



## ESPPADOM

Echanges financiers – prestataires pour les services à domicile auprès des personnes en perte d'autonomie

Programme soutenu par  
la Caisse nationale de solidarité pour l'autonomie

# EVOLUTION DU SCHEMA XSD DU MESSAGE ORDER

**Résumé** : Ce document détaille les évolutions du schéma xsd de description du message Order décidées par les ateliers techniques du 15 janvier et du 19 février 2016

Statut document de travail  
Version 0.2  
Date 21/1/2016  
Auteurs EDESS – Philippe AMELINE, François ROUGERIE

### Evolutions

V 0.0	21/01/2016	Version initiale, suite à l'atelier du 15/01/2016
V 0.1	28/01/2016	
V 0.2	20/02/2016	Prise en compte des évolutions décidées lors de l'atelier du 19/02/2016

# Table des matières

1	Contexte .....	3
2	Cohérence des types .....	4
2.1	Pour les dates .....	4
2.2	Pour les codes issus de listes prédéfinies .....	4
3	Redondance avec les informations de prestations.....	6
4	Modification de cardinalité .....	7
4.1	Blocs optionnels d'informations diverses .....	7
4.2	Optionnalité du prestataire .....	7
5	Evolution de la commande .....	8
6	Evolution des Personnes .....	9
6.1	Ajout de la civilité.....	9
6.2	Ajout d'une adresse postale aux contacts.....	9
6.3	Ajout du secteur géographique .....	9
6.4	Ajout d'un bloc d'informations additionnelles .....	10
7	Evolution de la prestation .....	11
7.1	Cadre d'intervention .....	11
7.2	Evolution du type d'acte .....	11
7.3	Ajout des actes.....	11
7.4	Gestion des évolutions .....	12
8	Evolution des fichiers.....	13
8.1	Nommage.....	13
8.2	Généralisation .....	13

# 1 CONTEXTE

---

Le schéma xsd du message ORDER décrit l'arborescence des balises qui portent les informations contenues dans le schéma, la cardinalité de ces balises ainsi que les informations de formatage des données qu'elles hébergent.

Modifier le schéma xsd est une opération délicate, puisqu'elle peut obliger à modifier les systèmes déjà fonctionnels. Il est donc fondamental de ne le faire qu'à bon escient, et, si nécessaire, le plus tôt possible afin de limiter le nombre de sites impactés.

Ce document détaille les modifications proposées lors de la réunion technique du 15 janvier 2016 et entérinés lors de la réunion du 19 février 2016. Il a pour vocation de servir de manuel d'instruction aux équipes ayant déjà mis en œuvre la version précédente du standard, mais aussi, dans un esprit de « mémoire des organisations », de laisser une trace de ces évolutions et des raisons qui les ont motivées.

## 2 COHÉRENCE DES TYPES

### 2.1 POUR LES DATES

La première règle de cohérence des types consiste à **utiliser le type DateMandatoryDateTimeType pour toutes les dates.**

```
<xsd:simpleType name="DateMandatoryDateTimeType">  
  <xsd:union memberTypes="DateTimeType DateType"/>  
</xsd:simpleType>
```

Les balises modifiées sont listées dans le tableau ci-dessous, avec leur ancien type.

Balise modifiée	Signification	Type précédent
CrossIndustryOrder / CIExchangedDocumentContext / ApplicationSpecifiedCIDocumentContextParameter / CommitteeDateTime	Date de la décision d'accord de la prise en charge	TextType
CrossIndustryOrder / CIOHSupplyChainTradeTransaction / ApplicableCIOHSupplyChainTradeDelivery / ShipToCITradeParty / BirthDate	Date de naissance du bénéficiaire	TextType

Ces deux dates ont probablement déjà été spécifiées dans le passé sous forme de DateMandatoryDateTimeType puisque ce type était encore présent sous forme de commentaire dans le fichier AUX\_UNECE\_RAM\_8p0\_ESPPADOM\_Order\_V00\_02.xsd :

```
<xsd:element name="CommitteeDateTime" type="UNECE_UDT_9p0:TextType" /> <!--  
UNECE_QDT_8p0:DateMandatoryDateTimeType-->
```

```
<xsd:element name="BirthDate" type="UNECE_UDT_9p0:TextType" minOccurs="0"/> <!--  
UNECE_QDT_8p0:DateMandatoryDateTimeType -->
```

Il a été discuté de conserver la balise BirthDate sous forme de texte afin de pouvoir inscrire une date de naissance en année lunaire. Le fait que la date de naissance soit un élément important d'identitovigilance (donc qu'il est important de la représenter dans un format précis) et la très faible fréquence des dates non standards a amené à ne pas tenir compte de cet argument.

