

HORAE : feuille de route de Gwenaëlle Patat (document de discussion)

Table des matières

1	Rappel de l'offre de stage	2
1.1.1	Contexte institutionnel ou entrepreneurial.....	2
1.1.2	Notes sur l'encadrement.....	3
1.1.3	Présentation du projet.....	3
1.1.4	Livrable(s) technique(s).....	4
1.1.5	Livrable(s) scientifique(s)	4
1.1.6	Compétences mises en application.....	4
2	Prendre connaissance	5
2.1	contenu de la Dropbox.....	5
2.2	https://redmine.teklia.com/projects/horae	5
3	Structuration de métadonnées.....	5
3.1	Leroquais.....	5
3.1.1	Catalogue océrisé.....	5
3.1.2	Définir la cible	5
3.1.3	Définir les outils	26
3.1.4	Reprise	33
3.2	Cursus	52
3.2.1	Modéliser les informations de cursus	52
3.2.2	Importer les données dans une base MySQL.....	58
4	Annotation et validation	85
4.1.1	Rubriques	Erreur ! Signet non défini.
4.1.2	Niveaux 1-2-3	125
4.1.3	Type de pages	125
4.1.4	Type de zone	125
4.1.5	Textes.....	125
5	Ergonomie avec Oxygen XML.....	125
5.1	Principes	125

5.2	CSS	126
5.2.1	Liste déroulante fermée	127
5.2.2	Recherche d'information dans le document	127
5.2.3	Recherche d'information	128
5.2.4	Interface	130
5.2.5	Actions	130
5.3	Framework et actions	133
5.3.1	Créer des actions	133
5.3.2	Syntaxe (une fois créé)	133
5.4	Mise en oeuvre	133
5.4.1	Vérifier réactivité du logiciel	133
5.4.2	CSS+action 1	134
5.4.3	CSS+Action2	182
6	Exploitation d'informations : usages liturgiques	182
6.1	Réseau des usages	182
6.2	Histoire des manuscrits	182
6.3	182
6.4	182

1 Rappel de l'offre de stage

APPEL A STAGES DU MASTER

« TECHNOLOGIES NUMERIQUES APPLIQUEES A L'HISTOIRE »

1.1.1 Contexte institutionnel ou entrepreneurial

Dans le cadre de ses différents projets, la section de Paléographie latine de l’Institut de Recherche et d’Histoire des Textes offre trois stages pour l’année 2020, dont un commun avec la section de diplomatique.

L’Institut de recherche et d’histoire des textes (IRHT) se consacre à la recherche fondamentale sur les manuscrits médiévaux et les imprimés anciens. L’histoire des textes écrits dans les principales langues de culture du pourtour méditerranéen, latin, langues romanes, hébreu,

grec, copte, syriaque, arabe, y est traitée dans tous ses aspects : supports matériels de l'écrit, écriture et décoration, contenu textuel, iconographie, diffusion et réception.

Centre de compétences de renommée internationale, l'IRHT réunit des spécialistes de haut niveau dans tous les champs de l'histoire érudite des écritures, des livres, des bibliothèques, des textes et de l'art de l'enluminure. Leurs enquêtes philologiques, lexicographiques, historiques, paléographiques et codicologiques ont pour objectifs de dater et localiser les manuscrits, d'identifier et établir les textes qu'ils véhiculent, de préparer l'édition critique des textes et de reconstruire l'histoire de leur production, de leur circulation, de leur audience et de leurs usages de l'Antiquité au début de la Renaissance. Elles donnent lieu à la production de nombreux instruments de travail mis à la disposition de tous les chercheurs.

L'IRHT est un centre pionnier des humanités numériques (publication du *Médiéviste et l'ordinateur* dès 1979). La section de paléographie pilote depuis 2012 des projets de pointe en matière d'analyse des écritures, de lecture et d'analyse des sources par ordinateur (reconnaissance d'écriture manuscrite), au niveau national et international,

1.1.2 Notes sur l'encadrement

Le cadre humain et institutionnel est très favorable : en lien avec les équipes de l'IRHT, en particulier celles des projets en cours, les stagiaires pourront approfondir leur connaissances des formats et de bonnes pratiques (EAD et TEI), celles des sources médiévales et modernes et leur pratique de logiciels clés du domaine (Oxygen, TXM, etc.). Ils pourront également bénéficier des liens avec les autres partenaires de Biblissima.

L'équipe d'accueil est celle de la section de Paléographie latine qui pilote plusieurs projets nationaux et internationaux dans le domaine des humanités numériques, en particulier le projet *HORAE Hours : Recognition, Analysis, Edition* visant à l'étude du texte des livres d'heures et le projet *HOME History of Medieval Europe* visant à la création d'un environnement de recherche portant sur les informations transmises par les chartes et les cartulaires médiévaux. Ces projets mettent en relation les humanités avec des chercheurs en linguistique et en analyse d'image appliquant des méthodes d'apprentissage machine pour la classification, l'identification et l'analyse des textes et images de manuscrits médiévaux.

La ou le stagiaire sera accompagné(e) dans sa montée en compétences dans la compréhension des manuscrits médiévaux et la compréhension des outils d'analyse, et aura l'occasion de mieux connaître les processus de numérisation dans le domaine patrimonial.

1.1.3 Présentation du projet

Cette proposition de stage s'intègre dans les développements du projet *HORAE Hours : Recognition, Analysis, Edition* (ANR-17-CE38-0008), dans lequel la société privée Teklia et le laboratoire de linguistique LS2N sont associés à l'IRHT pour classifier les types de pages et les types d'éléments graphiques sur la page, transcrire automatiquement le texte des livres d'heures numérisés, identifier les éléments textuels élémentaires, identifier les structures, analyser la circulation des textes de dévotion au bas Moyen Âge.

Le ou la stagiaire aura à mener les opérations suivantes :

1. participer au recensement des livres d'heures (dépouillement de catalogues et alimentation d'une base de données) ;
2. annoter et valider les résultats de l'analyse automatique (classification des pages, des zones de pages, de texte, etc.) ;
3. structurer et normaliser les métadonnées, en particulier métadonnées de date et de lieu : création du géoréférencement et exploitation de données floues (date incertaine, intervalles) ;
4. créer structurer des bases textuelles et descriptives ;
5. selon les compétences préalables du ou de la stagiaire, contribuer à l'enrichissement des métadonnées concernant les manuscrits avec des données historiques originales, par exemple pour les manuscrits dont le premier destinataire est une femme, ou contribuer à l'analyse des données (analyse de similitude)

1.1.4 Livrable(s) technique(s)

1. Corpus d'images avec annotations validées
2. Transformations pour enrichir automatiquement les données de géolocalisation
3. Transformations pour enrichir semi-automatiquement les données de chronologie
4. Transformations pour créer un corpus textuel structuré à partir de données non structurées.

1.1.5 Livrable(s) scientifique(s)

1. Rapport sur les rapports géographiques et chronologiques des entités selon leur type (personne physique, personne morale, objet)
2. Corpus textuel et descriptif structuré
3. Visualisations géoréférencées et visualisations de chronologie tenant compte des intervalles de production
4. Analyse de similitudes textuelles

1.1.6 Compétences mises en application

Le ou la stagiaire accroîtra ses connaissances sur et sa maîtrise concernant : (1) l'histoire du livre manuscrit médiéval et la dévotion ; (2) les technologies d'apprentissage machine appliquées à l'analyse d'image et à l'analyse textuelle.

Il ou elle sera accompagné dans son apprentissage des problématiques de modélisation et formalisation, ici appliquée aux problématiques de datation et de géolocalisation pour des entités diverses (personnes morales et physiques, objets, dates floues, multiples ou incertaines).

Les compétences techniques mises en œuvre seront l'utilisation de bases Heurist et de son schéma intégré, la conversion des données par XSLT, pour récupérer des données et les enrichir semi-automatiquement, l'annotation avec interface et schéma spécifiques, la

création et la structuration de corpus textuels avec XSLT ou autres langages ; apprentissage et mise en œuvre d'outils dédiés à l'analyse géographique ou à l'analyse de similitudes pour visualisation heuristique des données des sciences humaines.

2 Prendre connaissance

2.1 contenu de la Dropbox

A la racine :

1. document d'accueil
2. feuille de route / document de suivi
3. dossiers (ne pas modifier le contenu, mais regarder et comprendre sans enregistrer d'éventuelles modifications)
 - **1-Docs soumis** : document définissant le projet
 - **WP1 Corpus** : documents de dépouillement pour établir le recensement des livres d'heures
 - **WP1 Modèle**
 - éditions de textes = corpus de référence
 - tables des matières « toc »
 - feuilles XSLT
 - **WP4 Savoirs interface**

2.2 <https://redmine.teklia.com/projects/horae>

3 Structuration de métadonnées

Passer de données semi-structurées à données structurées : « enrichissement »

3.1 Leroquais

3.1.1 Catalogue océrisé

Volume principal et supplément : ..\Dropbox\HORAE\WP1Corpus\Leroquais

Format Word

3.1.2 Définir la cible

TEI ? Autre ?

Format ?

Premières observations du dossier Leroquais (catalogue des livres d'heures conservés à la BnF) :

Je fais l'hypothèse d'une probable automatisation d'une conversion du document Word vers un format TEI, avec une balise <div> par manuscrit puis des balises <p>. La balise <appInfo> semble indiquer des informations quant à l'outil utilisé pour la conversion et le traitement du document Word, mais j'ai

rarement croisé cette balise. Si la structuration du catalogue ainsi encodé paraît pensée pour une future sortie en HTML, la richesse des métadonnées n'est pas mise en valeur par le présent encodage.

- ⇒ DS: toujours préciser de quoi on parle :
 - Donc ici, quel document Word vers quel document xml
 - Oui, par oxgarage <https://oxgarage.tei-c.org/#>
- ⇒

Les informations relatives aux manuscrits mériteraient en effet d'être détaillées dans une structure TEI qui utilise la balise <msDesc> et ses enfants.

- ⇒ oui

Par ailleurs, la conversion semble avoir échoué en certains points : nous avons un attribut rend="Error" à la ligne 116 qui apparaît de manière récurrente, et les signes < et >, qui désignent respectivement les caractères interdits < et > en XML, apparaissent de manière étrange dans le texte.

- ⇒ Pour comprendre un document et ses caractéristiques, il faut retourner au document source et voir les données du document source

Définir le format cible : donc ici, c'est la structuration du <msDesc/>

Le format des dates pourrait aussi être harmonisé : à la ligne 32 du premier fichier, la date semble être accompagnée de l'heure, ce qui n'est pas le cas à la ligne 12. Toutefois la date où est précisée l'heure correspond à un changement dans l'encodage (<revisionDesc>) et ne renvoie pas au document encodé lui-même, donc il peut être pertinent de garder l'heure pour la traçabilité du document numérique créé. L'auteur de la modification peut alors aussi être indiqué. En outre, nous pourrions normaliser les dates relatives aux manuscrits grâce à l'attribut "when" pour optimiser le référencement.

Est-il aussi possible de compléter ce qui apparaît comme inconnu dans le <publicationStmt> ? À moins que l'information ne soit vraiment pas accessible.

J'ai ensuite remarqué les fichiers .xsl qui indiquent une transformation des fichiers vers des fichiers TEI valides contenant la balise <msDesc> à laquelle j'avais pensé.

J'ai aussi des questions quant au fichier .xsd. S'il définit un type de schéma XML qui me rappelle la structuration d'une ODD, je n'ai pas été confrontée à ce type de fichier pendant mes cours.

- ⇒ Oui, effectivement, c'est le format de base de déclaration des schémas en XML
- ⇒ ODD : schéma + description : one document does it all
 - ODD :
 - peut-on contraindre un élément petit-fils ?
 - par exemple restreindre les valeurs d'attribut de <p/> à l'intérieur d'un élément <origPlace/>

Il faudrait automatiser la structuration des métadonnées dans un encodage adapté à la description des manuscrits, qui puisse rendre ainsi hommage à la richesse des métadonnées. Le premier format qui me vient en tête est donc la TEI.

⇒ 1 paragraphe, 1 idée.

Nous pouvons définir plusieurs cibles : la structuration des métadonnées du catalogue dans un format xml TEI, qui sera ensuite transformé en document html via une feuille de transformation xsl. L'HTML me semble être la sortie finale pour pouvoir publier les ressources en ligne.

⇒ D'où vient l'idée de l'html ? pourquoi est-ce que ce serait l'objectif ?

⇒ PRIMAUTE de l'INFORMATION : STRUCTURATION ET QUALITÉ

L'aspect inventaire du catalogue pourrait aussi nous faire penser à l'encodage d'instruments de recherche en xml EAD, ce qui regrouperait les informations de manière moins fine, ou du moins adaptée à la description du manuscrit.

⇒ 1) ce n'est pas du tout un inventaire (systématique)

⇒ 2) le format n'est pas adapté : pourquoi le proposer.

⇒ Formulation à améliorer : 3 idées :

- C'est un inventaire : oui/non, parce que (a), (b), (c)
- Si oui, alors, le format EAD est adapté
- On évalue l'intérêt des formats en concurrence : EAD et TEI, avec des arguments (a), (b), (c)....
- Et on tranche.
- Ici, la capacité descriptive fera préférer TEI

Toutefois, cela aurait pour intérêt de mettre en avant l'énumération des manuscrits cataloguée en gardant la description détaillée pour un fichier qui serait propre à chaque manuscrit, avec sa transcription.

Utiliser le format de structure générale et le format <msDesc/> en particulier de
<https://www.dropbox.com/s/pqsvtz5lttvowg4/ECMEN19.xml?dl=0>

Rédiger l'ODD qui va avec pour <msDesc/> : être le plus contraignant pour n'autoriser que ce que l'on trouve dans ce catalogue précisément.

En matière d'éléments et d'attributs et de valeurs d'attributs.

16/04

Warning/Error : (Schematron permet de générer des avertissements, donc DS suppose que ODD aussi, à vérifier)

Legacy data (données héritées) : elles ne sont jamais parfaites, elles ne sont jamais complètes ; il faut élaborer une stratégie qualité qui permet une amélioration progressive, mais qui ne bloque pas l'avancement

La TEI est extrêmement souple ; une homogénéisation peut être nécessaire, mais on peut aussi y renoncer : c'est un choix d'équipe, d'institution, de collaborations

L'erreur empêche de transformer le document, alors qu'il peut être valide

L'ordre des éléments, qui est utile dans la saisie, n'est généralement pas exploité lors des transformations si le nom des éléments est différent.

Par ex. : @class : dans tel projet, il est systématique, il faut donc le rendre obligatoire ; mais il est très spécifique à un unique projet et ne pourra pas être rendu obligatoire pour le projet d'à côté.

Référents / ODD/ Xpath

- Extract-paths

Réflexions personnelles sur l'ODD du catalogue des manuscrits datés du fonds français dans le cadre du projet ECMEN

Lier les pointeurs à la référence “xml:id” correspondante via un chemin Xpath. Exemple de syntaxe (à vérifier) :

```
</sequence>
</content>
<attlist>
  <!-- Attribut "class" rendu obligatoire pour l'élément "msItem". -->
  <attDef ident="class" mode="change" usage="req">
    <valList>
      <!-- Restriction des valeurs de l'attribut "class" aux "xml:id" déclarées dans les éléments <category>. | -->
      <valItem ident="#xpath{//tei:classDecl//tei:category[@xml:id]}"/>
    </valList>
  </attDef>
  <attDef ident="corresp" mode="delete"/>
  <attDef ident="compte" mode="delet"/>
```

Typeage des données pour les valeurs d'attributs :

- teidata.numeric pour les attributs avec des chiffres, soit “quantity”, “unit”, “columns”, et “n”
- teidata.pointer pour les attributs contenant une URI avec l'identifiant ark de la BnF, soit “ref”, “url” et “target”

Contraintes :

- suppression de l'attribut “status” dans l'élément <msDesc>
- attribut “class” dans l'élément <msItem> rendu obligatoire
- ordre dans séquence d'éléments au sein d'un élément parent, comme c'est (pour l'instant) le cas dans <msIdentifier>, <msContents> et <msItem>

Avertissements pour améliorer la complétude et l'homogénéisation des données, mais qui ne rendent pas pour autant le document TEI invalide (le monde n'est ni tout noir ni tout blanc mais gris). Possible de les insérer dans l'ODD grâce à la balise **<sch:report test="tei:model[@output]" role="warning">** (à tester):

- dans le cas d'une absence d'éléments au sein d'une séquence, mais uniquement sur les éléments considérés comme essentiels pour l'analyse propice au projet (<repository> dans <msIdentifier> ? <rubric> dans <msItem> ?...à compléter). Si les données ne sont pas connues, mettre un attribut “type='absent'” par exemple.

