

SHACL

SHApe Constraints Language

Thomas Francart, **SPARNA**

thomas.francart@sparna.fr

SHACL est un « langage pour valider un graphe RDF à l'aide d'un ensemble de contraintes ».

Les définitions SHACL sont une
« **description des données**
qui satisfont ces contraintes »

Validation

Génération de formulaires

Génération de code informatique

SHACL est aussi exprimé en **RDF** !

L'espace de nom SHACL est

<http://www.w3.org/ns/shacl#>

Avec le préfixe typique **sh:**

La spécification SHACL est à

<https://www.w3.org/TR/shacl/>

et à

<https://www.w3.org/TR/shacl-af/>

(advanced features)

```
ex:PersonShape a sh:NodeShape ;
  sh:targetClass ex:Person ;
  sh:property [
    sh:path ex:numero_seco ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
    sh:pattern "^\\d{3}-\\d{2}-\\d{4}$" ;
  ] ;
  sh:property [
    sh:path ex:travaille_a ;
    sh:class ex:Company ;
    sh:nodeKind sh:IRI ;
  ] ;
  sh:closed true ;
  sh:ignoredProperties ( rdf:type ) .
```

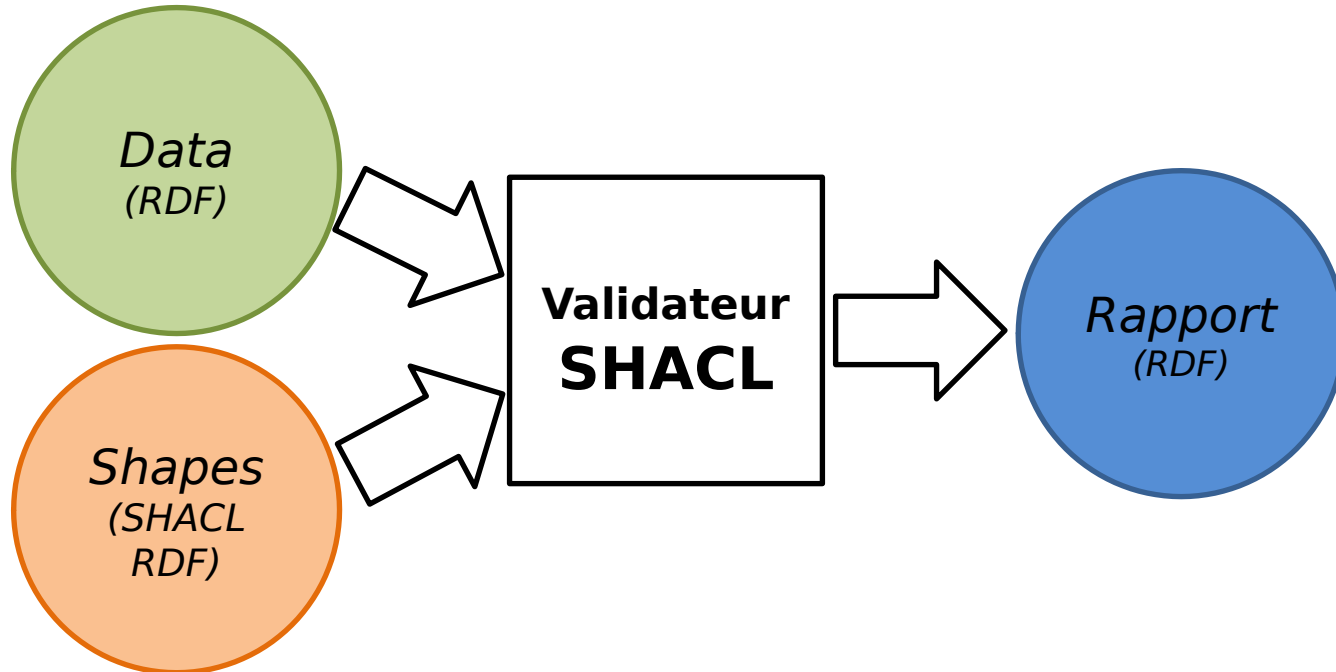
S'applique aux
Personnes

Maximum 1 numéro de
sécu, de la forme
xxx-xx-xxxx

«travaille_a» doit être
une URI avec la classe
« Company »

Les personnes ne peuvent
avoir aucune autre propriété à
part numero_seco, travaille_a,
et rdf:type

La validation de données RDF avec des Shapes se fait par un **validateur SHACL** et produit un **rapport de validation** en RDF.



Les « **Targets** » des Shapes (cibles)

Une Shape s'applique à un ensemble de

« **focus nodes** »

qui sont déterminés par la

« **target definition** »

de la Shape



La définition la plus simple d'une Target est de **cibler toutes les instances d'une classe** avec **sh:targetClass**.

