

```

<?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://fsb.belgium.be/prove"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
targetNamespace="http://fsb.belgium.be/prove">
  <types>
    <s:schema xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:fmo="http://fsb.belgium.be/prove/fmop100"
xmlns:fmo2="http://fsb.belgium.be/prove/fmop290" elementFormDefault="qualified"
targetNamespace="http://fsb.belgium.be/prove">
      <s:import namespace="http://fsb.belgium.be/prove/fmop100"/>
      <s:import namespace="http://fsb.belgium.be/prove/fmop290"/>
      <s:element name="ping">
        <s:complexType>
          <s:sequence/>
        </s:complexType>
      </s:element>
      <s:element name="pingResponse">
        <s:complexType>
          <s:sequence>
            <s:element name="pingResult" type="s:string" minOccurs="0"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="testSOAPFault">
        <s:complexType>
          <s:sequence/>
        </s:complexType>
      </s:element>
      <s:element name="testSOAPFaultResponse">
        <s:complexType>
          <s:sequence/>
        </s:complexType>
      </s:element>
      <s:element name="fmop100">
        <s:complexType>
          <s:sequence>
            <s:element ref="fmo:fmop100"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="fmop100Response">
        <s:complexType>
          <s:sequence>
            <s:element ref="fmo2:fmop290"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
    <!-- edited with XML Spy v4.2 U (http://www.xmlspy.com) by Paul Stijfhals
(Recherche) -->
    <!-- edited with XMLSPY v2004 rel. 4 U (http://www.xmlspy.com) by De Baets
(de baets) -->
    <xs:schema xmlns:ent="http://fsb.belgium.be/prove/enterprise"
xmlns:not="http://fsb.belgium.be/prove/notary"
xmlns="http://fsb.belgium.be/prove/fmop100"

```

```

xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://fsb.belgium.be/prove/fmop100"
elementFormDefault="qualified">
  <xs:import namespace="http://fsb.belgium.be/prove/notary"/>
  <xs:import namespace="http://fsb.belgium.be/prove/enterprise"/>
  <xs:annotation>
    <xs:documentation xml:lang="en">Schema version: 1.2
      Date: 2005-07-20
      Author: Yves Jonckheere</xs:documentation>
  </xs:annotation>
  <xs:element name="fmop100">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="not:notary"/>
        <xs:element name="client_application_id" type="xs:string"
minOccurs="0" maxOccurs="1"/>
        <xs:element name="enterprise">
          <xs:complexType>
            <xs:choice>
              <xs:element name="enterprise_number"
type="ent:enterprise_number"/>
              <xs:element name="enterprise_extern_number"
type="xs:string"/>
              <xs:sequence>
                <xs:element name="enterprise_name" type="xs:string"
minOccurs="0"/>
                <xs:element name="enterprise_juridical_form"
type="xs:string" minOccurs="0"/>
                <xs:element name="enterprise_street_code" type="xs:string"
minOccurs="0"/>
                <xs:element name="enterprise_street_number"
type="xs:string" minOccurs="0"/>
                <xs:element name="enterprise_post_code" type="xs:string"
minOccurs="0"/>
              </xs:sequence>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
  <!-- edited with XML Spy v4.2 U (http://www.xmlspy.com) by Paul
Stijfhals (Recherche) -->
  <!-- edited with XMLSPY v2004 rel. 4 U (http://www.xmlspy.com) by De Baets
(de baets) -->
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://fsb.belgium.be/prove/fmop290"
xmlns:iso="http://fsb.belgium.be/common/isocodes"
xmlns:ent="http://fsb.belgium.be/prove/enterprise"
targetNamespace="http://fsb.belgium.be/prove/fmop290"
elementFormDefault="qualified">
    <xs:import namespace="http://fsb.belgium.be/prove/enterprise"/>
    <xs:import namespace="http://fsb.belgium.be/common/isocodes"/>
    <xs:annotation>
      <xs:documentation xml:lang="en">Schema version: 1.2
        Date: 2005-07-20
        Author: Yves Jonckheere</xs:documentation>
    </xs:annotation>
    <xs:element name="fmop290">
      <xs:complexType>
        <xs:sequence>
          <xs:choice>

