GUIDELINES FOR XML EDI EXCHANGES

TECHNICAL DOCUMENT

VERSION 1.2 – 28 FEBRUARY 2006

GENERAL DIRECTORATE OF CUSTOMS AND EXCISE

SUB-DIRECTORATE C

Translation for information purposes only; in the event of any discrepancy or difference of interpretation, the French version shall prevail.

RECORD OF CHANGES

Date	Amended Topics	Comments
12/04/2005	All	
27/02/2006	All, harmonisation	Version 1.2

Contents

1. Introduction

This document is for Customs' EDI partners' development departments. Herein they will find all the specifications they need to process EDI exchanges.

2. Exchanges with EDI Partners, Guidelines and Definitions

2.1. Definitions

- *Operator:*

A compagny with one or several business relations with Customs

- *Connection Provider:*

A business providing a technical service handling the EDI routing of an operator's functional messages to an EDI window of Customs' e-procedures

An operator may handle these operations without resorting to the services of a connection provider.

- *Software Publisher:*

A business publishing software comprising a functionality enabling the exchange of EDI messages with one or several Customs e-procedures

- *EDI Partner:*

The terms apply to an operator using EDI to submit declarations to Customs, an EDI connection provider or a software publisher.

- *Interchange Agreement*

A special EDI mode agreement assigned to EDI partners once the connection with the EDI window is done, and to each EDI software certification. The agreement contains the email addresses where Customs must send its technical messages for the different procedures (business agreement). If a provider uses several different addresses for the same procedure, provider will have several agreements.

The Interchange Agreement identifier in the connection envelope is the element <InterchangeAgreementId>.

- *Technical Message*

In the document, the technical message is a message in the 'container' sense: it is the exchange unit between a connection provider and the DGDDI (General Directorate of Customs and Excise). A technical message has a **connection envelope**.

- *Functional Message*

In the document, the functional message is a message in the application sense (a simplified declaration, for instance). The functional message has a **message envelope**.

- *Transaction Identifier*

This is a unique mandatory number managed by the **EDI partner**. The number makes it possible to locate all the incoming or outgoing functional messages under the same reference, leading to changes of state of the same object, ranging from its initial state (creation) to its end state. For instance, the transaction identifier of the functional message creating an early SID (simplified import declaration) will be the same as the identifier for the functional message validating the same SID.

Customs e-procedures will not generate any transaction identifiers or sequence numbers. When an e-procedure sends a functional message to an EDI partner, it will refer to the partner's transaction number.

- *Sequence Number*

The sequence number gives the order of a functional message in a transaction, thus enabling the EDI messaging system (MAREVA) to deliver functional messages in the right order to the e-procedure (viz. the functional validation message must not be delivered before the functional creation message). The operator assigns the sequence number. Customs does not provide a number in the envelope of its functional messages transmitted to operators.

The sequence number must begin with zero.

To ensure optimum operations for exchanges with EDI partners, Customs will set up mechanisms to identify any operators and connection providers involved in an exchange so that exchanges are properly routed. The mechanisms will be implemented in a dedicated EDI messaging system (MAREVA) thanks to the information provided in the connection and message envelopes.

2.2. Transaction Kinetics

OUTGOING is from the EDI partner to Customs.

INCOMING is from Customs to the EDI partner.

The successful integration of an OUTGOING message will systematically generate an INCOMING message to the sender, i.e. a functional receipt acknowledgment.

A failure will generate a notification of an INCOMING message notifying the rejection of the outgoing message.

2.3. Implementation

MAREVA is a messaging system processing EDI application messages to all Customs e-procedures from EDI partners. MAREVA handles all EDI exchanges to and from a Customs e-procedure.

3. XML Modelling

3.1. Schema Construction Guidelines

XML schemas are used for modelling**.** There are two types of schemas:

Generic schemas for describing all kinds of application messages (reusable bricks)

Specific schemas for describing application messages

The document only deals with generic schemas.

A generic schema describes the basic types that are the basis for assigning types to the elementary data. In the basic types, the definitions of the identifiers and the codes stored in the reference tables or special to the e-procedure can also be found.
A generic schema describes the elements that are used to describe the simple structures involved in message composition.
A generic schema describes the components used to describe the complex structures involved in message composition.

A generic schema describes the parameters of the connection envelope.

