

## **Modification du schéma « XMLFR-ext.xsd »**

SIES, 21 novembre 2012

### Clarification de la définition de l'objet habilitation.

Dans un fichier CDM, un objet « habilitation » concerne un et un seul diplôme habilité ou dont l'habilitation est demandée, c'est-à-dire une mention (Master ou Licence), une licence-pro ou un autre diplôme.

Dans les versions 1.0.9.x, la manière dont l'objet était modélisé ne reflétait pas toujours ce choix (problème de cardinalité ou de choix d'organisation des éléments).

Les présentes modifications ont pour objet de mettre le modèle en conformité avec le choix d'organisation explicité ci-dessus.

### Fichier CDMFR.xsd

Dans le schéma CDMFR, il ne pouvait y avoir qu'un seul objet habilitation. Avec la définition ci-dessus, ceci pose problème dans le cas où le fichier contient la description de plusieurs formations habilitées (ensemble de l'offre de formation d'un établissement, par exemple) : dans ce cas, le fichier CDM doit comporter autant d'objets habilitation qu'il y a de diplômes habilités (ou dont l'habilitation est demandée).

### Action

- Pour l'élément « habilitation » rattaché directement à la racine (CDM) du document, précision sur la cardinalité : ajout de l'attribut « maxOccurs='unbounded' » pour indiquer qu'il est possible d'avoir de 0 à plusieurs éléments de ce type, un par diplôme concerné par une habilitation.

### Fichier CDMFR-ext.xsd

#### **Modification de la définition du type complexe « habilitationType » :**

### Actions

- Encapsulation des éléments « field », « licPro » et « otherDiploma » dans une balise « choice » (avec choix unique et obligatoire). Ceci permet d'avoir un objet « habilitation » qui ne concerne qu'un seul diplôme : une mention de master ou de licence (field), une licence professionnelle (licPro) ou un autre diplôme. Cardinalité du « choice » mise explicitement à [1-1]
- La modification ci-dessus a comme conséquence d'obliger à transformer l'élément « choice » principal de « habilitationType » en séquence. Ce qui est de toute façon plus logique que de laisser un « choice » illimité (qui permet de mettre tous les éléments plusieurs fois, et dans n'importe quel ordre (ce qui est contraire à la définition de l'objet habilitation (cf début du document)).

- Cardinalités des éléments « field », « licPro » et « otherDiploma » supprimées : le fait d'avoir englobé ces 3 éléments dans un élément « choice » fait que cela rend obligatoire la présence d'un et un seul élément parmi les trois.
- Ajout de l'élément « domainName » au niveau de l'habilitation (cardinalité [0-n]) afin de permettre d'indiquer quel est le domaine général du diplôme, dans le cas où il y aurait des domaines différents (au niveau cohabilitation spécialité, par exemple) mais où ils ne seraient pas tous repris au niveau de la mention. (Par exemple, mention en SHS comportant une spécialité cohabilitée sur un autre établissement en ALL).

### Propositions

- Dans « OtherDiploma », suite à ces modifications, il doit être possible d'enlever l'élément « refProgram », dans la mesure où l'information est déjà présente dans « habiliObject ». Il doit aussi être possible d'enlever l'élément « refOrgUnit » car l'information se trouve dans l'objet « cohabilitation ». A voir.

### **Elément « habiliRequest ».**

A priori, il n'y a qu'une demande d'habilitation par formation.

Il n'est donc pas nécessaire de laisser la possibilité de mettre plusieurs éléments de ce type dans une habilitation.

### Actions

- Cardinalité passée à [0-1].