-dans le cas d'une normalisation de certains attributs à encourager soit pour la cohérence de l'encodage (par exemple si l'attribut "ref" ne renvoie pas à un "xml:id" répertorié), soit pour systématiser des informations considérées comme incontournables dans le cadre du projet (par exemple l'attribut "xml:lang" dans l'élément "textLang", avec comme valeur le code langue correspondant à la norme ISO 639, avec laquelle il est possible d'indiquer le moyen français via la valeur "fro" ou l'ancien français avec la valeur "frm" ; ou encore l'attribut "script" dans <scriptNote> qui indique le type d'écriture).

Suggestions à compléter au fur et à mesure de la personnalisation de l'ODD.

17/04

Résultats du premier test de l'ODD :

- Attribut "status" sur le <msDesc> bel et bien supprimé
- Tentative de faire un avertissement sur les <msItem> quand il n'y a pas l'attribut "class" et qu'il ne fait pas référence aux "xml:id" souhaités, soit ceux notés dans les éléments <category> du <teiCorpus>. **Il faut faire une condition via schematron et une xsl pour récupérer la liste d'id en valeurs possibles. Tentative encore à l'état d'expérimentation.**
- Problème sur les <msItem> qui contiennent parfois plusieurs œuvres. Il est plus pertinent de mettre plusieurs <msItem> par <msContents> si nécessaire, car l'élément est créé pour décrire un œuvre dans son individualité au sein du contenu intellectuel d'un manuscrit. La séquence d'éléments définie dans l'ODD à l'intérieur du <msItem> devient alors correcte, bien qu'il faille laisser la possibilité de rentrer plusieurs <title> et <author> dans le cas de traductions et versions linguistiques différentes. On peut donc rajouter un attribut "xml:lang" à ces deux éléments.

Suite des réflexions sur l'ODD à personnaliser :

- Question du caractère obligatoire des attributs importante pour la normalisation, avec la question sous-jacente de la pertinence pour le projet et de la disponibilité des données pour pouvoir les remplir.
- On peut aussi rendre obligatoire des séquences d'éléments pour l'homogénéisation (cf. le cas de l'élément <supportDesc> qui ne contient pas systématiquement un élément <support> qui contient lui-même un élément <material>). **Possible de le faire avec un framework d'Oxygen (à tester) :**
 - supportDesc non régulier => faire une transformation pour récupérer les informations existantes de façon homogène (ajouter automatiquement support/material)

- ergonomie de la saisie : si on le rend obligatoire, il est mieux de proposer aux utilisateurs que, au moment où ils indiquent l'attribut `@material='parch'`, alors, l'élément fils `<support><material>Parchemin</material></support>` soit inséré à la suite, ou que la valeur « Parchemin » soit ajoutée si les éléments sont déjà créés.
- -Attention à la restriction des valeurs d'attributs. Conserver celles autorisées par la TEI en plus de celles personnalisées. La restriction est pertinente que si elle permet une homogénéisation, comme dans le cas des unités de mesure (cf. "mm" pour l'attribut "unit" qui se retrouve essentiellement dans les éléments `<dimensions>` et `<measure>`).

Travail d'amélioration de l'ODD à poursuivre en vue d'une homogénéisation du catalogue de manuscrits du projet ECMEN.

20/04

Tests de vérification du chemin des références de l'attribut "class" dans les `<msItem>` vers l'`xml:id` correspondant :

```
<!-- Restriction des valeurs de l'attribut "class" aux "xml:id" déclarés dans les éléments <category>. La valeur de l'attribut "class" renvoie en effet aux catégories de livres définies dans l'élément "category". Message d'alerte si l'attribut "class" n'apparaît pas. -->
<vallItem
    <constraintSpec
        <constraint>
            <sch:rule
                <!--<xsl:value-of
                    select="string-join('#',tei:teiCorpus//tei:classDesc/tei:taxonomy/tei:category[@xml:id])"/>-->
                    <sch:assert
                        test="count[contains(@class, string-join('#',tei:teiCorpus//tei:classDesc/tei:taxonomy/tei:category[@xml:id]))) = 1">Please specify the type of book.</sch:assert>
                </sch:rule>
            </constraint>
        </constraintSpec>
    </vallItem>
```

Tentative avec schematron dans l'ODD, la suggestion des xml-id correspondants doit passer par une règle en xslt avec la condition "xsl:if", sous la forme suivante :

```
<!-- EXEMPLE (liste de valeurs de l'attribut @class au sein de l'élément <msItem>)
    <!--
        <xsl:if
            test="$Element
                =
                'msItem'">
```

```

<item      value=      "string_join('#',      'ancestor::teiCorpus//classDesc/taxonomy/category[@xml:id]')"/>
</xsl:if>->-->

```

Questionnements autour de la normalisation des langues dans un catalogue de manuscrits.
Suggestion suivante :

```

<!-- Attribut "xml:lang" rendu obligatoire sur l'élément "title", afin notamment d'indiquer le type de version linguistique
dans l'élément "msItem". Sa valeur doit être remplie selon le code ISO 639 pour des question d'homogénéisation.
-->
<attDef ident="xml:lang" mode="change" usage="req"/>

```

Mais cela n'est pas pertinent dans le cadre du projet ECMEN qui répertorie des manuscrits du fonds français, donc la majorité des titres sont en français (ou en ancien français). La manipulation représente donc une perte de temps pour les catalogueurs. En outre, il est plus intéressant de jouer sur les conditions dans l'élément <textLang>, notamment dans le cadre du projet HORAE qui s'inscrit dans une histoire globale. Toutefois, cette information quant à la normalisation de la langue est plus pertinente dans les notices qu'au sein même du catalogue au format TEI.

Suite des réflexions sur les contraintes à apporter à l'ODD du catalogue d'Ecmen :

-Définition d'une séquence d'éléments dans l'élément “collation” comme ceci :

```

<elementSpec           ident="collation"           mode="change">
    <!-- L'élément "collation" peut contenir la liste d'éléments suivants. Attention à bien mettre les bonnes
        informations dans les éléments correspondants pour une meilleure structuration des données. -->
    <content>
        <sequence>
            <elementRef   key="desc"           minOccurs="0"           maxOccurs="1"/>
            <elementRef   key="formula"         minOccurs="0"           maxOccurs="1"/>
            <elementRef   key="p"              minOccurs="0"           maxOccurs="unbounded"/>
            <elementRef   key="q"              minOccurs="0"           maxOccurs="unbounded"/>
            <elementRef   key="signatures"       minOccurs="0"           maxOccurs="1"/>
            <elementRef   key="catchwords"       minOccurs="0"           maxOccurs="1"/>
        </sequence>
    </content>

```

Cela permet une homogénéisation de l'implémentation de l'information, qui apparaît parfois encodée de diverses manières :

```

<teiCorpus>
  <TEI>
    <fileDesc>
      <sourceDesc>
        <listWt>
          <witness>
            <msDesc>
              <physDesc>
                <objectDesc>
                  <supportDesc>
                    <collation>
                      <desc>Majoritairement des quaternions, et un quinion et un binion.</desc>
                    <formula>
                      <hi rend="sub">1-20</hi><hi rend="sup">f.160</hi>
                      <hi rend="sub">21</hi><hi rend="sup">f.170</hi>
                      <hi rend="sub">22</hi><hi rend="sup">f.174</hi>
                    </formula>
          </supportDesc>
        </listWt>
      </sourceDesc>
    </fileDesc>
  </TEI>
</teiCorpus>

```

Dans le premier exemple ci-dessus, les réclames ne se trouvent pas entre les balises "catchwords, et le nombre de feuillets par cahier n'est pas encodé dans une balise particulière, tandis que dans le deuxième exemple, la description des cahiers est encodée dans une balise <desc>.

-Interrogation autour de la liste de valeurs à restreindre dans l'attribut "type" de l'élément "desc" :

```

<elementSpec ident="desc" mode="change">
  <attList>
    <attDef ident="type" mode="change">
      <!-- Définition d'une liste de valeurs restriées pour l'attribut "type". -->
      <valList mode="add">
        <valItem>
        <valItem>
        <valItem>
      </valList>
    </attDef>

```

Dans l'encodage du catalogue, "pricking" apparaît une fois avec un "s" :

```

<teiCorpus>
  <TEI>
    <fileDesc>
      <sourceDesc>
        <listWt>
          <witness>
            <msDesc>
              <physDesc>
                <objectDesc>
                  <layoutDesc>
                    <layout columns="1" writtenLines="34 35">
                      <desc type="ruling" rend="à la mine de plomb" corresp="f. 31v">30, [4, 75], 20 x 11, [5, 141, 5], 38</desc>
                      <desc type="prickings">Plaques visibles pour les rectrices majeures verticales</desc>
                    </layout>
      </sourceDesc>
    </fileDesc>
  </TEI>
</teiCorpus>

```

Nous avons fait le choix d'enlever le "s" dans l'ODD. De plus, nous nous demandons s'il faudrait traduire "schéma" en anglais pour des questions d'homogénéité.

Question : le document rng généré depuis l'ODD peut-il intégrer les règles schematron formulées dans l'ODD.xml ? **Question à creuser.**

04/05

Rappel de la syntaxe d'insertion d'une règle schematron dans l'ODD :

```

<elementSpec ident="provenance" mode="change">
  <constraintSpec ident="provStruct" scheme="isoschematron">
    <constraint>

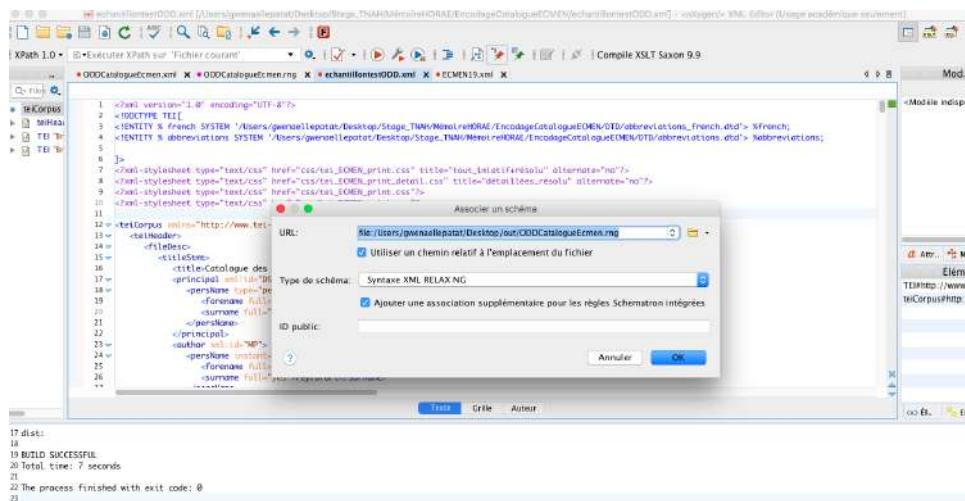
```

```

<!--<rule xmlns="http://purl.oclc.org/dsdl/schematron" context="//tei:provenance" role="warning">>-->
<assert      xmlns="http://purl.oclc.org/dsdl/schematron" test      ="(child::tei:orgName
or
or
and
Provenance must contain (orgName or placeName or persName) and (p)</assert>
<!--</rule>-->
</constraint>
</constraintSpec>

```

Exemple implémenté par M.Stutzmann. L'élément `<provenance>` doit au moins contenir l'élément `<orgName>`, `<placeName>` ou `<persName>` et l'élément `<p>`. L'intérêt de cette méthode est qu'un message personnalisé est généré en cas de non-conformité à la règle (à la différence d'une simple définition de `<sequence>` dans l'`<elementSpec>`). La balise `<assert>` affiche le message si la valeur de son `@test` est "false", tandis que la balise `<report>`, qui peut se trouver à la même place qu'`assert`, affiche le message si la valeur de son `@test` est "true". À noter que la balise `<rule>` n'est pas obligatoire.



Pour que la règle schematron soit associée à l'ODD, nous apprenons grâce à la documentation du logiciel d'Oxygen qu'au moment de l'association du `rng`, il faut aussi associer le schéma des règles schematron pour obtenir les associations suivantes :

```

<?xml-model href="percent.rng" type="application/xml" schematypens="http://relaxng.org/ns/structure/1.0"?>
<?xml-model href="percent.rng" type="application/xml" schematypens="http://purl.oclc.org/dsdl/schematron"?>

```

Toutefois, cela affiche une erreur en rouge et non pas un simple warning en jaune.

De même, si la règle schematron suivante ne répond pas au désir d'autocompletion des bonnes valeurs d'attributs dans l'`@class` de `<mstitem>`, elle permet d'afficher un message d'erreur en cas de référence au mauvais `@xml:id` :

```

<elementSpec ident="msItem" mode="change">
  <constraintSpec ident="class" scheme="isoschematron">
    <!-- Restriction des valeurs de l'attribut "class" aux "xml:id" déclarés dans les éléments <category>.
        La valeur de l'attribut "class" renvoie en effet aux catégories de livres définies dans l'élément "category".
      <Message d'alerte si l'attribut "class" n'apparaît pas. -->
      <constraint>
        <!--<rule xmlns="http://purl.oclc.org/dsdl/schematron" context="//tei:msItem" role="warning">-->
      <assert xmlns="http://purl.oclc.org/dsdl/schematron"
        test="substring-after(@class, '#') = ancestor::tei:teiCorpus/tei:teiHeader/tei:encodingDesc/tei:classDecl/tei:taxonomy/tei:category/@xml:id">
        Please select a valid class</assert>
      </constraint>
    </constraintSpec>
  </constraintSpec>

```

À noter que la règle est à insérer directement à l'intérieur de l'<elementSpec>.

Problème sur les valeurs de l'attribut type dans <desc>. La valeur par défaut suggérée est “deprecationInfo”. Cela est dû à une définition de valeurs qui ne sont pas propre à la TEI. Il faut donc les implémenter avec l'@ns suivant et le mode “add” :

```

<elementSpec ident="desc" mode="change">
  <attList>
    <attDef ident="type" mode="change">
      <!-- Définition d'une liste de valeurs restreintes pour l'attribut "type", qui ne font pas partie de la TEI. -->
    <valList mode="add" type="closed" ns="http://www.example.com/ns/nonTEI">
      <valItem ident="pricking"/>
      <valItem ident="ruling"/>
      <valItem ident="schéma"/>
    </valList>
  </attList>

```

07/05

Travail sur la définition de la cible

Il s'agit de définir une structure interopérable en respectant l'ODD définie à propos du projet Ecmen. L'objectif est d'être le plus proche possible du document source en conservant toutes les données.

Pour définir la forme du document cible, prendre trois notices à encoder à la main afin de faire la liste des points à vérifier lors de la transformation.

Nous voyons qu'il existe deux sources pour le catalogue de Leroquais : Les_livres_d'heures_manuscrits_de_[...]Leroquais_Victor_bpt6k15129612_notices_tout.docx et Supplément_aux_livres_d'heures_manuscrits_notices.docx. **Initiative de prendre des notices aussi dans le supplément.** Dans un premier temps, nous avons vérifié les métadonnées du catalogue pour bien remplir le <teiHeader> de l'ensemble du document.