On signalera que le « numéro de sécurité sociale » (NIR) peut, lorsque le mois de naissance n'est pas connu, contenir 62 ou 63 au lieu d'une donnée issue de l'habituel intervalle 01 – 12. Ce type de convention n'est, a priori, pas compatible avec le type xsd:date puisque les recommandations du W3C précisent que « The value of each numeric-valued property (other than timeOnTimeline) is limited to the maximum value within the interval determined by the next-higher property. For example, the day value can never be 32, and cannot even be 29 for month 02 and year 2002 (February 2002). » : <https://www.w3.org/TR/xmlschema-2/#dateTime>

S'il fallait pouvoir accepter des dates « exotiques », la solution serait peut-être de créer un nouveau type qui ajoute le type xsd:string à l'union « DateTimeType DateType » du style DateMandatoryDateTimeType.

### 2.2 POUR LES CODES ISSUS DE LISTES PRÉDÉFINIES

Les codes issus de listes prédéfinies, comme le GIR ou le type de prise en charge doivent utiliser le type CodeQualifiedType qui permet de désigner en attributs la liste ou la classification au sein de laquelle a été choisi le code fourni en paramètre, ainsi que, de façon optionnelle, le libellé correspondant à ce code.

Dans l'ancien schéma, toutes les balises du bloc ContextShipTo qui décrit le cadre de prise en charge (APA, PCH ou autre, GIR, etc) utilisaient le type TextType alors qu'elles doivent toutes véhiculer un code issu d'une liste prédéfinie. Il est proposé de les transformer en CodeQualifiedType.

```
<xsd:complexType name="ContextShipToType">
  <xsd:sequence>
    <xsd:element name="GIR" type="TextType" minOccurs="0" />
    <xsd:element name="NaturePrestation" type="TextType" minOccurs="0" />
    <xsd:element name="CadreIntervention" type="TextType" minOccurs="0" />
    <xsd:element name="TypeBeneficiaire" type="TextType" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
```

### 3 REDONDANCE AVEC LES INFORMATIONS DE PRESTATIONS

Le réel objet du message Order est de véhiculer une liste de prestations. Cette liste est décrite au sein du bloc *IncludedCIOLSupplyChainTradeLineItem*. Deux blocs d'informations, situées en dehors de ce bloc décrivaient des données qui font référence aux prestations :

Balise	Signification
CrossIndustryOrder / CIOHSupplyChainTradeTransaction / ApplicableCIOHSupplyChainTradeAgreement / PriceListReferencedCIReferencedDocument	Liste de tarifs ou catalogue applicable à tous les services et produits commandés  Contient une référence à un document externe
CrossIndustryOrder / CIOHSupplyChainTradeTransaction / ApplicableCIOHSupplyChainTradeDelivery / DeliveryCIDeliveryInstructions	Instructions optionnelles pour l'exécution des services et de leur livraison.  Contient le profil des employés devant intervenir ainsi qu'une information temporelle au format RRULE

La balise DeliveryCIDeliveryInstructions était redondante puisqu'on la retrouve au sein de chaque bloc de description de prestation.

CrossIndustryOrder / CIOHSupplyChainTradeTransaction / IncludedCIOLSupplyChainTradeLineItem / SpecifiedCIOLSupplyChainTradeDelivery / DeliveryCIDeliveryInstructions

La balise PriceListReferencedCIReferencedDocument n'était pas présente au sein des lignes de prestations, donc non directement redondante, mais elle induisait la possibilité que des données de description de la prestation puissent être présentes en dehors du message, et donc a fortiori en dehors du bloc IncludedCIOLSupplyChainTradeLineItem qui décrit une prestation.

Ces deux balises, de toute façon optionnelles, rendaient le message inutilement complexe en posant le problème de l'**application de règles de préséance entre information globale et information locale**.

S'imposaient-elles à chaque ligne de prestation, en remplacement ou en complément des données existantes, ou bien complétaient-elles les prestations qui ne contiennent pas ces données ? On voit bien le danger qu'il y aurait eu à ne pas répondre de façon très précise à ces questions (avec la difficulté supplémentaire, pour la balise PriceListReferencedCIReferencedDocument de faire « participer au message » un document externe de format inconnu).