```

```

        <xs:element name="enterprise_found" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="enterprise_number"
type="ent:enterprise_number"/>
                    <xs:element name="enterprise_status_code"
type="xs:string"/>
                    <xs:element name="enterprise_juridical_form_code"
type="xs:string"/>
                    <xs:element name="enterprise_names" maxOccurs="unbounded">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="enterprise_name_language_code"
type="iso:language"/>
                                <xs:element name="enterprise_denom_begin_date"
type="xs:date" minOccurs="0"/>
                                <xs:element name="enterprise_denom_end_date"
type="xs:date" minOccurs="0"/>
                                <xs:choice>
                                    <xs:element name="enterprise_social_name"
type="xs:string"/>
                                    <xs:element name="enterprise_commercial_name"
type="xs:string"/>
                                    <xs:element name="enterprise_abbreviation"
type="xs:string"/>
                                </xs:choice>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="enterprise_address"
maxOccurs="unbounded">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="enterprise_address_type"
type="xs:string"/>
                                <xs:element name="enterprise_address_street_code"
type="xs:string" minOccurs="0"/>
                                <xs:element name="enterprise_address_street_name"
type="xs:string" minOccurs="0"/>
                                <xs:element name="enterprise_address_number"
type="xs:string" minOccurs="0"/>
                                <xs:element name="enterprise_address_box"
type="xs:string" minOccurs="0"/>
                                <xs:element name="enterprise_address_city"
type="xs:string" minOccurs="0"/>
                                <xs:element name="enterprise_address_postcode"
type="xs:string" minOccurs="0"/>
                                <xs:element name="enterprise_address_niscode"
type="xs:string" minOccurs="0"/>
                                <xs:element name="enterprise_address_state"
type="xs:string" minOccurs="0"/>
                                <xs:element name="enterprise_address_country_code"
type="iso:country" minOccurs="0"/>
                                <xs:element name="enterprise_address_details"
type="xs:string" minOccurs="0"/>
                                <xs:element name="enterprise_address_phone_number"
type="xs:string" minOccurs="0"/>
                                <xs:element name="enterprise_address_fax_number"
type="xs:string" minOccurs="0"/>
                                <xs:element name="enterprise_address_email"
type="xs:string" minOccurs="0"/>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>

```

```

        </xs:element>
        <xs:element name="enterprise_capital" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:decimal">
              <xs:minExclusive value="0"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="enterprise_capital_monetary_unit"
type="ent:kbo_monetary_unit" minOccurs="0"/>
        <xs:element name="enterprise_begin_date" type="xs:date"
minOccurs="0"/>
        <xs:element name="enterprise_end_date" type="xs:date"
minOccurs="0"/>
        <xs:element name="enterprise_stop_code" type="xs:string"
minOccurs="0"/>
        <xs:element name="enterprise_duration" type="xs:string"
minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="error">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="id" type="xs:string"/>
        <xs:element name="code" type="xs:string"/>
        <xs:element name="description" type="xs:string"/>
        <xs:element name="suggested_action" type="xs:string"
minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:choice>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ent="http://fsb.belgium.be/prove/notary"
xmlns:per="http://fsb.belgium.be/prove/person"
xmlns:ent="http://fsb.belgium.be/prove/enterprise"
xmlns:iso="http://fsb.belgium.be/common/isocodes"
targetNamespace="http://fsb.belgium.be/prove/notary"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation xml:lang="en">Schema version: 1.1
      Date: 2005-07-26
      Author: Ignaz Wanders</xs:documentation>
  </xs:annotation>
  <xs:import namespace="http://fsb.belgium.be/common/isocodes"/>
  <xs:import namespace="http://fsb.belgium.be/prove/person"/>
  <xs:import namespace="http://fsb.belgium.be/prove/enterprise"/>
  <xs:element name="notary">
    <xs:annotation>
      <xs:documentation xml:lang="en">The general identification of a
notary.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="office_id" type="xs:string">
          <xs:annotation>
            <xs:documentation xml:lang="en">The general ID of the notary.
It is this ID that the FSB will use