Mais d'autres définition de cibles sont possibles

```
ex:PersonShape
  a sh:NodeShape ;
  sh:targetClass ex:Person ;
  ...
```

All Persons

```
ex:labelShape
  a sh:NodeShape ;
  sh:targetSubjectOf skos:preflabel ;
  ...
```

All subjects of
prefLabel

```
ex:worksForShape
  a sh:NodeShape
  sh:targetObjectOf ex:worksFor ;
  ...
```

All values of
worksFor

```
ex:PersonShape
  a sh:NodeShape ;
  sh:targetNode ex:Alice .
```

Only Alice

Pour trouver les instances, SHACL cherche aussi les instances des **sous-classes**.

```
ex:PersonShape  
  a sh:NodeShape ;  
  sh:targetClass ex:Person .
```

```
ex:Doctor rdfs:subClassOf ex:Person .  
ex:Maboul a ex:Doctor .
```

*ex:Maboul est une
cible de la Shape
ci-dessus*

Pour que ce mécanisme fonctionne, il faut que le validateur SHACL **connaisse les sous-classes** !

[sh:targetClass implicite]

Si une Shape est aussi une Classe
(`rdfs:Class`), alors toutes ses instances sont
implicitement ciblées

```
ex:Person  
  a rdfs:Class, sh:NodeShape .
```

```
ex:PersonShape  
  a sh:NodeShape ;  
  sh:targetClass ex:Person .
```

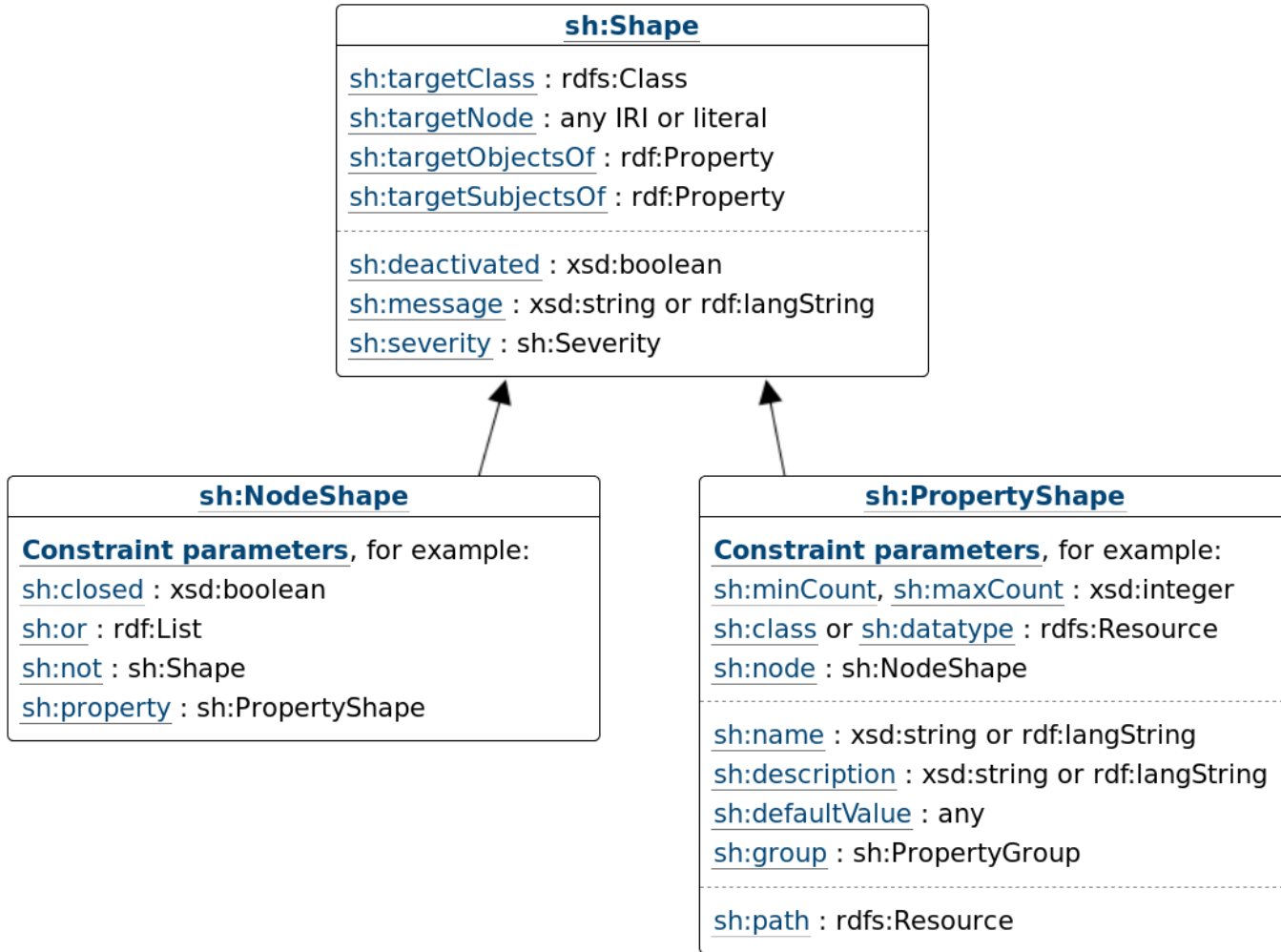
Equivalent à

Contraintes de **noeuds**
et
Contraintes de **propriétés**

Il y a **2 types de Shapes**

les **sh:NodeShape** spécifient des contraintes sur les nœuds (ressources)

