REPUBLIC OF CAMEROON

Peace-Work- Fatherland

University of Buea

REPUBLIC DU CAMEROUN

Paix-Travail-Patrie

Universite de Buea

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

CEF482

XML AND DOCUMENT CONTENT VALIDATION

GROUP 5

MODELING EMPLOYEE TYPES USING INHERITANCE AND SUBSTITUTION GROUPS

Course Instructor

Dr. SOP DEFFO Lionel Landry

Academic year 2023/2024

GROUP MEMBERS AND MATRICULE

NAME	MATRICULE
AHOUMO TEMATEU ROXANE PHILIPPINE	FE21A128
ALAIN NKEH MBUNKUR	FE21A133
ALEANU NTIMAEH ENOW	FE21A134
ANANFACK ZANGO ANGELA CARINE	FE21A137
BOCHUKE BABILA DANIEL	FE21A152
CHO WESLEY MUNGO	FE21A160
DJOUMESSI DONGMO RONSARD C	FE21A172
NKWETI MANGEM ANGCELIA	FE21A280
NYENTY PAUL EGBE	FE21A292
NZEMTEJUH SYLVANUS	FE21A296

INTRODUCTION

In the realm of XML schema design, inheritance and substitution groups are powerful mechanisms that provide flexibility and reusability. Inheritance allows one to define a base type with common elements, which can be extended by more specific types. Substitution groups enable different elements to be used interchangeably, thus supporting polymorphism within XML documents.

In this report, we detail the creation and use of an XML schema to model a hierarchy of employee types using inheritance and substitution groups. The schema supports the representation of full-time, part-time, and contractor employees, providing a structured and flexible framework for defining different employee categories.

SCHEMA DESIGN

The XML schema *(employee.xsd)* is designed to establish a structured hierarchy of employee types using inheritance and substitution groups. This design facilitates the representation of various employee categories while ensuring data integrity and consistency.

Base Type Definition

The schema begins by defining a base type, EmployeeType, which is abstract and contains elements common to all employee types:

```
<xs:complexType name="EmployeeType" abstract="true">
  <xs:sequence>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="ID" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

This abstract type cannot be instantiated directly but serves as a foundation for more specific employee types.

Derived Types

The schema then defines specific employee types by extending the base EmployeeType. Each derived type adds its unique elements:

1. Full-Time Employee:

```
<xs:complexType name="FullTimeEmployeeType">
  <xs:complexContent>
    <xs:extension base="EmployeeType">
      <xs:sequence>
        <xs:element name="AnnualSalary" type="xs:decimal"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

A full-time employee has an annual salary as an additional attribute. This type extends the base EmployeeType by adding the AnnualSalary element.

2. Part-time Employees:

```
<xs:complexType name="PartTimeEmployeeType">
  <xs:complexContent>
    <xs:extension base="EmployeeType">
      <xs:sequence>
        <xs:element name="HourlyRate" type="xs:decimal"/>
        <xs:element name="HoursPerWeek" type="xs:integer"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

A part-time employee is characterized by an hourly rate and the number of hours worked per week. This type extends the base EmployeeType by adding the HourlyRate and HoursPerWeek elements.

2. Contractor Employees:

```
<xs:complexType name="ContractorEmployeeType">
  <xs:complexContent>
    <xs:extension base="EmployeeType">
      <xs:sequence>
        <xs:element name="ContractRate" type="xs:decimal"/>
        <xs:element name="ContractDuration" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

A contractor employee has a contract rate and contract duration. This type extends the base EmployeeType by adding the ContractRate and ContractDuration elements.