```

```

        to identify a notary.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="lang" type="iso:language">
    <xs:annotation>
        <xs:documentation xml:lang="en">The language in which the
notary prefers to receive answers.
        Note: KBO/BCE currently only recognizes "nl" and
"fr" as valid
        languages.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="nrn" type="per:person_number">
    <xs:annotation>
        <xs:documentation xml:lang="en">The national registry number of
the notary.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="num_kbo_not" type="ent:enterprise_number">
    <xs:annotation>
        <xs:documentation xml:lang="en">The KBO/BCE enterprise number
of the notary.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="num_kbo_fed" type="ent:enterprise_number">
    <xs:annotation>
        <xs:documentation xml:lang="en">The KBO/BCE enterprise number
of the federation of notaries.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ent="http://fsb.belgium.be/prove/enterprise"
xmlns:iso="http://fsb.belgium.be/common/isocodes"
xmlns:adr="http://fsb.belgium.be/prove/address"
xmlns:per="http://fsb.belgium.be/prove/person"
targetNamespace="http://fsb.belgium.be/prove/enterprise"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:annotation>
        <xs:documentation xml:lang="en">version 1.2: added an optional field
enterprise number to the functions in case a person
        represents an enterprise. The schema is backwards
compatible.
    </xs:documentation>
    Schema version: 1.1
    Date: 2005-07-26
    Author: Ignaz Wanders</xs:documentation>
</xs:annotation>
<xs:import namespace="http://fsb.belgium.be/common/isocodes"/>
<xs:import namespace="http://fsb.belgium.be/prove/address"/>
<xs:import namespace="http://fsb.belgium.be/prove/person"/>
<xs:simpleType name="enterprise_number">
    <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.
    </xs:documentation>
    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:simpleType name="kbo_monetary_unit">
  <xs:annotation>
    <xs:documentation xml:lang="en">The monetary unit (or currency) the
capital of an enterprise is expressed in.
    It is expressed using an KBO code for
currency.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:pattern value="[A-Z]{2}|[A-Z]{3}"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="contactType">
  <xs:annotation>
    <xs:documentation xml:lang="en">Contact information for an enterprise
consists of an e-mail address,
    a phone number, and a fax number. This format is
based on the KBO
    properties within an "address".</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="fax-number" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">An optional facsimile number for
the address.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="phone-number" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">An optional phone number for the
address.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="email" type="adr:emailType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">An optional e-mail address for
the address.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:element name="soc_juridical_form">
  <xs:annotation>
    <xs:documentation xml:lang="en">The enterprise's juridical form, as
defined using KBO/BCE codes.
    The codes are three-digit codes, and language independent.
    Examples: 014 = SA (fr) or NV (nl)
              015 = SPRL (fr) or BVBA (nl)</xs:documentation>
  </xs:annotation>
</xs:simpleType>

```

```

        <xs:restriction base="xs:string">
            <xs:pattern value="\d{3}"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="soc_capital">
    <xs:annotation>
        <xs:documentation xml:lang="en">The capital of an enterprise.
            This should be accompanied by a monetary
unit.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:decimal">
            <xs:minExclusive value="0"/>
            <xs:totalDigits value="15"/>
            <xs:fractionDigits value="2"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="monetary_unit" type="ent:kbo_monetary_unit">
    <xs:annotation>
        <xs:documentation xml:lang="en">The monetary unit (or currency) the
capital of an enterprise is expressed in.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="soc_duration" type="xs:positiveInteger">
    <xs:annotation>
        <xs:documentation xml:lang="en">The duration of... (?), expressed in
years</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="soc_name">
    <xs:annotation>
        <xs:documentation xml:lang="en">The name of an enterprise:
            full name    maps onto code 001 in the KBO
            short name   maps onto code 002 in the KBO
            abbreviation maps onto code 003 in the KBO
            A name is language-dependent.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="soc_name_language_code" type="iso:language">
                <xs:annotation>
                    <xs:documentation xml:lang="en">The language of the enterprise
name</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="soc_name_full" type="xs:string">
                <xs:annotation>
                    <xs:documentation xml:lang="en">The full name of the
enterprise.
                        Corresponds to the code 001 in the
KBO/BCE.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="soc_name_short" type="xs:string" minOccurs="0">
                <xs:annotation>
                    <xs:documentation xml:lang="en">The short name of the
enterprise.
                        Corresponds to the code 002 in the
KBO/BCE.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

        <xs:element name="soc_acronym" type="xs:string" minOccurs="0">
          <xs:annotation>
            <xs:documentation xml:lang="en">The abbreviation of the
enterprise.
                Corresponds to the code 003 in the
KBO/BCE.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="day">
    <xs:annotation>
      <xs:documentation xml:lang="en">The day of a month. This field must
be between 1 and 31.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="31"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="month">
    <xs:annotation>
      <xs:documentation xml:lang="en">The month of a year. This field must
be between 1 and 12.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:positiveInteger">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="12"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="year">
    <xs:annotation>
      <xs:documentation xml:lang="en">The year, including century
indication.
                Only years between 1800 and 2199 are accepted. These boundaries
are set
                arbitrarily and are only used to check for accidental major
typos.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:positiveInteger">
        <xs:minInclusive value="1800"/>
        <xs:maxInclusive value="2199"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="soc_exercise_end">
    <xs:annotation>
      <xs:documentation xml:lang="en">The end of the accounting year is the
same for each year: only day
                and month required. There is no validation that the day within
that
                month is a valid date. Thus 30th February is accepted by the
schema.</xs:documentation>
    </xs:annotation>
  </xs:complexType>
</xs:sequence>