Les **sh:PropertyShape** spécifient les contraintes sur les (valeurs de) propriétés



Une NodeShape

spécifie les contraintes sur le nœud lui-même

```
skos:Concept
  a rdfs:Class, sh:NodeShape ;
  sh:nodeKind sh:IRI.
```

Le nœud
doit être
une IRI

Une PropertyShape

Spécifie les contraintes d'une propriété (ou **property path**)
indiqué par **sh:path**

```
ex:prefLabelShape
  a sh:PropertyShape ;
  sh:path skos:prefLabel;
  # (constraints on prefLabel here)
```

prefLabel
must ...

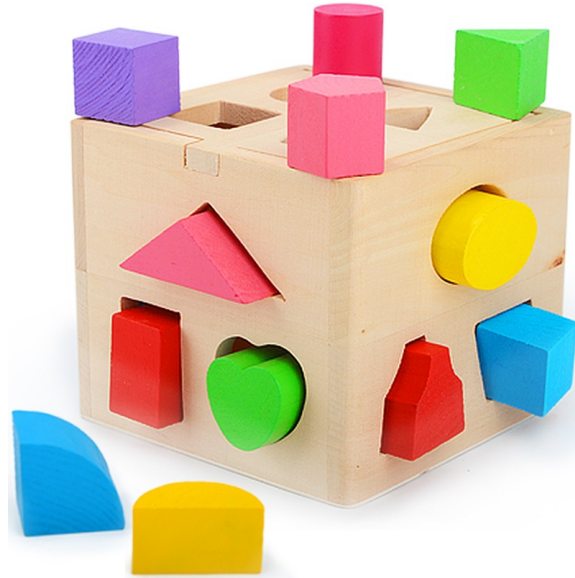
Les NodeShapes référencent leurs PropertyShapes via **sh:property**

```
skos:Concept ←  
  a rdfs:Class, sh:NodeShape ;  
  sh:nodeKind sh:IRI;  
  sh:property [  
    sh:path skos:prefLabel;  
    sh:uniqueLang true ;  
  ] ;  
  sh:property [  
    sh:path dct:created;  
    sh:minCount 1 ;  
  ] .
```

Node
Shape

Property
Shapes

Les contraintes de base fournies par SHACL



« Core Constraint Components »

sh:minCount / maxCount

Cardinalités min et max : il doit y avoir au moins / au plus X valeurs de la propriété

```
skos:Concept
  a rdfs:Class, sh:NodeShape ;
  sh:property [
    sh:path skos:prefLabel ;
    sh:minCount 1 ;
  ] ;
```

Un Concept doit avoir au moins 1 prefLabel

sh:class

La valeur d'une propriété doit être une instance de la classe spécifiée

```
ex:Person
  a rdfs:Class, sh:NodeShape
  sh:property [
    sh:path ex:worksFor ;
    sh:class ex:Company ;
  ] ;
```

Toutes les valeurs
de ex:worksFor
doivent être des
instances de
ex:Company

sh:datatype

La valeur d'une propriété doit avoir le datatype spécifiée

```
foaf:Person
  a rdfs:Class, sh:NodeShape ;
  sh:property [
    sh:path foaf:age ;
    sh:datatype xsd:integer ;
  ] ;
```

Les valeurs de
foaf:age
doivent avoir le
datatype
xsd:integer

sh:nodeKind

La valeur (propriété ou NodeShape) doit être une IRI, un Literal ou un blank node

```
skos:Concept
  a rdfs:Class, sh:NodeShape ;
  sh:property [
    sh:path skos:notation ;
    sh:nodeKind sh:Literal ;
  ] ;
```

Les valeurs de
skos:notation
doivent être des
Literal

```
skos:Concept
  a rdfs:Class, sh:NodeShape ;
  sh:nodeKind sh:IRI .
```

Les SKOS Concepts
doivent être des IRIs

sh:pattern

Vérifie qu'une URI (ou un littéral) correspond à une regex

```
foaf:Person
  rdfs:Class, sh:NodeShape ;
  sh:property [
    sh:path ex:ssn ;
    sh:pattern "^\\d{3}-\\d{2}-\\d{4}$" ;
  ] ;
```

Numéro de sécu
doit être
xxx-xx-xxxx

```
epono:MEP
  rdfs:Class, sh:NodeShape ;
  sh:pattern "^http://data.euoparl.europa.eu/person/MEP_.*$" ;
```

Les membres du
parlement doivent
avoir une URI de
cette forme

Exercise

```
@prefix sh: <http://www.w3.org/ns/shacl#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct: <http://purl.org/dc/terms/>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
```

```
foaf:Document
  a rdfs:Class, sh:NodeShape ;
  sh:nodeKind sh:IRI ;
  .
```

