



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

**Área Departamental de Engenharia de Electrónica e Telecomunicações e de
Computadores**

**Geração automática de API fluente Java com base em esquema
XML**

LUÍS CARLOS DA SILVA DUARTE

Licenciado em Engenharia Informática e de Computadores

**Relatório preliminar para obtenção do Grau de Mestre
em Engenharia Informática e de Computadores**

Orientador : Doutor Fernando Miguel Gamboa de Carvalho

Janeiro, 2018



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

**Área Departamental de Engenharia de Electrónica e Telecomunicações e de
Computadores**

**Geração automática de API fluente Java com base em esquema
XML**

LUÍS CARLOS DA SILVA DUARTE

Licenciado em Engenharia Informática e de Computadores

**Relatório preliminar para obtenção do Grau de Mestre
em Engenharia Informática e de Computadores**

Orientador : Doutor Fernando Miguel Gamboa de Carvalho

Janeiro, 2018



Introduction

This project has the objective of implementing a Java solution that allows the automatic generation of a fluent API based on a XML schema. This allows to automate the generation of classes, which are very similar most of the time, therefore saving time to the user and also eliminating human error on the said generation.

1.1 Use case

The use case that will be used to test and evaluate the solution will be the HTML5 XML schema. In this case we have multiple elements that share behavior and/or attributes that can be generated automatically.

The generated HTML5 elements API will then be used in the HtmlFlow API, which is also a fluent java API that is used to write well formed HTML files. At the moment the HtmlFlow library only supports a set of the HTML elements which were created manually and the rest of the library interacts with those elements in order to write the HTML files. With the help of the solution which will be developed in this thesis the HtmlFlow will support the whole HTML syntax.

1.2 Organization

This project will be divided in separate components, in order to keep a logic organization. The different components are explained in the following subsections. The interaction between the existing components is described in 1.1.

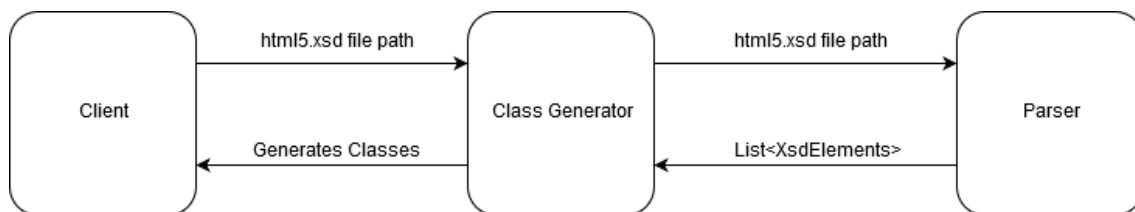


Figure 1.1: Components interaction

1.2.1 Parser

The parser will have the responsibility to read the XML schema element tree and extract the needed information in order to generate the respective classes. The result of the execution of this component should be a list of elements, each element representing a class that should be created and containing all the information needed for the creation.

1.2.2 Class Generator

The class generator will have the responsibility of generating classes based on the information received from the Parser. The class generator should request the parsing of a XML schema file and based on the Parser result, create the classes accordingly. Apart from that the generator should also create an infrastructure that will help the usage of the the resulting API.

To achieve the generation of the classes a tool named ASM will be used. This tool allows the manipulation of byte codes, allowing the generation of classes, methods and fields.

1.2.2.1 Supporting Infrastructure

The generated code will be supported by an infrastructure that mimics the syntax of XML schema files. The supporting infrastructure is shown in 1.2.

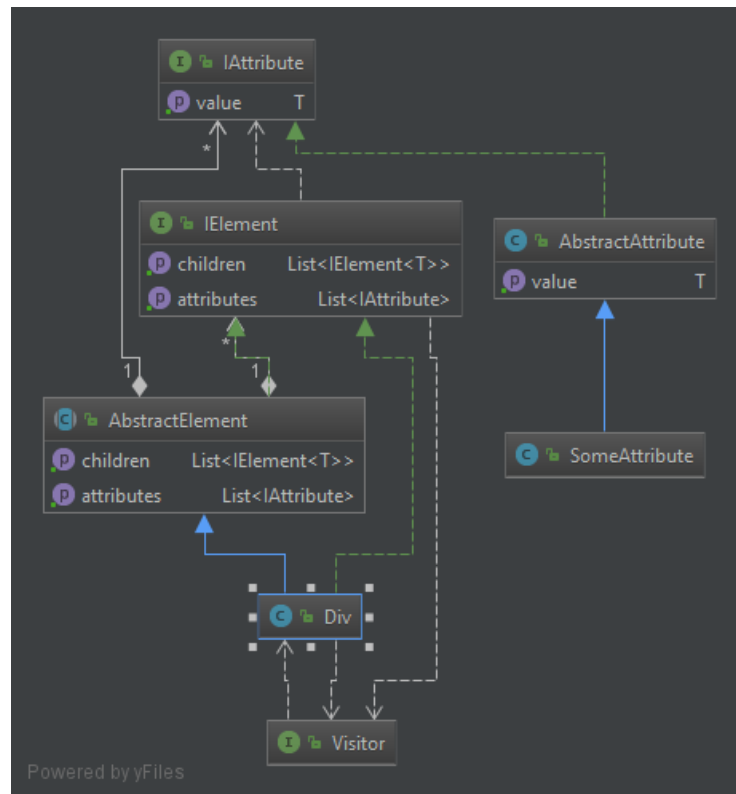


Figure 1.2: Supporting Infrastructure

All the generated APIs will have this classes which are independent of the contents of the parsed file. This infrastructure will then be extended by different type of classes, divided in four groups.

Elements

The elements are a group of classes that will be generated based in the existing xsd:elements. All these classes will extend AbstractElement. Each element will also contain the specific element code, which can include addition of elements, attributes or implementing group and element interfaces.

Attributes

The attributes are a group of classes that will be generated based in the existing xsd:attributes. All these classes will extend AbstractAttribute. Each attribute will have a type, that indicates the type of the value of the said attribute. Attributes will also enforce restrictions to their value if there are any explicitly described in the XML schema file.

Group Interface

The group interfaces are an addition that represent the `xsd:attributegroups`. In the XML schema these attributegroups indicate that a given element is allowed to have the said attributes, in the generated code the respective interfaces allow the addition of all the attributes present in the said group to the element attributes.

Element Interface

The element interfaces are similar to the group interfaces, the difference being that element interfaces allow the addition of other elements as children of the current element.

Visitors

In order to the generated API allow manipulation by the client all the generated elements implement the Visitor pattern, therefore the client of the API can implement its own Visitor class and specify the behavior of the visit methods.

1.2.3 Client

A separate component that will execute the class generator specifying the file name and the API name to be generated. This component will then contain the resulting API and can be used directly. This component can also implement tests to verify if the generation of the API implemented all the expected features.