

```

<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:conv="http://www.openuri.org/2002/04/soap/conversation/"
xmlns:cw="http://www.openuri.org/2002/04/wsdl/conversation/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:jms="http://www.openuri.org/2002/04/wsdl/jms/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:s0="http://www.openuri.org/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
targetNamespace="http://www.openuri.org/"
  <types>
    <s:schema xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:fent="http://temp.openuri.org/Fent.xsd" elementFormDefault="qualified"
targetNamespace="http://www.openuri.org/"
      <s:import namespace="http://temp.openuri.org/Fent.xsd"/>
      <s:element name="find">
        <s:complexType>
          <s:sequence>
            <s:element ref="fent:cbeEntityRequest"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="findResponse">
        <s:complexType>
          <s:sequence>
            <s:element ref="fent:cbeEntityReply"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:fent="http://temp.openuri.org/Fent.xsd"
xmlns:fentv1="http://temp.openuri.org/v1/Fent.xsd"
targetNamespace="http://temp.openuri.org/Fent.xsd"
elementFormDefault="qualified" attributeFormDefault="unqualified">
      <xs:import namespace="http://temp.openuri.org/v1/Fent.xsd"/>
      <xs:element name="reply" abstract="true"/>
      <xs:element name="request" abstract="true"/>
      <xs:element name="cbeEntityReply">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="fent:reply"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="cbeEntityRequest">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="fent:request"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:fentv1="http://temp.openuri.org/v1/Fent.xsd"
xmlns:cbe="http://fsb.belgium.be/WSKBO/datamodel/v1"
xmlns:fent="http://temp.openuri.org/Fent.xsd"
xmlns:com="urn://fsb.belgium.be/fphp/v3_0/common"
xmlns:iso="http://fsb.belgium.be/common/isocodes"
targetNamespace="http://temp.openuri.org/v1/Fent.xsd"
elementFormDefault="qualified" attributeFormDefault="unqualified">
      <xs:import namespace="http://fsb.belgium.be/WSKBO/datamodel/v1"/>

```

```

<xs:import namespace="http://temp.openuri.org/Fent.xsd"/>
<xs:import namespace="http://fsb.belgium.be/common/isocodes"/>
<xs:import namespace="urn://fsb.belgium.be/fphp/v3_0/common"/>
<xs:element name="request_v1" type="fentv1:EntityRequest_v1"
substitutionGroup="fent:request"/>
<!--redefined elements-->
<xs:element name="RequestEntityData" abstract="true"/>
<xs:element name="RequestEntityData_v1"
type="fentv1:EntityRequestDataType_v1"
substitutionGroup="fentv1:RequestEntityData">
  <xs:annotation>
    <xs:documentation>Specific request data's</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="EntitySelectionFilter" abstract="true"/>
<xs:element name="EntitySelectionFilter_v1"
type="fentv1:EntitySelectionFilter_v1"
substitutionGroup="fentv1:EntitySelectionFilter">
  <xs:annotation>
    <xs:documentation>contains every criterion to select a set of subject
to be returned to the requester. It is directly linked to the proportionality
of a request.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="EntityValueFilter" abstract="true"/>
<xs:element name="EntityValueFilter_v1"
type="fentv1:EntityValueFilter_v1"
substitutionGroup="fentv1:EntityValueFilter">
  <xs:annotation>
    <xs:documentation>contains every information of which fields should
be returned to the requester. It is directly linked to finality of a
request.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="CommonValueFilter" abstract="true"/>
<xs:element name="CommonValueFilter_v1"
type="fentv1:CommonValueFilterType_v1"
substitutionGroup="fentv1:CommonValueFilter">
  <xs:annotation>
    <xs:documentation>common domains of value</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="CommonValueFilter_v2"
type="fentv1:CommonValueFilterType_v2"
substitutionGroup="fentv1:CommonValueFilter">
  <xs:annotation>
    <xs:documentation>common domains of value</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="EnterpriseValueFilter" abstract="true"/>
<xs:element name="EnterpriseValueFilter_v1"
type="fentv1:EnterpriseValueFilterType_v1"
substitutionGroup="fentv1:EnterpriseValueFilter">
  <xs:annotation>
    <xs:documentation>enterprise domains of value</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Context" abstract="true"/>
<xs:element name="Context_v1" type="fentv1:ContextType_v1"
substitutionGroup="fentv1:Context">
  <xs:annotation>
    <xs:documentation>enterprise domains of value</xs:documentation>
  </xs:annotation>

```

```

</xs:element>
<!--redefined schemas-->
<xs:complexType name="EntityRequest_v1">
  <xs:annotation>
    <xs:documentation>a fentv1 request</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="RequestId" type="xs:string"/>
    <xs:element ref="fentv1:Context"/>
    <xs:element ref="fentv1:RequestEntityData"/>
  </xs:sequence>
  <xs:attribute name="ReplyVersion" use="required">
    <xs:simpleType>
      <xs:annotation>
        <xs:documentation>version 1.1: initial reply
version</xs:documentation>
        <xs:documentation>version 1.2: version with the ability for
extended function information.</xs:documentation>
      </xs:annotation>
      <xs:restriction base="com:Version">
        <xs:enumeration value="1.1"/>
        <xs:enumeration value="1.2"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
<xs:complexType name="EntityRequestDataType_v1">
  <xs:annotation>
    <xs:documentation>business data's used to process the request of an
enterprise research</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="fentv1:EntitySelectionFilter"/>
    <xs:element ref="fentv1:EntityValueFilter"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="EntitySelectionFilter_v1">
  <xs:annotation>
    <xs:documentation>criteria's and values used to identify the
requested enterprises</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="CbeNumber" type="cbe:CbeEntityNumber"
maxOccurs="1">
      <xs:annotation>
        <xs:documentation>known CBEnumber to get access to detailed
informations.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ContextType_v1">
  <xs:annotation>
    <xs:documentation>business data's used to process the request of an
enterprise research</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Nrn" type="com:PersonNumber"/>
    <xs:element name="OrganizationUnit">
      <xs:annotation>
        <xs:documentation>Final organization unit. CBE refered it as
"instrumenteerde administratie"</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="AgencyCode">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:length value="10"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="DepartmentCode">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:length value="10"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Language" type="iso:language" minOccurs="0">
  <xs:annotation>
    <xs:documentation xml:lang="en">The language in which any textual
fields are
                                expressed.</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="EntityValueFilter_v1">
  <xs:annotation>
    <xs:documentation>Flags to describe which fields (domains) of an
enterprise should be returned back. The request will be rejected if an
unauthorized value has been asked. The reason of a rejection</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="fentv1:CommonValueFilter"/>
    <xs:element ref="fentv1:EnterpriseValueFilter" minOccurs="0"/>
    <xs:element name="IncludeDates" type="xs:boolean" minOccurs="0">
      <xs:annotation>
        <xs:documentation>include validity dates inside the
reply</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="IncludeDescriptions" type="xs:boolean"
minOccurs="0">
      <xs:annotation>
        <xs:documentation>include description of codes used inside the
reply</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="EnterpriseValueFilterType_v1">
  <xs:annotation>
    <xs:documentation>Specific flags for an enterprise
entity</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="JuridicalSituation" type="xs:boolean"
default="false" minOccurs="0">
      <xs:annotation>
        <xs:documentation>get the juridical situation</xs:documentation>
      </xs:annotation>
    </xs:sequence>