A generic schema describes the parameters of the message envelope.

For each type of application message:

A schema describes the **structure** of the application message

Complex messages might be broken down into several schemas

3.1.1 The 'TypeBase' Schema

It only contains simple structures (simpleType):

Type Base	Definition
ShortText	A string with a maximum of 35 characters
LongText	A string with a maximum of 260 characters
Date	Date
Time	Time
Indicator	Boolean indicator
Amount	Monetary amount expressed (explicitly or not) in a currency, without decimals
DecimalAmount	Monetary amount expressed (explicitly or not) in a currency, with decimals
Measure	Numeric information determined by calculation or counting combined with a measurement unit
Quantity	Number of non-monetary units
Rate	Currency rate: expresses an amount in the currency compared to the reference currency (euro) For instance, 1 euro equals 1.31459 US dollars
Quota	Quota applied to a tax

Percent Percentage

Excerpt

```
<xs:simpleType name="MontantDecimal">
<xs:annotation>
<xs:documentation>monetary amount with decimals</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:decimal">
<xs:totalDigits value="18"/>
<xs:fractionDigits value="2"/>
</xs:restriction>
</xs:simpleType>
```

The “BasicType” schema also contains types of reference codes and identifiers.

Example 1: description of an agreement (data referring to a repository table)

```
<xs:simpleType name="RefAgrement">
<xs:annotation>
<xs:documentation>Agreement identifier</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
<xs:maxLength value="12"/>
</xs:restriction>
</xs:simpleType>
```

Example 2: description of an identifier (identifier of a PDI declaration)

```
<xs:simpleType name="IdentifiantDec">
<xs:annotation>
<xs:documentation>Declaration Identifier</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
<xs:maxLength value="22"/>
</xs:restriction>
</xs:simpleType>
```

3.1.2 The ‘DouaneElements’ Schema

It describes the simple elements used to build the message. The schema only contains simple structures (simpleType). It refers to the TypeBase schema (instruction “include”).

Excerpt

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:include schemaLocation="TypeBase.xsd"/>
<xs:element name="refdsi" type="IdentifiantDec">
<xs:annotation>
<xs:documentation>SID declaration reference</xs:documentation>
</xs:annotation>
</xs:element>
```

3.1.3 The “Composants” Schema

It describes the components used to build the messages. The schema only contains complex structures (complexType). It refers to the DouaneElements schema (instruction “include”).

Excerpt

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:include schemaLocation="DouaneElements.xsd"/>
<!--Documents-->
<xs:complexType name="TDocuments">
<xs:sequence maxOccurs="unbounded">
```

```
<xs:element name="Document" type="TDocument"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="TDocument">
<xs:sequence>
<xs:element name="doc" type="RefDoc"/>
<xs:element name="refdoc" type="TexteCourt"/>
<xs:element name="datdoc" type="Date"/>
<xs:element name="indd48" type="Indicateur"/>
<xs:element name="mntd48" type="Montant" minOccurs="0"/>
<xs:element name="deld48" minOccurs="0">
<xs:simpleType>
<xs:restriction base="Quantite">
<xs:totalDigits value="2"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
```

3.1.4 The 'ParametresConnexion' Schema

It describes the structure common to all the messages including the technical parameters required for the proper routing of the messages to the MAREVA messaging system.

Excerpt

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:complexType name="TEnveloppeConnexion">
```

3.1.5 The 'ParametresMessage' Schema

It describes the structure common to all the messages where all the parameters common to all the application messages are included.

Excerpt

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:complexType name="TEnveloppeMessage">
```

3.2. Conventions

3.2.1. The Set of Characters

The default set of characters is UTF-8. The set is not appropriate for Latin characters. The option is to use the ISO-8859-1 set of characters.

Example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

3.2.2. The Number of the XML Version

The **version** attribute indicates the version of the XML language that is used. It has the value 1.0.

```
<xs:element name="URL" type="xs:anyURI"/>
```

3.2.3. Naming the Schemas

When several names are combined to make the name of one schema, the first letter of each name must be written with a capital letter.

Message**P**arametres
Message**D**si

3.2.4. Naming and Organising Complex Structures

The iterative structures are systematically framed by tags. This method facilitates the identification of groups in the XML file.