## 4 MODIFICATION DE CARDINALITÉ

### 4.1 BLOCS OPTIONNELS D'INFORMATIONS DIVERSES

Le message Order contient plusieurs blocs optionnels destinés à recevoir des données qui n'ont pas nécessairement de cohérences entre elles :

Balise	Signification
CrossIndustryOrder / CIExchangedDocumentContext / ApplicationSpecifiedCIDocumentContextParameter	Informations qui précisent le contexte d'application de ce document.  Ne contient que la date de décision de l'accord de la prise en charge qui était obligatoire
CrossIndustryOrder / CIOHSupplyChainTradeTransaction / ApplicableCIOHSupplyChainTradeDelivery / ShipToCITradeParty / ContextShipTo	Précisions sur le bénéficiaire.  Ne contenait que des données obligatoires : le GIR, la nature de prestation (APA, PCH...), le cadre d'intervention (mandataire...) et le type de bénéficiaire (enfant, handicapé...)

Ce type de blocs optionnels qui apportent un « **complément d'informations en vrac** » **ne doit pas contenir d'information obligatoire**. Dans le cas contraire, préciser une information oblige à en préciser d'autres qui ne sont éventuellement pas pertinentes dans son contexte ; typiquement, le bloc *ContextShipTo* obligeait, même en cas de prestation de type PCH (prise en charge du handicap) à faire apparaître un score GIR qui n'est validé que chez la personne âgée.

Le bloc optionnel *ApplicationSpecifiedCIDocumentContextParameter* ne contient qu'une seule information (obligatoire) et aurait donc pu être laissé en l'état. En tant que bloc de contexte, il a pourtant vocation à évoluer et il est plus raisonnable de ne pas lui conserver une valeur obligatoire qui posera problème dans le futur.

### 4.2 OPTIONNALITÉ DU PRESTATAIRE

En toute rigueur, une commande désigne toujours le prestataire à qui elle est confiée. Par ailleurs, la volonté a été fortement exprimée lors de l'atelier que le message Order puisse ne pas être limité à la passation d'une commande, par exemple, par les organisations en amont, pour **décrire un plan d'aide ou un plan de charge non encore affecté**. Cette extension du cadre d'application du message pourrait amener à rendre optionnelle la balise :

CrossIndustryOrder / CIOHSupplyChainTradeTransaction / ApplicableCIOHSupplyChainTradeAgreement / SellerCITradeParty

Cependant, la rendre optionnelle risquerait de rendre valide de véritables messages de commande dont le bloc de description du prestataire a été accidentellement omis. Il a été convenu de la laisser obligatoire et de laisser les utilisateurs qui souhaitent utiliser Order dans un cadre où le prestataire n'a pas été défini à décider d'une convention d'utilisation des deux balises obligatoires ID et name, soit en les laissant vides, soit en définissant un couple de valeurs spécifique pour les cas où le prestataire n'est pas attribué

## 5 EVOLUTION DE LA COMMANDE

---

La nature de la prestation (APA, PCH...) était une information présente dans le bloc de contexte du bénéficiaire :

CrossIndustryOrder / CIOHSupplyChainTradeTransaction / ApplicableCIOHSupplyChainTradeDelivery / ShipToCITradeParty / ContextShipTo

En pratique, c'est un descriptif du plan d'aide et elle a été déplacée dans le bloc

CrossIndustryOrder / CIOHExchangedDocument

Qui devient ainsi :

```
<xsd:complexType name="CIOHExchangedDocumentType">
  <xsd:sequence>
    <xsd:element name="ID" type="UNECE_UDT_9p0:IDUnqualifiedType"/>
    <xsd:element name="NaturePrestation" type="UNECE_UDT_9p0:CodeQualifiedType"/>
    <xsd:element name="IssueDateTime" type="UNECE_QDT_8p0:DateMandatoryDateTimeType"/>
    <xsd:element name="IncludedCINote" type="UNECE_RAM_8p0:CINoteType" minOccurs="0"/>
    <xsd:element name="EffectiveCISpecifiedPeriod" type="UNECE_QDT_8p0:CISpecifiedPeriodType"/>
  </xsd:sequence>
</xsd:complexType>
```

