

Agence du Numérique en Santé

9, rue Georges Pitard – 75015 Paris

T. 01 58 45 32 50

esante.gouv.fr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Validation sémantique | | | | |
| Poc :Schematron /ph-schematron | | | | |
| Statut : En cours | | | Classification : Restreinte | | | Version : v0.1 |

Destinataires

|  |  |
| --- | --- |
| Prénom / Nom | Entité / Direction |
| Tous les collaborateurs |  |
|  |  |
|  |  |

Documents de référence

Historique du document

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Version | Rédigé par | | Vérifié par | | Validé par | |
| 0.1 | P.NOM | Le 01/03/2018 | P.NOM | Le JJ/MM/AA | P.NOM | Le JJ/MM/AA |
| Motif et nature de la modification : **Création du document** | | | | | |
|  |  |  |  |  |  |  |
| Motif et nature de la modification : | | | | | |
|  |  |  |  |  |  |  |
| Motif et nature de la modification : | | | | | |
|  |  |  |  |  |  |  |
| Motif et nature de la modification : | | | | | |
|  |  |  |  |  |  |  |
| Motif et nature de la modification : | | | | | |
|  |  |  |  |  |  |  |
| Motif et nature de la modification : | | | | | |
|  |  |  |  |  |  |  |
| Motif et nature de la modification : | | | | | |
|  |  |  |  |  |  |  |
| Motif et nature de la modification : | | | | | |

**SOMMAIRE**

[1. Introduction 3](#_Toc73630070)

[1.1. Périmètre 3](#_Toc73630071)

[1.1.1. Document CDA 3](#_Toc73630072)

[1.1.2. Jeux de valeurs 3](#_Toc73630073)

[1.1.3. Fichier de terminologie 3](#_Toc73630074)

[2. VAlidation par schématron 4](#_Toc73630075)

[2.1. Schématron 4](#_Toc73630076)

[2.1.1. Implémentation 4](#_Toc73630077)

[2.1.2. Code 5](#_Toc73630078)

[2.2. Librairie «ph-schematron » 6](#_Toc73630079)

[2.2.1. Implémentation 6](#_Toc73630080)

[2.2.2. Code 6](#_Toc73630081)

[2.2.3. Utilisation 7](#_Toc73630082)

[Annexe : pour les annexes en fin de document 7](#_Toc73630083)

# Introduction

## Périmètre

La solution a pour objectif de vérifier que les codes utilisés dans les documents CDA et les jeux de valeurs existent bien dans les terminologies. Pour faire cette vérification, l’outil doit s’appuyer sur les fichiers de terminologies produits par le CGTS.

### Document CDA

Les données codées par rapport une terminologie se retrouvent dans les balises :

* Code :  
  L’attribut « code » représente le code dans la terminologie.  
  L’attribut « displayName» représente le libellé du code  
  L’attribut « codeSystem» représente l’identifiant de la terminologie  
  L’attribut « codeSystemName» représente le nom la terminologie.
* Value :  
  L’attribut « type » prend alors la valeur « CE » ou « CD »

L’attribut « code » représente le code dans la terminologie.  
L’attribut « displayName» représente le libellé du code  
L’attribut « codeSystem» représente l’oid de la terminologie  
L’attribut « codeSystemName» représente le nom la terminologie.

Question sur les types « CV » sur le cr bio chicungunya

Pour identifier la terminologie, c’est l’identifiant de la terminologie qui sera utilisé. L’attribut « codeSystemName» est optionnelle.

Les unités codées en UCUM sont identifiés dans la balise suivante :

* value avec l’attribut type qui prend la valeur « PQ »

### Jeux de valeurs

Les éléments codés par rapport une terminologie se retrouvent dans les balises :

* Concept:  
  L’attribut « code » représente le code dans la terminologie.  
  L’attribut « displayName» représente le libellé du code  
  L’attribut « codeSystem» représente l’oid de la terminologie  
  L’attribut « codeSystemName» représente le nom la terminologie.

Pour identifier la terminologie, c’est l’identifiant de la terminologie qui sera utilisé.