```

```

        <xs:element ref="ent:day"/>
        <xs:element ref="ent:month"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="begin_exceptional_exercise" type="xs:date">
    <xs:annotation>
        <xs:documentation xml:lang="en">The begin date of the exceptional
exercise.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="end_exceptional_exercise" type="xs:date">
    <xs:annotation>
        <xs:documentation xml:lang="en">The end date of the exceptional
exercise.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="soc_assembly_date">
    <xs:annotation>
        <xs:documentation xml:lang="en">The annual assembly should take place
in a certain month of the year.
            The day within the month is not required. The year is not
required.
            If the day and/or year are provided, there is no validation
that it is
            a valid date. Thus 30th February 2005 is accepted by the
schema.</xs:documentation>
    </xs:annotation>
<xs:complexType>
    <xs:sequence>
        <xs:element ref="ent:month"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="function">
    <xs:annotation>
        <xs:documentation xml:lang="en">People or enterprises can be
registered with an enterprise having a certain
            function.</xs:documentation>
    </xs:annotation>
<xs:complexType>
    <xs:sequence>
        <xs:element name="code">
            <xs:annotation>
                <xs:documentation xml:lang="en">The code of the function.
                    Function codes are five digits long, and identified
as "quality"
                    within a function.
                    Examples are: 00001 Founder (Fondateur/Oprichter)
                    10002 Administrator
(Administrateur/Bestuurder)</xs:documentation>
            </xs:annotation>
<xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:pattern value="\d{5}"/>
            </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="dt_begin" type="xs:date">
    <xs:annotation>
        <xs:documentation xml:lang="en">The start date of the validity
of the function.</xs:documentation>
    </xs:annotation>
</xs:element>

```

```

        </xs:annotation>
    </xs:element>
    <xs:element name="dt_end" type="xs:date" minOccurs="0">
        <xs:annotation>
            <xs:documentation xml:lang="en">The end date of the validity of
the function. This field is optional.
                If the field is given, a corresponding stop code
must be provided.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="stop_code" minOccurs="0">
        <xs:annotation>
            <xs:documentation xml:lang="en">The stop code for a function is
mandatory if an end date is given.
                the stop code defines the reason for ending the
function.
                Stop codes are either a one-letter or a three-digit
code
                Example stop codes are: A, B, C, D, E, F, X, 003,
004, 005, 010, ...</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:pattern value="[A-Z]"/>
                <xs:pattern value="\d{3}"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:choice>
        <xs:annotation>
            <xs:documentation xml:lang="en">The function can be held by
either a person, or another enterprise</xs:documentation>
        </xs:annotation>
        <xs:element name="person">
            <xs:annotation>
                <xs:documentation xml:lang="en">This element is used when the
function is held by a person</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="nrn" type="per:person_number">
                        <xs:annotation>
                            <xs:documentation xml:lang="en">The national registry
number of the person.</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    <xs:element name="first_name" type="xs:string">
                        <xs:annotation>
                            <xs:documentation xml:lang="en">The first name of the
person.</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    <xs:element name="last_name" type="xs:string">
                        <xs:annotation>
                            <xs:documentation xml:lang="en">The last name of the
person.</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    <xs:element name="contact" type="ent:contactType"
minOccurs="0">
                        <xs:annotation>
                            <xs:documentation xml:lang="en">Information of contact
person.</xs:documentation>