## 6 EVOLUTION DES PERSONNES

---

### 6.1 AJOUT DE LA CIVILITÉ

Les personnes n'avaient pas d'information de civilité (Madame, Monsieur), une balise optionnelle est ajoutée à cette fin au type CITradePartyType :

```
<xsd:complexType name="CITradePartyType">
  <xsd:sequence>
  [...]
  <xsd:element name="Civility" type="UNECE_UDT_9p0:CodeQualifiedType" minOccurs="0"/>
  [...]
  </xsd:sequence>
</xsd:complexType>
```

### 6.2 AJOUT D'UNE ADRESSE POSTALE AUX CONTACTS

Le bloc de description de la personne permet de déclarer une liste de contacts. Ces contacts étaient définis sous la forme :

```
<xsd:complexType name="CITradeContactType">
  <xsd:sequence>
    <xsd:element name="PersonName" type="TextType" minOccurs="0"/>
    <xsd:element name="TypeCode" type="CodeQualifiedType" minOccurs="0"/>
    <xsd:element name="TelephoneCIUniversalCommunication" type="CIUniversalCommunicationNumberType" minOccurs="0"/>
    <xsd:element name="MobileTelephoneCIUniversalCommunication" type="CIUniversalCommunicationNumberType"
minOccurs="0"/>
    <xsd:element name="EmailURICIUniversalCommunication" type="CIUniversalCommunicationUnqualifiedURIType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

Un contact était ainsi quelqu'un à qui on pouvait téléphoner ou expédier un mail, mais pas écrire ou rendre visite ; la nécessité de prévoir des contacts à qui on puisse, par exemple, envoyer des factures, nous a amené à ajouter une adresse postale ; le bloc devient ainsi :

```
<xsd:complexType name="CITradeContactType">
  <xsd:sequence>
    <xsd:element name="PersonName" type="TextType" minOccurs="0"/>
    <xsd:element name="TypeCode" type="CodeQualifiedType" minOccurs="0"/>
    <xsd:element name="TelephoneCIUniversalCommunication" type="CIUniversalCommunicationNumberType" minOccurs="0"/>
    <xsd:element name="MobileTelephoneCIUniversalCommunication" type="CIUniversalCommunicationNumberType"
minOccurs="0"/>
    <xsd:element name="EmailURICIUniversalCommunication" type="CIUniversalCommunicationUnqualifiedURIType" minOccurs="0"/>
    <xsd:element name="PostalAddress" type="CITradeAddressType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

### 6.3 AJOUT DU SECTEUR GÉOGRAPHIQUE

Certains départements sont subdivisés en secteurs au sein desquels le bénéficiaire est pris en charge par des prestataires spécifiques. Cette information manquait au sein du message et elle a été ajoutée au sein du bloc ContextShipTo qui devient :

```
<xsd:complexType name="ContextShipToType">
  <xsd:sequence>
    <xsd:element name="GIR" type="CodeQualifiedType" minOccurs="0"/>
    <xsd:element name="TypeBeneficiaire" type="CodeQualifiedType" minOccurs="0"/>
    <xsd:element name="GeographicSector" type="IDQualifiedType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

## 6.4 AJOUT D'UN BLOC D'INFORMATIONS ADDITIONNELLES

La variété de pratique des donneurs d'ordre justifie d'apporter une certaine flexibilité au bloc ContextShipTo, par exemple pour héberger des informations comme le « numéro de dossier papier » en usage dans certains départements. Plutôt que de multiplier les balises dédiées, au risque de rendre le message confus, il a été décidé de créer un bloc de données additionnelles sous la forme :

```
<xsd:complexType name="CIAdditionalInformationType">
  <xsd:sequence>
    <xsd:element name="Type" type="CodeQualifiedType"/>
    <xsd:element name="Label" type="TextType" minOccurs="0"/>
    <xsd:element name="Content" type="TextType"/>
  </xsd:sequence>
</xsd:complexType>
```

Il est ainsi possible d'inclure tout type d'information d'intérêt local, en indiquant son type au sein d'une liste ESPPADOM, en fournissant le libellé précis tel qu'il s'exprime localement, et, enfin, en indiquant le contenu de cette variable pour ce bénéficiaire.

Le bloc ContextShipTo devient ainsi :

```
<xsd:complexType name="ContextShipToType">
  <xsd:sequence>
    <xsd:element name="GIR" type="CodeQualifiedType" minOccurs="0"/>
    <xsd:element name="TypeBeneficiaire" type="CodeQualifiedType" minOccurs="0"/>
    <xsd:element name="GeographicSector" type="IDQualifiedType" minOccurs="0"/>
    <xsd:element name="AdditionalInformation" type="CIAdditionalInformationType" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