```

```

        </xs:element>
        <xs:element name="ForeignBranchOffices" type="xs:boolean"
default="false" minOccurs="0">
        <xs:annotation>
            <xs:documentation>get the foreign branch
offices</xs:documentation>
        </xs:annotation>
        </xs:element>
        <xs:element name="Publications" type="xs:boolean" default="false"
minOccurs="0">
        <xs:annotation>
            <xs:documentation>get the publications in the belgian
monitor</xs:documentation>
        </xs:annotation>
        </xs:element>
        <xs:element name="Bankaccounts" type="xs:boolean" default="false"
minOccurs="0">
        <xs:annotation>
            <xs:documentation>Set to true if the client wants to have the
bank accounts of the enterprise
returned in the response.</xs:documentation>
        </xs:annotation>
        </xs:element>
        <xs:element name="Finances" type="xs:boolean" default="false"
minOccurs="0">
        <xs:annotation>
            <xs:documentation>Set to true if the client wants to have the
finances information of the enterprise
included in the response. Default is
false.</xs:documentation>
        </xs:annotation>
        </xs:element>
        <xs:element name="LinkedEnterprise" type="xs:boolean" default="false"
minOccurs="0">
        <xs:annotation>
            <xs:documentation>Set to true if the client wants to have the
links with other enterprises in case
of merges. Default is false.</xs:documentation>
        </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="CommonValueFilterType_v1">
    <xs:annotation>
        <xs:documentation>Common flags for any entity (enterprise or business
unit)</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="BasicDatas" type="xs:boolean" default="true">
            <xs:annotation>
                <xs:documentation>default : true. Please set it false to ignore
those fields : status, type, denominations, address, juridicalForms [see
response fields].</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Activities" type="xs:boolean" default="false"
minOccurs="0">
            <xs:annotation>
                <xs:documentation>get the nacebel codes</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Functions" type="xs:boolean" default="false"
minOccurs="0">

```

```

        <xs:annotation>
            <xs:documentation>get the Functions</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="ExtendedFunctionsInfo" type="xs:boolean"
default="false" minOccurs="0">
        <xs:annotation>
            <xs:documentation>get the detailed Functions</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="Permissions" type="xs:boolean" default="false"
minOccurs="0">
        <xs:annotation>
            <xs:documentation>get the permissions</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="CommonValueFilterType_v2">
    <xs:annotation>
        <xs:documentation>Common flags for any entity (enterprise or business
unit)</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="BasicDatas" type="xs:boolean" default="true">
            <xs:annotation>
                <xs:documentation>default : true. Please set it false to ignore
those fields : status, type, denominations, address, juridicalForms [see
response fields].</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Activities" type="xs:boolean" default="false"
minOccurs="0">
            <xs:annotation>
                <xs:documentation>get the nacebel codes</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Functions" type="xs:boolean" default="false"
minOccurs="0">
            <xs:annotation>
                <xs:documentation>get the Functions</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="ExtendedFunctionsInfo" type="xs:boolean"
default="false" minOccurs="0">
            <xs:annotation>
                <xs:documentation>get the detailed Functions</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Permissions" type="xs:boolean" default="false"
minOccurs="0">
            <xs:annotation>
                <xs:documentation>get the permissions</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<!--reply-->
<xs:element name="reply_v1" type="fentv1:EntityReplyDatas_v1"
substitutionGroup="fent:reply"/>
<xs:element name="Enterprise" abstract="true"/>
<xs:element name="Enterprise_v1" type="fentv1:EnterpriseType_v1"
substitutionGroup="fentv1:Enterprise"/>

```

```

    <xs:element name="Warning" abstract="true"/>
    <xs:element name="Warning_v1" type="fentv1:Warning_v1"
substitutionGroup="fentv1:Warning"/>
    <xs:element name="Error" abstract="true"/>
    <xs:element name="Error_v1" type="fentv1:Error_v1"
substitutionGroup="fentv1:Error"/>
    <xs:element name="Backend" abstract="true"/>
    <xs:element name="EntityCommonInfo" abstract="true"/>
    <xs:element name="EntityCommonInfo_v1"
type="fentv1:EntityCommonDataType_v1"
substitutionGroup="fentv1:EntityCommonInfo"/>
    <xs:element name="Function" abstract="true"/>
    <xs:element name="Function_v1" type="fentv1:FunctionType_v1"
substitutionGroup="fentv1:Function"/>
    <xs:element name="heldByPerson" abstract="true"/>
    <xs:element name="heldByPerson_v1" type="fentv1:Person_v1"
substitutionGroup="fentv1:heldByPerson"/>
    <xs:element name="heldByPerson_v2" type="fentv1:Person_v2"
substitutionGroup="fentv1:heldByPerson"/>
    <xs:element name="Person" abstract="true"/>
    <xs:element name="Person_v1" type="cbe:functionExecutorPerson"
substitutionGroup="fentv1:Person"/>
    <xs:element name="PersonDetails" abstract="true"/>
    <xs:element name="PersonDetails_v1" type="fentv1:Summary_v1"
substitutionGroup="fentv1:PersonDetails"/>
    <xs:complexType name="EntityReplyDatas_v1">
      <xs:annotation>
        <xs:documentation>List of business entities</xs:documentation>
      </xs:annotation>
      <xs:choice>
        <xs:sequence>
          <xs:element ref="fentv1:Enterprise"/>
          <xs:element ref="fentv1:Warning" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:sequence>
          <xs:element ref="fentv1:Error" minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
      </xs:choice>
    </xs:complexType>
    <xs:complexType name="EnterpriseType_v1">
      <xs:annotation>
        <xs:documentation>atomic enterprise (entity
realization)</xs:documentation>
      </xs:annotation>
      <xs:sequence>
        <xs:element name="Number" type="cbe:CbeEntityNumber" minOccurs="0"/>
        <xs:element name="Type" type="cbe:CbeEnterpriseType">
          <xs:annotation>
            <xs:documentation>EPP/EPM. isBasicDatas</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="JuridicalForm" type="cbe:JuridicalForm"
minOccurs="0">
          <xs:annotation>
            <xs:documentation>isBasicDatas</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="JuridicalSituation" type="cbe:JuridicalSituation"
minOccurs="0"/>
        <xs:element name="ForeignBranchOffice"
type="cbe:ForeignBranchOfficeType" minOccurs="0">
          <xs:annotation>