```

```

        </xs:annotation>
    </xs:element>
    <xs:choice>
        <xs:element name="belgian-address" type="adr:belgian-
addressType">
            <xs:annotation>
                <xs:documentation xml:lang="en">The address of the
person.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="foreign-address" type="adr:foreign-
addressType">
            <xs:annotation>
                <xs:documentation xml:lang="en">The address of the
person.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:choice>
    <!-- added 20070131 by IW: person holding function
represents a company -->
    <xs:element name="enterprise_num"
type="ent:enterprise_number" minOccurs="0">
        <xs:annotation>
            <xs:documentation xml:lang="en">The enterprise number
if the person represents a company.</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="enterprise">
    <xs:annotation>
        <xs:documentation xml:lang="en">This element is used when the
function is held by an enterprise</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="enterprise_num"
type="ent:enterprise_number">
                <xs:annotation>
                    <xs:documentation xml:lang="en">The enterprise
number.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="social_name" type="xs:string">
                <xs:annotation>
                    <xs:documentation xml:lang="en">The name of the
enterprise.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:choice>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
    <!-- edited with XML Spy v4.0.1 U (http://www.xmlspy.com) by Paul
Stijfhals (Recherche) -->
    <xs:schema xmlns:iso="http://fsb.belgium.be/common/isocodes"
xmlns:xs="http://www.w3.org/2001/XMLSchema"

```

```

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
    </xs:documentation>
  </xs:annotation>

```

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 three-letter code "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
XA	900	stateless (when applied to nationalities)
XB	901	not yet proven (when applied to nationalities)
XC	992	moved to abroad (when applied to 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

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:documentation>
 </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:documentation>

```
</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:per="http://fsb.belgium.be/prove/person"
  targetNamespace="http://fsb.belgium.be/prove/person"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation xml:lang="en">Schema version: 1.0 RC1
      Date: 2005-04-08
      Author: Ignaz Wanders</xs:documentation>
  </xs:annotation>
  <xs:simpleType name="person_number">
    <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:annotation>
<xs:restriction base="xs:string">
  <xs:pattern value="\d{11}"/>
</xs:restriction>
</xs:simpleType>
</xs:schema>
<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: 2.2
      Date: 2006-12-19
      Author: Ignaz Wanders

      Changes in version:
      2.2 removed the four-digit constraint from the streetcode.
      (apparently, alphanumeric streetcodes exist)</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>
    </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">
      <!--
      restriction removed to allow alphanumeric streetcodes
      -->
      <xs:pattern value="\w+"/>
    </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: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:complexContent>
        <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 Bijkuis in België voor buitenlandse
onderneming</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
    <xs:annotation>
        <xs:documentation xml:lang="fr">Example:

```

	001	Adresse du siège social / du Domicile
	002	Adresse d'unité d'établissement
	003	Succursale d'une entreprise

```

</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: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" type="iso:country">
      <xs:annotation>
        <xs:documentation xml:lang="en">The ISO country code of the
address. This field is mandatory.</xs:documentation>
      </xs:annotation>
    </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

```

address. and adds a few codes which uniquely define a Belgian
street codes These codes are NIS codes for municipalities, and
independent for streets. The use of codes makes the address
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>

```

```

</types>
<message name="pingSoapIn">
  <part name="parameters" element="s0:ping"/>
</message>
<message name="pingSoapOut">
  <part name="parameters" element="s0:pingResponse"/>
</message>
<message name="testSOAPFaultSoapIn">
  <part name="parameters" element="s0:testSOAPFault"/>
</message>

```

```

<message name="testSOAPFaultSoapOut">
  <part name="parameters" element="s0:testSOAPFaultResponse"/>
</message>
<message name="fmop100SoapIn">
  <part name="parameters" element="s0:fmop100"/>
</message>
<message name="fmop100SoapOut">
  <part name="parameters" element="s0:fmop100Response"/>
</message>
<portType name="findEnterpriseSoap">
  <operation name="ping">
    <input message="s0:pingSoapIn"/>
    <output message="s0:pingSoapOut"/>
  </operation>
  <operation name="testSOAPFault">
    <input message="s0:testSOAPFaultSoapIn"/>
    <output message="s0:testSOAPFaultSoapOut"/>
  </operation>
  <operation name="fmop100">
    <input message="s0:fmop100SoapIn"/>
    <output message="s0:fmop100SoapOut"/>
  </operation>
</portType>
<binding name="findEnterpriseSoap" type="s0:findEnterpriseSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document"/>
  <operation name="ping">
    <soap:operation soapAction="http://fsb.belgium.be/prove/ping"
style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="testSOAPFault">
    <soap:operation soapAction="http://fsb.belgium.be/prove/testSOAPFault"
style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="fmop100">
    <soap:operation soapAction="http://fsb.belgium.be/prove/fmop100"
style="document"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
<service name="findEnterprise">
  <port name="findEnterpriseSoap" binding="s0:findEnterpriseSoap">
    <soap:address
location="http://fsb.belgium.be:80/fmop/1.0/ws/findEnterprise.jws"/>
  </port>
</service>

```

</definitions>