Example 1: Schema describing the Documents structure

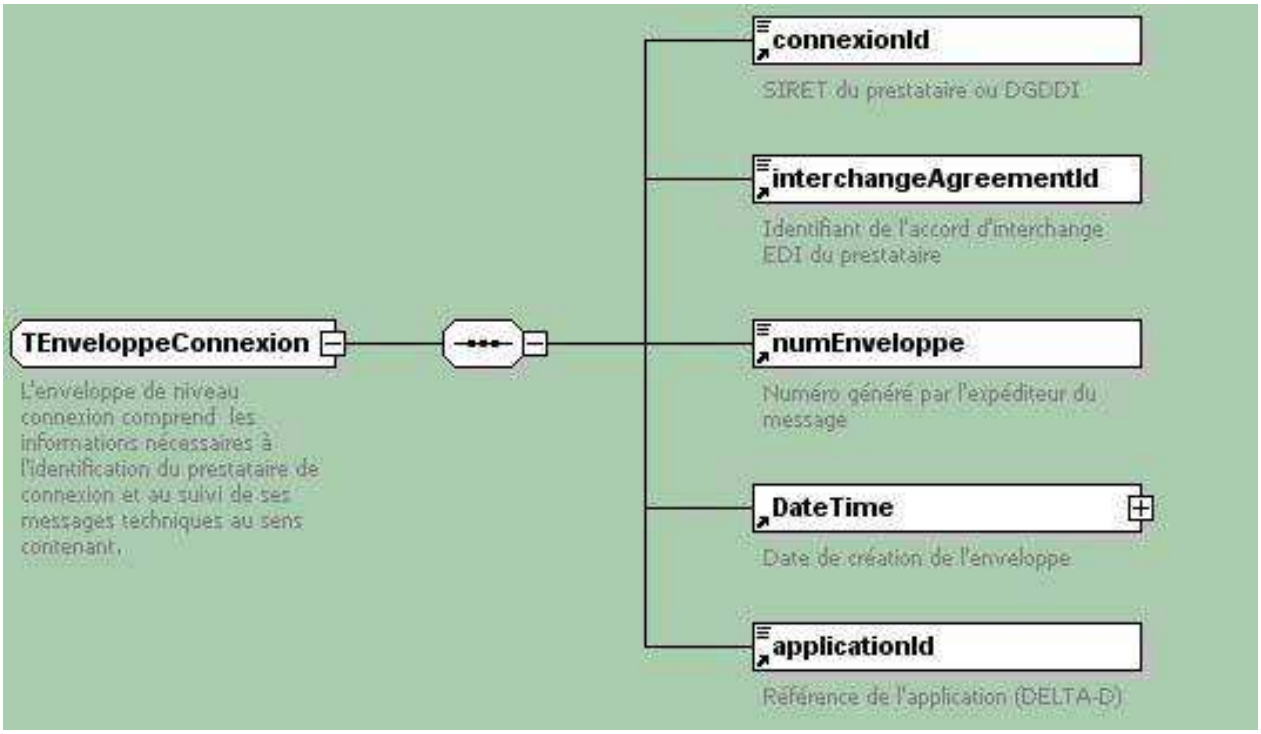
```
<xs:element name="Document" type="TDocument" miniOccurs="0">
        <xs:annotation>
          <xs:documentation> This data applies to SID and SID completed levels</xs:documentation>
          </xs:annotation>
        </xs:element>
<!--Documents-->
        <xs:complexType name="TDocuments">
          <xs:sequence maxOccurs="unbounded">
                <xs:element name="Document" type="TDocument"/>
          </xs:sequence>
        </xs:complexType>
<xs:complexType name="TDocument">
        </xs:sequence>
        <xs:element ref="doc"/>
        <xs:element ref="refdoc"/>
        <xs:element ref="datdoc"/>
        <xs:element ref="indd48"/>
        <xs:element ref="mntd48" minOccurs="0"/>
        <xs:element ref="deld48" minOccurs="0"/>
<xs:sequence>
</xs:complexType>
```

Example 2: Rendering of matching XML file

```
 <Documents>
        <Document>
                <doc>Stri</doc>
                <refdoc>String</refdoc>
                <datdoc>00/00/0000</datdoc>
                <indd48>1</indd48>
                <mntd48>30/mntd48>
                <deld48>1</deld48>
        </Document>
</Documents>
```

