UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

Model Driven Engineering
2024-2025

Assignment 4

EMF-Based 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

Airport management systems are complex infrastructures that require meticulous design to manage resources such as terminals, airside areas, gates, and flights. Using model-driven engineering (MDE) principles, this project leverages EMF to define a metamodel that encapsulates the airport's operational domain. The metamodel specifies entities, relationships, and constraints, enabling efficient validation and behavior testing.

The system is developed to include derived fields and validation constraints for real-world relevance. For example, the isInternationalattribute of a flight is derived from its associated city, and operations such as findGateByName encapsulate domain-specific logic.

Metamodel Design

The metamodel is specified in Ecore, representing the structural hierarchy of an airport. Key components include:

1. **Terminal**: The central entity managing various functional and spatial components, such as airside areas, landside areas, and transportation services.
2. **Airside and Landside**: Subdivisions of the terminal, responsible for operations such as gate management and passenger transitions.
3. **GateArea**: A crucial component for managing boarding gates and linking flights.
4. **Flight**: Represents individual flights with attributes like flight number, type (arrival or departure), and derived attributes like isInternational.
5. **Validation Rules**: Integrated into the metamodel using Object Constraint Language (OCL). For instance, ValidTime ensures that flight times conform to the HH:mm format.

To validate its applicability, the metamodel is instantiated in two implementations:

1. **Debrecen International Airport**: This scenario emphasizes features such as multiple floors, Schengen and non-Schengen gate areas, and comprehensive landside and airside facilities.
2. **Pescara Airport Terminal 1**: This implementation highlights a simpler structure with a focus on a single-floor terminal, basic gate area configurations, and essential transportation services.

Implementation

Model Instantiation and Object Validation

The system uses the EMF-generated factory classes to instantiate model elements. For instance, the AirportFactory is utilized to create Terminal, GateArea, Flight, and other entities. Object validation is performed using the EMF Diagnostician to ensure compliance with defined constraints.

The following code illustrates the creation and validation of model objects:

```
Terminal terminal = factory.createTerminal();
terminal.setName("Main Terminal");
validateObject(terminal);
```

Validation results are presented using diagnostic messages. This ensures that all objects conform to their constraints before deployment.

Derived Attributes and Operations

The metamodel includes derived attributes such as isInternational in the Flight class, which evaluates whether a flight is international based on its associated city. Additionally, the GateArea class features operations like calculateTotalGates and findGateByName, implemented in Java for runtime execution.

For example, the findGateByName operation searches for a gate with a specific name within a GateArea:

```
Gate foundGate = gateArea.findGateByName("Gate A");
```

Testing Derived Fields

A test suite validates the correctness of derived attributes and operations. The DerivedFieldsTest class demonstrates how changes in the model impact derived fields and ensures they behave as expected.

```
flight.setCity("LocalCity");
System.out.println("Is International: " + flight.isIsInternational()); // Expected: false
```

Behavior Testing

Functional behavior is tested using unit tests, such as validating the total number of gates in a GateArea and searching for specific gates. The GateAreaTest class ensures that methods like calculateTotalGates yield accurate results.

Results and Validation

The system effectively demonstrates the utility of EMF for modeling complex domains. Validation tests confirm that constraints such as flight time format and valid counter associations are enforced. Derived attributes provide real-time insights into the system's state, ensuring accurate and consistent behavior.

Testing the GateArea operations resulted in the following outcomes:

- calculateTotalGates correctly counted gates.
- findGateByName successfully located gates by their names.