### **Sous-élément « habiliRequest / habiliTiming »**

### Actions

- Les éléments « habiliCreatDate » et « habiliRenewDate » sont isolés du reste de la séquence dans un élément de type « choice », ce qui ne permet de mettre que l'un ou l'autre, alors que, fonctionnellement, rien n'interdit d'avoir les 2 (une date de première habilitation, et une date de dernier renouvellement). ➔ suppression du nœud « choice » pour laisser les éléments directement dans la séquence.
- Modification de la cardinalité de l'élément « habiliCreatDate » (date à laquelle la formation décrite a été habilitée pour la première fois) : cardinalité passée de [1-1] à [0-1] car dans le cas d'une demande d'habilitation d'un nouveau diplôme, cette date n'est pas connue.
- Modification de la cardinalité des l'élément « habilitOpeningWhishDate » et « habiliWhishDuration » (Respectivement date d'ouverture et durée d'habilitation souhaitée par l'établissement) : cardinalités passées de [1-1] à [0-1] (S'il s'agit d'une formation habilitée, il n'y a pas une date d'ouverture et une durée souhaitées, mais une date et une durée accordées.
- Modification de la cardinalité de l'élément « habiliValidDuration » (Durée pour laquelle la formation est habilitée) : cardinalité passée de [1-1] à [0-1] car dans le cas d'une demande d'habilitation, cette durée n'est pas encore connue.

- Mise en conformité de la cardinalité de l'élément « hbiliRenewDate » et de la documentation associée : d'un côté, il y avait une cardinalité multiple, ce qui permet d'avoir de 0 à n dates, de l'autre il est écrit, je cite « Dans le cas d'un renouvellement, date du dernier renouvellement de l'habilitation ». Ce qui est intéressant, c'est la date du dernier renouvellement, et non la succession de ces dates. ➔ La cardinalité est mise à [0-1], pour être conforme au commentaire.

#### **Elément « cohabilitation » :**

L'établissement support/déposant du diplôme doit aussi figurer ici. Et s'il n'y a pas de cohabilitation, il y aura uniquement l'établissement qui délivre seul le diplôme concerné par l'habilitation, ce qui implique que la présence d'au moins une balise cohabilitation.

De plus, l'élément cohabilitation contient lui-même une liste d'établissements (« orgUnit ») avec leur rôle (support ou cohabilité). Il ne me semble donc pas nécessaire qu'il y ait plusieurs objets « cohabilitation », un seul devrait suffire à décrire les cohabilitations d'un diplôme (à traiter ultérieurement, pour l'instant, on laisse la possibilité d'avoir plusieurs balises de ce type).

#### Actions

- Cardinalité passée de [0-n] à [1-n]

#### **Elément « field » (définition)**

Corrections sur les cardinalités des éléments de la définition de l'objet « field ».  
Cet élément est utilisé pour décrire une mention (et une seule).

#### Actions

- Cardinalité de la séquence principale mise à [1-1] (il y avait [0-n]).
- Cardinalité de l'élément « fieldName » mise à « défaut » i.e. [1-1] (il y avait un 0 en borne inférieure, ce qui fait que le nom n'était pas obligatoire).
- Cardinalité de l'élément « coHabilitation » dans les spécialités mise à [1-n] (à la place de [0-n]) pour les mêmes raisons que dans la branche principale (cf ci-dessus).

#### **Elément « habiliObject » (Formation(s) concernées par cette habilitation)**

Cet élément a une cardinalité [1-n].

Il contient un ou plusieurs éléments « refProgram » (cardinalité [1-n] ici aussi).

Il peut y avoir plusieurs éléments parce que les éléments référencés sont des objets « program » et qu'une formation peut être composée de plusieurs objets « program ».

Le fait d'avoir 2 éléments imbriqués avec une cardinalité [1-n] fait qu'il est possible de décrire une même chose de plusieurs façons différentes. Cela amène de la souplesse, mais aussi de la complexité. Cela ne me semble pas nécessaire ici.

Cf - ANNEXE 1 pour un exemple.

#### Action

- Passage de la cardinalité du niveau supérieur (habiliObject) à [1-1].

**Mise à jour des documentations** (voir fichier XSD) sur les éléments suivants :

- field
- licPro
- otherDiploma (ajout de « ... ou licence-Pro » à la fin)
- cohabilitation

### **Carte heuristique FreePlane**

Mise à jour à partir de la version 2.0.8.