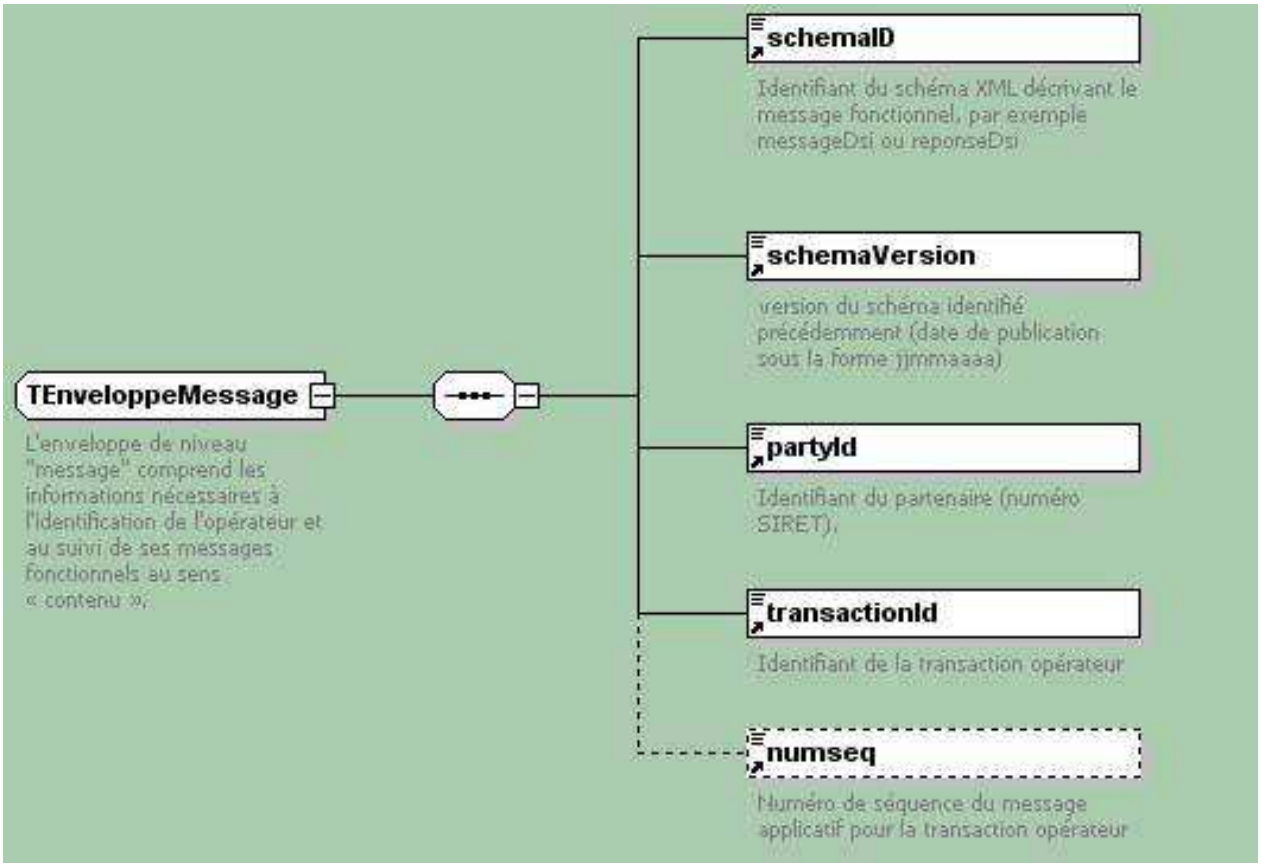
4. Envelope Processing

4.1. ParametresConnexion, Diagram



TEnveloppeConnexion
The envelope at connection level includes the information required for connection provider identification and follow-up of its technical messages in the container sense.
connexionID
Provider's SIRET or DGDDI
interchangeAgreementId
Identifier of the provider's EDI interchange agreement
numEveloppe
Number generated by message sender
DateTime
Envelope creation date
applicationId
Application reference (DELTA-D)

4.2. MessageParameters, Diagram



TEnveloppeMessage
The envelope at 'message' level includes the information required for operator identification and follow-up of its functional messages in the 'content' sense.
schemaID
Identifier of XML schema describing the functional message, viz. MessageDsi or reponseDsi
schemaVersion
Version of the schema identified earlier (publishing date as ddmmyyyy)
partyId
Partner's identifier (SIRET number)
transactionId
Operator transaction identifier
numseq
Sequence number of the application message for the operator transaction

4.3. Rules for Use

4.3.1. Building the Outgoing Envelope from the Operator's System

All the data are mandatory.

