

MDE Assignment 5

Group members: Sadrul Mizan Nashid, MD Ariful Islam, Elmar Karimov

Part 5.1

For this part, we have added new attributes, changed the name of the concepts and also added new concepts. We also deleted some concepts to make it simple. Below is a screenshot of a new metamodel.

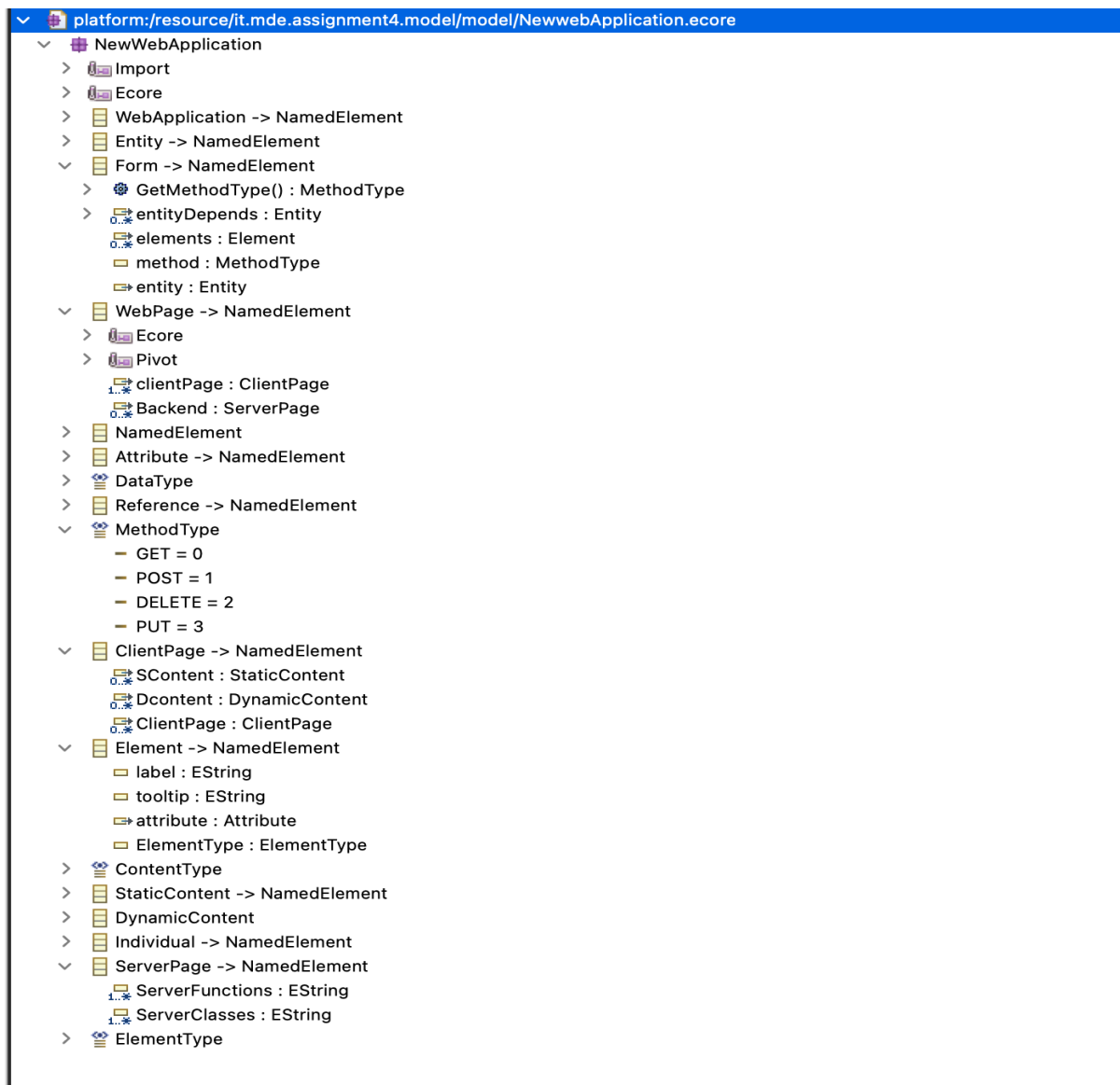


Fig 1: New webapplication metamodel in Ecore.

