

# Document et Web Sémantique

Durée : 3h00

## Documents autorisés :

- Supports de cours annotés ;
- Sites web : <http://www.w3.org/>, <http://www.w3schools.com/>, <http://schema.org/>, <http://www.swi-prolog.org/>, <https://developers.google.com/structured-data/testing-tool/>

## Remarques :

- Veuillez lire attentivement les questions avant de répondre.
- Le barème donné est un barème indicatif qui pourra évoluer lors de la correction.

## Informations

### Avant de commencer l'examen

Connectez vous à la machine et suivez les instructions suivantes :

- copiez l'archive `Genealogie.tar.gz` qui se trouve dans `/opt/files` dans votre répertoire (commande `cp`) ;
- après l'avoir décompressée (commande `tar` avec les options `zxvf`), renommez le répertoire en lui donnant comme nom : votre nom suivi de votre prénom, sans espace, sans caractères accentués et en minuscule sauf pour les premières lettres du prénom et du nom. Par exemple, si « Paul Du Villaré » devait passer cet examen, il nommerait son répertoire « DuvillarePaul » ;
- configurer le proxy (`asi-examen.insa-rouen.fr`, port 80) de votre navigateur web (sous Firefox, menu « Édition » puis « Préférences », onglet « Avancé » et « Réseau », puis bouton « Paramètres » enfin « Configuration manuelle du proxy », cocher aussi « utiliser ce serveur proxy pour tous les protocoles ») afin de pouvoir accéder aux sites précédemment listés.

### À la fin de l'examen

Cinq minutes avant la fin de l'examen :

- assurez vous que tous les fichiers sont au format `UTF-8` (c'est par défaut le cas) ;
- créez une archive de votre répertoire (commande `tar` avec les options `zcvf`), que vous nommerez comme le répertoire (avec le suffixe `.tar.gz`) ;
- supprimer l'utilisation du proxy (« Pas de proxy ») ;
- déposez sur moodle votre archive (cours « Document et Web sémantique », section « examens »).

## Contexte

L'objectif de cet examen est de travailler sur la généalogie présentée par la figure 1 où les flèches représentent les filiations.

### 1 Validation d'un document XML (5 points)

L'archive donnée contient entre autres :

- le fichier XML `genealogie.xml` qui décrit les personnes et leurs filiations ;

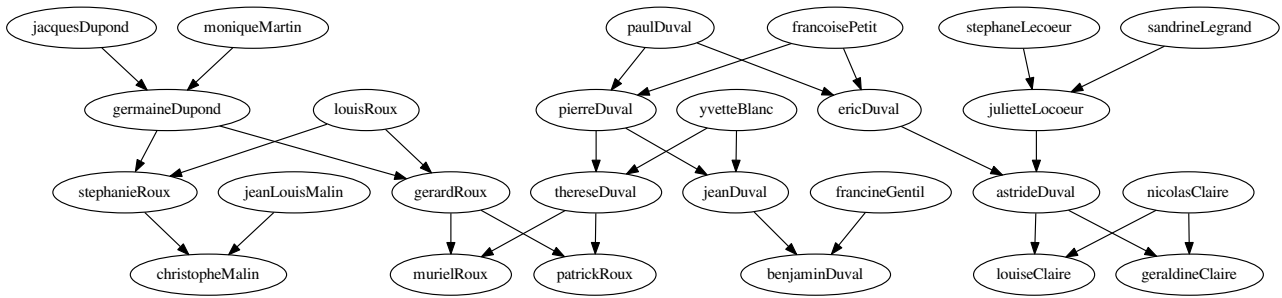


FIGURE 1 – Généalogie étudié dans cet examen

— le fichier XSD `genealogie.xsd` qui permet de valider le fichier XML précédent (utilisation du logiciel `xmllint`).

1. Pourquoi utilise-t-on les groupes dans ce XSD ?

*Répondre à cette question dans le fichier `Reponse-1-1.txt`.*

2. Les codes XSD décrivant les éléments `homme` et `femme` sont les même. Quelle amélioration pouvez vous apporter ?

*Répondre à cette question dans le fichier `Reponse-1-2.txt`.*

À partir du fichier `genealogie.xsd`, Vous créez un nouveau fichier XSD (`genealogieAmeliorree-1.xsd`) qui code cette amélioration.

3. Par défaut les éléments fils de l'élément `personnes` sont obligatoirement des éléments `hommes` puis des éléments `femmes`. Comment peut on avoir successivement aussi bien des hommes que des femmes ?

*Répondre à cette question dans le fichier `Reponse-1-3.txt`.*

Créez un nouveau fichier XSD (`genealogieAmeliorree-2.xsd`) à partir du fichier `genealogie.xsd` qui code cette amélioration. Testez le sur le fichier `genealogieHommesFemmesMelanges.xml`.

## 2 Transformation en XML et HTML (6 points)

Lorsque vous répondez aux questions ci dessous, vous ferez attention à :

- avoir des entêtes XML ou HTML correcte ;
- obtenir un document XML/HTML convenablement indenté ;
- minimiser le nombre de templates ;
- être aussi générique que possible ;
- ne pas utiliser l'élément `for-each`.

Vous utiliserez `saxonb-xslt` pour effectuer vos transformations XSL.