Connection envelope data	Definition	Comments
<connexionId>	Provider's identifier	SIRET number
<interchangeAgreementId>	Interchange agreement identifier	Identifier notified in the contract signed with the DGDDI
<numEnveloppe>	Technical message identifier	Generated by the transmitting system
<DateTime>	Transmission timestamp	Timestamp of message creation
<applicationId>	E-procedure identifier	Identifier code of the e-procedure found in the interchange agreement

Message envelope data	Definition	Comments
<schemaId>	Identifier of the application message schema	For instance, DecImp
<schemaVersion>	Identifier of the schema version	For instance, V1.1
<partyId>	Operator identifier	Operator's SIRET number
<transactionId>	Transaction identifier	Generated by the transmitting system
<numseq>	Sequence number	Generated by the transmitting system

4.3.2. Building the Outgoing Envelope from Customs' System

Connection envelope data	Definition	Comment
<connexionId>	DGDDI identifier	"DGDDI" generated by MAREVA
<interchangeAgreementId>	Interchange agreement identifier	Data transmitted to MAREVA by DELTA (the interchange agreement identifier is known by DELTA because a message transmitted by Customs is always an answer to a message received from an operator).
<numEnveloppe>	Technical message identifier	Number generated by MAREVA
<DateTime>	Transmission timestamp	Timestamp of message creation, completed by MAREVA
<applicationId>	E-procedure identifier	Transmitted by DELTA

Message envelope data		
<schemaId>	Identifier of the application message schema	For instance, 'ReponseDsi' transmitted by DELTA
<schemaVersion>	Identifier of the schema version	For instance, V1.1
<partyId>	Operator identifier	Operator's SIRET number transmitted by DELTA
<transactionId>	Transaction identifier	Transmitted by DELTA (the transaction identifier is known by DELTA because a message transmitted by Customs is always an answer to a message received from an operator)
<numseq>	Sequence number	Not used

4.3.3. DELTA and MAREVA Processing Rules

A 'PEDI Agreement" relationship is created in the ROSA repository, making it possible to store the required technical information about a provider.

MAREVA

Incoming (receiving messages from the provider):

- Controls the structure of the connection envelope
- Controls the validity of the <ConnexionId> <InterchangeAgreementId> pair and any signature compared to the certificate via the PEDI relationship
- Manages the technical exchanges with the connection provider thanks to the provider's <NumEnveloppe>

- Controls the sequencing of the functional messages thanks to the headings of the functional message envelope <EnveloppeMessage>:

<PartyId>
<TransactionId>
<Numseq>

- Sends the interchange agreement number (<InterchangeAgreementId>) and the functional message with its envelope (<EnveloppeMessage>) to DELTA. MAREVA only transmits a functional message if the sequencing is correct. If not, it keeps the message until it can deliver the messages in the right order.

Outgoing (sending messages to the provider):

- Receives from DELTA a parameter and the functional message with its <EnveloppeMessage> completed by DELTA, except for the <Numseq> heading that is not completed for outgoing messages
- Upgrades the data of the connection envelope – The <InterchangeAgreementId> data is parameterised by DELTA
- Handles the routing of technical messages thanks to the element <mel>, which has been obtained by querying the ROSA PEDI relationship based on the <InterchangeAgreementId> (parameterised by DELTA) and the name of the application that was known at time of DELTA - MAREVA connection.
- Manages the technical exchanges with the connection provider thanks to <NumEnveloppe> generated by MAREVA.

DELTA

Incoming:

- Receives the following data from MAREVA and stores them:
  - <InterchangeAgreementId>
  - The functional message
  - The message envelope linked to the functional message.

Outgoing:

- Builds the message envelope of the functional message (all the headings are completed except for <Numseq>)

- Sends the following to MAREVA:
  - The functional message envelope (<EnveloppeMessage>) completed except for <Numseq> that is not used. The TransactionId is the one for the declaration that is the subject of Customs' message. The Transaction Id was recorded by DELTA when it received the operator's message
  - The related functional message
  - The <interchangeAgreementId> parameter

- The identity of the e-procedure accessing MAREVA is known at the time of the connection to MAREVA (therefore, the e-procedure name does not have to be parameterised).