- On décrit des Documents (foaf:Document)
- « Tous les documents doivent être des IRI »
- On travaille avec le Dublin Core étendu avec le préfix « dct: »

- Ajoutez les contraintes suivantes :

- 1) Tous les documents doivent avoir au moins un titre (dct:title) ;
- 2) Tous les documents peuvent avoir un auteur (dct:creator) qui est une Personne (foaf:Person) ;
- 3) Tous les documents doivent avoir une et une seule date de création (dct:created) qui est un literal de type « xsd:date » ;

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct: <http://purl.org/dc/terms/>.
@prefix sh: <http://www.w3.org/ns/shacl#>.
```

```
foaf:Document
  a rdfs:Class, sh:NodeShape ;
  sh:nodeKind sh:IRI ;
  sh:property [
    sh:path dct:title ;
    ...
  ] ;
  sh:property [
    sh:path dct:creator ;
    ...
  ] ;
  sh:property [
    sh:path dct:created ;
    ...
  ] ;
.
```

3 contraintes
donc 3 blocs
« sh:property »

Tous les documents doivent avoir au moins un titre (dct:title)

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.  
@prefix dct: <http://purl.org/dc/terms/>.  
@prefix sh: <http://www.w3.org/ns/shacl#>.  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
```

```
foaf:Document  
  a rdfs:Class, sh:NodeShape ;  
  sh:nodeKind sh:IRI ;  
  sh:property [  
    ...  
  ] ;
```

Tous les documents doivent avoir au moins un titre

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct: <http://purl.org/dc/terms/>.
@prefix sh: <http://www.w3.org/ns/shacl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
```

```
foaf:Document
  a rdfs:Class, sh:NodeShape ;
  sh:nodeKind sh:IRI ;
  sh:property [
    sh:path dct:title ;
    sh:minCount 1;
  ] ;
```

.

Tous les documents peuvent avoir un auteur qui est une Personne

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct: <http://purl.org/dc/terms/>.
@prefix sh: <http://www.w3.org/ns/shacl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
```

```
foaf:Document
  a rdfs:Class, sh:NodeShape ;
  sh:nodeKind sh:IRI ;
  sh:property [
    sh:path dct:title ;
    sh:minCount 1;
  ] ;
  sh:property [
    sh:path dct:creator ;
    ...
  ] ;
.
```

Tous les documents peuvent avoir un auteur qui est une Personne

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct: <http://purl.org/dc/terms/>.
@prefix sh: <http://www.w3.org/ns/shacl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
```

```
foaf:Document
  a rdfs:Class, sh:NodeShape ;
  sh:nodeKind sh:IRI ;
  sh:property [
    sh:path dct:title ;
    sh:minCount 1;
  ] ;
  sh:property [
    sh:path dct:creator ;
    sh:class foaf:Person ;
    sh:nodeKind sh:IRI ;
  ] ;
```

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct: <http://purl.org/dc/terms/>.
@prefix sh: <http://www.w3.org/ns/shacl#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
```

```
foaf:Document
  a rdfs:Class, sh:NodeShape ;
  sh:nodeKind sh:IRI ;
  sh:property [
    sh:path dct:title ;
    sh:minCount 1;
  ] ;
  sh:property [
    sh:path dct:creator ;
    sh:class foaf:Person ;
    sh:nodeKind sh:IRI ;
  ] ;
  sh:property [
    sh:path dct:created ;
    ...
  ] ;
```

Tous les documents
ont une et une
seule date de
création qui est une
date


```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct: <http://purl.org/dc/terms/>.
@prefix sh: <http://www.w3.org/ns/shacl#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
```

```
foaf:Document
  a rdfs:Class, sh:NodeShape ;
  sh:nodeKind sh:IRI ;
  sh:property [
    sh:path dct:title ;
    sh:minCount 1;
  ] ;
  sh:property [
    sh:path dct:creator ;
    sh:class foaf:Person ;
    sh:nodeKind sh:IRI ;
  ] ;
  sh:property [
    sh:path dct:created ;
    sh:datatype xsd:date ;
    sh:nodeKind sh:Literal ;
    sh:minCount 1;
    sh:maxCount 1;
  ] ;
```

Tous les documents
ont une et une
seule date de
création qui est une
date

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct: <http://purl.org/dc/terms/>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix ex: <http://example.org/>.
```

```
ex:doc1 a foaf:Document;
  dct:title "Support de formation SHACL" ;
  dct:created "2020-11-30"^^xsd:date ;
  dct:creator ex:Thomas ;
```

```
.
```

Ce graphe est-il valide ?
Pourquoi ?

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct:  <http://purl.org/dc/terms/>.
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#>.
@prefix ex:   <http://example.org/>.
```

```
ex:doc1 a foaf:Document;
  dct:title "Support de formation SHACL" ;
  dct:created "2020-11-29"^^xsd:date ;
  dct:created "2020-11-30"^^xsd:date ;
  dct:creator ex:Thomas ;
```

.

```
ex:Thomas a foaf:Person .
```

Ce graphe est-il valide ?
Pourquoi ?