Ajout d'un nœud intermédiaire « ? » entre « habilitation » et les trois nœuds « field », « licPro » et « otherDiploma » pour représenter l'élément « choice » qui englobe ces 3 éléments dans la version 1.0.10. Cardinalité du nœud à [1-1].

Cardinalité de l'élément « coHabilitation » dans les spécialités mis à [1-n]. Même modification que dans le schéma, pour avoir à la fois l'établissement habilité et les cohabilités.

Cardinalité de « licPro » mise à [0-1] (et non [0-n]) : ces cardinalités sont indiquées dans la carte, mais uniquement pour mémoire : la cardinalité du « choice » ci-dessus implique qu'il y a un et un seul élément parmi les 3.

Correction du nom du nœud « licDiplomaTypeValueSet » (majuscule pour le « s » de « Set ») et passage du texte en couleur rouge, pour indiquer qu'il s'agit d'une liste de valeurs.

Mise en cohérence des cardinalités dans « otherDiploma »

« refOrgUnit » est à [0-1] dans le fichier XSD et à [1-n] sur la carte.

« refProgram » est à [0-1] dans le fichier XSD et à [0-n] sur la carte.

Pour les 2 éléments, on met [0-n], ce qui laisse le plus de liberté (en plus, ces éléments peuvent aussi être mis dans « habiliObject » et « coHabilitation »

Cardinalité de refProgram dans la définition du specialityType mise à [1-n] pour conformité avec le XSD.

Branche cohabilitation : Ajout d'un nœud intermédiaire entre « domainName » et « fixedDomain » / « openDomain » pour figurer l'élément « choice ».

Valeurs possibles dans domainNameValueSet mises à jour.

Ajout d'une branche « domainName » dans « habilitation » pour les domaines au niveau du diplômes (qui peuvent être différents de ceux du niveau cohabilitation). Branche ajoutée entre « habiliObject » et « partnerShip » (même format que dans « cohabilitation »).

### **Définition de « partnershipDataType »**

- Correction cardinalité de l'élément « refOrgUnit » dans « partnershipDataType » : mise à [1-n] pour conformité avec le XSD.

- Correction cardinalité de l'élément « country » : mise à défaut ([1-1]) pour être conforme XSD.
- Remplacement du nom du type « partnershipTypeValueSet » par « partnershipValueSet » dans la branche « partnerShip / partnerShipType ».

#### **Branche « habiliRequest »**

- Dans « habilitTiming », suppression du nœud « ? » car l'élément « choice » autour de « habiliCreatDate » et « habiliRenewDate » a été enlevé (voir plus haut la raison).
- Changement des cardinalités de tous les éléments composant la séquence « habilitTiming » : mises à [0-1] pour tous.
- Ajout d'un nœud intermédiaire dans « / habiliRequest / habiliReqType » pour indiquer le type d'objet utilisé (habiliStatusValueType)
- Ajout d'un nœud intermédiaire dans « / habiliRequest / habiliStatus » pour indiquer le type d'objet utilisé (habiliReqTypeValueType).
- Liste des valeurs pour le type « habiliStatusValueSet » : majuscules enlevées (pour être conforme au schéma).
- Liste des valeurs pour le type « habiliReqTypeValueSet » modifiée pour mise en conformité avec les valeurs indiquées dans le schéma.
- Ordre des rubriques dans « habiliReqDates » modifié pour mise en conformité avec le schéma.

Copie de la définition de partnerShipDataType dans les nœuds « partnership » et « i13nProgPartner ».

Attribut « Organization » : modification du nom du type d'objet : il s'agit de « organizationValueSet » et non de « organizationValueType ».

Ajout des attributs « id », « ident » et « language » dans la définition de « habilitation ».

#### **Version 1.1.1 (SIES le 07/12/12)**

Les 2 modifications portent sur les cardinalités de 2 éléments dans la définition du type complexe pour les spécialités (specialityType) dans le fichier « CDMFR-ext.xsd ».