Pour l'instant, choix d'une notice qui n'est pas encore totalement structurée. Il s'agit de la première ("1-Livre d'heures et missel franciscains 1380", Les_livres_d'heures_manuscrits_de_[...]Leroquais_Victor_bpt6k15129612_notices_tout.doc x, pp.1-7). Nous avons ensuite fait le choix de mettre un @xml:id par notice pour mieux s'y retrouver :

teiCorpus	TEI	Facsimile
40 ▾	<distributor> https://catalogue.bnf.fr/ark:/12148/cb36797672n </distributor>	Bibliothèque
41 ▾	<availability>	Nationale de France</availability>
42 ▾	<licence>Domaine Public</licence>	
43 ▾	</availability>	
44 ▾	</publicationStmt>	
45 ▾	</bibFull>	
46 ▾	</sourceDesc>	
47 ▾	</fileDesc>	
48 ▾	</teiHeader>	
49 ▾		
50 ▾		
51 ▾	<TEI xml:id="BNF_msLat_757">	
52 ▾	<teiHeader>	
53 ▾	<fileDesc>	
54 ▾	<titleStmt>	
55 ▾	<title>1. LIVRE D'HEURES ET MISSEL FRANCISCAINS, 1380</title>	
56 ▾	</titleStmt>	
57 ▾	<publicationStmt>	
58 ▾	<place>Paris</place>	
59 ▾	</publicationStmt>	
60 ▾	<sourceDesc>	
61 ▾	<listMts>	
62 ▾	<mt>	
63 ▾	<desc>	
64 ▾	<msIdentifier>	
65 ▾	<settlement>Paris</settlement>	
66 ▾	<repository>Bibliothèque nationale de France</repository>	
67 ▾	<idno>msLat. 757</idno>	
68 ▾	<altIdentifier>	
69 ▾	<idno>4299</idno>	
70 ▾	<altIdentifier>	
71 ▾	<msContent>	
72 ▾	<summary>Ce manuscrit présente une combinaison assez rare du livre d'Heures et du missel. [...]. Le calendrier (fol. 3 v° à 15) est	
73 ▾		
74 ▾		

La composition des livres d'heures étant réputée complexe, avec des pièces qui se succèdent et qui s'imbriquent selon les niveaux, le plus long dans la structuration est la décomposition des informations fournies par Leroquais en plusieurs `<msItem>`. Ces `<msItem>` pourront être doté de l'`@class` qui renverront à des classes répertoriées dans le cadre du projet Horae. Ces classes seront encodées dans les balises `<category>` du `<teiHeader>` principal. Par ailleurs, quand les informations de la notice ne nous paraissent pas claires ou assez précises, nous vérifions systématiquement dans le manuscrit numérisé à l'adresse suivante : <https://gallica.bnf.fr/ark:/12148/btv1b8470209d/f1.image>. Si chaque notice est par la suite associée à un facsimile, les liens vers le manuscrit numérisé et les folios concernés seront encodés. Cela nous a ainsi permis de distinguer ce qui relevait de la rubrique, de l'incipit ou de l'explicit.

Il faut aussi faire attention à ce que certaines données aient une forme normalisée. Nous avons donc fait le choix d'écrire la foliotation ainsi : `<locus>ff.16r-222r et ff.376v-432r</locus>`. Nous avons également supprimé les guillemets pour les citations.

Nous avons ressenti le besoin d'être attentif aux possibles fautes dans l'OCR, notamment dans les citations en latin où nous trouvons parfois des abréviations. Par exemple, « *Oratio. Benedicta sit hora illa qua Deus homo annunciatus est... — ... et impleatur desiderium meum. Arnen.* » à la p.2 de Les_livres_d'heures_manuscrits_de_[...]Leroquais_Victor_bpt6k15129612_notices_tout.doc x. C'est bien "Amen" dans le catalogue de Leroquais. On trouve aussi : « Domine Deus omnipotens, fac me Ma. et 2V. fortes et stabiles contra omnes, inimicos nostros... — 429 ...et animas corporum nostrorum. » à la p.3 du même document, où "2V" correspond en fait à "N." dans le catalogue original.

Les erreurs sur les caractères spéciaux dues à l'océrisation nous interroge sur l'encodage des abréviations dans le format cible. Nous les avons ici encodées directement dans le texte, avec la référence numérique du caractère pour qu'XML puisse lire l'unicode correspondant :

```

242 <msItem>
243   <locus>ff.149r-154r</locus>
244   <title>Litaines</title>
245   <note>aucun saint local</note>
246   <explicit><locus>f.152r</locus>...<choice>
247     <abbr>¶#8479;</abbr>
248     <expan>Répons</expan>
249   </choice> Oremus pro ministro nostro,<choice>
250     <abbr>¶#8483;</abbr>
251     <expan>Verset</expan>
252   </choice> Dominus conservet eum et vivificet eum...</explicit>
253 </msItem>
...

```

Toutefois, il est probable que ces abréviations soient par la suite soit encodées dans une DTD propre au projet, soit signalées au sein de l'encodage du catalogue avec une mention sur les choix de transcription.

Sur la cature précédente, nous avons un doute d'encodage, car le passage cité pour les Litaines ne correspond pas vraiment à l'explicit, étant donné que les répons et verset cités ne se trouvent pas à la toute fin de la pièce liturgique. Nous avons cependant fait ce choix car les points de suspensions indiqués par Leroquais peuvent laisser penser que le texte se prolonge jusqu'à la fin de la pièce. L'ODD définie et les tei guidelines ne proposent pas de balise adéquate pour ce genre de citation, car elle ne peut non plus figurer dans une balise `<p>`, la citation n'est pas un paragraphe. L'ODD quant à elle laisse le choix entre `<rubric>`, `<incipit>` ou `<explicit>`.

Il faudrait par ailleurs **changer une contrainte dans l'ODD** :

<!-- Définition d'une séquence d'éléments dans l'élément `<msItem>`. Aucun des éléments ci-dessous n'apparaissent systématiquement. Par ailleurs, on peut retrouver plusieurs fois la séquence dans un même `<msItem>`.

Le titre et l'auteur peuvent apparaître plusieurs fois pour signaler d'éventuels traductions dans plusieurs langues. -->

```

<content>
<sequence preserveOrder="true">
  <elementRef key="locus" minOccurs="0" maxOccurs="1"/>
  <elementRef key="author" minOccurs="0" maxOccurs="unbounded"/>
  <elementRef key="title" minOccurs="0" maxOccurs="unbounded"/>
  <elementRef key="rubric" minOccurs="0" maxOccurs="1"/>
  <elementRef key="incipit" minOccurs="0" maxOccurs="1"/>
  <elementRef key="explicit" minOccurs="0" maxOccurs="1"/>
  <elementRef key="colophon" minOccurs="0" maxOccurs="1"/>

```

```
<elementRef key="finalRubric" minOccurs="0" maxOccurs="1"/>
<elementRef key="note" minOccurs="0" maxOccurs="1"/>
</sequence>
</content>
```

Nous voyons ici qu'un <msItem> ne peut contenir que zéro à une rubrique, alors que certaines pièces, si je ne m'abuse, peuvent contenir plusieurs rubriques. Il faudrait donc changer le "1" par "unbounded".

11/05

Erreur sur l'OCR p.4 du document

Les_livres_d'heures_manuscrits_de_[...].Leroquais_Victor_bpt6k15129612_notices_to ut.docx : "écriture" et pas "ccriture".

ODD : **Affiner les contraintes du <physDesc> ?**

12/05

Ne pas faire un <TEI> par notice comme dans pour le projet Ecmen, mais un <msDesc> contenu dans <text>, <body> et <listBibl>. Mettre un @xml:id sur le <msDesc> pour s'y retrouver dans les notices ?

Erreur d'océrisation p.4 du document

Les_livres_d'heures_manuscrits_de_[...].Leroquais_Victor_bpt6k15129612_notices_to ut.docx : Jean Galéas Visconti et non pas Jean-Gaiéas Visconti.

Choix de structure la mention de répliques du manuscrit dans les balises <additional> et <surrogates>, avec des <bibl> ou <msDesc> selon ce que est décrit.

Principes à ne pas oublier lors de la définition du format cible :

- Refuse the temptation to guess
- La structure prime sur le contenu
- La nature de l'information prime sur la présentation
- L'objectif est d'automatiser une transformation

Mise au point sur les premiers choix d'encodage

Débat sur la place à accorder à l'édition électronique. Dans un projet, il faut savoir reconnaître les contributions telles qu'elles sont. Ici, l'édition électronique du catalogue de Victor Leroquais s'inscrit sous sa paternité, sa contribution scientifique est importante. L'édition électronique s'inscrit également sous la paternité du Consortium TEI qui nous édicte tout un panel de balises avec une sémantique bien précise pour répondre à un souci d'intéropérabilité. Il faut donc bien réfléchir à ce que l'on met dans les balses `<edition>` et `<responStmt>`. Cette dernière se réfère à la notion de responsabilité. Cela demande de mesurer où l'on se situe, en tant qu'encodeur, dans la chaîne d'interaction que recouvre la responsabilité intellectuelle. Le principal rôle de l'encodeur est d'expliciter l'implicite du texte sur lequel il travaille, de révéler sa structure.

/!\ La balise `<distributor>` se réfère à un concept contemporain sur la diffusion du texte. Il ne peut s'appliquer à l'édition de 1927. De même, attention à l'usage de l'`@ facets`. Il sert à faire un lien vers une image du contenu de l'élément.

Pour la structure générale des notices, il est mieux de garder une balise `<TEI>` par notice, comme pour Ecmen, et pour que l'ODD définie puisse s'appliquer. Pour rendre le document valide avec TEI all, mettre un `<text>`, un `<body>` puis un `<p>` à l'intérieur de chaque balise `<TEI>`.

Il est important de respecter la structure de Leroquais. Par exemple, l'ancienne côte, qui pourrait logiquement prendre sa place dans un <altIdentifier>, est ici citée avec la description du feuil de garde, ce qui signifie que l'ancienne côte doit être encodée avec la description des premières pages. Par ailleurs, le projet ne requiert pas la connaissance des anciennes cotes, donc il n'est pas la peine de les structurer.

En ce qui concerne les en-têtes des notices, l'information est éclatée entre un <head>, un <msContents> et un <date> dans <history>. Ce qui donne sur le papier : “1. LIVRE D’HEURES ET MISSEL FRANCISCAINS. 1380”, devient en xml :

```
<head>1.</head>
<msContents>
    <summary>LIVRE D’HEURES ET MISSEL FRANCISCAINS</summary>
[...]
<history>
    <origin>
        <origDate>1380</origDate>
[...]
```

Si une citation ne concerne ni un incipit ni un explicit, la mettre dans un <quote>.

Ne pas mettre f. ou ff. dans <locus>, car nous travaillons par défaut sur des folios. On peut toutefois préciser un @scheme=”page” si nous nous trouvons sur des pages.

!& <rubric> en TEI n'a pas un sens codicologique, la balise s'applique aux titres initiaux, donc le début du texte, et pas nécessairement à des rubriques ailleurs dans le texte. Attention à **favoriser l'encodage sémantique le plus juste**.

Le premier paragraphe de la notice mentionne les textes qui seront décrits : ne pas les mettre dans des <msItem> englobant tous les autres, car cela sera impossible à automatiser.

Vérifier que les paragraphes de l'OCR sont bien ceux de l'édition originale. En général, un paragraphe correspond à un <msItem>. Pour les <msItem> imbriqués, matcher sur les numéros de folios, sur les tirets longs pour diviser les parties, et les

tirets longs précédés ou suivis de points de suspension pour les incipits et les explicits.

De plus, l'italique signale les rubriques.

/!\ Ne pas restituer les abréviations en caractère décimal avec une hypothèse de transcription. Cela s'applique pour la transcription du texte mais pas pour une notice. D'autant plus qu'il est difficile de déterminer ce qui correspond à un répons, une réponse, un verset ou un versicule. Il faut donc restituer stricto sensu ce que Leroquais a écrit.

Certains *<locus>* seront à recréer avec la validation du schéma défini dans l'ODD.

Mettre les répliques d'un manuscrit dans la balise *<filiation>* d'un *<msItem>* ou bien dans *<history>*, *<origin>* et *<note>*

Dans *<history>*, la balise *<origin>* renvoie à la création du manuscrit avant son acquisition, tandis que la balise *<provenance>* fait référence aux possession, à l'appartenance.

/!\ Ne perdre aucun mot.

14/05

Quelques erreurs d'océrisation à la p.6 du document Les_livres_d'heures_manuscrits_de_[...]Leroquais_Victor_bpt6k15129612_notices_to ut.docx : “318v<>, 321, 322v<>” où <> correspond à ° ; “la première moitié du xvi® siècle”.

Construction de la notice 1 et éléments qui peuvent servir pour l'automatisation

La première ligne de la notice en **majuscule** correspond à un *<head>*, un *<summary>* et un *<origDate>*.

La deuxième ligne avec une police plus petite correspond au lieu de conservation et à la cote, soit un *<repository>* et un *<idno>*.

Le premier paragraphe signale un résumé du contenu, donc à mettre dans une balise <summary>.

Les paragraphes suivants peuvent correspondre à des <msItem> dans un <msContents> avec des <rubric> etc., dans <history> etc. -> faire attention à la mention de **folios** avec des **chiffres** ou des **lettres majuscules**, aux **tirets longs**, aux **Majuscules**, aux **guillemets**, à l'**italique**, aux **points de suspensions**, pour matcher les bonnes informations dans les bonnes balises.

Les **paragraphes** avec des **mentions de folios mais moins de tirets longs** peuvent davantage concernés les balises <history>, <scriptDesc>, <note> etc. Et moins le <msContents>.

Saut de ligne avec un paragraphe en **écriture plus petite** concernant le <physdesc>. Chaque nouveau paragraphe peut correspondre à un changement d'information dans le <physDesc> avec <decoDesc>, <bidingDesc> etc.

Les dernières informations séparées avec un **tiret long** renvoient à une bibliographie à mettre dans <additional> et <listBibl>.

15/05

Attention au document océrisé : perte d'informations (cf. p.232 notice 112 où il manque une ligne entre "Saint-Just Mart." et "abbé") + "Arnen" au lieu de "Amen" + "Pareil." au lieu de "Parch." + transformation de certains chiffres dans numérotation de la notice en lettres.

Mention de l'écriture parfois du point de vue linguistique et pas paléographique -> attention pour la récupération des infos car l'une va dans <history> et l'autre dans <ScriptDesc>.

La structure de la deuxième notice test choisie (n°112 dans Les_livres_d'heures_manuscrits_de_[...]Leroquais_Victor_bpt6k15129612_notices_to

ut.docx p.232) confirme celle de la première : les tirets longs séparent des <msItem>, l'avant-dernier paragraphe concerne l'analyse historique et le dernier la décoration, les dimensions etc. Il n'y a toutefois pas de bibliographie dans cette notice.

18/05

Cf. Travail sur les chemins XPath à garder pour entraîner le modèle de reconnaissance Horae, avec la question du choix des abréviations pas toujours faciles à transcrire, et de ce que l'on conserve comme informations pour l'entraînement.

À vérifier : précision des 3 volumes de notices de Leroquais dans métadonnées, et précision des pages de la notice en question ? Cela sera difficile à automatiser si on part de l'OCR. Après avoir tester l'ODD sur les notices tests, il faudra peut-être la modifier (cf. contraintes dans <msItem>, car suivre l'ordre de la notice de Leroquais, ce qui facilite l'automatisation, implique de recréer parfois plusieurs fois la même balise)

Erreurs OCR dans

Les_livres_d'heures_manuscrits_de_[...]\Leroquais_Victor_bpt6k15129612_notices_to ut.docx :

- “I)une autre main” pour ‘D’une autre main” p.675
- “Arnen” pour Amen quasiment à chaque fois -> **correction à automatiser**
- “Pareil.” pour “Parch.” souvent -> **correction à automatiser**

Idées pour structurer automatiquement les informations :

- Jouer sur les points de suspension pour les citations (s'ils sont à la fin c'est un <incipit>, s'ils sont au début c'est un <explicit>, s'ils sont au début et à la fin c'est une <quote>)
- Les mots coupés d'un tiret puis d'un espace sont en fin de ligne dans le le livre original -> **tiret et espace à supprimer dans automatisation**
- Ce qui précède une citation avec ":" = <note>
- Dans description esthétique du manuscrit, on peut faire un paragraphe par tiret long dans <decoNote> pour séparer les informations d'ordre différent, comme le suggère les notices de Leroquais
- Une fois les <locus> créés, il faudra enlever les "Fol." et "fol.", remplacer les v° par v. et mettre r. si le nombre n'est suivi de rien.
- Lorsque des informations de nature différente en TEI sont dans un même paragraphe chez Leroquais, jouer sur les mentions de "lignes" et "mill." pour les balises <extent> et <dimensions> ; s'il est question de "siècle" le mettre dans <origDate>.