### 2.1 Nombre d'enfants (2 points)

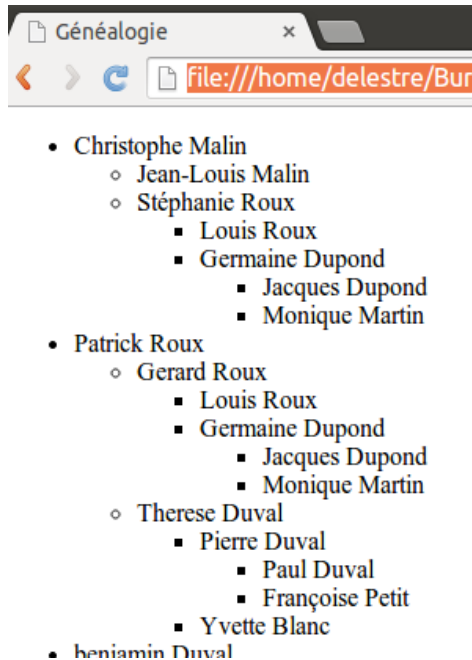
Complétez la feuille XSL `ajoutNbEnfants.xsl` qui permet d'ajouter le nombre d'enfants qu'a chaque personne (le fichier XML `genealogieAvecNbEnfants.xml` est un exemple de résultats obtenus avec l'utilisation de cette feuille).

### 2.2 Une page web avec microdata (4 points)

Développez la feuille XSL `html.xsl` qui permet de produire une page HTML à partir du fichier `genealogieAvecNbEnfants.xml` tel que présentée par la figure 2 : pour chaque personne, la page présente son arbre généalogique sous forme de listes non ordonnées. Cette page Web contiendra

aussi des microdata du schéma `schema.org` décrivant les personnes (classe `Person`), c'est-à-dire leur nom (`familyName`), prénom (`givenName`), leur date de naissance (`birthDate`), et leur date de décès (`deathDate`) si nécessaire, ainsi que leur relation de parenté direct (`parent`). Vous testerez la validité de vos microdata à l'aide de l'outil google (<https://developers.google.com/structured-data/testing-tool/>)

## Rendu de la page HTML



## Interprétation des microdata (Outil de Google)

Person 3	
<b>givenName:</b>	Stéphanie
<b>familyName:</b>	Roux
<b>birthDate:</b>	1971
<b>parent [Person]:</b>	
<b>name:</b>	Louis Roux Germaine Dupond Jacques Dupond Monique Martin

Person 4	
<b>givenName:</b>	Louis
<b>familyName:</b>	Roux
<b>birthDate:</b>	1930
<b>deathDate:</b>	2009

Person 5	
<b>givenName:</b>	Germaine
<b>familyName:</b>	Dupond
<b>birthDate:</b>	1939
<b>parent [Person]:</b>	
<b>name:</b>	Jacques Dupond Monique Martin

Person 6	
<b>givenName:</b>	Jacques
<b>familyName:</b>	Dupond

FIGURE 2 – Rendu de la page et interprétation des microdatas

## 3 Web des données liées (5 points)

### 3.1 Obtention des triplets RDF (4 points)

On suppose que les personnes sont identifiées par des URI dont la nomenclature est `http://genealogie.org/ID` où ID est l'attribut `id` d'une personne.

Développez une feuille XSL (avec les mêmes contraintes que celle de la partie 2) qui permet de produire le fichier turtle `genealogie-produit.ttl` (à l'image du fichier `genealogie.ttl` donné) décrivant les personnes et leurs relations de parenté (utilisation du schéma, de la classe et des propriétés de la question précédente). Vous testerez ce fichier en le chargeant dans le logiciel Protégé (version 5.0beta)<sup>1</sup> disponible dans le répertoire `/opt`.

### 3.2 Requête SPARQL (1 point)

Dans le fichier `Requete-3.txt`, vous développerez une requête SPARQL permettant d'obtenir les noms et prénoms de tous les personnes qui sont grands-parents de la généalogie sur la base de triplets

1. Pour rappel une fois chargé, si le fichier est modifié, Protégé vous propose automatiquement de le recharger

RDF contenus dans `genealogie.ttl`. Vous testerez votre requête avec le logiciel `arq` disponible dans le répertoire `/opt/apache-jena-2.12.1/bin`.

## 4 Inférence prolog (5 points)

On suppose posséder une feuille XSL `enProlog.xsl` qui permet d'obtenir des faits prolog à partir du fichier `genealogie.xml`. Le résultat de cette transformation vous est donné (`genealogie.pl`).

Vous devez maintenant développer `genealogieRegles.pl` qui proposera les prédicats suivants :

1. *fil*<sub>2</sub> tel que `fil(Personne,Fils)` est vrai lorsque `Fils` est le fils de `Personne`.
2. *grandParent*<sub>2</sub> tel que `grandParent(GdParent,Personne)` est vrai lorsque `GdParent` est un grand parent de `Personne`.
3. *petitEnfant*<sub>2</sub> tel que `petitEnfant(PetitEnfant,Personne)` est vrai lorsque `PetitEnfant` est un petit enfant de `Personne`.
4. *frereSoeur*<sub>2</sub> tel que `frereSoeur(FS,Personne)` est vrai lorsque `FS` est un frère ou une soeur de `Personne`.
5. *cousin*<sub>2</sub> tel que `cousin(Cousin,Personne)` est vrai lorsque `Cousin` est un cousin germain ou une cousine germaine (de premier rang) de `Personne`.