## 7 EVOLUTION DE LA PRESTATION

---

### 7.1 CADRE D'INTERVENTION

Le cadre d'intervention (prestataire, mandataire...) était indiqué dans le bloc de contexte du bénéficiaire :

```
CrossIndustryOrder / CIOHSupplyChainTradeTransaction / ApplicableCIOHSupplyChainTradeDelivery / ShipToCITradeParty / ContextShipTo
```

En pratique, un même plan d'aide peut inclure des prestations dans les différents modes, et il a été décidé de déplacer cette information au sein du bloc de description de la prestation. Le sous-bloc cible est celui qui, au sein de la prestation, véhicule le contexte financier :

```
CrossIndustryOrder / CIOHSupplyChainTradeTransaction / IncludedCIOLSupplyChainTradeLineItem / SpecifiedCIOLSupplyChainTradeAgreement
```

Ce bloc devient ainsi :

```
<xsd:complexType name="CIOLSupplyChainTradeAgreementType">
  <xsd:sequence>
    <xsd:element name="NetPriceProductCITradePrice" type="CITradePriceType"/>
    <xsd:element name="CadreIntervention" type="CodeQualifiedType"/>
  </xsd:sequence>
</xsd:complexType>
```

### 7.2 EVOLUTION DU TYPE D'ACTE

Au sein du bloc CITradeProductType qui définit la prestation à réaliser, avec son identifiant et son libellé, l'identifiant était du type IDQualifiedType, ce qui n'était pas compatible avec le choix au sein de la liste ESPPADOM\_SERVICE. Le type sera donc désormais CodeQualifiedType :

```
<xsd:complexType name="CITradeProductType">
  <xsd:sequence>
    <xsd:element name="ID" type="CodeQualifiedType"/>
    <xsd:element name="Name" type="TextType"/>
  </xsd:sequence>
</xsd:complexType>
```

### 7.3 AJOUT DES ACTES

L'évolution vers des « plans d'aide qualitatifs » amène les conseils départementaux à décrire, au sein d'une prestation générale, par exemple « Aide à domicile », un ensemble d'actes, comme « aide aux repas du lundi au vendredi », « aide à la toilette du lundi au vendredi », « aide au lever le WE ».

Ces actes ne sont pas assimilables à des prestations au sens où elles ne portent pas d'informations financières. Il n'existait pas de façon de les décrire au sein de l'ancien message.

Ce besoin a entraîné l'ajout au bloc IncludedCIOLSupplyChainTradeLineItem d'une nouvelle balise **SpecifiedCITradeProductDetails** optionnelle et multiple :

```
<xsd:complexType name="CIOLSupplyChainTradeLineItemType">
  <xsd:sequence>
    <xsd:element name="AssociatedCIOLDocumentLineDocument" type="CIOLDocumentLineDocumentType"/>
    <xsd:element name="SpecifiedCIOLSupplyChainTradeAgreement" type="CIOLSupplyChainTradeAgreementType"/>
    <xsd:element name="SpecifiedCIOLSupplyChainTradeDelivery" type="CIOLSupplyChainTradeDeliveryType"/>
    <xsd:element name="SpecifiedCIOLSupplyChainTradeSettlement" type="CIOLSupplyChainTradeSettlementType" minOccurs="0"/>
    <xsd:element name="SpecifiedCITradeProduct" type="CITradeProductType"/>
    <xsd:element name="SpecifiedCITradeProductDetails" type="CITradeDetailType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="SpecifiedCITradeLineChange" type="CITradeLineChangeType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

Le type CITradeDetailType permet de décrire un acte, en lui attribuant un identifiant unique, un identifiant au sein de la liste ESPPADOM\_SERVICE, un libellé, et un bloc DeliveryInstruction identique à celui de la

prestation qui permet de définir la quantité et les instructions de mise en œuvre (type d'intervenant et calendrier).