### Fichier de terminologie

Le CGTS fournie les terminologies dans un format pivot « rdf ». Ainsi, on retrouve de manière systématique les codes dans la balise :

* skos:notation

Actuellement, les terminologies disponibles dans ce format sont :

* CIM10 :   
  OID : 2.16.840.1.113883.6.3
* CISP-2 :  
  OID : 2.16.840.1.113883.6.139
* ADICAP :

OID : 1.2.250.1.213.2.11

Pour la terminologie « loinc », un fichier RDF a été créé à partir du csv disponible en télechargement sur le site :

* <https://loinc.org/downloads/>

Pour UCUM, un fichier xml a été créer à partir du du fichier csv disponible sur le site de UCUM.

# VAlidation par schématron

## Schématron

Le Schematron est un langage permettant de valider la structure d'un document XML par une liste d'assertions. Schematron permet de définir des contraintes complexes, que d'autres langages de validation XML ne permettent pas.

### Implémentation

Un schématron unique a été créé pour valider les documents CDA ou les JDV.

La variable « path\_terminologie» permet de définir le répertoire dans lequel se trouve les terminologies.

Les fichiers RDF des terminologies doivent respecter la règle de nommage suivante : oidTerminologie.rdf.  
Par exemple pour la CIM 10, le nom du fichier sera : 2.16.840.1.113883.6.3.rdf

La vérification de l’existence du code est faite par la règle abstraite « is-code-exist».

La fonction «is-code-exist».» est appelé dans les contextes suivants :

* //hl7:code
* //hl7:value[@xsi:type='CD' or @xsi:type='CE']
* //svs:Concept

### Code

<?xml version=**"1.0"** encoding=**"UTF-8"**?>

<schema xmlns=**"http://purl.oclc.org/dsdl/schematron"** xmlns:hl7=**"urn:hl7-org:v3"** xmlns:sch=**"http://purl.oclc.org/dsdl/schematron"** xmlns:local=**"http://art-decor.org/functions"** queryBinding=**"xslt2"**>

<title>**Verification sémantique**</title>

<ns uri=**"urn:hl7-org:v3"** prefix=**"hl7"**/>

<ns uri=**"urn:hl7-org:v3"** prefix=**"cda"**/>

<ns uri=**"http://www.w3.org/2001/XMLSchema-instance"** prefix=**"xsi"**/>

<ns uri=**"http://www.w3.org/2001/XMLSchema"** prefix=**"xs"**/>

<ns uri=**"urn:oid:1.3.6.1.4.1.19376.1.3.2"** prefix=**"lab"**/>

<ns uri=**"http://www.w3.org/XML/1998/namespace"** prefix=**"xml"**/>

<ns uri=**"urn:ihe:pharm"** prefix=**"pharm"**/>

<ns uri=**"urn:hl7-org:sdtc"** prefix=**"sdtc"**/>

<ns prefix=**"rdf"** uri=**"http://www.w3.org/1999/02/22-rdf-syntax-ns#"**/>

<ns prefix=**"skos"** uri=**"http://www.w3.org/2004/02/skos/core#"**/>

<ns prefix=**"svs"** uri=**"urn:ihe:iti:svs:2008"**/>

<pattern>

<title>**Verification sémantique**</title>

<!--Chemin vers le repertoire des terminologie-->

<let name=**"path\_terminologie"** value=**"'../terminologie/'"**/>

<!--Régle abstraite qui verifie de l'exitance d'un code dans la terminologie-->

<rule abstract=**"true"** id=**"is-code-exist"**>

<let name=**"theCode"** value=**"@code"**/>

<let name=**"thenNameCode"** value=**"@displayName"**/>

<let name=**"theCodeSystem"** value=**"@codeSystem"**/>

<let name=**"theCodeSystemName"** value=**"@codeSystemName"**/>

<assert role=**"error"** test=**"not (doc-available(concat($path\_terminologie, $theCodeSystem,'.rdf')) and not(exists(doc(concat($path\_terminologie, $theCodeSystem,'.rdf') )//skos:notation[text()= string($theCode)])))"**>

**CodeSystem : "**<value-of select=**"$theCodeSystem"**/>**"**

**CodeSystemName : "**<value-of select=**"$theCodeSystemName"**/>**"**

**Code : "**<value-of select=**"$theCode"**/>**"**

**DisplayName :** <value-of select=**"$thenNameCode"**/>

**Ce code n'est pas retouvé dans la terminologie**

</assert>

</rule>

<!--Document CDA : Verification sur la balise "code"-->

<rule context=**"//hl7:code"**>

<sch:extends rule=**"is-code-exist"**/>

</rule>

<!--Document CDA : Verification sur la balise "value" avec l'attribut type "CD" ou "CE"-->