19/05

Il est bien que l'ODD soit contraignante, mais il faut toujours se demander avant de supprimer un élément s'il n'a pas lieu d'être maintenu. Cf. La balise <note> qui peut toujours être utile pour une explication apportée. Toutefois, son usage peut-être contraint grâce à des attributs, une classification, une sous-structuration, une interdiction des balises <p>, etc.

Il faut aussi se poser la question de l'équivalence des éléments : à quoi servent-ils ? Qu'est-ce qui peut en faire usage ? Cf. Le cas de <head>.

Il faut pouvoir répéter les <msItem> dans un <msContents>, qu'ils puissent s'imbriquer entre eux, et qu'ils contiennent au moins un <title> ou un <incipit>. Dans ce dernier cas, **insérer un règle schematron dans l'ODD ?**

Toujours justifier l'insertion de plusieurs <incipit>, <explicit>, <note> etc. Dans un <msItem> selon le format.

Pbs à régler dans l'ODD :

- Texte refusé dans <layout>
- Faire en sorte que <note> puisse aller n'importe où dans <msItem> et pas dans un ordre car cela peut être gênant pour l'automatisation

20/05

Le but de l'automatisation est d'avoir la notice la plus juste et la plus précise possible selon les données que nous disposons. Cf. Question des <msItem> imbriqués : comment l'automatiser ? Est-ce possible à chaque fois ?

Trois paramètres sont essentiels dans cet exercice :

- Le document source
- La spécification du format
- L'intelligence personnelle

Dans un projet d'humanités numériques, il faut avoir conscience de quelle information on dispose et comment l'exploiter. Le projet HORAE est construit sur une méthode originale qui consiste à obtenir une bonne information à partir d'une information imparfaite.

25/05

Relecture de la proposition de modélisation :

!\\ Les `<note>` ne sont pas acceptées partout en TEI ! Par exemple, elles ne peuvent être l'enfant direct d'un `<msContents>`.

Le `<msItem>` est une partie d'une partie dans une partie de la notice.

Ne pas mettre le numéro de la notice dans un `<head>` mais dans un `@n` sur la balise `<TEI>`.

Le processus est dépersonnalisé, il s'agit de délimiter ce qu'il est possible de faire.

L'arbre de décision résume le résultat obtenu selon la méthode employée. Il permet de décrire sa réflexion sur des conditions : comment se présente le document source, ce que l'on crée et comment.

En ce qui concerne la modélisation de l'encodage (cf. Modélisation-note-msItem+d.docx), les locus dans les `<msItem>` sont intéressants en ce qu'ils répondent à des enjeux codicologiques. On peut par exemple comparer les heures de la Vierge calibrée par Obsecro Te pour savoir le taux d'abrévement relatif et donc la capacité d'alphanumerisation du lecteur.

Ainsi, le `<locus>` est obligatoire pour les feuillets extrêmes d'un `<msItem>`. Il se trouve être alors le premier enfant de `<msItem>`, mais il devient optionnel dans une `<note>`.

La balise `<note>` permet d'ajouter des commentaires sur le contenu encodé comme sur l'encodage (**? -> pas mieux d'utiliser dans ce dernier cas la syntaxe propre aux commentaires entre <!-- -->**).

Homogénéiser les structures des `<msItem>` et définir des cas-types. Par exemple, `<note>` figure seule, ou avec un `<locus>` si l'information existe, uniquement dans une description de feuillets blancs, manquants ou de lacunes.

Bien distinguer l'information de l'utilisation et de l'encodage.

L'arbre de décision doit refléter la structure des données, afin de voir ce qu'il faut prendre en compte dans les futurs programmes de l'encodage automatisé.

27/05

L'arbre de décision doit refléter davantage la structure du document source.

Continuation des tests pour matcher les informations : bien vérifier qu'il n'y a pas eu de pertes de données lors de la transformation.

L'automatisation paraît particulièrement difficile pour les <msItem> et leurs balises enfants : trop d'exceptions, notamment dans les citations, pour que l'information soit correctement encodée.

3.1.3 Définir les outils

Pour la conversion automatisée du catalogue vers le format TEI, deux choix s'imposent :

- Partir du document word océrisé et encoder à coup de regex, entre autres.
- Partir de la pré-structuration en TEI réalisée via Oxbagage et créer un code en XSLT ou en Python.

27/05

Liens vers documentation pour utiliser regex en Python sur un document Word et générer un document xml :

- <https://docs.python.org/fr/2.7/library/re.html#re.match> (package re)
- <https://docs.python.org/fr/3/tutorial/controlflow.html> (fonction range pour <msItem>)
- <https://www.geeksforgeeks.org/python-format-function/> (fonction .format{} pour <msItem>)
- <https://www.fil.univ-lille1.fr/~marvie/python/chapitre4.html> (génération d'un xml)

```

Entrée [31]: 1 import re
2 import docx2txt
3 import lxml
4 import xml.etree.ElementTree as ET
5
6 notices = docx2txt.process('/Users/gwenaellepataf/Desktop/Stage_TNAH/MémoireHORAE/Catalogue_VI/noticesTestsoocr.docx')
7
8 paragraphe = re.search('.*\n.*\n', notices)
9
10
11 #from lxml import etree
12
13 #teiCorpus = etree.Element("teiCorpus")
14
15 #print(etree.tostring(teiCorpus, xml_declaration=True, encoding="UTF-8", pretty_print=True))
16 #encodage_notices = ET.ElementTree(teiCorpus)
17 #encodage_notices.write("/Users/gwenaellepataf/Desktop/Stage_TNAH/MémoireHORAE/Catalogue_VI/testnoticesauto.xml")

```

<re.Match object; span=(0, 48), match='1. LIVRE D'HEURES ET MISSEL FRANCISCAINS. 1380\n'>

Entrée []:

28/05

Finition de l'arbre de décision suivant la notice océrisée de Leroquais : tests de la dangerosité des regexs et ce sur quoi on peut matcher. Utilisation de regex101 :

The screenshot shows the regex101 interface with the following details:

- REGULAR EXPRESSION:** `/([Lacune|lacune]) entre(.+?)\.(?{1})(\s+\r-?)?/gm`
- TEST STRING:** A block of Latin text from a liturgical book, including various headings and numbered sections.
- EXPLANATION:**
 - Matches the character `L` literally (case sensitive).
 - 1st Capturing Group: `([Lacune|lacune]) entre(.+?)\.(?{1})(\s+\r-?)?`
 - 2nd Capturing Group: `([Lacune|lacune])`
 - 1st Alternative `Locune`: Matches the characters `Locune` literally (case sensitive).
 - 2nd Alternative `lacune`: Matches the characters `lacune` literally.
- MATCH INFORMATION:**

Match	Full match	Group 1
1	23140-23165	Locune entre 108 et 110.
2	23141-23165	Locune entre 115 et 116.
3	23141-23147	Locune
	23153-23164	108 et 110
- QUICK REFERENCE:** A sidebar with common regex symbols and their meanings.

Solutions Python/XSLT à approfondir.

09/06

Documentation	regex	en	XSLT	:
https://www.irif.fr/~carton/Enseignement/XML/Cours/XSLT/index.html				;
https://www.xml.com/pub/a/2003/06/04/tr.html				;
https://stph.scenario-community.org/contribs/doc/cdt/xslt2/co/expression_reguliere.html				

Comment prendre un document océrisé au format word dans XSLT ?

Possibilité de créer des variables pour les différents états du document, puis de faire des select et de jouer avec les modes.

Faire une action à la fois et les numéroter. Vérifier à chaque fois l'intégrité des données, et la validité au format TEI, donc attention à l'imbrication des balises.

Chaque décision de l'arbre correspond à une action.

Commencer par le plus englobant pour affiner ensuite. 4 parties principales se distinguent :

- Le summary
- Le msContents
- Le physDesc
- L'history

Copier intégralement le document source et faire une action à chaque fois, soit ajouter les balises autour des bonnes information. Documenter chacune de ses actions.

Comment matcher la notice en entier ?

Attention aux différents types de tiret (c'est un petit entre 2 numéros de notice)

Tentative avec Python

Code pour récupérer le numéro de la notice :

```
num_notice = re.search('\n((\d|\d\d|\d\d\d)[0-9]+(-[0-9]+)?)(\.).*?)([A-Z]{3,})+\n', notices)
print(num_notice.group(1))
```

On capture seulement la référence “112” et pas le “1” et le “313”.

S la formule suivante fonctionne sur regex101 : ^(([0-9]|\d|\d\d|\d\d\d)[0-9]+(-[0-9]+)?)(\.).*?)([A-Z]{2,3})+

The screenshot shows the regex101.com interface. On the left, there are save/share options, flavor selection (PCRE (PHP)), tools like code generator and regex debugger, and a sponsor section for Work Happens. The main area shows a regular expression search for 'LIVRE D'HEURES ET MISSEL FRANCISCAINS' in a test string. The regex is `/^((0-9|I|l)+(0-9|l)?)(\.,)*(A-Z){2,3}\./`. The results panel shows 3 matches, 1040 steps, and an explanation of the regex components. The explanation details the use of capturing groups and quantifiers. The match information panel shows the full match and its components.

Elle ne renvoie rien via le parseage du doc océrisé en Python.

Problème avec le “^” dans regex en python : **comment matcher le début de la ligne ?**
 Comment faire pour que les diverses occurrences repérées sur regex101 soit aussi prises en compte en Python ? En effet, la regex suivante ne renvoie qu'un titre sur les trois notice quand on utilise python : `titre_notice = re.search('[0-9|I|l]+\\.(.*[A-ZÉÀÙ]+)\\.', notices)`

```

Entrée [31]: 1 import re
2 import docx2txt
3 import lxml
4 import xml.etree.ElementTree as ET
5
6 notices = docx2txt.process('/Users/gwenaellepata/Desktop/Stages_TNAH/MémoireHORAE/Catalogue_VL/TestEncodeNoticePython.docx')
7
8 paragraphes = re.findall('.*\n', notices)
9 num_notice = re.search('((0-9|I|l)+(0-9|l)?)(\.,)*(A-Z){2,3}\.', notices)
10 titre_notice = re.search('([0-9|I|l]+\.(.*[A-ZÉÀÙ]+)\.|\n)', notices)
11 print(titre_notice.group(1))
12
13
14 """
15 from lxml import etree
16
17 teiCorpus = etree.Element("teiCorpus")
18 TEI = etree.SubElement(teiCorpus, "TEI")
19 TEI.set("n", num_notice.group(1))
20
21 print(etree.tostring(teiCorpus, xml_declaration=True, encoding="UTF-8", pretty_print=True))
22 encodage_notices = ET.ElementTree(teiCorpus)
23 """

```

LIVRE D'HEURES ET MISSEL FRANCISCAINS

```

Out[31]: '<?xml version="1.0" encoding="UTF-8"?><teiCorpus><TEI n="1"><!-- notice --><!-- notice --><!-- notice --></TEI></teiCorpus>'
```

10/06

Étapes de la structuration des informations dans les notices :

- 1) Générer les notices dans leur intégralité
- 2) Délimiter les notices une par une (Titre + contenu)
- 3) Délimiter les structures les plus englobantes (titre, contenu, description, biblio), ce qui correspond en structuration TEI à <summary>, <msContents>, <physDesc> et <additional>, avec parfois un historique (<history>) entre le contenu et la description matérielle

Le document océrisé est une chaîne unifiée de caractères, donc les wildcard “^” et “\$” ne peuvent fonctionner.

Faire précéder le caractère d'échappement “\” par “r”.

Le texte est riche en combinaisons alphanumériques, donc procéder avec des regexs trop génériques est dangereux.

Code pour diviser les notices et associer un titre à son contenu :

```
****

import re
import docx2txt

notices = docx2txt.process('/Users/gwenaellepatat/Desktop/Stage_TNAH/MémoireHORAE/Catalogue_VL/noticesTestocr.docx', 'r')

#Capture des titres de notices dans l'ordre stockées dans un dictionnaire
livres_heures={}
for char in range(1, 320):
    try:
        titre=re.search(r"\n"+'((\| ){1,2})?'+ str(char) + "\." + '.*?[A-ZÉÈÙÀ]{2,}.*?\n', notices).group(0)
        livres_heures[char]=titre
        #print(titre)
    except: #Exceptions utiles pour détecter les possibles dénominations irrégulières
        #print(char)
        continue

#Utilisation des valeurs du dictionnaire, soit les titres de notices, pour diviser le texte et associer le contenu à chaque titre
regexPattern = '|'.join(map(re.escape, livres_heures.values()))
contenu=re.split(regexPattern, notices)
```

```

#Zip de la liste pour créer une liste de tuples contenant les titres et leur contenu
structure_notices = list(zip(livres_heures.values(), contenu[1:]))

for char in structure_notices:
    fichier = open("/Users/gwenaellepatat/Desktop/Stage_TNAH/MémoireHORAE/Catalogue_VL/notice_Leroquais.txt", "a")
    fichier.write('<TEI> Titre : %s \n Content : %s </TEI> \n\n' % (char[0], char[1]))
    fichier.close()
    #print('<TEI> Titre : %s \n Content : %s </TEI> \n\n' % (char[0], char[1]))
    ****

```

Le texte peut ainsi être parcouru item par item, ce qui permet d'utiliser des regexs séquentielles moins complexes et de baisser le taux d'erreurs.

La regex pour récupérer les titres fonctionne pour les erreurs d'océrisation ("l" et "l" à la place de 1 dans certains numéros de notices), mais pas pour la seule exception de Les_livres_d'heures_manuscrits_de_[...]Leroquais_Victor_bpt6k15129612_notices_tout : "311-312. CREDO DU SIRE DE JOINVILLE ET LIVRE D'HEURES. XIIIe SIÈCLE FIN, ET XIVe SIÈCLE FIN". En effet, la notice ne comprend pas un mais deux numéros séparés par un tiret. Comme il s'agit de la seule exception, il est plus prudent de la corriger à la main, afin de ne pas créer d'erreurs dans la récupération des titres par ailleurs.

On peut générer un document de sortie au format .txt qui met en avant la structuration des notices. Les balises TEI peuvent être rajoutées dans un deuxième temps.

11/06

Le rajout des balises TEI se fera une fois qu'on aura tout récupéré.

Pour récupérer le contenu et diviser les informations, il faut faire une liste de liste en passant par un dictionnaire. Une fois que les titres et les contenus sont séparés, il suffit d'enquêter sur le contenu. La liste principale sert donc à la fois une entrée et une sortie dans le code. En effet, on ajoute systématiquement un nouvel élément à partir d'une itération sur le contenu.

Extrait :

```

#Zip de la liste pour créer une liste de tuples contenant les titres et leur contenu
structure_notices = list(zip(livres_heures.values(), contenu[1:]))

```

Le "contenu[1:]" signifie tout ce qui suit l'index 1, car le texte est séparé en nombre de titre + 1 (pour le contenu), donc on utilise une tranche pour associer le contenu à son titre.

Le zip() est utile pour faire un join entre deux listes de la même longueur.