Part 5.2

For this part, we have used ATL tools for model to model transformation. In the ATL file we have created some helper context and rules for the transformation. Below is the screenshot of ATL file.

```
-- @path webApplication=/it.mde.assignment4.model/model/webApplication.ecore
-- @path NewwebApplication=/it.mde.assignment4.model/model/NewwebApplication.ecore

module NewsWebToMagazine;
create OUT : NewwebApplication from IN : webApplication;

⊖ helper context webApplication!Attribute def : attributeType() : String =
    if self.type = 'text' then
        true
    else
        false
    endif;

⊖ helper context NewwebApplication!ServerPage def : ServerClassType() : String =
    self.ServerClasses = 'Text';

⊖ rule WebApp {
    from
        s: webApplication!WebApplication
    to
        t : NewwebApplication!WebApplication (
            name <- s.name
        )
}

⊖ rule WebEntity {
    from s: webApplication!Entity
    to t: NewwebApplication!Entity (
        name <- s.name
    )
}

⊖ rule WebAttribute {
    from s: webApplication!Attribute
    to t: NewwebApplication!Attribute (
        name <- s.name,
        dataType <- s.type,
        isPK <- s.isPrimaryKey
    )
}

⊖ rule WebPageRule {
    from s: webApplication!Page
    to t: NewwebApplication!WebPage(
        name <- s.name
    )
}
```

Fig 2: ATL file for M2M transformation.

Part 5.3:

For model to text transformation, we used Acceleo to make the SQL queries. We have created two files and generated the SQL for the model we created. Below are the screenshot of those files.

```
[comment encoding = UTF-8 /]
[module generate('http://www.example.org/webApplication')]
[import it::mde::assignment5::acceleo::main::generateSQL /]

@ [template public generateElement(aWebApplication : WebApplication)]
[comment @main/]
[file (aWebApplication.name, false, 'UTF-8')]
    The application name is [aWebApplication.name/]
    This contains the following entities:
    [for (entity : Entity | aWebApplication.entities)]
        [entity.generateSQL()/]
    [/for]

[/file]
[/template]
```

Fig 3: acceleo file for WebApplication.

```
[comment encoding = UTF-8 /]
[module generateSQL('http://www.example.org/webApplication')]

@ [template public generateSQL(anEntity : Entity)]
[comment @main/]
[file (anEntity.name, false, 'UTF-8')]
    CREATE TABLE [anEntity.name/] (
        [for (it : Attribute | anEntity.attributes) separator(',')] [it.name/] ([it.type/]) [/for]
        [for (it : Attribute | anEntity.attributes ->select(e | e.isPrimaryKey = true))]
        PRIMARY KEY ([it.name/])
    [/for]
    );

    SELECT [for (it : Attribute | anEntity.attributes) separator(',')] [it.name/][[/for] FROM [anEntity.name/];

    SELECT [for (it : Attribute | anEntity.attributes ->select(e | e.isPrimaryKey = true)) separator(',')] [it.name/][[/for] FROM [anEntity.name/];

    SELECT [for (it : Attribute | anEntity.attributes) separator(',')] [it.name/][[/for] FROM [anEntity.name/] WHERE [for (it : Attribute | anEntity.attributes) separator(',')] [it.name/] = ? [/for]

[/file]
[/template]
```

Fig 4: acceleo file for entity.

```
| CREATE TABLE News (
    news_id (int) , title (text) , content (text) , publish_date (date) , pictures (images)
    PRIMARY KEY (news_id)
);

SELECT news_id, title, content, publish_date, pictures FROM News;

SELECT news_id FROM News;

SELECT news_id, title, content, publish_date, pictures FROM News WHERE news_id = ? ;
```

Fig 5: generated SQL queries.