```

```

        <xs:documentation>branch office in Belgium for external
society.</xs:documentation>
    </xs:annotation>
</xs:element>
    <xs:element name="BusinessUnit" type="cbe:BusinessUnitType"
minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>should not contain details for sub-
BusinessUnit.</xs:documentation>
    </xs:annotation>
</xs:element>
    <xs:element name="Publication" type="cbe:PublicationType"
minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>publication in the Belgian
Monitor</xs:documentation>
    </xs:annotation>
</xs:element>
    <xs:element name="Date" type="cbe:EnterpriseDateType" minOccurs="0">
    <xs:annotation>
        <xs:documentation>isBasicDatat</xs:documentation>
    </xs:annotation>
</xs:element>
    <xs:element ref="fentv1:EntityCommonInfo" minOccurs="0"/>
    <xs:element name="Capital" type="xs:double" minOccurs="0"/>
    <xs:element name="Currency" type="xs:string" minOccurs="0"/>
    <xs:element name="Juridical" type="xs:boolean" minOccurs="0"/>
    <xs:element name="BankAccount" type="cbe:BankAccount" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Finances" type="cbe:Finances" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="EntityCommonDataType_v1">
    <xs:annotation>
        <xs:documentation>Domains containing all general information about an
of
        enterprise and/or a business unit. iow: common fields
        a business entity (common between all
realization).</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Status" type="cbe:StatusType" minOccurs="0"/>
        <xs:element name="Denomination" type="cbe:DenominationType"
minOccurs="0" maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>isBasicDatat</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Address" type="cbe:AddressType" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>isBasicDatat</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Activity" type="cbe:ActivityType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element ref="fentv1:Function" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="LinkedEnterprise" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>

```

```

        <xs:documentation>enterprise with KBO-known
relationship</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="cbe:LinkedEnterpriseType"/>
        </xs:complexContent>
    </xs:complexType>
</xs:element>
<xs:element name="Permission" type="cbe:AuthorizationType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="Validity" type="cbe:ValidityType" minOccurs="0">
    <xs:annotation>
        <xs:documentation>isBasicDatas</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="FunctionType_v1">
    <xs:sequence>
        <xs:choice>
            <xs:element name="heldByEnterprise" type="cbe:CbeEntityNumber">
                <xs:annotation>
                    <xs:documentation>The enterprisenumber of the enterprise
                    holding the
function</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element ref="fentv1:heldByPerson_v1"/>
        </xs:choice>
        <xs:element name="Type" type="cbe:FunctionCodeType">
            <xs:annotation>
                <xs:documentation>CBE code</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="TypeDescription" type="xs:string" minOccurs="0">
            <xs:annotation>
                <xs:documentation>if required</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="ValidityPeriod" type="cbe:Period" minOccurs="0"/>
        <xs:element name="Stop" type="cbe:CbeStopCode" minOccurs="0">
            <xs:annotation>
                <xs:documentation>if stopped, CBE code</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="StopDescription" type="xs:string" minOccurs="0">
            <xs:annotation>
                <xs:documentation>if required</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Summary_v1">
    <xs:annotation>
        <xs:documentation xml:lang="en">The summary provides limited current
information.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="source" type="fentv1:Register"/>
        <xs:element name="number" type="com:PersonNumber"/>
        <xs:element name="status" type="xs:integer">
            <xs:annotation>

```

```

        <xs:documentation>Possible values:
            0: OK
            1: error
            3: person not found
            4: person scratched from register: please check other
register for more up-to-date results
            5: person referenced: please check again with referenced
number
            6: back-end access violation</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="gender" type="iso:gender"/>
<xs:element name="lastName" type="xs:string"/>
<xs:element name="firstName" type="xs:string" minOccurs="0"/>
<xs:element name="middleName" type="xs:string" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Second and following
firstname</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="birthDate" type="com:Date" minOccurs="0"/>
<xs:element name="birth_place_nis_code" type="xs:string"
minOccurs="0"/>
<xs:element name="birth_place_description" type="xs:string"
minOccurs="0"/>
<xs:element name="birth_country_code" type="iso:country"
minOccurs="0"/>
<xs:element name="nationality_code" type="iso:country"
minOccurs="0"/>
<xs:element name="isAlive" type="xs:boolean" minOccurs="0"/>
<xs:element name="marital_status_code" type="xs:string"
minOccurs="0"/>
<xs:element name="ad_street_code" type="xs:string" minOccurs="0"/>
<xs:element name="ad_street_name" type="xs:string" minOccurs="0"/>
<xs:element name="ad_street_number" type="xs:string" minOccurs="0"/>
<xs:element name="ad_post_box" type="xs:string" minOccurs="0"/>
<xs:element name="ad_nis_code" type="xs:string" minOccurs="0"/>
<xs:element name="ad_post_code" type="xs:string" minOccurs="0"/>
<xs:element name="ad_city" type="xs:string" minOccurs="0"/>
<!-- for unstructured (declared) address data -->
<xs:element name="ad_declared_free_format" type="xs:string"
minOccurs="0"/>
<!-- for unstructured (foreign) address data -->
<xs:element name="ad_foreign_free_format_1" type="xs:string"
minOccurs="0"/>
<xs:element name="ad_foreign_free_format_2" type="xs:string"
minOccurs="0"/>
<xs:element name="ad_foreign_free_format_3" type="xs:string"
minOccurs="0"/>
<!-- for unstructured (foreign) declared address data -->
<xs:element name="ad_foreign_declared_free_format_1" type="xs:string"
minOccurs="0"/>
<xs:element name="ad_foreign_declared_free_format_2" type="xs:string"
minOccurs="0"/>
<xs:element name="ad_foreign_declared_free_format_3" type="xs:string"
minOccurs="0"/>
<!-- end of free-format address fields -->
<xs:element name="ad_country_code" type="iso:country" minOccurs="0"/>
<xs:element name="ad_country_name" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:simpleType name="Register">
    <xs:annotation>

```

```

        <xs:documentation xml:lang="en">Enumeration of the possible
registers. Both the Dutch and the French language are recognized.
        The application using these codes should make sure that the
Dutch and French values are considered equal.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:enumeration value="KSZ">
            <xs:annotation>
                <xs:documentation xml:lang="nl">Kruispuntbank der Sociale
Zekerheid.</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="BCSS">
            <xs:annotation>
                <xs:documentation xml:lang="fr">Banque Carrefour de la Securite
Sociale.</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="RR">
            <xs:annotation>
                <xs:documentation xml:lang="nl">Rijksregister.</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="RN">
            <xs:annotation>
                <xs:documentation xml:lang="fr">Registre
Nationale.</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="KBO">
            <xs:annotation>
                <xs:documentation xml:lang="nl">Kruispuntbank van
Ondernemingen.</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="BCE">
            <xs:annotation>
                <xs:documentation xml:lang="fr">Banque Carrefour des
Entreprises.</xs:documentation>
            </xs:annotation>
        </xs:enumeration>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="Person_v1">
    <xs:sequence>
        <xs:element ref="fentv1:Person"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Person_v2">
    <xs:sequence>
        <xs:choice>
            <xs:element ref="fentv1:Person"/>
            <xs:element ref="fentv1:PersonDetails"/>
        </xs:choice>
        <xs:element ref="fentv1:Warning" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Error_v1">
    <xs:annotation>
        <xs:documentation>Definition of the error object.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Id" type="xs:string">