Pour division du contenu :

- Regex suivante incertaine pour le contenu des manuscrits -> r"(^Bibliothèque nationale,){1}.*?\n+" + '(.?*\n.?)+' + r"^(Pareil\.{1}|Parch\.{1})" **À creuser**
- Contenu sur la description matérielle capturée :

```
#Capture des descriptions matérielles des manuscrits
for index, char in enumerate(structure_notices):
    try:
        physDesc=re.search(r"(Pareil\.|Parch\.).{1}((.*?)\n)*(Rel\.|Rcl\.|Demi\.-reliure){1}(.*?)(-\?)?", char[1]).group(0) # char[1] est le contenu dans notre tuple
        structure_notices[index].append(physDesc) #Modification de la liste principale
        #print(physDesc)
    except :
        print(index)
        Continue
```

- Contenu sur la bibliographie capturée :

```
#Capture de la bibliographie
for index, char in enumerate(structure_notices):
    try:
        additional=re.search('^(Rel\.|Rcl\.|Demi\.-reliure){1}(.*?)\.\s+\-\s+([A-ZÉÈÀÙ]{1}.*)\n', char[1]).group(3) # char[1] est le contenu dans notre tuple
        structure_notices[index].append(additional)#Modification de la liste principale
        #print(additional)
    except :
        print(index)
        Continue
```

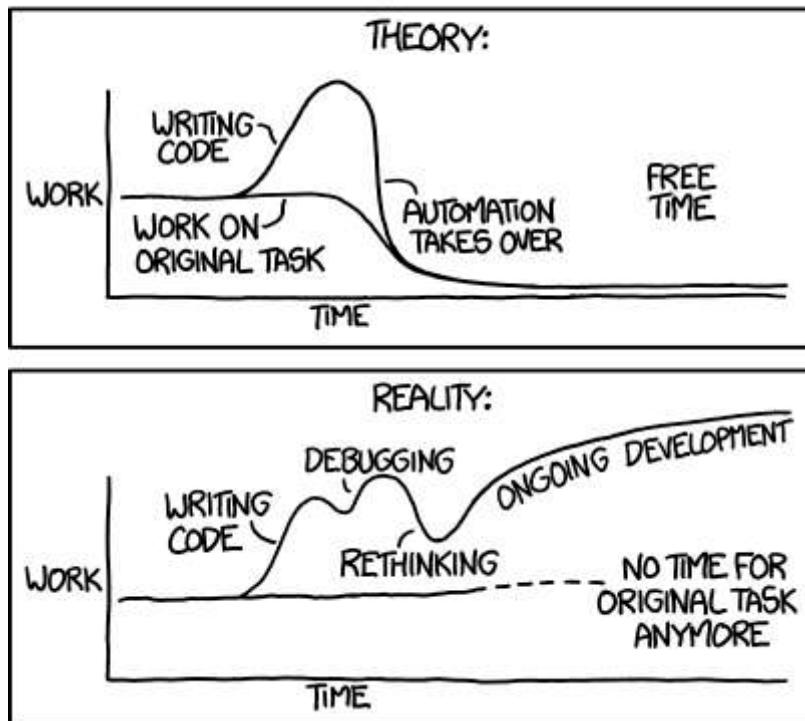
Toutefois attention -> certains noms sont en petite capitales et Python les détecte comme des lettres minuscules, donc attention à la regex.

- Regex partiellement correcte mais délicate pour ce qui concerne l'historique du manuscrit car sauts de lignes créés par des sauts de pages et capture de la description physique pour la notice 313.

3.1.4 Reprise

3.1.4.1 Is your journey necessary ?

**"I SPEND A LOT OF TIME ON THIS TASK.
I SHOULD WRITE A PROGRAM AUTOMATING IT!"**



<https://xkcd.com/1319/>

Quand l'automatisation n'est pas plus rapide que la main, c'est qu'on a pris le mauvais chemin.

3.1.4.2 Analyse de la notice (bis repetita)

La notice est toujours structurée de la façon suivante :

- 1) numéro d'ordre [obligatoire / mandatory]
- 2) résumé de contenu [obligatoire / mandatory]
- 3) date [obligatoire / mandatory]
- 4) cote [obligatoire / mandatory]

- 5) contenu [obligatoire / mandatory]
- 6) historique (optionnel)
- 7) description physique [obligatoire / mandatory]
- 8) description du décor (souvent dans un ou plusieurs paragraphes propres, mais si le décor est peu important, dans le même paragraphe)

- 9) reliure [obligatoire / mandatory]
- 10) bibliographie (optionnel, dans le même paragraphe)

Les éléments 1-5 peuvent se créer en une seule fois.

Ensuite, on compte le nombre de paragraphes avant la notice suivante ; et on regarde combien de paragraphes répondent à une regex de type « commence par ‘Rel.’ » ou de type « commence par ‘Parch’ » ; s'il n'y en a qu'un seul, on le fait précéder par le tag correspondant.

12/06

Les tuples sont immuables, donc transformer la liste de tuples en une liste de listes pour pouvoir rajouter les éléments délimités.

/!\ Éviter le .txt en format de sortie, car on risque de perdre la structure, et il n'est ici pas besoin d'enregistrer les données à chaque étape. Pour conserver la structure de la liste finale, on peut utiliser en format de sortie un csv ou un json à partir desquels on peut générer d'autres formats comme la TEI ou RDF.

Dans le cas de la structuration des notices, où l'on veut passer de données semi-structurées à des données structurées, se demander ce qui est le plus rentable : passer par un code python ou xslt ou bien le faire sur le document source lui-même au format .docx ? Ici, structurer à la main permet de récupérer plus vite certaines informations dangereuses à récupérer avec des regexs, en tenant compte de critères comme la taille de la police ou l'enchaînement des paragraphes.

Dans un premier temps, corriger les erreurs d'OCR en vérifiant leur récurrence :

- suppression des réclames
- remplacement de “Arnen” par “Amen”
- remplacement des “l” et des “l” dans n° de notices par 1
- remplacement de “Pardi.” par “Parch.” (notice 36)
- remplacement de mi ll. par mill. (notice 56)
- remplacement de xme par xiiie
- remplacement de Rei. En Rel. (notice 81)
- remplacement de Kel. En Rel.

10 mots-clés résument la structuration de la notice (cf. Supra). Tenir compte de l'ordre et des paragraphes du document pour faire apparaître la structure.

Dans Word, utiliser la fonction Rechercher-Replacer et créer des styles. Adapter les outils à ce que l'on souhaite faire. On peut jouer sur tous les critères de mise en forme (ex. : § avec le mot “usage” de police 16 = partie sur l'historique).

Mettre en avant la structure de la notice avec des styles. Il faudra ensuite les récupérer (via XSLT ou Python ?) pour les convertir vers le format cible en TEI (cf. Oxgarage).

Particularités du document source :

- Certains n° de notices ne sont pas suivies de . (cf. 29 et 46)
- Notice 152 -> le titre n'a pas été pris en compte car la police est différente (justification centrée et pas à gauche)
- Irrégularités dans les dates à la fin des titres -> mention du siècle avec précision entre parenthèses entre telle date et telle date
- Pas de date dans le titre de la notice 265

Vérification de la structuration à poursuivre.

15/06

Point sur transformations avec Python :

On utilise souvent python en faisant des transformations vers texte plat, donc les métadonnées de style ne sont pas gardées. On peut toutefois utiliser les packages python_docx et docxtxt pour séparer le texte en italique et en gras du texte normal, ou obtenir la taille de la police.

Cependant avec l'OCR, cela peut donner des résultats déstructurés, donc utiliser davantage les séries de folios en début de paragraphe avec une condition if et une fonction startswith(). C'est une fois seulement que les n° de folios sont séparés que l'on peut jouer sur les métadonnées de style.

Vérification des styles dans l'OCR :

/!\ Variations sur les tirets -> cf. notice 104 avec 2 petits tirets au lieu d'un tiret long avant la bibliographie.

Suppression des réclames.

Correction de l'OCR :

- Anien -> Amen
- Eel. -> Rel.
- Parchb. -> Parch.
- Inill. -> mill.
- min. -> mill.

Structuration de la description matérielle :

Les parties à séparer sont :

- Le matériau utilisé (parchemin, papier ou vélin) -> FAIT
- Nombre de folios -> Problème de regex
- Nombre de colonnes -> FAIT
- Dimensions -> FAIT
- Description de la décoration -> **Comment encadrer tous les paragraphes concernés par un seul élément <decoDesc> ?**
- Reliure -> **Problème sur les styles de l'OCR** car le style associé à la reliure englobe parfois la bibliographie dont le style n'apparaît pas donc reprendre cela dans l'OCR pour que les deux styles ne se chevauchent pas /!\

Choix d'utiliser xslt à partir du document océrisé convertie en xml TEI P5 via Oxgarage.

Vérification des matchs sur 3 notices tests. Dans une premier temps, création d'une feuille xsl par match -> **comment faire plusieurs match sur un même xsl:template ? Avec des conditions xsl:if, xsl:when et/ou xsl:for-each ?**

16/06

Correction des styles dans l'OCR structuré (superposition style Reliure et style Références bibliographiques). Re-génération d'un document au format xml via Oxgarage.

Suite Structuration de la description matérielle :

- Reliure -> problème réglé (matcher à la fois sur le `<p rend='reliure'>` et `<hi rend='Reliure_Car'>`)
- Description de la décoration -> problème semi-résolu. Différents tests, le plus satisfaisant encadre bien les différents `<p>` concernés par un `<decoNote>`, mais est-ce un problème si les autres `<p>` en-dehors de la description de la décoration ne sont pas générés ?
- Nombre de folios -> malgré différents tests (changement de regexs, prendre ce qui n'est pas matché en dehors du matériau et des dimensions du manuscrit), information pas matchée. **À creuser.**

17/06

Structuration du contenu des manuscrits :

- Résumé du contenu -> FAIT en partie (manque le **1er § optionnel** s'il ne s'agit pas de la description du feuillet de garde ou des anciennes cotes)

- Les différentes unités textuelles -> FAIT en partie (qu° autour de comment structurer à l'intérieur d'un § : **condition avec tests** d'infériorité ou de supériorité avec tels et tels folios)
- Les folios -> FAIT en partie car risques d'erreurs avec les regex donc **à structurer en dernier une fois les autres parties délimitées ?**
- Les titres de parties -> FAIT en partie
- Les rubriques -> **à creuser**
- Les incipit
- Les citations
- Les explicit
- Les notes/commentaires

18/06

Tuto XSLT/Math : <https://haypo.developpez.com/tutoriel/xml/xslt/programmation/#LV-A>

Retour sur la structuration des msItem :

Faire une condition mathématique qui dit que si le numéro de folio est plus grand ou strictement égal que le premier folio de la tranche et plus petit ou strictement égal au dernier folio de la tranche, alors c'est un même <msItem>. En revanche, si le n° de folio sort de cette tranche définie en début de paragraphe, on crée un nouvel <msItem> : **comment le mettre en application en XSLT ?**

Essai en partant d'un document source dont les n° de folios sont encadrés par la balise <num> et l'@value pour pouvoir les matcher plus facilement dans la feuille xsl. /!\ ce document source n'a plus les métadonnées de style à l'intérieur des paragraphes concernés par la transformation (ceux se rapportant au contenu des manuscrits), donc à utiliser uniquement pour structurer les <msItem>. De plus, les folios indiqués par une lettre capitale ne sont pas pris en compte (sinon trop d'informations sont encadrées par <num> qui prend déjà trop, notamment les jours du calendrier), et il faut prendre en compte l'information "v°". **À creuser.**

Structuration des titres :

Fait en grande partie, il manque les titres précédés d'un folio indiqué par une lettre capitale (une regex les prenant en compte matche bcp trop d'informations, tous les S. etc.). Toutefois problèmes pour récupérer en entier les titres traversés par un saut de page dans le document océrisé.

Structuration de la partie <rubric> :

Faire une condition qui dit que quand le texte en italique est précédé de guillemet ouvrant et d'un début de citation, alors il faut créer un élément <rubric>. Tentative d'utiliser la fonction substring-after() avec comme arguments l'élément <hi> et le début de la citation : problème de syntaxe pour imbriquer utilisation d'une regex, d'une variable et d'une condition xsl:if. **À creuser.**

19/06

Structuration de l'incipit :

Matcher ce qui est entre le premier guillemet ouvrant et les 3 points de suspension suivis d'un tiret, en enlevant le <hi rend='italic'>. Attention car on matche dans ce cas uniquement les incipit se terminant par des points de suspension.

De plus, pour ne pas générer d'erreurs inutilement, il faut **mettre les calendriers à part dans le document source** : les faire ressortir via une transformation xsl ou une structuration par style dans word.

Point général :

Définir l'ordre des opérations via l'enchaînement des éléments que l'on créé directement dans la feuille xsl à l'intérieur d'un template. De toute manière, les opérations suivent la structure du document source, donc bien matcher les bons templates.

Utiliser la version 3 d'xslt et copier la structure du document source dans le document de sortie pour s'assurer de ne pas perdre d'informations.

Mettre l'@namespace pour chaque élément : **à vérifier.**

Penser aux fonctions relatives aux chaînes de caractère, comme start-with(), pour ne pas utiliser systématiquement des regexs.

/!\ à la division entre le nombre de folios, de colonnes ou l'état matériel du manuscrit (<condition>), car il faut tokéniser la phrase contenant l'information pour garder des parties et pas d'autres selon ce que l'on veut matcher (regex ou fonction tokenize()).

Corriger les paragraphes créés par des sauts de page dans le document source.

Penser aux <xsl:apply-templates/> pour copier ce qui est dans le document source (application des mêmes règles à l'intérieur d'un template).

Pb des <msItem> :

- 1) Diviser par §
- 2) Diviser par tiret (attention aux citations, jouer sur les guillemets)
- 3) Jouer sur les n° de folios (cf. xsl:for-each, fonction tokenize() sur page et tiret)
- 4) Créer a posteriori les <locus> en ajoutant les @from et les @to. => Si la valeur de <locus> est contenu dans la tranche de <locus> précédents, alors il faut inclure le <msItem> à l'intérieur.

Continuation de la structuration du <physDesc> :

/!\ Une fois le document transformé, supprimer les <p/> vides de <decoNote> créés par l'OCR.

3 possibilités dans la matière utilisé :

- Parchemin (@material='parch')
- Papier (@material='paper')
- Vélin (@material='vellum' ?) -> **Non, il s'agit de parchemin également**

22/06

/!\ au format cible -> ajouter l'élément <condition> ; certains @ ; l'élément <layout> quand nous n'avons qu'une colonne ("à longues lignes dans la notice").

Dans l'xsl, créer la structure au niveau des <div> du document source, et faire ensuite des apply-templates avec un TEI par <div>.

Point sur les problèmes :

Au niveau de la description matérielle et de la décoration :

- Problème de regex pour matcher la bonne information dans le <layout> (l'ensemble des informations dans <p @rend='Codico'> sont matchées), seuls les manuscrits à une colonne sont bien encodés.
- Problème de regex pour matcher les bonnes informations dans l'élément <condition>. Elle fonctionne seulement pour la notice test 112 ("plusieurs lacunes"), et avec une condition test elle ne fonctionne pour aucune. Le choix de copier l'ensemble des information du <p @rend='Codico'> n'est pas conforme à ce que signifie l'élément <condition> en TEI mais permet d'éviter une perte d'informations.
- Des <p/> vides sont créés à la fin de chaque <decoNote> -> faciles à enlever au moment du nettoyage final de l'xml cible.

Au niveau de l'historique et des éléments contextuels :

- Dans <origDate>, on peut facilement récupérer la date mentionnée dans le titre de la notice, mais plus difficilement dans un paragraphe. Si l'on joue par exemple sur la mention du mot “date” à l'intérieur d'un paragraphe, cela peut créer des erreurs car l'information est parfois échappée dans plusieurs paragraphes qui ne contiennent pas tous la mention “date”.
- Problème de regex pour matcher les informations relatives à l'élément <provenance>, car jouer sur la mention de “possesseurs” ne fonctionne pas.

Au niveau des calendriers (msItem traité à part pour faciliter la structuration) :

- Comment faire une variable qui prend ce qui n'est pas matché par la regex dans le premier paragraphe (donc en-dehors de la foliation et du titre) afin d'injecter les informations non-matchées dans la balise <note> ? Choix (pour l'instant) de tout prendre dans <note> afin de ne pas avoir de pertes malgré la redondance d'informations.
- Les <title> sont tantôt bons tantôt vides , car, quand le titre ne suit pas la foliation, il n'est pas matché. Si on enlevait cette condition, on aurait trop de risques d'erreurs.
- Pour les <locus>, on peut effacer a posteriori lors du nettoyage de l'xml de sortie les chiffres suivant le premier point.

23/06

L'espace de nom déclaré dans l'élément parent s'applique aux éléments enfants, donc pas obligatoire de le répéter à chaque création d'éléments.

/!\ Pour alléger le code, son temps de réactivité et sa consommation énergétique, faire des variables qui regroupent la valeur d'attribut commune au contenu d'un élément.

Mettre un '.' dans @select de <xsl:value-of> pour éviter les doublons dans les résultats.

Pour rendre la structure de sortie plus précise, on peut agir en plusieurs phases de transformation. La première étape est déjà d'obtenir les grandes lignes d'une structure tei correcte que l'on a défini dans le format cible. On ne peut pas (tout de suite) automatiser la récupération des informations contenues dans <origDate> lorsque l'information est à l'intérieur de §, ou bien celles contenues dans <provenance>.