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct:  <http://purl.org/dc/terms/>.
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#>.
@prefix ex:   <http://example.org/>.
```

```
ex:doc1 a foaf:Document;
  dct:title "" ;
  dct:created "2020-11-30"^^xsd:date ;
  dct:creator ex:Thomas ;
```

.

```
ex:Thomas a foaf:Person .
```

Ce graphe est-il valide ?
Pourquoi ?

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct:  <http://purl.org/dc/terms/>.
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#>.
@prefix ex:   <http://example.org/>.
```

```
ex:doc1 a foaf:Document;
  dct:title "Support de formation SHACL" ;
  dct:created "2020-11-30" ;
  dct:creator ex:Thomas ;
```

```
.
```

```
ex:Thomas a foaf:Person .
```

Ce graphe est-il valide ?
Pourquoi ?

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct:  <http://purl.org/dc/terms/>.
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#>.
@prefix ex:   <http://example.org/>.
```

```
ex:doc1 a foaf:Document;
  dct:title "Support de formation SHACL" ;
  dct:created "2020-11-30"^^xsd:date ;
  dct:creator [
    a foaf:Person ;
    foaf:name "Thomas" ;
  ] .
```

Ce graphe est-il valide ?
Pourquoi ?

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct:  <http://purl.org/dc/terms/>.
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#>.
@prefix ex:   <http://example.org/>.
```

```
ex:doc1 a foaf:Document;
  dct:title "Support de formation SHACL" ;
  dct:created "2020"^^xsd:date ;
  dct:creator ex:Thomas ;
```

.

```
ex:Thomas a foaf:Person .
```

Ce graphe est-il valide ?
Pourquoi ?

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct:  <http://purl.org/dc/terms/>.
@prefix sh:   <http://www.w3.org/ns/shacl#>.
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
```

```
foaf:Document
  a rdfs:Class, sh:NodeShape ;
  sh:nodeKind sh:IRI ;
  sh:property [
    sh:path dct:title ;
    sh:minCount 1;
  ] ;
  sh:property [
    sh:path dct:creator ;
    sh:class foaf:Person ;
    sh:nodeKind sh:IRI ;
  ] ;
  sh:property [
    sh:path dct:created ;
    sh:datatype xsd:date ;
    sh:nodeKind sh:Literal ;
    sh:minCount 1;
    sh:maxCount 1;
  ] ;
.
```

Ajoutez une nouvelle Shape pour spécifier que les URIs des Personnes (foaf:Person) doivent être de la forme

<http://example.org/person/....>


```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct: <http://purl.org/dc/terms/>.
@prefix sh: <http://www.w3.org/ns/shacl#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
```

```
foaf:Person
  a rdfs:Class, sh:NodeShape ;
  sh:pattern "^http://example.org/person/.*$" ;
.
```

Ajoutez une nouvelle Shape pour spécifier que les URIs des Personnes (foaf:Person) doivent être de la forme

[http://example.org/person/....](http://example.org/person/...)

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix dct:  <http://purl.org/dc/terms/>.
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#>.
@prefix ex:   <http://example.org/>.
```

```
ex:doc1 a foaf:Document;
  dct:title "Support de formation SHACL" ;
  dct:created "2020-11-30"^^xsd:date ;
  dct:creator ex:Thomas,
    <http://example.org/person/Olivier> ;
```

.

```
ex:Thomas a foaf:Person .
```

```
<http://example.org/person/Olivier> a foaf:Person .
```

Ce graphe est-il valide ?
Pourquoi ?

Outils

API

- TopQuadrant SHACI API :

<https://github.com/TopQuadrant/shacl>

Java, API Jena, implémentation de référence des éditeurs de la recommandation, complète, pas forcément performante

- RDF4J SHACL :

<https://rdf4j.org/documentation/programming/shacl/>

Java, API RDF4J, moins complète mais plus performante, validation à l'insertion/suppression (vs. Validation du graphe complet)

Playgrounds

- SHACL Playground : <https://shacl.org/playground/>
 - ~ Interface de test très technique (pas très user-friendly)
- SHACL Play : <http://shacl-play.sparna.fr>
 - ~ Orienté end-user : formulaire de soumission, rapport de validation
 - ~ Bonus : diagramme des shapes, badges, permalinks, etc.
 - ~ Testons en copiant-collant les exemples précédents

Support de SHACL dans les triplestores

- GraphDB supporte SHACL :
 - ~ <https://graphdb.ontotext.com/documentation/free/shacl-validation.html>
- Jena Fuseki supporte SHACL :
 - ~ <https://jena.apache.org/documentation/shacl/index.html>