Substitution Groups

Substitution groups are used to allow instances of different derived types to be substituted wherever the base type element Employee is referenced. The base element Employee is defined as abstract, and specific employee elements are included in the substitution group:

```
<xs:element name="Employee" type="EmployeeType" abstract="true"/>
<xs:element name="FullTimeEmployee" type="FullTimeEmployeeType"
substitutionGroup="Employee"/>
<xs:element name="PartTimeEmployee" type="PartTimeEmployeeType"
substitutionGroup="Employee"/>
<xs:element name="ContractorEmployee" type="ContractorEmployeeType"
substitutionGroup="Employee"/>
```

- The Employee element is defined as abstract and serves as the head of the substitution group.
- FullTimeEmployee, PartTimeEmployee, and ContractorEmployee elements are included in the substitution group, allowing them to replace Employee wherever it appears in the XML document.

Root Element

The root element Employees is designed to contain a sequence of Employee elements, allowing the inclusion of various employee types within the same document:

```
<xs:element name="Employees">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Employee" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

- The Employees element acts as a container for a collection of Employee elements.
- The maxOccurs="unbounded" attribute allows multiple Employee elements (of any type within the substitution group) to be included, supporting a diverse set of employee records in a single XML document.

Combining the above codes into a single code:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!-- Define the base type for employees -->
  <xs:complexType name="EmployeeType" abstract="true">
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="ID" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <!-- Define full-time employee type -->
  <xs:complexType name="FullTimeEmployeeType">
    <xs:complexContent>
      <xs:extension base="EmployeeType">
        <xs:sequence>
          <xs:element name="AnnualSalary" type="xs:decimal"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <!-- Define part-time employee type -->
  <xs:complexType name="PartTimeEmployeeType">
    <xs:complexContent>
      <xs:extension base="EmployeeType">
        <xs:sequence>
          <xs:element name="HourlyRate" type="xs:decimal"/>
          <xs:element name="HoursPerWeek" type="xs:integer"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

```
    <!-- Define contractor employee type -->
    <xs:complexType name="ContractorEmployeeType">
        <xs:complexContent>
            <xs:extension base="EmployeeType">
                <xs:sequence>
                    <xs:element name="ContractRate" type="xs:decimal"/>
                    <xs:element name="ContractDuration" type="xs:string"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <!-- Define elements for each employee type using substitution groups -->
    <xs:element name="Employee" type="EmployeeType" abstract="true"/>

    <xs:element name="FullTimeEmployee" type="FullTimeEmployeeType"
substitutionGroup="Employee"/>
    <xs:element name="PartTimeEmployee" type="PartTimeEmployeeType"
substitutionGroup="Employee"/>
    <xs:element name="ContractorEmployee" type="ContractorEmployeeType"
substitutionGroup="Employee"/>

    <!-- Root element -->
    <xs:element name="Employees">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Employee" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Example XML Document