Idées pour affiner progressivement le document obtenu en sortie :

- Affiner le <teiHeader> du <teiCorpus>
- Enlever les <p/> et les <bibl/> vides
- Enlever les espaces parasites

/!\ Nettoyer implique avoir sali, ici il s'agit d'affiner ce qui est généré par certaines erreurs d'océrisation.

Difficulté à créer des `<msItem>` dans des `<msItem>` : faire un `<msItem>` par paragraphe concerné puis diviser avec les tirets -> pb avec la fonction `tokenize()`, car le 1er argument doit correspondre à un seul item. **D'abord diviser les paragraphes à partir des tirets sans faire de `<msItem>` imbriqués ?**

24/06

Attention à la validité TEI du document transformé (cf @namespace à attribuer aux éléments les plus englobants) et à sa conformité avec le format cible.

Faire des apply-templates dans la définition de la structure du format cible souhaité. Ces apply-templates renvoient à des templates spécifiques à chaque partie récupérée du document source. Cette structure de code est plus propre et permet de faire des modifications plus facilement.

Attention : trop d'apply-templates imbriqués peut créer des problèmes lors des boucles.

Point sur les `<msItem>` :

- Locus -> bonne récupération des chiffres souhaités mais surplus après le 1er point (récupération de tous les chiffres) donc réduire le nombre de matchs possibles (**pas plus de 3 chiffres pour éviter de récupérer les dates -> vérifier nombre de chiffres pour fol.**) + récupérer les **lettres capitales** quand cela est possible
- Title -> a priori ok mais attention aux **sauts de ligne**
- Rubric -> faire une variable qui récupère l'italique non pas dans les paragraphes de `<msItem>` mais dans les `<msItem>` hypothétiques séparés par des -. Faire une **condition avec un `<xsl:for-each-group>`** ?

(Pour les citations, choix de récupérer les structures les plus sûres dans un premier temps mais s'appuyer sur les points de suspension n'est pas une science exacte.)

- Inicpit -> récupération de ceux qui commencent par un guillemet et qui se terminent par des points de suspension
- Quote -> récupération de celles commençant et se terminant par points de suspensions

- Explicit -> récupération de ceux commençant par points de suspension et se terminant par un guillemet
- Note -> récupération de l'ensemble de l'information matchée par la variable \$hypothèse_msItem pour ne rien perdre. **Vérifier ce qui est matché dans <note>**

Jouer sur les tirets implique de trop diviser et de créer des **msItem fautifs** -> les regrouper lors d'une deuxième passe, selon les n° de folios, les informations présentes ou absentes de leur structure interne.

25/06

Mieux vaut ne rien créer que de créer des choses fausses. On peut donc récupérer certaines informations en plusieurs passes (cf. <rubric> : on récupère les informations communes à tout un paragraphe dans chaque subdivision de <msItem>, donc affiner la condition). Ne pas oublier la fonction normalize-space() pour enlever les espaces indésirables.

Suite structuration msItem (notices tests) :

- Pour les calendriers : les folios sont bien matchés dans <locus>. Les titres sont également bien dans la balise <title> sauf lorsque le nom du calendrier ne suit pas immédiatement la foliotation (cf. Notice 2), mais il est trop dangereux de rendre cette condition hypothétique dans les regexs, sachant qu'ils sont bien matchés dans la majorité : **mieux vaut ne rien créer du tout ou bien créer certaines balises <title> vides et la majorité remplie correctement ?**
- Pour les autres <msItem> :
 - Pour les feuillets, laisser ceux indiquant une citation avec la citation. Ces feuillets seront matchés à l'intérieur de la citation. /!\ les folios marqués d'une majuscule ou d'une minuscule ne sont pas systématiquement récupérés. Si la majorité est correctement encodée, certaines exceptions ne sont pas matchées, pourquoi ?...Problème de regex ?
 - Pour les titres, ils sont correctement machés. Seuls ceux suivis d'un folio indiqué par une minuscule ne le seront pas, car ajouter cette possibilité pourrait générer trop d'erreurs.
 - Pour les différents types de citations, plusieurs problèmes se posent. Faire attention à ne pas récupérer dans les <incipit> ce qui est déjà dans <rubric>. Penser à **récupérer les n° de folios**. Si l'on peut jouer sur les guillemets et les points de suspension, voir dans quelle mesure on peut se passer de ces critères

pour encoder semi-automatiquement. **Séparer les citations regroupées dans une même <note> à partir des n° de folios.**

29/06

Faire des apply-templates dans l'ossature générale du document de sortie, mais garder les éléments les plus englobants qui encadrent plusieurs apply-templates.

Pas obliger de faire une regex pour <summary> (même si le texte encodé est ainsi matché sans point et espaces) car on peut prendre ce qui entre <head> (ou faire un mode head) et faire des templates vides pour les enfants directs.

De même pour <locus> et <title> où l'on peut faire des substring-after et des substring-before, mais affiner pour bien matcher, donc cela vaut-il vraiment plus le coût que les regex ?

Attention : les paragraphes ne structurent pas systématiquement des <msItem> imbriqués. On peut toutefois créer une variable pour garder en mémoire ce qui appartient à un même paragraphe.

Travailler sur les conditions une par une : ce qui précède, ce qui suit, la ponctuation.

Re-structuration des paragraphes éclatés :

Dans les notices papiers de Leroquais, les paragraphes concernant le contenu du manuscrit commencent par Fol., Feuillet ou un chiffre. Ceux ne débutant pas de cette manière dans l'OCR sont à fusionner avec le paragraphe précédent. Toutefois attention -> certains sont mal-coupés sur les chiffres (cf. l.127 notice 1), donc vérifier à la main.

Une fois qu'on a matché les paragraphes mal-découpés, il faut les reconstituer, comment ?

Correction des paragraphes commençant par 0. Supprimer les balises <p> fautives.

Fusion des paragraphes suivis d'un paragraphe débutant avec un tiret -> attention à la restitution de paragraphes en double et aux tirets pas fusionnés car eux-mêmes précédés d'un paragraphe avec un tiret. Vérifier à la main sur les notices transformées.

Affiner le code pour faire de même avec les calendriers.

01/07

/!\ Quantifier les problèmes.

En tout, 1656 paragraphes ne commencent pas par ‘F’ ou par un chiffre.

Liste des paragraphes océrisés problématiques à restructurer :

- Ceux débutant par un tiret
- Ceux débutant par une minuscule
- Ceux débutant par un guillemet ouvrant

Il est plus délicat de restructurer des citations mises en exergue en jouant sur les seules majuscules, car la majuscule est par définition ce qui distingue le début d'un paragraphe. **Que faire dans ce cas-là ?**

Suppression des paragraphes parasites de l'OCR, indiquant notamment les sauts de page, comme celui-ci :

```
<p><hi style="color:red; font-size: 14pt; font-weight:bold;" rend="ERROR">□</hi><note place="margin" type="conversion" resp="#teitodocx"><hi rend="docxError">unable to handle picture here, no embed or link</hi></note></p>
```

605 paragraphes contiennent un <hi> avec l’@rend=’Error’ et 532 balises <note> contiennent un <hi> avec l’@rend=’docxError’.

La suppression de ces paragraphes dus à l’océrisation rend les codes de transformation d’autant plus efficaces.

Ne pas oublier de transformer les ‘I’ et les ‘I’ en 1 dans les numéros de notices.

02/07

!\\ Dans un code de transformation, plutôt que de faire un regex complexe, il est préférable de faire des extractions étapes par étapes (cf. Code python Sergio) afin de contrôler les éventuelles sources d’erreurs. Aller du général vers le spécifique, sachant qu’il reste toujours des exceptions.

Travail sur correction des erreurs d’OCR du document source :

- Suppression des <p/> vides (494 corrections automatisées)
- Fusion des paragraphes commençant par une minuscule avec les paragraphes précédents (environ 172 corrections à la main)
- Fusion des paragraphes commençant par des points de suspension avec les paragraphes précédents (correction automatisée)

- Fusion des paragraphes commençant par des guillemets avec les paragraphes précédents (correction automatisée)
- Fusion des citations commençant par une majuscule avec les paragraphes précédents (correction à la main) -> **Continuer la phase de vérification.**

Correction des "l" et des "L" dans les n° de notices. **Il faudra de même penser à corriger les locus.**

Une fois les notices structurées, à voir si l'on peut améliorer la structure de certains paragraphes mal-découpés hors du <msContents>.

03/07

Dernières vérifications des corrections apportées au document source pour :

- Les paragraphes commençant par une majuscule
- Les paragraphes commençant par un numéro
- La structure des paragraphes décrivant les calendriers

Vérification qu'il n'y ait pas de pertes d'informations.

Lors du code de transformation vers le document cible, il faudra penser à matcher les éléments enfants (cf les <hi> parfois autour des folios par exemple) mais aussi les <hi> avec l'@rend et le suffixe –Car dans la valeur d'attribut (c'est le cas pour les informations présentes à la fois dans des paragraphes entiers ou dans une partie du paragraphe, comme la description des calendriers et des reliures).

06/07

Vérification et simplification du code :

- @n ok
- <idno> ok
- <summary> ok -> le seul risque est que soit pris un premier paragraphe qui concerne la description du feuillet de garde et qui ne commencerait pas par 'F'
- Balises concernant la codicologie -> ok
- Balises concernant la décoration -> ok
- Balises concernant la reliure -> ok
- Balises concernant l'historique -> ok
- Balises concernant les références bibliographiques -> ok Mais penser a posteriori à **supprimer les <bibl/> vides** créer par la division sur les tirets
- <msItem> :

- Pour les folios, bonne récupération des couples de ff, mais **améliorer la normalisation pour les versos et améliorer la récupération d'un seul folio**
- **Distinguer les <title> des citations**
- /!\ Ne pas créer de <msItem> quand il n'y a pas de foliotation - > certains sont à fusionner
- /!\ **Régler le problème de la répétition des calendriers** -> quand un <hi rend='Calendrier_Car'> est encadré par une balise <p rend='Texte du Corps (2)'>, le premier paragraphe est répété deux fois dans la notice, donc faire une condition pour ne pas récupérer ce qui est entre <hi rend='Calendrier_Car'> dans les premiers paragraphes concernés.

07/07

Une fois le document encodé, plusieurs étapes dans la vérification de l'intégrité des données :

- Lecture continue avec le mode auteur d'oxygen et le framework défini pour le tei msDesc afin de rendre la lecture plus lisible pour vérifier à la fois la cohérence du fichier produit et l'absence de pertes d'informations
- Lecture ciblé en sélectionnant les informations déplacées par rapport au document d'origine (cf. Date -> dans titre de la notice papier mais dans <origDate> de la notice tei) et en comparant le document source transformé et le document produit grâce aux fonctionnalités de Word.

Vérification et simplification du code (suite):

- <msItem> :
 - Pour les folios, presque ok -> reste à **normaliser** dans une deuxième passe **les espaces pour les versos et à supprimer les folios fautifs** dûs à la sur-division des msItem. **Rajouter les folios marqués par une minuscule** (trop dangereux à générer automatiquement) et **vérifier ceux marqués par une majuscule**. /!\ pour les **calendriers**, vérifier folios pas matchés et les **ajouter** (cf. Notice 8) -> la suppression des 45 balises <hi rend='FontSize-15pt' xml-space='preserve'> n'a pas réglé le problème de capture.
 - Distinction des <title> ok mais parfois récupération de ce qui concerne la balise <note> donc **à vérifier et à supprimer si besoin**.

- Citations -> récupération des structures les plus évidentes et les plus sûres (guillemet ouvrant pour <incipit>, guillemet fermant pour <explicit> et points de suspension de part et d'autre pour <quote>). Les citations qui demandent de la compréhension et qui ne contiennent **pas de signes typographiques distinctifs particuliers seront encodés lors d'une deuxième passe**.
- <rubric> -> pas générés, **comment récupérer l'italique ?**
- /!\\ Ne pas créer de <msItem> quand il n'y a pas de foliotation -> certains sont à fusionner ou à imbriquer lors d'une deuxième passe
- /!\\ Lors de la vérification de l'intégrité des données, **ajouter les informations si elles manquent, supprimer les doublons** (cf. <hi rend='Calendrier'> ou premier paragraphe qui apparaît aussi dans le <summary>) et **corriger les éventuelles dernières erreurs d'ocr** (cf. "l" et "I" dans <locus>).

08/07

Choix de conserver l'ensemble du contenu du <msItem> dans la balise <note> afin d'avoir une vision d'ensemble de l'information matchée.

Problèmes à régler/réglés :

- Les **folios** de part et d'autre de 'à' ou de 'et' (couples de folios) indiqués comme **verso** s'affichent dans l'@from ou to avec un 'r' et le v° n'est pas remplacé par v -> cela n'est pas dû au pipe qui sert à indiquer deux possibilités (équivalent de 'ou'), l'enlever empêche les choses et alourdit le code. Les folios v° sont ainsi encodés comme si le v° n'était pas détecté. Est-ce à cause de la suppression du v° par rien dans la fonction replace() ? Tentative avec replace() v° par v dans un xsl:when et ajout de 'r' pour tous les autres cas (xsl:otherwise), mais cela crée le même problème...
- La **récupération de l'italique** n'est pas possible directement dans le template qui définit les <msItem>, car ils sont divisés sur les tirets par la fonction tokenize() puis les citations sont appelées grâce à une variable qui utilise la fonction substring-after() -> les fonctions relatives aux chaînes de caractère ne conservent que le texte et pas l'encodage du document source, donc il faut définir un nouveau template pour matcher l'italique. Reste à l'appeler correctement à l'intérieur de la structure des <msItem> -> **call-template ? Apply-templates ?**

- Faire des apply-templates dans la structure des `<msItem>` pour améliorer la lisibilité du code -> à **approfondir**
- Les 11 occurrences de numéros de folios à l'intérieur de citations (entre des points de suspensions) sont correctement encodés dans les `<locus>`
- Vérifier les folios des calendriers dans le document de sortie
- Penser à exclure l'italique des autres balises autres que `<rubric>` afin de ne pas avoir de répétitions -> utilisation d'une variable.

09/07

Problème du match de l'italique et de la récupération des `<rubric>` :

Tentatives avec apply-templates et mode pour insérer l'information dans la structure des msItem divisés sur les tirets => pb de saturation de la mémoire d'Oxygen qui a été augmentée. Trop d'apply-templates provoque aussi des problèmes de boucles. Toutefois, cela fait boucler l'information sur le paragraphe. Si l'on divise les `<msItem>` sur les paragraphes, l'information est beaucoup plus complexe à récupérer (plus la chaîne de caractère est longue, plus les regexs sont dangereuses). Toutefois, diviser sur les tirets fait perdre l'information de l'italique. La principale difficulté est donc qu'une partie des informations pour les msItem est plus facile à récupérer en divisant sur les tirets (l'utilisation de variables à l'intérieur permet de faire des regexs plus simples pour matcher correctement les informations dans la majorité des cas), tandis que l'italique ne peut se récupérer sur cette division (car la fonction tokenize() renvoie du texte sans encodage), donc l'italique ne pourrait se matcher qu'à l'échelle du paragraphe. Que faire ?

Idée de générer deux feuilles de sortie avec un id par paragraphe pour les msItem : l'une récupère toutes les informations avec uniquement les `<rubric>` dans les `<msItem>`, l'autre le reste des informations du msItem sans les `<rubric>`. Il faudrait ensuite fusionner, via python par exemple, les msItem qui ont le même @corresp dans les deux fichiers -> possible ? Si cela est possible, un des risques est la répétition d'information (si un incipit contient par exemple ce qui est contenu dans `<rubric>`), mais cela peut se corriger à la main.

Dans les documents de sortie, certaines informations relatives à la codicologie sont mal encodées à cause d'erreurs d'OCR (dans les chiffres notamment) -> à corriger lors d'une relecture.

10/07

Tuto pour fusionner deux documents xml :

- Via xslt :
http://www.journaldunet.com/developpeur/ressource/xml/xml_merge.shtml ->
fonction document() ; <http://pageperso.lif.univ-mrs.fr/~pierre-alain.reynier/XML/files/cours3.pdf> (p.42-43)
- Via python : <https://stackoverflow.com/questions/15921642/merging-xml-files-using-pythons-elementtree>
- Via XQuery : <https://stackoverflow.com/questions/55683844/xquery-combine-two-xml-files>

Attention à où on se place dans l'arborescence pour les apply-templates : un apply-templates qui renvoie à une division sur les paragraphes ne peut s'appliquer dans une structure divisée sur les tirets.