Utiliser SHACL Play pour valider les exemples ci-dessus

<https://shacl-play.sparna.fr/play/validate>

Copier-coller un exemple RDF et le contenu du fichier de Shapes (ou uploader le fichier de Shapes)

Validation results of 1 shapes

Download validation report in [CSV](#) [Turtle](#) [RDF/XML](#)

1 Violation

Violation 1 "Value must be an instance of foaf:Person" (dct:creator) [see details](#)

Violation 1 "Value must be an instance of foaf:Person" (dct:creator)

- Shape : 55e2b831f907dfb5fdde9c93fbf612c5
- Constraint : sh:ClassConstraintComponent

Focus node	Property or path	Value
ex:doc1	dct:creator	ex:Thomas

Des badges de validation SHACL pour votre Github

ep-graph-poc

Dataset 1 Controlled Vocabularies

Direct link to SHACL SVG diagram : https://shacl-play.sparna.fr/play/view?url=https://github.com/sparna-git/ep-graph-poc/raw/main/01-Dataset1_CV/04-SHACL/Dataset1_CV-Shapes.ttl

Civility : shacl-play 19 violations

Gender : shacl-play Conformant

Country : shacl-play 4 violations

Concept_Status : shacl-play 1 violation

Dataset 2 - MEP

Direct link to SHACL SVG diagram : https://shacl-play.sparna.fr/play/view?url=https://github.com/sparna-git/ep-graph-poc/raw/main/02-Dataset2_MEP/04-SHACL/Dataset2_MEP-Shapes.ttl

MEP : shacl-play Conformant

Assistants : shacl-play Conformant

PoliticalGroups : shacl-play 3 violations

Editer / maintenir des Shapes

Editeurs

- TopBraid Composer :
 - ~ <https://www.topquadrant.com/topbraid-composer-install/>
 - ~ Editeurs de la recommandation SHACL
 - ~ Intègre l'API sus-mentionnée

Solution de maintenance « low-tech »

- Tableau Excel + convertisseur de Excel vers RDF

~ SKOS Play Excel 2 RDF converter :

<https://labs.sparna.fr/skos-play/convert>

- Cocher « Ignorer les post-traitements SKOS »

~ Prendre exemple sur le template fourni ou sur les Shapes

OpenArchaeo :

<http://openarchaeo.huma-num.fr/federation/resources/shapes/OpenArchaeo-Shapes.xlsx>

Une feuille avec les NodeShapes

	A	B	C	D	E	F	G
1	URI du Graphe	http://openarchaeo.huma-num.fr/federation/shapes/openarchaeo-shapes					
2	@prefix	crm	http://www.cidoc-crm.org/cidoc-crm/				
3	@prefix	crmarch	http://www.lcs.forth.gr/isl/CRMarchaeo/				
4	@prefix	crmsci	http://www.lcs.forth.gr/isl/CRMsci/				
5	@prefix	crmba	http://www.lcs.forth.gr/isl/CRMba/				
6	@prefix	sh	http://www.w3.org/ns/shacl#				
7	@prefix	oash	federation/shapes/openarchaeo-shapes/				
8							
9	Cette feuille spécifie la liste de <u>Shapes</u> des classes d' <u>OpenArcheo</u>						
10							
11	<i>L'URI de la Shape</i>	<i>Le type de la Shape</i>	<i>L'ensemble des objets contrôlés par cette Shape</i>	<i>Un libellé pour la Shape</i>	<i>L'ordre dans lequel la Shape sera affichée</i>	<i>La description textuelle de la Shape</i>	<i>Le type de nœuds qui doivent répondre à cette Shape (toujours sh:IRI)</i>
12	URI	rdf:type	sh:targetClass	rdfs:label@fr	sh:order^^xsd:Integer	rdfs:comment@fr	sh:nodeKind
13	oash:Site	sh:NodeShape	crm:E27_Site	E27 Site		Il est SOUHAITABLE que les instances de E27_Site soient des IRIs (pas des 1 nœuds anonymes)	sh:IRI
14	oash:PreferredIdentifier	sh:NodeShape	crm:E42_Identifier	E42 Identifier (valeurs de P48 has_preferred_identifier)		Il est SOUHAITABLE que les instances de E42_Identifier soient des IRIs (pas 2 des nœuds anonymes)	sh:IRI
15	oash:EncounterEvent	sh:NodeShape	crmsci:S19_Encounter_Event	S19 Encounter Event		Il est SOUHAITABLE que les instances de S19_Encounter_Event soient des 3 IRIs (pas des nœuds anonymes)	sh:IRI
16	oash:Person	sh:NodeShape	crm:E21_Person	E21 Person		Il est SOUHAITABLE que les instances de E21_Person soient des IRIs (pas 4 des nœuds anonymes)	sh:IRI
17	oash:LegalBody	sh:NodeShape	crm:E40_Legal_Body	E40 Legal Body		Il est SOUHAITABLE que les instances de E40_Legal_Body soient des IRIs 5 (pas des nœuds anonymes)	sh:IRI
18						Il est SOUHAITABLE que les instances	