5. XML Message Implementation

5.1. Schema Construction Principle

All XML messages have a generic structure and a specific part matching the functional content of the message.
A schema describing a message systematically refers to the schemas describing the connection envelope and the message envelope.
All messages are described by at least two schemas making it possible to comply with the generic structure.

5.2. EDI Partners' Outgoing Messages

A technical message transmitted by an EDI partner may convey several functional messages of the same type. This rule may be adjusted for some e-procedures.

[diagram]

DsiMessage **ConnectionEnvelope** **Messages** **Message** **MessageEnvelope** **Declaration**
Declaration data

Source code Example of MessageDecImp schema

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- edited with XMLSPY v5 rel. 3 U (http://www.xmlspy.com) by daniel (dgd) --> <!--Version 2.00, 1 February 2006-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:include schemaLocation="DecImp.xsd"/>
<xs:element name="MessageDecImp">
<xs:complexType>
<xs:sequence>
<xs:element name="EnveloppeConnexion" type="TEnveloppeConnexion"/>
<xs:element name="Messages" type="TMessages"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:complexType name="TMessages">
<xs:sequence>
<xs:element name="Message" type="TMessage" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="TMessage">
<xs:sequence>
<xs:element name="EnveloppeMessage" type="TEnveloppeMessage"/>
<xs:element ref="Declaration"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

5.3. Customs' Outgoing Messages

The rules found in paragraphs 5.1. and 5.2. are still applicable.

[diagram]

Dsianswer Message **ConnectionEnvelope** **Messages** **Message** **MessageEnvelope** **DeclarationAnswer**

Source code example of MessageReponseDec schema

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- edited with XMLSPY v5 rel. 3 U (http://www.xmlspy.com) by daniel (dgd) -->
<!--Version 2.00, 1 February 2006-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:include schemaLocation="ReponseDec.xsd"/>
<xs:element name="MessageReponseDec">
<xs:complexType>
<xs:sequence>
<xs:element name="EnveloppeConnexion" type="TEnveloppeConnexion"/>
<xs:element name="Messages" type="TMessages"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:complexType name="TMessages">
<xs:sequence>
<xs:element name="Message" type="TMessage" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="TMessage">
<xs:sequence>
<xs:element name="EnveloppeMessage" type="TEnveloppeMessage"/>
<xs:element ref="ReponseDeclaration"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
```

5.4. Examples of Generated XML Messages

Message sent by an operator

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MessageDecImp xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="MessageDecImp.xsd">
<EnveloppeConnexion>
<connexionId>85520050700447</connexionId>
<interchangeAgreementId>00000001</interchangeAgreementId>
<numEnveloppe>191001</numEnveloppe>
<DateTime>
<date>19/10/05</date>
<time>11:35:01</time> </DateTime>
<applicationId>DELTA-D-F</applicationId>
</EnveloppeConnexion>
<Messages>
<Message>
<EnveloppeMessage>
<schemaID>MessageDecImp</schemaID>
<schemaVersion>1.1</schemaVersion>
<partyId>85520050700447</partyId>
<transactionId>191001</transactionId>
<numseq>0</numseq>
</EnveloppeMessage>
</ Declaration>
Body of application message
</ Declaration>
</Message>
</Messages>
</MessageReponseDsi>
```

ReponseDec Message sent by Customs

```
<MessageReponseDec>
<EnveloppeConnexion>
<connexionId>DGDDI</connexionId>
<interchangeAgreementId>00000001</interchangeAgreementId>
<numEnveloppe>9618</numEnveloppe>
<DateTime>
<date>19/10/05</date>
<time>10:56:04</time>
</DateTime>
<applicationId>DELTA-D-F</applicationId>
</EnveloppeConnexion>
<Messages>
<Message>
<EnveloppeMessage>
<schemaID>ReponseDec</schemaID>
<schemaVersion>1.1</schemaVersion>
<partyId>85520050700447</partyId>
<transactionId>191001</transactionId>
</EnveloppeMessage> <>
< ReponseDeclaration>
Body of application message
</ReponseDeclaration>
</Message>
</Messages>
</MessageReponseDsi>
```