```

```

        <xs:annotation>
            <xs:documentation>An identifier for the error. This is generally
the type of exception, such as
                NullPointerException, XmlException,
etc.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="Code" type="xs:string">
        <xs:annotation>
            <xs:documentation>The error code.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="Description" type="xs:string">
        <xs:annotation>
            <xs:documentation>The description of the error
code.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="StackTrace" type="xs:string" minOccurs="0">
        <xs:annotation>
            <xs:documentation>The stacktrace of the error. This element is
purely technical in nature, and is only of
                use in rare occasions.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="SuggestedAction" type="xs:string" minOccurs="0">
        <xs:annotation>
            <xs:documentation>The suggested action to take after the error
has been received.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="Cause" type="fentv1:Error_v1" minOccurs="0">
        <xs:annotation>
            <xs:documentation>The cause of the error could be another error.
If so, that error is provided here.
                There can be a whole stack of errors. (Beware of infite
loop when using this element.)</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element ref="fentv1:Backend" minOccurs="0">
        <xs:annotation>
            <xs:documentation>This element can be used to include the backend
error or warning.</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="Warning_v1">
    <xs:annotation>
        <xs:documentation>Definition of the error object.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Id" type="xs:string">
            <xs:annotation>
                <xs:documentation>An identifier for the error. This is generally
the type of exception, such as
                    NullPointerException, XmlException,
etc.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Code" type="xs:string">
            <xs:annotation>
                <xs:documentation>The error code.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>

```

```

        </xs:annotation>
    </xs:element>
    <xs:element name="Description" type="xs:string">
        <xs:annotation>
            <xs:documentation>The description of the error
code.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element ref="fentv1:Backend" minOccurs="0">
        <xs:annotation>
            <xs:documentation>This element can be used to include the backend
error or warning.</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:schema>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://fsb.belgium.be/WSKBO/datamodel/v1"
xmlns:addr="http://fsb.belgium.be/prove/address"
xmlns:iso="http://fsb.belgium.be/common/isocodes"
xmlns:person="http://fsb.belgium.be/common/PersonNumber"
targetNamespace="http://fsb.belgium.be/WSKBO/datamodel/v1"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:import namespace="http://fsb.belgium.be/prove/address"/>
    <xs:import namespace="http://fsb.belgium.be/common/isocodes"/>
    <xs:import namespace="http://fsb.belgium.be/common/PersonNumber"/>
    <xs:simpleType name="CbeEntityNumber">
        <xs:annotation>
            <xs:documentation>Crossroad Bank of Enterprise number, used to
identify an
                unique enterprise or businessUnit. It is composed by
10
                digits (the first one could be avoided if equal to
0). A
                0 on the first of the 10 digits represents an
unit.</xs:documentation>
                enterprise and a 2 represents a business
        </xs:annotation>
        <xs:restriction base="xs:long"/>
    </xs:simpleType>
    <xs:simpleType name="CbeAddressType">
        <xs:annotation>
            <xs:documentation>type code to identify the address. 001-
>headquarter,
                002->business unit, 003->foreign branch
office</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="CbeDenominationType">
        <xs:annotation>
            <xs:documentation>type code to identify different denomination. 001->
                social denomination, 002 ->abbreviation, 003 ->
                commercial denomination</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
            <xs:enumeration value="001"/>
            <xs:enumeration value="002"/>
            <xs:enumeration value="003"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="CbeStatusCode">

```

```

        <xs:annotation>
            <xs:documentation>enterprise or business unit status code. Example :
                active, juridically created, inactive,
...</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="CbeStopCode">
        <xs:annotation>
            <xs:documentation>Reference to the reason why the entity has been
stopped</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="CbeLinkBusinessType">
        <xs:annotation>
            <xs:documentation>Reference to the link between enterprises (sister,
                mother, ...)</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="CbeEnterpriseType">
        <xs:annotation>
            <xs:documentation>Reference to the type of
enterprise</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
            <xs:enumeration value="ELP">
                <xs:annotation>
                    <xs:documentation>Enterprise Legal Person</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="EPP">
                <xs:annotation>
                    <xs:documentation>Enterprise Physical Person</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="CbeJuridicalFormCode">
        <xs:annotation>
            <xs:documentation>Reference to the juridical form of the enterprise
                (anonymous corporation, ...)</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="CbeJuridicalFormOfficialAbbreviationCode">
        <xs:annotation>
            <xs:documentation>Reference to the</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="CbeJuridicalSituationCode">
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="CbeJuridicalSituationAdministrationCode">
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="CbeActivityVersion">
        <xs:restriction base="xs:string">
            <xs:enumeration value="2003"/>
            <xs:enumeration value="2008"/>
        </xs:restriction>

```

```

</xs:simpleType>
<xs:simpleType name="CbeActivityTypeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="P">
      <xs:annotation>
        <xs:documentation>Principal activity</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="S">
      <xs:annotation>
        <xs:documentation>Secondary activity</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="H">
      <xs:annotation>
        <xs:documentation>Help activity</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="CbePublicationType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="CbePublicationSubject">
  <xs:annotation>
    <xs:documentation>subject of a publication</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="CbeNacebelCode">
  <xs:annotation>
    <xs:documentation>nacebel code used by CBE</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:complexType name="ForeignBranchOfficeType">
  <xs:annotation>
    <xs:documentation>Data's of a foreign branch office, regarding
address and
          validity.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Address" type="addr:addressType"/>
    <xs:element name="ValidityPeriod" type="tns:Period" minOccurs="0"/>
    <xs:element name="StopCode" type="tns:CbeStopCode" minOccurs="0">
      <xs:annotation>
        <xs:documentation>if stopped. CBE code</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="StopDescription" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>if required, Attention, liÃ© Ã l'existence de
attribut.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="EnterpriseDateType">
  <xs:annotation>
    <xs:documentation>Important dates of an enterprise : inscription and
age.</xs:documentation>
  </xs:annotation>

```

```

<xs:sequence>
  <xs:element name="InscriptionDate" type="xs:dateTime" minOccurs="0">
    <xs:annotation>
      <xs:documentation>if available</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="Duration" type="xs:int" minOccurs="0">
    <xs:annotation>
      <xs:documentation>if available, in year</xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="JuridicalForm">
  <xs:annotation>
    <xs:documentation>Juridical Form information</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="FormCode" type="tns:CbeJuridicalFormCode">
      <xs:annotation>
        <xs:documentation>CBE?</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="CodeDescription" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>if required</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="OfficialCodeDescription" type="xs:string"
minOccurs="0">
      <xs:annotation>
        <xs:documentation>if required</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="OfficialAbbreviation"
type="tns:CbeJuridicalFormOfficialAbbreviationCode" minOccurs="0">
      <xs:annotation>
        <xs:documentation>if available</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="ValidityPeriod" type="tns:Period" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="JuridicalSituation">
  <xs:annotation>
    <xs:documentation>Juridical Situation information</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="SituationCode"
type="tns:CbeJuridicalSituationCode"/>
    <xs:element name="CodeDescription" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>if required</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="AdministrationCode"
type="tns:CbeJuridicalSituationAdministrationCode" minOccurs="0"/>
    <xs:element name="ValidityPeriod" type="tns:Period" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DenominationType">
  <xs:annotation>
    <xs:documentation>denomination information</xs:documentation>
  </xs:annotation>

