



UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

Model Driven Engineering 2024-2025

Assignment 5

Transformations - Airport Management System

Name	Matriculation number	Email
Prachi	301124	prachi.prachi@student.univaq.it
Eric Rode	301145	eric.rode@student.univaq.it
David Aron Vigh	301150	davidaron.vigh@student.univaq.it

Introduction

This report documents the execution of a series of tasks focusing on model transformations using the ATL (Atlas Transformation Language) and Aceleo frameworks. The aim is to explore refactoring, model-to-model (M2M) transformations, and model-to-text (M2T) transformations to support model evolution and code generation in model-driven engineering.

Task 1: Refactoring the Metamodel

The first task involved refining the metamodel from a previous assignment (A4) to create an updated version. Refactoring included deletion, addition, renaming, and structural adjustments to make the implemented domain more realistic. The following refinements were implemented:

1. Deletion of concepts: The Customs class and its reference from Arrival were removed, along with the TransportationServices class and TransportationTypes enumeration. These removals simplified the metamodel by eliminating redundant concepts.
2. Addition of concepts: A new class, VIPArea, was introduced with attributes loungeType (of type EString) and capacity (of type EInt). Additionally, a new City class was created with attributes name (of type EString) and country (of type EString), increasing the model's ability to represent geographic and operational details.
3. Concept renaming: The ParkingArea class was renamed to CarParkingArea to provide a more descriptive and precise naming convention.
4. Structural refinements: A reference from Terminal to VIPArea was added with a multiplicity of 0..*, allowing terminals to host multiple VIP areas. The Flight.city attribute was replaced with a reference to the newly created City class, improving the abstraction for representing city-related information.

The changes were implemented in this order:

- Removed class Customs
- Reference for Customs from Arrival is removed
- Deleted TransportationServices
- Deleted TransportationTypes enumeration
- Reference for TransportationServices from Terminal is removed
- New class VIPArea created, with attributes: loungeType: EString, capacity: EInt
- Renamed ParkingArea to CarParkingArea
- Reference created from Terminal to VIPArea, with multiplicity: 0..*
- New City class created, with attribute name (EString) and country (EString)
- Replaced Flight.city with a reference to City class

Task 2: Model-to-Model Transformation with ATL

The second task required the development of a model-to-model (M2M) transformation using ATL to migrate models conforming to the original metamodel to the refined version. The transformation mapped elements from the old metamodel to their counterparts in the updated version while addressing changes such as renamed classes and newly introduced structures.

For example:

- Instances of the **ParkingArea** class in the original metamodel were migrated to **CarParkingArea** in the refined metamodel.
- The **Flight.city** attribute was transformed into a reference to the **City** class, necessitating the creation of **City** instances for each unique city name in the original model.

The ATL transformation script ensured consistency and correctness during the migration process, preserving data integrity while adapting the model to the new schema.

Task 3: Model-to-Text Transformation with Acceleo

The third task focused on **Acceleo**, a model-to-text transformation tool, to generate a modular and user-friendly HTML documentation of an airport terminal's infrastructure and operations. The primary objective is to organize detailed terminal data into a set of interconnected HTML files, each focusing on a specific aspect of the terminal, rather than consolidating all information into a single lengthy file.

The process begins with the creation of a main file, **TerminalIndex.html**, which serves as an entry point and navigation hub for the entire documentation. This file provides an overview of the terminal and includes hyperlinks to individual section-specific HTML files. These sections include **Drop-off**, **Airside**, **Landside**, **Flights**, **Parking Areas**, and **Transportation Services**. Each section is represented in a separate file (e.g., **DropOff.html**, **Airside.html**, etc.), containing detailed data extracted from the model.