Une feuille avec les PropertyShapes

12	URI de la contrainte	Le prédicat ou path sur lequel la contrainte s'applique	La Shape à laquelle la contrainte est attachée	La description textuelle de la contrainte	La cardinalité minimum que le prédicat ou le path doit avoir	La cardinalité maximum que le prédicat ou le path doit avoir	Classe attendue comme valeur de la propriété, quand il n'y en a qu'une	Classes attendues comme valeur de la propriété, quand il y en a plusieurs	Le type de nœud que les valeurs devraient avoir (sh:IRI or sh:Literal)	Filigrane
13	URI	sh:path	^sh:property(separator=",")	sh:description	sh:minCount^^xsd:integer	sh:maxCount^^xsd:integer	sh:class	sh:or	sh:nodeKind	
21	oash:P21	crm:P70i_is_documented_in	oash:Site	Le P70i_is_documented_in DOIT être un E31_Document			crm:E31_Document			
22										
23	Contraintes sur les Evenements de découverte (S19_Encounter_Event)									
24	oash:P24	crm:P2_has_type	oash:EncounterEvent	Le P2_has_type sur un événement de découverte DOIT être unique, et typé avec une et une seule URI des <u>Pactols</u>		1				
25	oash:P25	crm:P14_carried_out_by	oash:EncounterEvent	Le P14_carried_out_by sur un Evénement de découverte DOIT être un E21_Person			crm:E21_Person			
26	oash:P26	crmsci:O19_has_found_object	oash:EncounterEvent	Le O19_has_found_object DOIT être soit un B1_Built_Work, E25_Man-Made_Feature, E22_Man-Made_Object ou A2_Stratigraphic_Volume_Unit				([sh:class crmba:B1_Built_Work] [sh:class crm:E25_Man-Made_Feature] [sh:class crm:E22_Man-Made_Object] [sh:class crmarch:A2_Stratigraphic_Volume_Unit])		
27	oash:P27	crm:P8_took_place_on_or_within	oash:EncounterEvent	Le P8_took_place_on_or_within DOIT être un Site			crm:E27_Site			
28	oash:P28	[sh:alternativePath (crm:P8_took_place_on_or_within [sh:inversePath crm:P8i_witnessed])]	oash:EncounterEvent	Le P8_took_place_on_or_within ou son inverse DOIT être unique		1				
29	oash:P29	crm:P48_has_preferred_identifier	oash:EncounterEvent	découverte doit être unique et être un E42_Identifier			1 crm:E42_Identifier			
30	oash:P30	crm:P70i_is_documented_in	oash:EncounterEvent	Le P70i_is_documented_in DOIT être un E31_Document			crm:E31_Document			
31	Contraintes sur les Personnes (E21_Person)									
32	oash:P32	crm:P14i_performed	oash:Person	Le P14i_performed sur une Personne DOIT être soit un S19_Encounter_Event soit un E65_Creation				([sh:class crmsci:S19_Encounter_Event] [sh:class crm:E65_Creation])		
33	oash:P33	[sh:alternativePath (crm:P14i_performed [sh:inversePath crm:P14_carried_out_by])]	oash:Person	Une Personne DOIT avoir au moins un P14i ou un ^P14		1				
34	oash:P34	crm:P107i_is_current_or_former_member	oash:Person	Le P107i_is_current_or_former_member DOIT être un E40_Legal_Body			crm:E40_Legal_Body			

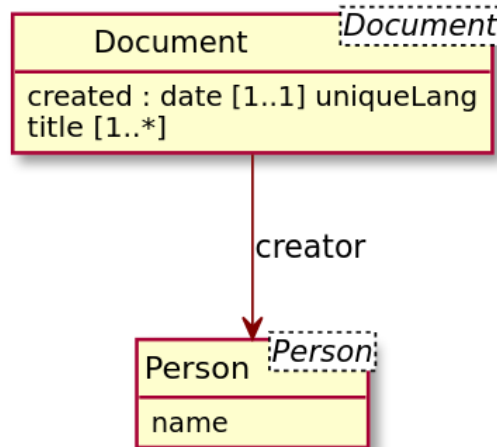
Exercice de manipulation du tableau Excel

- Ajoutez des contraintes sur les foaf:Person :
 - ~ Les Personnes doivent avoir 1 et 1 seul foaf:name
 - ~ Les Personnes peuvent avoir un foaf:firstName et un foaf:familyName
- Reconvertir en ttl avec le convertisseur en ligne
 - ~ <https://labs.sparna.fr/skos-play/convert>
 - Cocher « Ignorer les post-traitements SKOS »
- Revalider un exemple de fichier de données avec SHACL-Play