```

```

</xs:annotation>
<xs:sequence>
  <xs:element name="DenominationCode" type="tns:CbeDenominationType"
minOccurs="0">
    <xs:annotation>
      <xs:documentation>CBE code</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="CodeDescription" type="xs:string" minOccurs="0">
    <xs:annotation>
      <xs:documentation>if required</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="Language" type="iso:language" minOccurs="0"/>
  <xs:element name="Value" type="xs:string" minOccurs="0">
    <xs:annotation>
      <xs:documentation>if required</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="ValidityPeriod" type="tns:Period" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="PublicationType">
  <xs:annotation>
    <xs:documentation>publication information</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="ActNumber" type="xs:string"/>
    <xs:element name="Subject" type="tns:CbePublicationSubject">
      <xs:annotation>
        <xs:documentation>CBE code</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="SubjectDescription" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>if required</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Language">
      <xs:annotation>
        <xs:documentation>warning : FSB iso-code</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="iso:language">
            <xs:attribute name="CBEoldLanguageCode">
              <xs:annotation>
                <xs:documentation>deprecated : should be avoided
                                                                    if
possible</xs:documentation>
              </xs:annotation>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="Type" type="tns:CbePublicationType">
      <xs:annotation>
        <xs:documentation>CBE code</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Date" type="xs:dateTime"/>
  </xs:sequence>

```

```

</xs:complexType>
<xs:complexType name="ActivityType">
  <xs:annotation>
    <xs:documentation>activities information, using formally nace
code</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="AdministrationCode" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>CBE code</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="NacebelCode" type="tns:CbeNacebelCode"/>
    <xs:element name="ActivityType" type="tns:CbeActivityTypeType"
minOccurs="0"/>
    <xs:element name="NacebelCodeDescription" type="xs:string"
minOccurs="0">
      <xs:annotation>
        <xs:documentation>if required</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="ValidityPeriod" type="tns:Period" minOccurs="0"/>
    <xs:element name="Version" type="tns:CbeActivityVersion"
minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Period">
  <xs:annotation>
    <xs:documentation>timestamp. for example from September 1, 2006 to
January
                    5, 2007</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Begin" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="End" type="xs:dateTime" minOccurs="0">
      <xs:annotation>
        <xs:documentation>if the entity has been
stopped</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="AuthorizationType">
  <xs:sequence>
    <xs:element name="PermissionCode" type="xs:string"/>
    <xs:element name="RegistrationDateKBO" type="xs:dateTime"
minOccurs="0"/>
    <xs:element name="ValidityPeriod" type="tns:Period" minOccurs="0"/>
    <xs:element name="PhaseCode" type="xs:string" minOccurs="0"/>
    <xs:element name="Duration" type="xs:double" minOccurs="0"/>
    <xs:element name="DistributedAdministrationCode" type="xs:string"
minOccurs="0"/>
    <xs:element name="StoppingCode" type="tns:CbeStopCode"
minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="BankAccount">
  <xs:sequence>
    <xs:element name="BankAccountNumber" type="xs:string" minOccurs="0"/>
    <xs:element name="InstrumentalAdministration" type="xs:string"
minOccurs="0"/>
    <xs:element name="UsagePurposeCode" type="xs:string"/>
    <xs:element name="IBAN" type="xs:string" minOccurs="0"/>

```

```

        <xs:element name="ValidityPeriod" type="tns:Period" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Finances">
    <xs:sequence>
        <xs:element name="ValidityPeriod" type="tns:Period" minOccurs="0"/>
        <xs:element name="FiscalYearEndDay" type="xs:integer"/>
        <xs:element name="FiscalYearEndMonth" type="xs:integer"/>
        <xs:element name="ExceptionalFiscalYearStart" type="xs:date"
minOccurs="0"/>
        <xs:element name="ExceptionalFiscalYearEnd" type="xs:date"
minOccurs="0"/>
        <xs:element name="MonthAnnualMeeting" type="xs:integer"
minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="EntityType">
    <xs:annotation>
        <xs:documentation>Business entity. An entity is either a enterprise
or a
                business unit. Both of these have common fields and
                specifics fields</xs:documentation>
    </xs:annotation>
    <xs:choice>
        <xs:element name="Enterprise" type="tns:EnterpriseType"/>
        <xs:element name="BusinessUnit" type="tns:BusinessUnitType"/>
    </xs:choice>
</xs:complexType>
<xs:complexType name="EntityCommonDataType">
    <xs:annotation>
        <xs:documentation>Domains containing all general information about an
of
                enterprise and/or a business unit. iow: common fields
                a business entity (common between all
realization).</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Status" type="tns:StatusType" minOccurs="0"/>
        <xs:element name="Denomination" type="tns:DenominationType"
minOccurs="0" maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>isBasicDatas</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Address" type="tns:AddressType" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>isBasicDatas</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Activity" type="tns:ActivityType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="Function" type="tns:FunctionType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="LinkedEnterprise" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>enterprise with KBO-known
relationship</xs:documentation>
            </xs:annotation>
        </xs:complexType>
        <xs:complexContent>
            <xs:extension base="tns:LinkedEnterpriseType"/>

```

```

        </xs:complexContent>
    </xs:complexType>
</xs:element>
<xs:element name="Permission" type="tns:AuthorizationType"
minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Validity" type="tns:ValidityType" minOccurs="0">
        <xs:annotation>
            <xs:documentation>isBasicDatas</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="AddressType">
    <xs:annotation>
        <xs:documentation>Specific data's used by CBE for the address of an
entity</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Address" type="addr:addressType" minOccurs="0">
            <xs:annotation>
                <xs:documentation>If the address is removed (for privacy for
address
instance), the details of the former
address
should not be seen, else details are
visible</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="ValidityPeriod" type="tns:Period" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="LinkedEnterpriseType">
    <xs:annotation>
        <xs:documentation>Data's of a linked-enterprise</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Number">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="tns:CbeEntityNumber"/>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
        <xs:element name="LinkType" type="tns:CbeLinkBusinessType">
            <xs:annotation>
                <xs:documentation>CBE code</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="LinkTypeDescription" type="xs:string"
minOccurs="0">
            <xs:annotation>
                <xs:documentation>if required</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="ValidityPeriod" type="tns:Period" minOccurs="0"/>
        <xs:element name="StopCode" type="tns:CbeStopCode" minOccurs="0"/>
        <xs:element name="StopDescription" type="xs:string" minOccurs="0">
            <xs:annotation>
                <xs:documentation>if required</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="IsChild" type="xs:boolean" use="required"/>
</xs:complexType>

