Software Architectures: Model-to-Model Transformation

Koen Bellemans Rob Poelmans

19 May 2017

1 Introduction

We will start by describing the solution for the QTSContinents transformation. As it is the simplest one, and the QLSContinents transformation is a small expansion build on this solution.

2 QTSContinents

The transformation will take an input xml file, structured in the format described in Continents.ecore, and create an output xml file structured as described in QTSContinents.ecore. This transformation will follow the rule given in Continent2QTSContinent.atl.

This rule takes one continent at a time, and uses some helper functions to transform the data. In our case each helper will loop over the countries belonging to that continent, and accumulate their values for a particular field. In the case of population, this accumulated value is used as the total population for that continent. For gdp, costOfLiving, and povertyLine we need an average, so before returning the accumulator, it is divided by the number of countries belonging to the continent.

3 QLSContinents

The QLSContinent transformation is largely analogues with the previous one. The difference lies in the return type of the helpers. Where they returned floats previously, they return strings now. We achieve this by creating a let, which calculates the desired value as done in QTSContinent, and then runs it through a nested if structure to return the string value associated with the calculated numerical one.