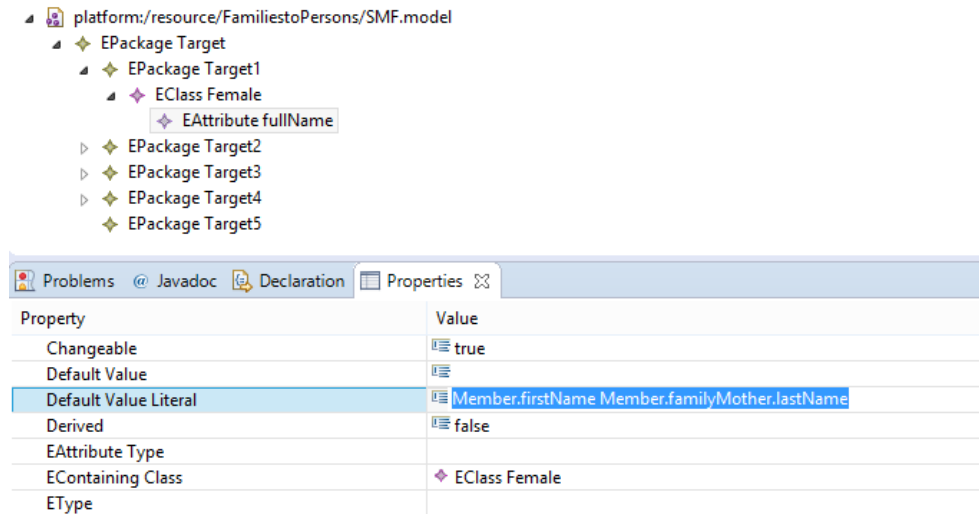


Figure 5- The generated SMF

Do this checking for three transformations provided. Then, fill out the related question in the questionnaire.

### Interactive mode

Run SEET for one of the transformations and select the “Interactive” mode. Answer the questions. In this case by your selection, one execution path is selected and symbolically run (Figure 6).

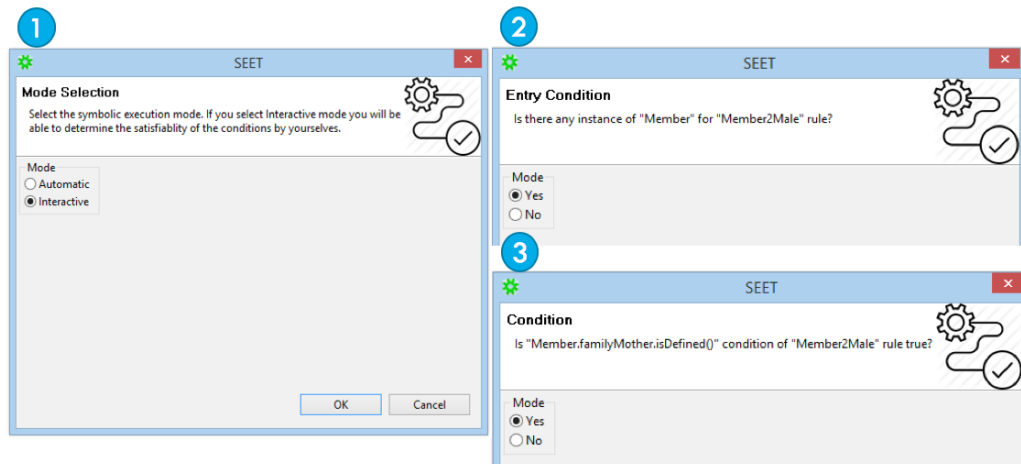


Figure 6- Interaction with the tester: 1) Selecting execution mode, 2) Asking satisfiability of the entry condition, 3) Asking the satisfiability of the condition

Fill out the remaining questions in the questionnaire.

## Appendix

The Families2Persons transformation determines the gender of family members. Figure 7 shows both the source and target metamodels of this transformation. The source metamodel (Families) has a class named Family. This class has a lastName attribute and can have Members. Any Member has the firstName attribute and can be mother, father, daughter or son of the Family. The target metamodel (Persons) has a root class named Person, which has a fullName attribute and can be a Male or a Female.

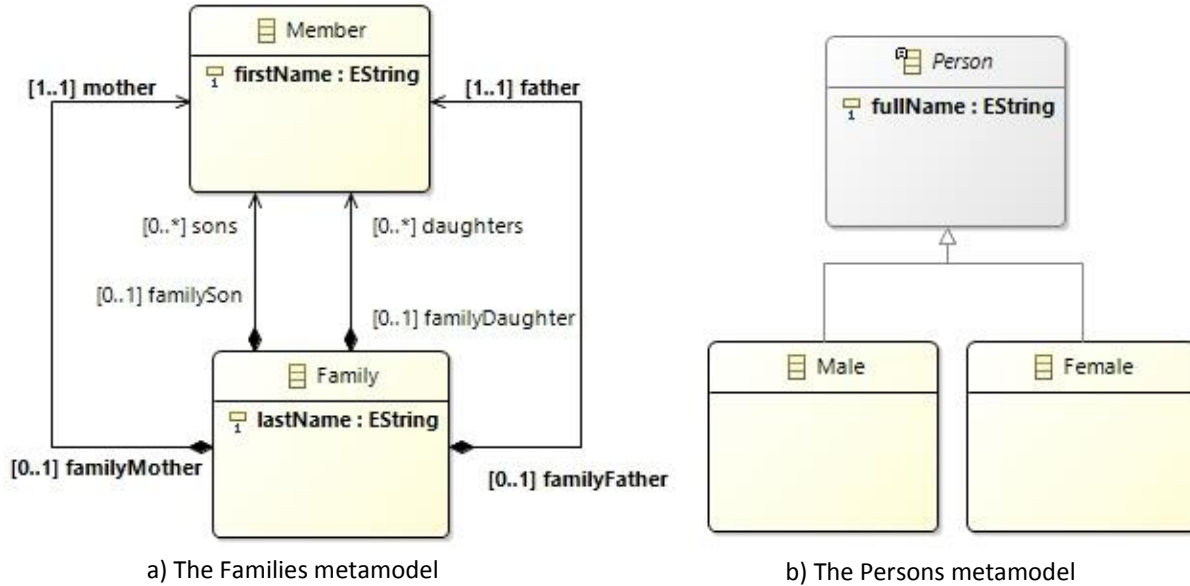


Figure 7- Families and Persons metamodels