**Elément « cohabilitation » dans les spécialités.** La modification annoncée ci-dessus (page 3) n'avait pas été mise en œuvre dans le schéma : c'est maintenant chose faite.

##### Action

Passage à une cardinalité [1-n].

**Elément « partnerShip » dans les spécialités:** il n'y a pas obligatoirement de partenariat pour toutes les spécialités.

La modification avait été effectuée au niveau général de l'habilitation et dans la carte heuristique, mais pas dans le schéma.

##### Action

- Passage à une cardinalité [0-n]



## ANNEXE 1 – Exemples pour l'élément habiliObject

Dans les versions 1.0.9.x, pour une habilitation qui concerne 3 programmes, on peut décrire les choses ainsi (les 3 sont valables)

:

1°) Imbrication n / 1

```
<habiliObject>
  <RefProgram>FRUAI01HA22<RefProgram>
<habiliObject>
<habiliObject>
  <RefProgram>FRUAI01HA33<RefProgram>
<habiliObject>
<habiliObject>
  <RefProgram>FRUAI01HA44<RefProgram>
<habiliObject>
```

Ou comme ceci

2°) Imbrication 1 / n

```
<habiliObject>
  <RefProgram>FRUAI01HA22<RefProgram>
  <RefProgram>FRUAI01HA33<RefProgram>
  <RefProgram>FRUAI01HA44<RefProgram>
<habiliObject>
```

Ou comme ceci

3°) Imbrication mixte

```
<habiliObject>
  <RefProgram>FRUAI01HA22<RefProgram>
<habiliObject>
<habiliObject>
  <RefProgram>FRUAI01HA33<RefProgram>
  <RefProgram>FRUAI01HA44<RefProgram>
<habiliObject>
```

Dans la version 1.0.10, la modification effectuée impose d'avoir le modèle n°2.

```
<habiliObject>
  <RefProgram>FRUAI01HA22<RefProgram>
  <RefProgram>FRUAI01HA33<RefProgram>
  <RefProgram>FRUAI01HA44<RefProgram>
<habiliObject>
```

Remarques Claude (le 3 décembre 2012)

### **Code du domaine**

Le code du domaine de la mention habilitée n'est pas précisé dans l'élément cohabilitation : seul son nom est fourni.

L'affichage dans les catalogues de formation des établissements ne va pas utiliser l'élément habilitation et ne disposera pas de cette information très importante (utilisée par tous les catalogues que j'ai observés) pour classer ses formations.

Correction simple proposée : ajouter le code du domaine dans la liste des programCode d'un élément program (très mineur car ne change pas la structure du programme) les 2 types :

- licProNatCodeValueSet : + modif de programCodeValueSet et programLibelValueSet avec la valeur licProNat
- licProSectActValueSet+ modif de programCodeValueSet et programLibelValueSet avec la valeur licProSectAct
- domainCodeValueSet+ modif de programCodeValueSet et programLibelValueSet avec la valeur domain

(fait)

### **Exemples de l'annexe 1**

Attention l'exemple tel qu'il est écrit n'est pas valide (mais ce n'est pas grave tant que ce document n'est pas publié) : il faudrait utiliser @ref pour mettre la valeur. L'utilisation d'un @ref et non du @idRef est obligatoire si le program n'est pas présent dans le document mais dans une autre demande de validation d'un autre établissement.

(pas d'actions)

### **OtherDiploma**

Je suis d'accord avec la proposition d'enlever (dans le futur) les éléments refOrgUnit et refProgram

(pas d'actions)

### **Sous-élément « habiliRequest / habiliTiming »**

L'utilisation de xs :all à la place de la séquence apporterait de la souplesse en n'imposant pas l'ordre des éléments. xs :all est possible ici car les cardinalités sont toutes [0-1] (fait)

### **Dans la RUR**

Il manque la rédaction de la règle habilitation R16

Proposition : (fait)

Seul l'un des trois éléments doit être présent (et dans ce cas être unique) :

- field
- mention
- otherDiploma

Ajouter aussi : « cf RU hab-R16 » pour licencePro et autres diplômes(fait)