```

```

<xs:complexType name="ValidityType">
  <xs:annotation>
    <xs:documentation>Validity of an information by CBE, iow : date and
reason
                                of a closure if occurred.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="ValidityPeriod" type="tns:Period" minOccurs="0"/>
    <xs:element name="StopCode" type="tns:CbeStopCode" minOccurs="0">
      <xs:annotation>
        <xs:documentation>CBE code</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="StopDescription" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>if required</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="EnterpriseType">
  <xs:annotation>
    <xs:documentation>atomic enterprise (entity
realization)</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Number" type="tns:CbeEntityNumber" minOccurs="0"/>
    <xs:element name="Type" type="tns:CbeEnterpriseType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>EPP/EPM. isBasicDatas</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="JuridicalForm" type="tns:JuridicalForm"
minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>isBasicDatas</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="JuridicalSituation" type="tns:JuridicalSituation"
minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ForeignBranchOffice"
type="tns:ForeignBranchOfficeType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>branch office in Belgium for external
society.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="BusinessUnit" type="tns:BusinessUnitType"
minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>should not contain details for sub-
BusinessUnit.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Publication" type="tns:PublicationType"
minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>publication in the Belgian
Monitor</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Date" type="tns:EnterpriseDateType" minOccurs="0">
      <xs:annotation>

```

```

        <xs:documentation>isBasicDatas</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="EntityCommonInfo" type="tns:EntityCommonDataType"
minOccurs="0"/>
    <xs:element name="Capital" type="xs:decimal" minOccurs="0"/>
    <xs:element name="Currency" type="xs:string" minOccurs="0"/>
    <xs:element name="Juridical" type="xs:boolean" minOccurs="0"/>
    <xs:element name="BankAccount" type="tns:BankAccount" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Finances" type="tns:Finances" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="BusinessUnitType">
    <xs:annotation>
        <xs:documentation>atomic business unit (entity
realization)</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Number" type="tns:CbeEntityNumber" minOccurs="0"/>
        <xs:element name="EntityCommonInfo" type="tns:EntityCommonDataType"
minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="StatusType">
    <xs:sequence>
        <xs:element name="StatusCode" type="tns:CbeStatusCode">
            <xs:annotation>
                <xs:documentation>CBE status code isBasicDatas</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="CodeDescription" type="xs:string" minOccurs="0">
            <xs:annotation>
                <xs:documentation>if required</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="functionExecutorPerson">
    <xs:sequence>
        <xs:element name="personNumber" type="person:PersonNumber">
            <xs:annotation>
                <xs:documentation>The personnumber of the person holding the
function</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="FirstName" type="xs:string" minOccurs="0">
            <xs:annotation>
                <xs:documentation>The first name of the person holding the
function</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="LastName" type="xs:string" minOccurs="0">
            <xs:annotation>
                <xs:documentation>The last name of the person holding the
function</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="FunctionType">
    <xs:sequence>

```



```

        </xs:simpleContent>
    </xs:complexType>
</xs:element>
    <xs:element name="PhysicalPersonByName"
type="tns:PhysicalPersonByNameType">
    <xs:annotation>
        <xs:documentation>Should only be used if the identification
            numbers are unknown.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:choice>
</xs:complexType>
<xs:complexType name="PhysicalPersonByNameType">
    <xs:sequence>
        <xs:element name="FirstName" type="xs:string"/>
        <xs:element name="LastName" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="OnBehalfOf" type="tns:CbeEntityNumber">
    <xs:annotation>
        <xs:documentation>If the physical person has a mandate to do such
            function on behalf of a moral person, this
should be
            documented</xs:documentation>
    </xs:annotation>
    </xs:attribute>
</xs:complexType>
<xs:simpleType name="FunctionCodeType">
    <xs:restriction base="xs:string"/>
</xs:simpleType>
</xs:schema>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:iso="http://fsb.belgium.be/common/isocodes"
targetNamespace="http://fsb.belgium.be/common/isocodes"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:annotation>
        <xs:documentation xml:lang="en">Schema version: 1.0 RC3
            Date: 2005-05-04
            Author: Ignaz Wanders</xs:documentation>
    </xs:annotation>
    <xs:simpleType name="language">
    <xs:annotation>
        <xs:documentation xml:lang="en">Language codes follow the ISO-639-1
two-letter code standards.
            For details, see http://www.loc.gov/standards/iso639-2/iso639jac.html

```

Validation rules:  
- two lower case letters

Regular expression: [a-z]{2}

Examples: nl, fr, de, en, es, ...

#### Notes.

1. The xs:language type has a facet ([a-zA-Z]{1,8})(-[a-zA-Z0-9]{1,8})\* which allows more than two characters. For example to denote regional languages.
2. For undetermined languages, ISO reserves the "und". For this reason, the two-letter restriction is not enforced

```

        in the schema.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:language"/>
</xs:simpleType>
<xs:simpleType name="country">
    <xs:annotation>
        <xs:documentation xml:lang="en">Country codes must be in the ISO-3166
two-letter format.
        For a complete list, see
        http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-
code-lists/index.html
        The FSB will translate the ISO code into specific KBO codes, if
required.
        The ISO standards permit certain two-letter codes to be
customized by users. This
        allows for the following codes, which can be used withing the
PROVE application:
        ISO CODE      KBO CODE      meaning
nationalities)      XA              900          stateless (when applied to
nationalities)      XB              901          not yet proven (when applied to
nationalities)      XC              992          moved to abroad (when applied
nationalities)      XD              995          at sea (international waters)
                    XE              999          undetermined

        Validation rules:
        - two upper case letters

        Regular expression: [A-Z]{2}

        Examples: BE, NL, FR, DE, GB, ES, ...
        (Note that UK is not a valid country code!)</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:pattern value="[A-Z]{2}"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="currency">
    <xs:annotation>
        <xs:documentation xml:lang="en">Currency codes must follow the ISO-
4217 code standards. These are three-letter codes
        derived from the ISO-639-1 two-letter country codes. For
details, see
        http://www.bsi-
global.com/British_Standards/currency/index.xalter

        Validation rules:
        - three upper case letters

        Regular expression: [A-Z]{3}

        Examples: EUR, GBP, USD, ...</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:pattern value="[A-Z]{3}"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="bic">
    <xs:annotation>
        <xs:documentation xml:lang="en">The type definition of a "Bank
Identifier Code" (BIC), specified in ISO 9362.