The content of each section-specific file is carefully structured with HTML elements such as headings, paragraphs, tables, and lists to ensure clarity and ease of navigation. For instance:

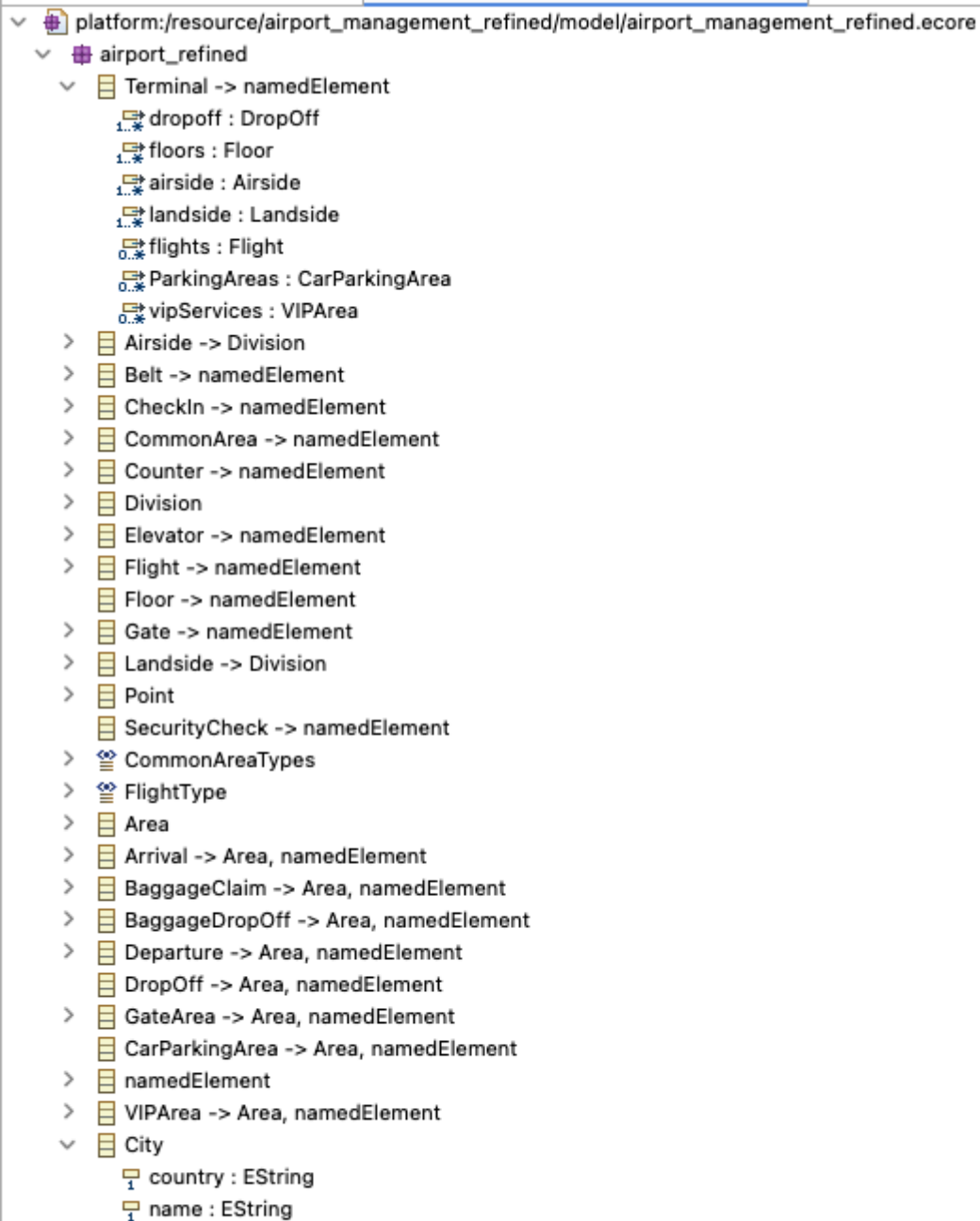
- The **Drop-off** section provides information about the terminal's drop-off area, including its name and the coordinates of its area corners.
- The **Airside** section details the gate area, including gate names, boarding statuses, flights, and the layout of the area corners.
- The **Landside** section includes both the departure and arrival areas, with information on baggage handling facilities, counters, customs, and layout details.

- The **Flights** section uses a table to list key information about scheduled flights, such as names, flight numbers, times, types, destination cities, and whether they are international.
- The **Parking Areas** and **Transportation Services** sections describe the layout, corners, and facilities related to parking and transport services.

Each of these files includes a link back to the `TerminalIndex.html`, ensuring seamless navigation across the documentation. By dividing the content into smaller files, the implementation not only reduces the size of individual files but also allows for easier updates and extensions to specific sections without impacting the entire documentation. This modular approach is ideal for maintaining large and complex datasets, as it promotes logical organization and simplifies data retrieval for users. Additionally, it supports scalability by enabling the inclusion of new sections or enhancements to existing sections with minimal effort.