<rule context=**"//hl7:value[@xsi:type='CD' or @xsi:type='CE']"**>

<sch:extends rule=**"is-code-exist"**/>

</rule>

<!--Document JDV : Verification sur la balise "Concept"-->

<rule context=**"//svs:Concept"**>

<sch:extends rule=**"is-code-exist"**/>

</rule>

<!--Document CDA : Verification sur la balise "value" pour UCUM -->

<rule context=**"//hl7:value[@xsi:type='PQ']"**>

<let name=**"theUnit"** value=**"@unit"**/>

<assert role=**"error"** test=**"(exists(doc(concat($path\_terminologie, 'ucum.xml') )//code[text()= string($theUnit)]))"**>

**Code : "**<value-of select=**"$theUnit"**/>**"**

**Ce code n'est pas dans UCUM**

</assert>

</rule>

</pattern>

</schema>

## Librairie «ph-schematron »

Un des objectifs du POC est d’évaluer cette librairie qui permet de faire de la validation sur schematron.  
Elle dispose entre autre de deux modes :

* Un mode « pure »  
  Avec une implemenatation pure java et donc de meilleur performance
* Un mode « xslt »  
  Avec une approche classique

Lien vers la librairie :   
<https://github.com/phax/ph-schematron>

### Implémentation

Une classe « validate » a été implémenté en java. Cette dernière utilises les méthodes de la librairie « ph-schematron» pour faire la validation.

Cette dernière offre aussi la possibilité de « parser » la réponse de la validation à travers une feuille XSL

### Code

//Mode pure à false

boolean modePure **=** **false;**

final ISchematronResource aResPure**;**

**if(**modePure**)**

aResPure **=** SchematronResourcePure**.**fromFile **(**fileShematron**);**

**else**

aResPure **=** SchematronResourceSCH**.**fromFile **(**fileShematron**);**

**if(!**aResPure**.**isValidSchematron **())**

**throw** **new** IllegalArgumentException **(**"Invalid Schematron!"**);**

final Document aDoc **=** aResPure**.**applySchematronValidation **(new** StreamSource **(**fCda**));**

//Validation et écriture du fichier

TransformerFactory tf **=** TransformerFactory**.**newInstance**();**

Transformer transformer**;**

transformer **=** tf**.**newTransformer**();**

transformer**.**transform**(new** DOMSource**(**aDoc**),** **new** StreamResult**(**outStream**));**

//Parsing XSl si les options sont présentes

**if** **(**commandLine**.**hasOption**(**"xsl"**)&&** commandLine**.**hasOption**(**"outputxsl"**)){**

final Source xmlSource **=** **new** javax**.**xml**.**transform**.**stream**.**StreamSource**(**commandLine**.**getOptionValue**(**"o"**));**

final Source xsltSource **=** **new** javax**.**xml**.**transform**.**stream**.**StreamSource**(**commandLine**.**getOptionValue**(**"xsl"**));**

final Result result **=** **new** javax**.**xml**.**transform**.**stream**.**StreamResult**(**commandLine**.**getOptionValue**(**"outputxsl"**));**

TransformerFactory tFactory **=** TransformerFactory**.**newInstance**();**

transformer **=** tf**.**newTransformer**(**xsltSource**);**

transformer**.**setParameter**(**"title"**,**fCda**.**getName**());**

transformer**.**transform**(**xmlSource**,** result**);**

**}**

### Utilisation

Lors de l’appel de la classe, les paramètres suivants peuvent être passés :

* -i : Fichier à valider (CDA ou JDV)
* -s : Fichier du schématron
* -o : Fichier de résultat de la validation
* -xsl : feuille xsl pour transformer le résultat de la validation
* - outputxsl : Fichier de resulat de la transformation XSL

Exemple :

java -cp ./\* validate.java -i -BIO\_v2.0\_Chikungunya.XML -s schematron\terminologie.sch -o output/CR-bio.svrl -xsl schematron/svrl-to-html.xsl -outputxsl output/crbio.html

### Conclusion

Le mode « pure » plus rapide ne reconnais pas les requêtes complexes dans les schematrons et donc non adapaté dans notre cas.