```

For details, see  
[http://www.swift.com/biconline/index.cfm?fuseaction=display\\_aboutbic](http://www.swift.com/biconline/index.cfm?fuseaction=display_aboutbic)

Validation rules:

- The length is 8 or 11 characters.
- First 4 chars are alphabetic and denote the bank code
- 5th and 6th char are an ISO country code
- 7th and 8th char are alphanumeric and denote the region within a country
- 9th - 11th char are the alphanumeric branch code
- A BIC code must be in upper case letters

Regular expression: `[A-Z]{6}[A-Z0-9]{2}([A-Z0-9]{3}){0,1}`

Examples: ABNAFRPP, GEBABEBB04A, ...

```
</xs:annotation>
<xs:restriction base="xs:string">
  <xs:pattern value="[A-Z]{6}[A-Z0-9]{2}([A-Z0-9]{3}){0,1}"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="iban">
  <xs:annotation>
    <xs:documentation xml:lang="en">The type definition of an
international bank account number (IBAN), specified in ISO 13616.
```

Validation rules:

- The length is up to 34 characters
- The first two characters are the ISO two-letter country code
- The 3rd and 4th character are numeric control digits
- The 5th to the last char are alphanumeric
- For the calculation of the control digits:
  - o move the first four chars to the end of the number
  - o convert each alphabetic char in the number to a digit according to a conversion table: A=10, B=11, C=12, ..., Y=34, Z=35
- Note: each letter is converted to two digits, so the number of chars increases
  - o calculate the mod 97 of the full number: it must be equal to one

Regular expression: `[A-Z]{2}\d{2}[A-Z0-9]{1,30}`

Examples: BE68539007547034, FR1420041010050500013M02606, GB29NWBK60161331926819

```
</xs:annotation>
<xs:restriction base="xs:string">
  <xs:pattern value="[A-Z]{2}\d{2}[A-Z0-9]{1,30}"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="gender">
  <xs:annotation>
    <xs:documentation xml:lang="en">The type definition for a gender must
follow the ISO 5128 specification
    The following data items and codes are used
      Not known      0
      Male           1
      Female         2
      Not specified 9</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:integer">
    <xs:maxInclusive value="2"/>
    <xs:minInclusive value="0"/>
```

```

    </xs:restriction>
  </xs:simpleType>
</xs:schema>
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:com="urn://fsb.belgium.be/fphp/v3_0/common"
targetNamespace="urn://fsb.belgium.be/fphp/v3_0/common"
elementFormDefault="qualified" attributeFormDefault="unqualified"
version="1.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">Definition of frequently used common
types.
      [Schema version: 1.0; Date: 2007-07-24; Author: Ignaz
Wanders]</xs:documentation>
    </xs:annotation>
    <xs:simpleType name="PersonNumber">
      <xs:annotation>
        <xs:documentation xml:lang="en">The type definition of the number of
a person.
          It is based on a string to avoid problems with leading zeroes.

          Validation rules:
          - The length is 11 digits, of which the last two are control
digits.
          - The first six digits are the birth date in the format yymmdd.
          - Let num1 = number(0:9) and num2 = number(9:11)
          - Then num2 = 97 - (num1 % 97) for birth dates up to 31/12/1999
            and num2 = 97 - ( (num1 + 2*10^9) % 97 ) for birth dates
later than 31/12/1999

          The modulus and birth date can not be captured in a regular
expression, but the
          basic check on the digits and the length are used in a regular
expression to validate the enterprise number.

          Note that birth dates are not always known, and exceptions to
the birth date rule
          exist. Therefore, the birth date should not be considered in a
validation rule.</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
          <xs:pattern value="\d{11}"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="EnterpriseNumber">
        <xs:annotation>
          <xs:documentation xml:lang="en">The type definition of the number of
an enterprise.
            It is based on a string to avoid problems with leading zeroes.

            Validation rules:
            - The length is 10 digits, of which the last two are control
digits.
            - The first digit is either a zero or a one.
            - Let num1 = number(0:8) and num2 = number(8:10)
            - Then num2 = 97 - (num1 % 97)

            The modulus can not be captured in a regular expression, but
the
            basic check on the digits and the length are used in a regular
expression to validate the enterprise
number.</xs:documentation>
          </xs:annotation>
          <xs:restriction base="xs:string">

```

```

    <xs:pattern value="[01]\d{9}"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="Date">
  <xs:annotation>
    <xs:documentation>Century, Year, Month, Day, based on the definition
of the national register. This type of date does not
necessarily need to be a valid date.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="century" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="18"/>
          <xs:enumeration value="19"/>
          <xs:enumeration value="20"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="year" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="\d{2}"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="month" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="\d{2}"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="day" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="\d{2}"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="EId">
  <xs:annotation>
    <xs:documentation>The electronic identity of
someone.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="certificate" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="Version">
  <xs:annotation>
    <xs:documentation xml:lang="en">The version number identification. A
version indication must be given with
those elements that are sensitive to versioning. Only those
indications are
allowed which are defined here.

There are several conventions, which must be respected by all
parties.

```

Minor versions are by definition backwards compatible. For example, the addition of an optional element is only a minor change; it does not break the contract with clients using an older version. A minor version update does not change the namespace.

A major version is by definition a version which is not backwards compatible. For example, the addition of a required field requires all clients to update their code. A major version update therefore changes the namespace.

Client and server code should be coded such that a major version can coexist with the previous version, at least for a short while.

Note that adding new versions to the enumeration is backwards compatible</xs:documentation>

```

</xs:annotation>
<xs:restriction base="xs:string"/>
</xs:simpleType>
</xs:schema>
<!-- edited with XMLSpy v2007 sp2 (http://www.altova.com) by vander
Schelden Yves-Alexandre (FEDICT) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:adr="http://fsb.belgium.be/prove/address"
xmlns:iso="http://fsb.belgium.be/common/isocodes"
targetNamespace="http://fsb.belgium.be/prove/address"
elementFormDefault="qualified" attributeFormDefault="unqualified">
<xs:annotation>
<xs:documentation xml:lang="en">Schema version: 2.1 Date: 2005-09-27
Author: Ignaz Wanders</xs:documentation>
</xs:annotation>
<xs:import namespace="http://fsb.belgium.be/common/isocodes"/>
<xs:simpleType name="emailType">
<xs:annotation>
<xs:documentation xml:lang="en">The general type definition of an e-
mail address. The
(simplified) regular expression pattern for an e-mail
address that is used is
(\w[\-.\_ \w]*\w@\w[\-.\_ \w]*\w\.\w{2,4})</xs:documentation>
.&_ \w]*\w\.\w{2,4})</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
<xs:pattern value="(\w[\-.\_ \w]*\w@\w[\-.\_ \w]*\w\.\w{2,4})"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="streetCodeType">
<xs:annotation>
<xs:documentation xml:lang="en">StreetCode must be compound of four
digits.</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
<xs:pattern value="\d{4}"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="nisCodeType">
<xs:annotation>
<xs:documentation xml:lang="en">NISCode must be compound of five
digits.</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">