Appendix

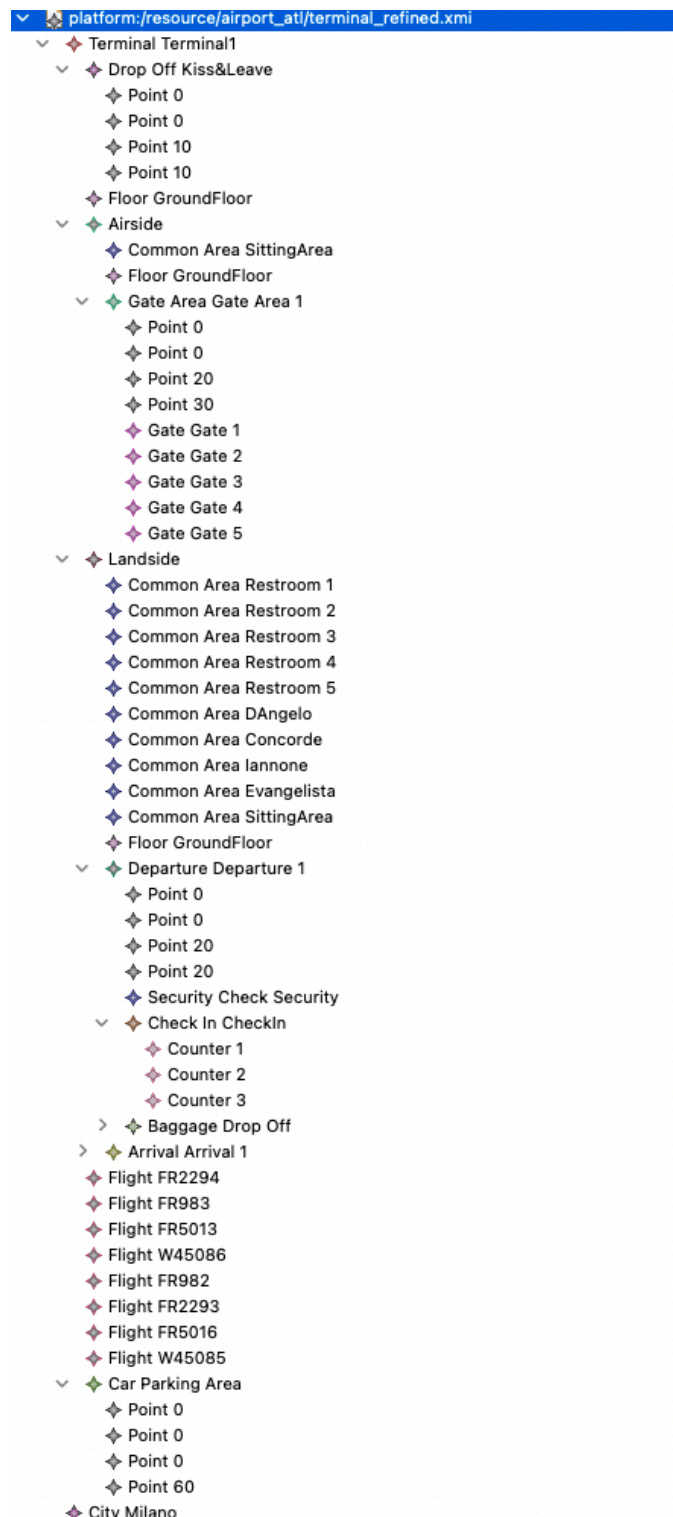
Refined metamodel



ATL transformation rules

```
1 -- Specify paths to the metamodels
2 -- @path AIRPORT=/airport_management/model/airport_management.ecore
3 -- @path REFINED=/airport_management_refined/model/airport_management_refined.ecore
4
5 module trafo;
6
7 -- Declare the input and output models
8 create OUT : REFINED from IN : AIRPORT;
9
10 -- Rule to transform Terminal from AIRPORT to REFINED
11 @rule Terminal2Terminal {
12   from
13     s : AIRPORT!Terminal
14   to
15     t : REFINED!Terminal (
16       name <- s.name
17       ,dropoff <- s.dropoff->collect(d | thisModule.DropOff2DropOff(d))
18       ,floors <- s.floors->collect(f | thisModule.Floor2Floor(f))
19       ,flights <- s.flights->collect(f | thisModule.Flight2Flight(f))
20       ,airside <- s.airside->collect(a | thisModule.Airside2Airside(a))
21       ,landside <- s.landside->collect(l | thisModule.Landside2Landside(l))
22       ,ParkingAreas <- s.ParkingAreas->collect(pa | thisModule.ParkingArea2CarParkingArea(pa))
23       ,vipServices <- s.TransportationServices->collect(ts | thisModule.TransportationServices2VIPArea(ts))
24     )
25 }
26
27 @lazy rule DropOff2DropOff {
28   from
29     s : AIRPORT!DropOff
30   to
31     t : REFINED!DropOff (
32       name <- s.name,
33       areaCorners <- s.areaCorners->collect(p | thisModule.Point2Point(p))
34     )
35 }
36
37 @lazy rule Floor2Floor {
38   from
39     s : AIRPORT!Floor
40   to
41     t : REFINED!Floor (
42       name <- s.name
43     )
44 }
45
46 @lazy rule Airside2Airside {
47   from
48     s : AIRPORT!Airside
49   to
50     t : REFINED!Airside (
51       commonAreas <- s.commonAreas->collect(ca | thisModule.CommonArea2CommonArea(ca)),
52       floor <- thisModule.Floor2Floor(s.floor),
53       gateArea <- s.gateArea->collect(ga | thisModule.GateArea2GateArea(ga))