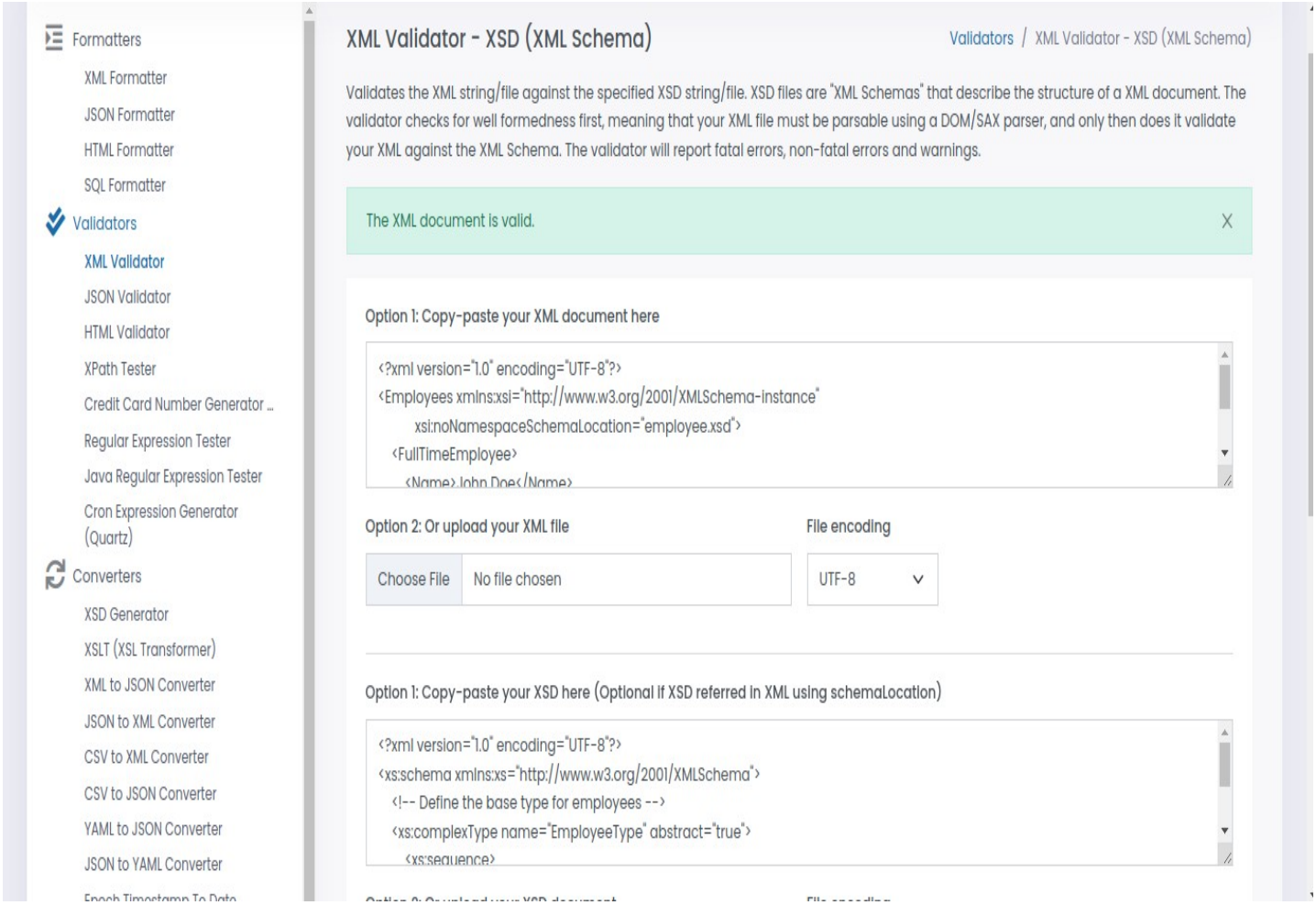
The example XML document *(employees.xml)* below illustrates the schema's usage by including different types of employees:

```
<?xml version="1.0" encoding="UTF-8"?>
<Employees xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="employee.xsd">
   <FullTimeEmployee>
      <Name>John Doe</Name>
      <ID>FT123</ID>
      <AnnualSalary>60000.00</AnnualSalary>
   </FullTimeEmployee>
   <PartTimeEmployee>
      <Name>Jane Smith</Name>
      <ID>PT456</ID>
      <HourlyRate>20.00</HourlyRate>
      <HoursPerWeek>25</HoursPerWeek>
   </PartTimeEmployee>
   <ContractorEmployee>
      <Name>Bob Johnson</Name>
      <ID>C789</ID>
      <ContractRate>50.00</ContractRate>
      <ContractDuration>6 months</ContractDuration>
   </ContractorEmployee>
</Employees>
```

VALIDATION

The XML document *(employees.xml)* was validated against the schema *(employee.xsd)* using the online XML validation tool, *FreeFormatter.com XML Validator*.

This was done to ensure that the XML document adhered to the defined structure and constraints, providing consistency and reliability in data representation.



CONCLUSION

The use of inheritance and substitution groups in the XML schema effectively models a hierarchy of employee types, offering a robust and extensible solution for managing various categories of employees. This approach ensures flexibility, allowing for future extensions and modifications with minimal changes to the overall schema structure.