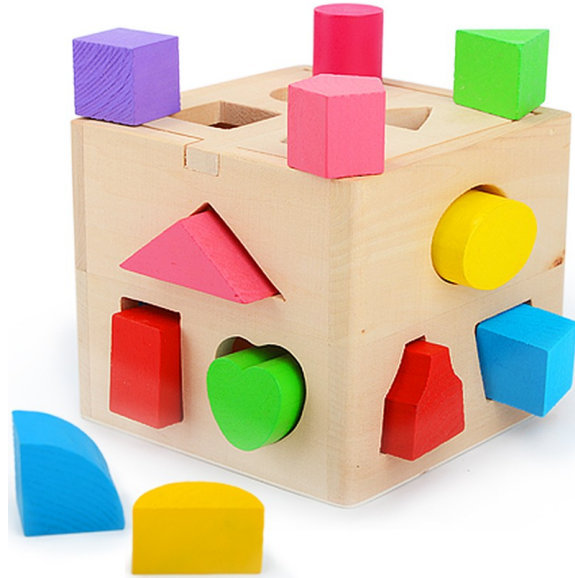
Voir des Shapes

SHACL Play viewer

- <https://shacl-play.sparna.fr/play/view>
- Uploader le fichier de Shapes converti



Les autres contraintes de base fournies par SHACL



« Core Constraint Components »

sh:closed

Pour travailler en monde fermé.

Les seules propriétés utilisées sont celles explicitement listées dans la Shape.

A combiner avec **sh:ignoredProperties** pour ignorer `rdf:type`

```
foaf:Person
  a sh:NodeShape, rdfs:Class ;
  sh:closed true ;
  sh:ignoredProperties (rdf:type) ;
  sh:property [
    sh:path ex:firstName ;
  ] ;
  sh:property [
    sh:path ex:lastName ;
  ] .
```

Les Personnes ont
seulement
firstName,
lastName, and
rdf:type

sh:node

Value must conform to the given Shape

```
ex:PersonShape
  sh:NodeShape ;
  sh:targetClass foaf:Person ;
  sh:property [
    sh:path ex:worksFor ;
    sh:node ex:CompanyShape ;
  ] .

ex:CompanyShape
  sh:NodeShape ;
  ...
.
```

Every value of
ex:worksFor must
be conformant with
CompanyShape

sh:[min|max]Length

Checks the length of a String

(inclusive).

```
ex:PasswordExampleShape
  a sh:NodeShape ;
  sh:targetNode ex:Bob, ex:Alice ;
  sh:property [
    sh:path ex:password ;
    sh:minLength 8 ;
    sh:maxLength 10 ;
  ] .
```

Passwords must
be between 8 and
10 characters.

sh:languageIn

Checks that the language of values are one of the specified languages

```
ex:IrelandLanguagesShape
  a sh:PropertyShape ;
  sh:path skos:prefLabel ;
  sh:languageIn ( "en" "ga" )
;
```

Checks that
prefLabels are
either in English
or Irish.

sh:uniqueLang

Checks that there is at most one value per language.

```
ex:PrefLabelShape
  a sh:PropertyShape ;
  sh:path skos:prefLabel ;
  sh:uniqueLang true ;
.
```

No 2 prefLabels
can have the
same language

sh:lessThan / sh:lessThanOrEquals

All values of a property are less than (or equal) all values of another property

```
ex:IntervalShape
  a sh:NodeShape ;
  sh:property [
    sh:path ex:startDate ;
    sh:lessThanOrEquals ex:endDate ;
  ] .
```

*An interval
must start
before it ends*

sh:hasValue / sh:in

At least one value is equal to the given value, or in the provided list.

```
ex:StanfordGraduate
  a sh:NodeShape ;
  sh:property [
    sh:path ex:alumniOf ;
    sh:hasValue ex:Stanford ;
  ] .
```

Stanford
graduates are
alumniOf
Stanford

```
ex:MyLittlePonyShape
  a sh:NodeShape ;
  sh:targetClass ex:littlePony;
  sh:property [
    sh:path ex:color ;
    sh:in ( ex:Pink ex:Purple ) ;
  ] .
```

Little ponies are
pink or purples

sh:equals / sh:disjoint

The set of values of a property is equal (*resp. disjoint*) with the set of values of another property.

```
ex:LabelsShape
  a sh:PropertyShape ;
  sh:path skos:prefLabel ;
  sh:disjoint skos:altLabel
```

prefLabels and
altLabels are
disjoint

```
ex>TitleShape
  a sh:PropertyShape ;
  sh:path foaf:name ;
  sh:equal skos:prefLabel .
```

foaf:name and
skos:prefLabels
are equal

sh:qualified[Min|Max]Count

sh:qualifiedValueShape

... minimum/maximum number of values
that satisfy a Shape

```
sh:property [  
  sh:path ex:parent ;  
  sh:minCount 2 ;  
  sh:maxCount 2 ;  
  sh:qualifiedMinCount 1 ;  
  sh:qualifiedValueShape [  
    sh:path ex:gender ;  
    sh:hasValue ex:female ;  
  ] ;  
] .
```

A Person must have
exactly 2 parents,
one of which is a
female

SPARNA

thomas.francart@ **sparna.fr**

Web de **données**
Structuration des **informations**
Accès aux **connaissances**