Expérience de transformation en 2 passes :

- Division des msItem sur les paragraphes en récupérant les <rubric> et l'ensemble du contenu dans <note>, ainsi que le reste de la strcuture du document cible
- Prendre le document de sortie en document d'entrée et diviser les msItem sur les tirets en les structurant
- Dans derniers documents de sortie, remettre les <rubric> dans les bons <msItem> en regardant les <note>

/!\ Dans l'ODD définie, rendre l'@quantity de <measure> et l'élément <dimensions> optionnelles (car pas infos pas présentes dans toutes les notices).

2 manuscrits sont analysés dans la notice 242 donc problème de validité TEI avec 2 <supportDesc>, 2 <layoutDesc> et 2 <bindingDesc>.

15/07 et 16/07

Vérification à la main de l'intégrité des données du document de sortie en 2 passe :

- Suppression des balises autofermantes (<bibl/>)
- Relecture

Cette étape est essentielle pour ce qui n'était pas automatisable (exceptions) :

- Les <rubric> : l'italique du document source concerne parfois des titres d'ouvrage dans les commentaires ou bien des termes en langue étrangère (ici en latin en l'occurrence) qui ne sont pas en début de pièce textuelle (cf. Récurrence de "sic"). De plus, certaines <rubric> s'allongent sur plusieurs morceaux de phrase divisés par un tiret, élément sur lequel nous avons divisé les <msItem>.

- Certains numéros de folios se suivent sans être séparés par un tiret, et donc ne sont pas toujours suivis d'un point.

À la fin de la vérification, il faudra appliquer l'ODD définie.

Cas du <layout> avec parfois une ou deux colonnes dans le même manuscrit : mettre "1 2" dans valeur de l'@columns (cf. Teiguidelines).

Cas des folios bis -> information à intégrer dans @n, @from ou @to de <locus> sous la forme suivante : "xxx_bis_r".

20/07

Dans code xsit, blocage pour problèmes de chaînes de caractères mutilées dans <title> et pour affichage du "v" dans @ from ou @to pour les tranches de folios.

Comment faire une condition sur folios pour les msItem imbriqués en XSLT ? Utiliser les @ from et @ to et faire des conditions avec les signes <= et >= ?

Dans les exceptions, /!\ aux "P." devant les n° de pages.

Poursuite de la re-structuration à la main (presque 1/3).

Possible de fusionner les <rubic> dans les msItem grâce à l'@ corresp de <msItem> avec XQuery d'après la requête suivante :

```

let          $tout           :=          doc("reprise/tout.xml")
let          $rubicDoc        :=          doc("reprise/rubric.xml")
return
for         $msItem          in          $tout//msItem[@corresp]
let          $corresp         :=          $msItem/@corresp/string()
return
for         $rubic            in          $rubicDoc//msItem[@corresp/string()=$corresp]//rubic
return
insert      $rubic           node       $rubic
as last into $msItem

```

On liste tous les <msItem> qui ont l'@corresp dans le document qui contient toutes les informations sauf les <rubic> ; on trouve son équivalent pour chaque @corresp dans le document qui liste toutes les <rubic> ; ensuite, pour chaque élément <rubic> partageant un <msItem> avec le même @corresp, on le copie dans le document qui récupère toutes les informations sauf les <rubic>.

21/07

Base de données avec variantes textuelles de livres d'heures de la fin du MA :
<http://www6.sewanee.edu/beyonduse/>

Référence : *Beyond Use* de J. Plummer et G. Clark

Remarque : en théorie, on peut laisser des informations strcuturées dans des documents séparés (cf. <rubric> d'un côté et reste des informations de l'autre), mais dans la pratique, en SHS, on privilégie la fusion des documents.

22/07

Autre manière de récupérer une information pour l'injecter dans un autre document : fonction doc() avec xslt.

Point sur le choix des données : Le lieu de conservation ne nous dit rien de l'origine et de la provenance ancienne des livres d'heures. Ils sont ici sélectionnés selon la disponibilité de la numérisation en IIIF. Les catalogues permettent de savoir la distorsion approximative du corpus avec les manuscrits connus pour le projet. Il faut en effet se rappeler que l'on travaille avec moins de 10% des sources en histoire médiévale, et que l'on peut avoir une mauvaise vision de ce qui est perdu. Reste donc la question insoluble des pertes inégales de manuscrits. L'intelligence artificielle permettra alors de distinguer les traditions communes ou qui diffèrent. Se pose donc la question de la représentativité dans la sélection : on prend le plus possible, et l'on corrige la déformation au besoin. Qu° des œuvres attestées mais non transmises (cf. Thomas Haye pour la littérature latine).

29/07

/!\ fol.i dans l'OCR = fol.1 chez Leroquais + livres d'heures avec des pages et non des folios -> ne pas mettre recto ou verso dans ce cas (notices 83, 149, 185, 290, 300, 310). Exceptions à ne pas automatiser + Cas des pertes de données dans l'OCR -> cf. Notice 151 avec manque entre "lesu piissime..." et redemptor...", manque rétabli.

Retouche du code :

- la création des balises <measure> et <dimensions> doivent être conditionnées car pas toujours les informations associées dans les notices, et leur création automatisée peut rendre le document invalide si elles sont vides
- Mettre 'col' et pas 'col.' dans les conditions (regex et string fonctions) car parfois emploi du mot 'colonnes" et non de l'abréviation 'col.'

31/07

Dans un **document de passation**, être le plus précis possible :

- Dater l'état du travail accompli
- Décrire les différents documents et bien nommer les fichiers, mettre en valeur la documentation
- Faire un plan récapitulant l'objet, la description des documents, les actions menées, les actions qu'il reste à accomplir en les illustrant avec des exemples

/!\ Pour notices, replacer la description des calendriers à l'intérieur de la notice (récupérer de manière différente que les autres <msItem>) selon sa foliation. Attention aux <finalRubric> (italique à la fin d'une citation), remettre le <locus> à chaque partie, remplacer les « o » par des « O » (erreurs d'OCR). Pour pb de foliation dans les tranches de folios (@from et @to), faire un rechercher-replacer « v° r » par « v ».

Pour alignement texte/image dans Transkribus, faire « Text2Image » dans « Tools ».

3.2 Cursus

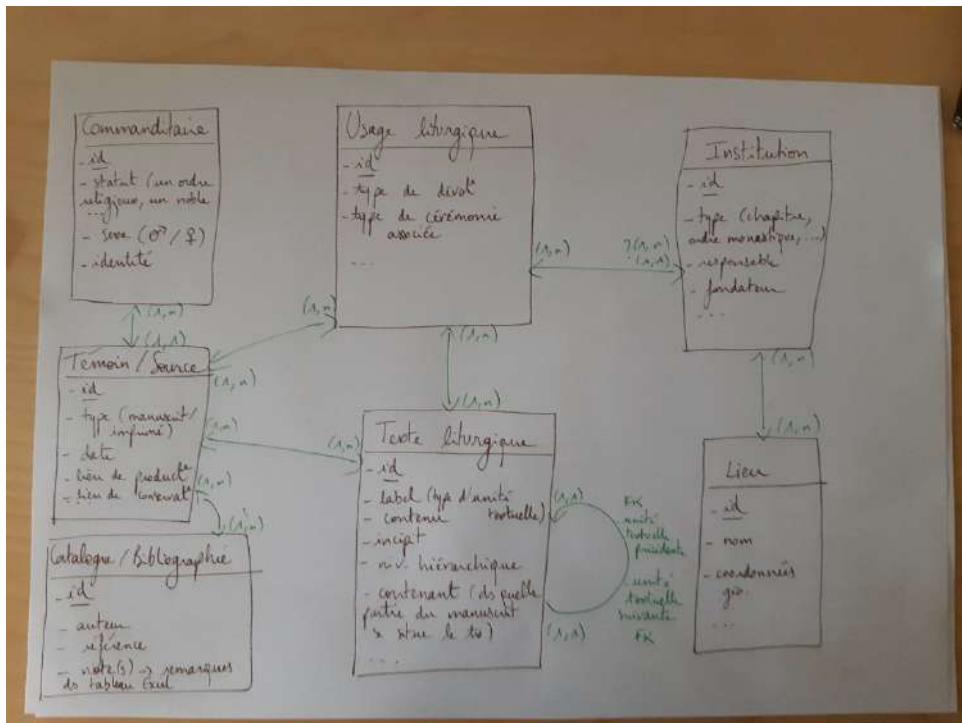
3.2.1 Modéliser les informations de cursus

21/04

Comment décrire un usage dans une base de données ?

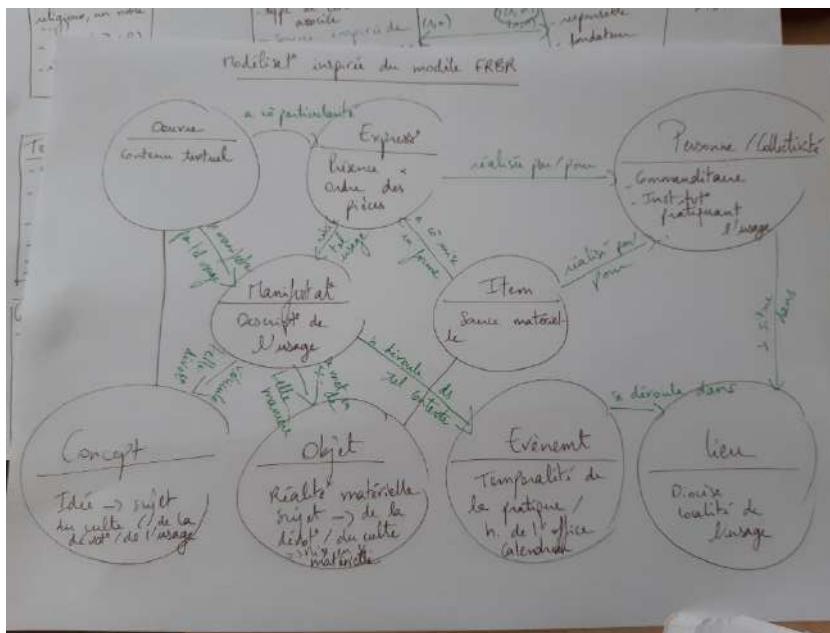
Dans le cas du Cursus, penser la modélisation à la fois de la hiérarchie des objets et leurs ordres (2 informations spécifiques à l'usage liturgique). La base de données doit répondre à la question : Qu'est-ce que le texte du livre d'heures, dans sa forme, son contenu, sa place au sein des autres textes, nous dit des usages liturgiques propres à une institution ?

Premier essai de modélisation :



Idée des clés étrangères pour rendre compte de la place de l'unité textuelle au sein de l'œuvre (table de relation une unité textuelle à une autre avec couple de deux clés étrangères, l'une renvoyant à l'unité textuelle précédente et l'autre à l'unité textuelle suivante), mais le faire pour un niveau de granularité le plus petit possible (une pièce au sein de l'œuvre mais pas l'œuvre elle-même). Par ailleurs, il est plus pertinent de rassembler le commanditaire et l'institution, car le premier peut aussi bien être une personne qu'une collectivité. De plus, les attributs de lieux de production et de conservation au sein de l'entité "Témoin" peuvent être changé en un lien vers l'entité "Lieu".

S'inspirer du modèle FRBR est intéressant pour notre réflexion, mais n'est pas tout à fait applicable dans le cadre de notre modélisation. Il faut l'adapter à ce que l'on cherche. Voici un schéma de notre modélisation inspiré du modèle FRBR :



On peut aussi s'inspirer de Georg Vogeler sur la modélisation de la séquence. Sa première proposition correspond à des clés étrangères sur des relations entre des textes ("hasStarted", "hasEnd", "isDocumentedIn"). Il présente également des triplets rdf, qu'il faudrait ici penser en SQL, ainsi qu'une liste à plat.

Essai d'une première implémentation dans la base Heurist sur les usages : intégrer à plat tous les niveaux, puis les niveaux de texte élémentaire et automatiser la tâche. Comment rendre compte des pièces au choix et des pièces multiples dans la description de l'usage ?

On peut typer la relation avec un vocabulaire spécifique pour préciser où l'on se trouve dans le cursus. On crée donc une relation entre l'usage et l'œuvre que l'on nomme "Cursus".

18/06

Cf. <http://heuristnetwork.org/>.

Service de base de données développé depuis 2005, Heurist permet de créer, gérer, analyser, visualiser, exporter, partager, publier un site web CMS (Content Management System, soit un système de gestion de contenu facilitant la création de site web à partir de modèles).

Points à aborder -> Créer les champs de la BD avant de les importer (cf. Script SQL create table puis Values : **analogie correcte** ?). Pour cela, dans le cas d'un import XML, créer la balise <detail> avec l'@conceptId, l'@name et l'@recordPointer. **Remédier à la limite de taille de fichier lors de l'import en xml ?**

Comparaison de 4 tests pour l'implémentation :

- UseTest1

Record View | Map-Timeline | Map new | Custom Reports | Export | Network Diagram | Crosstabs

Agde ID:402871

Type 89: UseTest1

Organisation: Agde
Office: Parvum
UselItem: Hours of the Virgin Level 2 Matins

Le champ UselItem revient à une liste sur le séquençage de pièces. Plus on clique sur le lien UselItem, plus l’arborescence du livre se déploie, jusqu’à arriver au niveau le plus fin (niveau 4). On peut donc rajouter autant de UselItem que l’on souhaite. S’il est spécifique à la relation créée, les œuvres restent génériques. Toutefois, cela n’est pas optimal du point de vue de l’ergonomie. La solution est en effet pratique pour enregistrer la séquence, mais si l’on supprime un UselItem, tout ce qui est en-dessous est supprimé.

Attention au vocabulaire utilisé dans la base Heurist pour typer les relations : cf. Celui de la séquence et du temporal.

- UseTest2

Record View | Map-Timeline | Map new | Custom Reports | Export | Network Diagram | Crosstabs

Agde ID:402861

Type 90: UseTest2

Organisation: Agde
Office: Parvum

RELATED

asSection	Hours of the Virgin
asSectionHoursOfTheVirgin	Matins
asSectionHoursOfTheVirgin-Matins	Hymn
asSectionHoursOfTheVirgin-Matins-Hymn	Quem terra pontus sidera colunt adorant praedicant trinam regentem machinam claustrum Mariae bajulat ; Cui luna sol et omnia deservunt per tempora perfusa caeli gratia gestant puerilae viscera ; Beata mater munere cuius supernus artifex mundum pugillo continens ventris sub arca clausus est ; Beata caeli nuntio fecunda sancto spiritu desideratur gentibus cuius per alium fusus est ; Jesu tibi sit gloria qui natus es de virgine cum patre et almo spiritu in sempiterna saecula ; Amen - Hymn

Cette solution permet de visualiser via les relations la succession des sections au sein de l’ouvrage, jusqu’à avoir accès au texte pour la plus fine granularité (la dernière relation ci-dessus). Cette idée est visuellement intéressante mais peut-être pas optimale du point de vue des imbrications dans les noms donnés aux relations.

Attention : la linéarité ne modélise pas la séquence qu'il faut donc modéliser par ailleurs. Ceci pose le problème de la modélisation appliquée à un logiciel. Si une information n'est pas modifiable, le problème vient du modèle sous-jacent.

- UseTest3

The screenshot shows a software interface with a navigation bar at the top containing 'Record View', 'Map-Timeline', 'Map new', 'Custom Reports', 'Export', 'Network Diagram', and 'Crosstabs'. The main area displays a record for 'Agde' with the ID '402868'. The record details include 'Type 91: UseTest3', 'Organisation: Agde', and 'Office: Parvum'. Below this, a 'RELATED' section lists two entries: 'asSectionHoursOfTheVirgin-Matins-Hymn' and 'asSectionHoursOfTheVirgin-Matins-Psalms'. The first entry is associated with a large block of Latin text, and the second with the text 'Benedicta tu in mulieribus et benedictus fructus ventris tui - Antiphon'.

Cette solution permet d'avoir directement accès aux niveaux qui ont du texte : **peut-être serait-il plus intéressant de conserver la présentation du UseTest2 en ajoutant seulement la dernière relation ci-dessus ?**

Réalité du texte dans l'usage. La relation vers la section apparaît, mais sans la hiérarchie.