```

```

        <xs:pattern value="\d{5}"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="descriptionType">
    <xs:annotation>
        <xs:documentation xml:lang="en">The type definition of a container
for descriptive
                                properties of an address. These includes names of
                                streets, municipalities, etc.

                                Descriptions are always language-
dependent.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="street" type="xs:string" minOccurs="0">
            <xs:annotation>
                <xs:documentation xml:lang="en">The street
name.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="municipality" type="xs:string" minOccurs="0">
            <xs:annotation>
                <xs:documentation xml:lang="en">The name of the
municipality.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="country" type="xs:string" minOccurs="0">
            <xs:annotation>
                <xs:documentation xml:lang="en">The name of the
country.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="details" type="xs:string" minOccurs="0">
            <xs:annotation>
                <xs:documentation xml:lang="en">Any additional details of the
address.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="language" type="iso:language" minOccurs="0">
            <xs:annotation>
                <xs:documentation xml:lang="en">The language in which any textual
fields are
                                expressed.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="faxnumber" type="xs:string" minOccurs="0"/>
        <xs:element name="cellPhone" type="xs:string" minOccurs="0"/>
        <xs:element name="emailAddress" type="xs:string" minOccurs="0"/>
        <xs:element name="telephoneNumber" type="xs:string" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="usageType">
    <xs:annotation>
        <xs:documentation xml:lang="en">The address usage. Used as a base
type.</xs:documentation>
    </xs:annotation>
</xs:complexType>
<xs:complexType name="kbo-usageType">
    <xs:annotation>
        <xs:documentation xml:lang="en">The address usage within the
KBO.</xs:documentation>
    </xs:annotation>
</xs:complexType>

```

```

    <xs:extension base="adr:usageType">
      <xs:sequence>
        <xs:element name="address-type" type="xs:string">
          <xs:annotation>
            <xs:documentation xml:lang="en">The type of
address</xs:documentation>
            <xs:documentation xml:lang="nl">Het type van het adres.
Voorbeelden: 001
                                                Adres Maatschappelijke
Zetel / Domicilie
                                                002 Adres
Vestigingseenheid 003 Bijhuis
                                                in België« voor
buitenlandse onderneming</xs:documentation>
            <xs:documentation xml:lang="fr">Example: 001 Adresse du
siège social /
                                                du Domicile 002 Adresse
d'unité
                                                d'établissement 003
Succursale d'une
                                                entreprise
@trangère</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="description" type="xs:string" minOccurs="0">
          <xs:annotation>
            <xs:documentation xml:lang="en">The description of the type
of address</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="base-addressType">
  <xs:annotation>
    <xs:documentation xml:lang="en">The address type defines basic
properties that are valid
for all addresses, such as house number, postbox, and
postcode.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="house-number" type="xs:string">
      <xs:annotation>
        <xs:documentation xml:lang="en">The house number. This is a
mandatory field. If
no house number is known, the letters
ZN (Dutch:
"zonder naam") should be used.
Note: due to a limitation in the
KBO/BCE, house
numbers may be truncated to four
characters, if
the KBO/BCE is involved in a server
transaction.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="postbox" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">The postbox is an optional
field.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>

```

```

</xs:element>
<xs:element name="postcode" type="xs:string">
  <xs:annotation>
    <xs:documentation xml:lang="en">The postcode. This is a mandatory
field.</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="country-code">
  <xs:annotation>
    <xs:documentation xml:lang="en">The ISO country code of the
address. This field
                                is mandatory.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="iso:country">
        <xs:attribute name="CBEoldCountryCode" type="xs:string">
          <xs:annotation>
            <xs:documentation>deprecated : should be avoided
                                if
possible</xs:documentation>
          </xs:annotation>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="description" type="adr:descriptionType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="kboUsage" type="adr:kbo-usageType" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="belgian-addressType">
  <xs:annotation>
    <xs:documentation xml:lang="en">The belgian-addressType is an address
and adds a few
                                codes which uniquely define a Belgian address. These
                                codes are NIS codes for municipalities, and street
                                codes
                                for streets. The use of codes makes the address
                                independent of language.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="adr:base-addressType">
      <xs:sequence>
        <xs:element name="streetcode" type="adr:streetCodeType"
minOccurs="0">
          <xs:annotation>
            <xs:documentation xml:lang="en">The street
code.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="niscode" type="adr:nisCodeType" minOccurs="0">
          <xs:annotation>
            <xs:documentation xml:lang="en">The NIS code of the
municipality.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="foreign-addressType">

```

```

    <xs:annotation>
      <xs:documentation xml:lang="en">A foreign address is an address
without using codes for
      municipalities and streets, we are forced to use
      descriptive text for those fields.</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="adr:base-addressType">
        <xs:sequence>
          <xs:element name="state" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
<xs:complexType name="addressType">
  <xs:choice>
    <xs:element name="base-address" type="adr:base-addressType"/>
    <xs:element name="belgian-address" type="adr:belgian-addressType"/>
    <xs:element name="foreign-address" type="adr:foreign-addressType"/>
  </xs:choice>
</xs:complexType>
</xs:schema>
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:per="http://fsb.belgium.be/common/PersonNumber"
targetNamespace="http://fsb.belgium.be/common/PersonNumber"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation xml:lang="en">Schema version: 1.0 Date: 2007-05-09
Author: Ignaz Wanders
      Modified by Yves-Alexandre vander
Schelden</xs:documentation>
  </xs:annotation>
  <xs:simpleType name="PersonNumber">
    <xs:annotation>
      <xs:documentation xml:lang="en">The type definition of the number of
a person. It is
      based on a string to avoid problems with leading
      zeroes.

      Validation rules: - The length is 11 digits, of which
      the last two are control digits. - The first six
      digits
      are the birth date in the format yymmdd. - Let num1 =
      number(0:9) and num2 = number(9:11) - Then num2 = 97
      -
      (num1 % 97) for birth dates up to 31/12/1999 and num2
      =
      97 - ( (num1 + 2*10^9) % 97 ) for birth dates later
      than
      31/12/1999

      The modulus and birth date can not be captured in a
      regular expression, but the basic check on the digits
      and the length are used in a regular expression to
      validate the enterprise number.

      Note that birth dates are not always known, and
      exceptions to the birth date rule exist. Therefore,
      the
      birth date should not be considered in a validation
      rule.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">

```

```

        <xs:pattern value="\d{11}"/>
    </xs:restriction>
</xs:simpleType>
</xs:schema>

</types>
<message name="findSoapIn">
    <part name="parameters" element="s0:find"/>
</message>
<message name="findSoapOut">
    <part name="parameters" element="s0:findResponse"/>
</message>
<portType name="enterpriseSoap">
    <operation name="find">
        <input message="s0:findSoapIn"/>
        <output message="s0:findSoapOut"/>
    </operation>
</portType>
<binding name="enterpriseSoap" type="s0:enterpriseSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document"/>
    <operation name="find">
        <soap:operation soapAction="http://www.openuri.org/find"
style="document"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
</binding>
<service name="enterprise">
    <port name="enterpriseSoap" binding="s0:enterpriseSoap">
        <soap:address
location="http://vilcanota:7002/Fent/1.0/ws/enterprise.jws"/>
    </port>
</service>
</definitions>

```