54     )
55 }
56
57 @lazy rule Landside2Landside {
58   from
59     s : AIRPORT!Landside
60   to
61     t : REFINED!Landside (
62       commonAreas <- s.commonAreas->collect(ca | thisModule.CommonArea2CommonArea(ca)),
63       floor <- thisModule.Floor2Floor(s.floor)
64       ,departureArea <- thisModule.Departure2Departure(s.departureArea)
65     )
66 }
```

Generated transformed model



Generated HTML files

Terminal: Terminal1

- [Drop-off](#)
- [Airside](#)
- [Landside](#)
- [Flights](#)
- [Parking Areas](#)
- [Transportation Services](#)

Drop-off

Name: Kiss&Leave

Area Corners:

- X: 0, Y: 10
- X: 0, Y: 20
- X: 10, Y: 10
- X: 10, Y: 20

[Back to Terminal Overview](#)

Airside

Floor: GroundFloor

Gate Area

Name: Gate Area 1

Total Gates: 5

Area Corners:

- X: 0, Y: 20
- X: 0, Y: 30
- X: 20, Y: 30
- X: 30, Y: 20

Gates:

- Gate Name: Gate 1, Boarding: true, Flight: FR2293
- Gate Name: Gate 2, Boarding: false, Flight: FR5013
- Gate Name: Gate 3, Boarding: false, Flight: invalid
- Gate Name: Gate 4, Boarding: false, Flight: invalid
- Gate Name: Gate 5, Boarding: false, Flight: invalid

[Back to Terminal Overview](#)

Flights

Name	Flight Number	Time	Type	City	International
FR2294	FR2294	10:30	departure	Milano	true
FR983	FR983	10:55	departure	London	true
FR5013	FR5013	10:55	departure	Brussels	true
W45086	W45086	16:15	departure	Tirana	true
FR982	FR982	10:30	arrival	London	true
FR2293	FR2293	10:30	arrival	Milano	true
FR5016	FR5016	15:25	arrival	Brussels	true
W45085	W45085	15:40	arrival	Tirana	true

[Back to Terminal Overview](#)

Transportation Services

- Name: Taxi, Type: Taxi
 - X: 0, Y: 60
 - X: 0, Y: 70
 - X: 60, Y: 10
 - X: 70, Y: 10
- Name: Bus, Type: Bus
 - X: 0, Y: 70
 - X: 0, Y: 80
 - X: 70, Y: 10
 - X: 80, Y: 10
- Name: Train, Type: Train
 - X: 0, Y: 80
 - X: 0, Y: 60
 - X: 60, Y: 10
 - X: 80, Y: 10

[Back to Terminal Overview](#)