```
<xsd:complexType name="CITradeDetailType">
  <xsd:sequence>
    <xsd:element name="DetailLineID" type="IDUnqualifiedType"/>
    <xsd:element name="ID" type="CodeQualifiedType"/>
    <xsd:element name="Name" type="TextType"/>
    <xsd:element name="DeliveryInstruction" type="CIOLSupplyChainTradeDeliveryType"/>
  </xsd:sequence>
</xsd:complexType>
```

## 7.4 GESTION DES ÉVOLUTIONS

Les révisions de plan d'aide sont fréquentes. Elles peuvent consister en l'ajout de nouvelles prestations, en l'arrêt de certaines prestations (avec une date d'arrêt) ou en la suspension de prestations (avec une date de début et une date de fin), par exemple dans le cas où le bénéficiaire est hospitalisé. Il doit même être possible de supprimer une prestation « non démarrée ».

Dans le cadre UN/CEFACT, la modification de commande donne lieu à un ensemble de messages spécifiques (Cross Industry Order Change) ; compte tenu du fait qu'un message Order véhicule essentiellement une liste de prestations, et qu'il était possible de créer des règles de mise en œuvre à la fois précises et simples, il a paru plus logique de gérer les évolutions au sein même du message.

Les règles sont les suivantes :

- 1) Toute prestation sans instruction de modification est considérée comme nouvelle.
- 2) Une prestation non démarrée peut être supprimée.
- 3) Une prestation peut être suspendue entre deux dates précises.
- 4) En dehors de la suspension, toute modification de prestation doit passer par l'arrêt de la prestation à modifier et la mise en place d'une nouvelle prestation, porteuse des évolutions souhaitées.

Pour gérer les évolutions, le bloc de description de la prestation possède désormais un bloc SpecifiedCITradeLineChange :

```
<xsd:complexType name="CIOLSupplyChainTradeLineItem">
  <xsd:sequence>
    <xsd:element name="AssociatedCIOLDocumentLineDocument" type="CIOLDocumentLineDocumentType"/>
    <xsd:element name="SpecifiedCIOLSupplyChainTradeAgreement" type="CIOLSupplyChainTradeAgreementType"/>
    <xsd:element name="SpecifiedCIOLSupplyChainTradeDelivery" type="CIOLSupplyChainTradeDeliveryType"/>
    <xsd:element name="SpecifiedCIOLSupplyChainTradeSettlement" type="CIOLSupplyChainTradeSettlementType" minOccurs="0"/>
    <xsd:element name="SpecifiedCITradeProduct" type="CITradeProductType"/>
    <xsd:element name="SpecifiedCITradeProductDetails" type="CITradeDetailType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="SpecifiedCITradeLineChange" type="CITradeLineChangeType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

Ce bloc permet de décrire le type d'évolution (suppression/arrêt/suspension), sa date de prise d'effet, sa date de fin d'effet en cas de suspension, son motif et un libellé qui permet de fournir un texte explicatif.

```
<xsd:complexType name="CITradeLineChangeType">
  <xsd:sequence>
    <xsd:element name="ChangeType" type="CodeQualifiedType"/>
    <xsd:element name="EffectDateTime" type="DateMandatoryDateTimeType"/>
    <xsd:element name="EndOfEffectDateTime" type="DateMandatoryDateTimeType" minOccurs="0"/>
    <xsd:element name="ReasonForChange" type="CodeQualifiedType"/>
    <xsd:element name="Label" type="TextType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

## 8 EVOLUTION DES FICHIERS

---

### 8.1 NOMMAGE

Les fichiers de schéma XML UN/CEFACT ont un nom du type <Schema Module Name>\_<Version Identifier>.xsd, où l'identifiant de version est représenté sous forme version majeure, version mineure, séparées par la lettre 'p' en remplacement de l'usuel point. Par exemple AUX\_UNECE\_RAM\_8p0.xsd. Les noms de fichiers xsd Esppadom évoluent vers le même formalisme, par exemple AUX\_UNECE\_RAM\_8p0\_ESPPADOM\_ORDER\_1p0.xsd.

### 8.2 GÉNÉRALISATION

Il faut parfois penser aux développeurs, et il est difficilement envisageable au long cours de décrire les mêmes types dans des fichiers spécifiques à chaque message (Order, Delivery et Invoice), ne serait-ce que parce qu'ils risquent de diverger au décours de leurs évolutions respectives.

Pour cette raison, les fichiers xsd non spécifiques d'un message deviennent généraux (garantissant que les mêmes objets sont réutilisables de message en message) et les informations descriptives de la personne sont désormais hébergées dans un fichier spécifique qui sera partagé par tous les messages.

Les fichiers qui portent cette généralisation sont :

AUX\_UNECE\_RAM\_8p0\_ESPPADOM\_PERSON\_1p0.xsd

AUX\_UNECE\_QDT\_8p0\_ESPPADOM\_1p0.xsd

AUX\_UNECE\_UDT\_9p0\_ESPPADOM\_1p0.xsd