- UseTest4

Record View Map-Timeline Map new Custom Reports Export Network Diagram Crosstabs

Agde ID:402874

Type 92: UseTest4

Organisation	Agde
Office	Parvum
useItem (cursus)	<p>Quem terra pontus sidera colunt adorant praedicant trinam regentem machinam claustrum Mariae bajulat ; Cui luna sol et omnia deserviunt per tempora perfusa caeli gratia gestant puerae viscera ; Beata mater munere cuius supernus artifex mundum pugillo continens ventris sub arca clausus est ; Beata caeli nuntio fecunda sancto spiritu desideratus gentibus cuius per alium fusus est ; Jesu tibi sit gloria qui natus es de virgine cum patre et almo spiritu in sempiterna saecula ; Amen Hymn</p> <p>Benedicta tu in mulieribus et benedictus fructus ventris tui Antiphon</p> <p>Domine, Dominus noster, quam admirabile est nomen tuum in universa terra ! Quoniam elevata est magnificentia tua super caelos. Ex ore infantium et lactentium perfecisti laudem propter inimicos tuos, ut destruas inimicum et ultorem. Quoniam video caelos tuos, opera digitorum tuorum, lunam et stellas quae tu fundasti. Quid est homo, quod memor es ejus ? aut filius hominis, quoniam visitas eum ? Minuisti eum paulominus ab angelis ; gloria et honore coronasti eum ; et constitueristi eum super opera manuum tuarum. Omnia subjecisti sub pedibus ejus, oves et boves universas, insuper et pecora campi, volucres caeli, et pisces maris qui perambulant semitas maris. Domine, Dominus noster, quam admirabile est nomen tuum in universa terra ! Psalm</p>

Solution sûrement la plus intéressante. Visuellement le texte apparaît comme dans la source, et si l'on clique dessus, on voit le texte comme une entité avec ses relations aux autres sections comme on peut le voir dans l'exemple UseTest2.

Nous avons les trois niveaux (l'usage, le UseItem, les relations). Modèle applicativement et intellectuellement supérieur. Pourquoi ? On se rapproche du modèle RDF où chaque relation nest déclarée q'une seule fois, donc la hiérarchie n'est décalrée qu'une seule fois.

Remarque : Mieux vaut typer la relation par "ImmediatelyFollows", car on part de l'entité vers ce qui a précédé, et dans notre lecture du livre de gauche à droite, on est sûr de ce qui précède mais pas de ce qui suit. Toutefois, nouveau problème : si l'ordre des sections prime sur l'ordre des pièces, se pose la question d'où mettre l'ordre de succession et s'il faut conserver les mêmes types de relation...**À creuser.**

24/04

Établissement d'une relation en miroir dans la base Heurist (changement dans le thesaurus des relations). Si Benedicta tu [...] : isContainedWithin : Hours of the Virgin, la relation réciproque est Hours of the Virgin : Contains : Benedicta tu [...].

Les données et le format doivent primer sur le logiciel. Le stockage des données s'établit selon le format le plus propre. Il faut ensuite trouver un logiciel et/ou une transformation pour afficher correctement les données d'une part, les requêter correctement d'autre part.

Dans notre modélisation, il faudrait rajouter le séquençage des sections en plus de l'ordre affiché au sein des œuvres. Le problème est en effet de représenter à la fois la succession au

niveau des pièces et la succession au niveau des sections. Cette dernière idée s'implémente ainsi dans la description de l'usage : Agde(use) :contains :Matins, Agde(use) :contains :Lauds, :contains(1) :precedes :contains(2).

Se pose ensuite la question des usages notés comme “-” pour les pièces non-renseignées, et comme “(Abs)” si la source utilisée ne contient pas l'information mais qu'elle est peut-être présente dans d'autres témoins. Les pièces absentes ou inconnues révèlent en effet la nécessité de choisir un format qu'on peut générer facilement si on met à jour le cursus et qu'on trouve une pièce non-renseignée.

3.2.2 Importer les données dans une base MySQL

Travailler sur des données homogènes !

Faire des liens vers ressources extérieures (cf. Références bibliographiques et catalogues de notices).

Suggestion pour l'automatisation :

- Nettoyage des données dans Cursus_HORAE_Listes.xlsx via Dataiku pour regénérer un CSV propre pour l'import des données
- Implémenter les données dans la base de données Heurist via Python (code pour récupérer des données dans le CSV)

22/04

Travail sur l'homogénéisation des données dans les tableaux sur le cursus (Cursus_HORAE_Listes_import_heurist.xlsx) et les œuvres (Export_stutzmann_horaе_t65_Work.csv). Régénération de CSV propre par Dataiku, mais échec de la jointure : le logiciel valide la condition d'égalité entre les deux colonnes "function" et "liturgical function", mais refuse la condition "startsWith" entre l'incipit dans le cursus et le texte dans l'œuvre...Finalement, jointure partielle des incipit au texte via la fonction "join column", comme ceci :

Problème : seuls les incipits strictement égaux à la colonne text ont été associés, pas de condition “startsWith”. Sur les 4412 lignes remplies par du texte, seules 175 ont un incipit.

Pb de la jointure à travailler. Il doit être possible de faire une condition startsWith. Choix de Dataiku car simplification du nettoyage de la donnée par accès à de nombreuses fonctionnalités (Regex, code Python, etc.). On peut extraire des données d'un CSV via Python, mais travailler sur 2 CSV simultanément paraît difficile, ou alors il faudrait créer un troisième CSV qui rassemble toutes les données pour comparer les colonnes à matcher. Cela nous fait passer par des étapes supplémentaires et laborieuses, alors que Dataiku permet justement de travailler sur des sets de données différents puis de les joindre sous certaines conditions.

23/04

Import du modèle de données (depuis l'exemple d'Agde) choisi en format JSON et XML, étude de leurs structures pour voir comment coder en Python pour générer un import des usages dans la base Heurist plus rapidement.

Problème sur l'export en JSON , car très peu d'informations sont développées :

```

{
  "heurist": {
    "records": [],
    "database": {
      "id": "8",
      "url": "https://heurist.huma-num.fr:443/heurist/",
      "db": "stutzmann_heurist"
    },
    "rectypes": []
  }
}

```

Paramètres à vérifier

Choix par défaut d'analyser le modèle sous le format XML. Plusieurs questions sur la génération des valeurs d'attributs pour savoir comment les coder en Python :

- Attribut depth sur l'élément <records>

Le chiffre indique le niveau depuis où est chargée la donnée, donc il augmente de un dans la relation depuis le document sélectionné.

Agde [usage] (niveau 0, id 402874), Agde [organisation] (niveau 1, id 402689), Agde [place] (niveau 2, id 386187)

Agde [usage] (niveau 0, id 402874), Benedicta tu [useItem] (niveau 1, id 402881), Benedicta tu [work] (niveau 2, id 395935), HORAE [bibl] (niveau 3, id 393553), O quam pulchra (niveau 4, id 399263 = une antienne qui a aussi pour [bibl] = HORAE)

- Attribut ConceptId sur l'élément <type>

Il est composé avant le tiret de l'id source de la base sur Horae, après le tiret de l'identifiant généré par la base. Le conceptId est utile pour transporter les données vers d'autres bases, **les rendre interopérables ?**

- Valeur de l'id de l'élément <workgroup>

À générer automatiquement ou bien généré par la base de donnée ?

Début de génération d'un document XML via la librairie Python LXML. Difficultés à faire des références à des schémas W3C dans les attributs de l'élément <hml> :

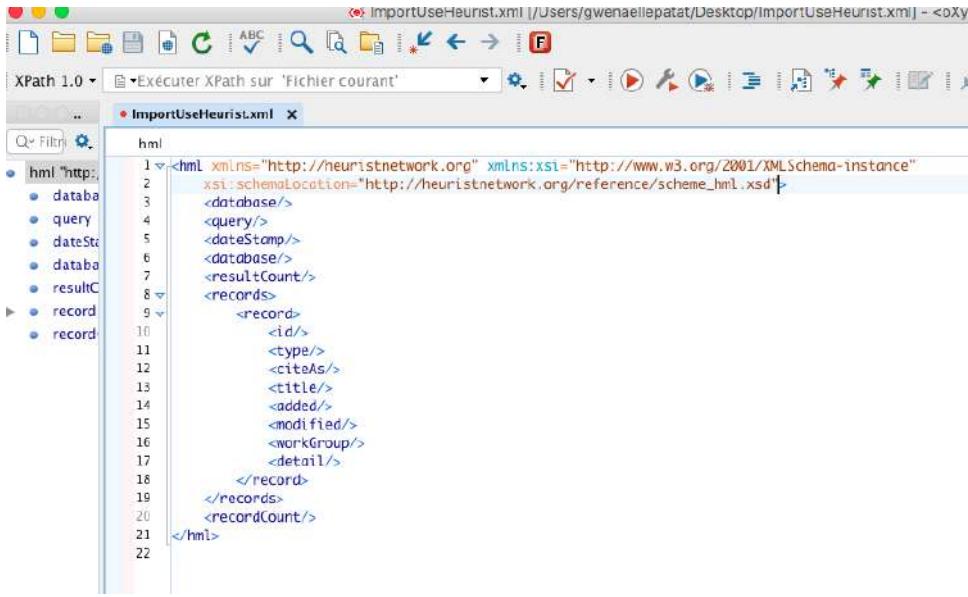
```

OPEN FILES
ImportUseXML.py
from lxml import etree
import xml.etree.ElementTree as ET
#Element racine
hml= etree.Element("hml")
#Attributs fixes de la balise hml
#Espace de nom vers la page d'accueil de la base Heurist
hml.set("xmlns", "http://heuristnetwork.org")
#Espace de nom pour schéma de référence d'XML : pas un attribut
hml.set("xmlns:xsi", "http://www.w3.org/2001/XMLSchema-instance")
#Sous-éléments de la balise hml
database = etree.SubElement(hml, "database")
query = etree.SubElement(hml, "query")
#Attribut xsi:schemaLocation
File "/Users/gwenaellepatat/Desktop/ImportUseXML.py", line 12, in <module>
    hml.set("xsi:schemaLocation", "http://heuristnetwork.org/reference/scheme_hml.xsd")
                                         ^
  File "/usr/lib/python3.6/xml/etree/ElementTree.py", line 815, in _set
    self._setAttribValue(attrib, value)
  File "/src/lxml/apihelpers.pxi", line 585, in lxml.etree._setAttributeValue
  File "/src/lxml/apihelpers.pxi", line 1731, in lxml.etree._attributeValidOrRaise
ValueError: Invalid attribute name 'xsi:schemaLocation'
[Finished in 0.6s with exit code 1]
[shell_cmd: python -u "/Users/gwenaellepatat/Desktop/ImportUseXML.py"]
[dir: /Users/gwenaellepatat/Desktop]
[path: /usr/bin:/bin:/usr/sbin:/sbin]

```

Traceback (most recent call last):
 File "/Users/gwenaellepatat/Desktop/ImportUseXML.py", line 12, in <module>
 hml.set("xsi:schemaLocation", "http://heuristnetwork.org/reference/scheme_hml.xsd")
 File "src/lxml/apihelpers.pxi", line 815, in lxml.etree._set
 File "src/lxml/apihelpers.pxi", line 585, in lxml.etree._setAttribValue
 File "src/lxml/apihelpers.pxi", line 1731, in lxml.etree._attributeValidOrRaise
ValueError: Invalid attribute name 'xsi:schemaLocation'
[Finished in 0.6s with exit code 1]
[shell_cmd: python -u "/Users/gwenaellepatat/Desktop/ImportUseXML.py"]
[dir: /Users/gwenaellepatat/Desktop]
[path: /usr/bin:/bin:/usr/sbin:/sbin]

Ils ont donc été générés à la main pour l'instant :



Code à poursuivre.

Liens vers Documentation XML/Python : <https://python.doctor/page-xml-python-xpath> ;
<https://lxml.de/tutorial.html> ; <https://docs.python.org/3/library/xml.etree.elementtree.html>

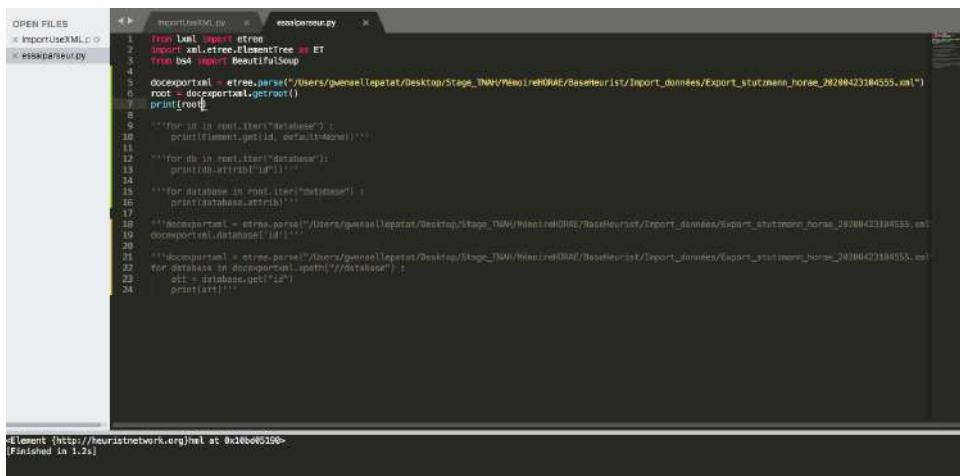
24/04

Les balises <detail> dans <record> pointent vers des entités référencées ailleurs dans la base.
 Il faut donc pouvoir générer à partir du code des relations vers les bonnes entités (Useltem

pour le Cursus par exemple), avec un identifiant généré automatiquement par Python, puis récupérer les données associées à partir du document CSV adéquat.

Le code Python doit en effet prendre compte des données générées par trois sources différentes :

- 1) Celles générées par la base Heurist. Ces données sont récupérables depuis le document d'export au format xml. Les résultats ont à présent été obtenus grâce à la librairie Python `untangle`. Nous avons en effet dans un premier temps voulu parser le document xml via la librairie “BeautifulSoup” et le parseur “etree”, mais, si le code trouvait l’élément racine, nous n’avons pas obtenu de résultats plus loin dans l’arborescence alors même que le code ne générait pas d’erreurs particulières.
Exemples de tentatives :



```
OPEN FILES
x ImportuseXML.py
x ImportuseXML.py
x essaiParseur.py

ImportuseXML.py
1 from xml import etree
2 import xmlelementtree.ElementTree as ET
3 from bs4 import BeautifulSoup
4
5 docexportxml = etree.parse("./Users/guennaellepetot/Desktop/Stagé_THA/MySqlHeurist/BaseHeurist/Import_données/Export_stuttmann_horne_28288423184555.xml")
6 root = docexportxml.getroot()
7 print(root)
8
9 """for id in root.iter("database"):
10     print(Element.getid, default=None))
11 """
12 """for db in root.iter("database"):
13     print(db.attrib["id"])
14 """
15 """for database in root.iter("database"):
16     print(database.attrib)
17 """
18 """docxml = etree.parse("./Users/guennaellepetot/Desktop/Stagé_THA/MySqlHeurist/BaseHeurist/Import_données/Stuttmann_horne_28288423184555.xml")
19 docxmlroot = docxml.getroot()
20 """
21 """docexportxml = etree.parse("./Users/guennaellepetot/Desktop/Stagé_THA/MySqlHeurist/BaseHeurist/Import_données/Export_stuttmann_horne_28288423184555.xml")
22 for database in docexportxml.iter("database"):
23     id = database.get("id")
24     print(id)"""

<Element {http://heuristnetwork.org}rel at 0x100d6510>
[Finished in 1.2s]
```

La librairie `untangle` permet quant à elle de transformer les données présentes dans le document xml en syntaxe Python. Les données (texte, attributs et valeurs d’attributs essentiellement) peuvent ensuite être appelées selon une syntaxe assez simple, comme le montre le code suivant :

```

OPEN FILES
ImportUseXML.py
importusehorist.py
escapeparcours.py

35     # Auto record < record = recordCount -1
36
37     # Ajout des éléments de la balise record
38     id = etree.SubElement(record, "id")
39     type = etree.SubElement(record, "type")
40     citeId = etree.SubElement(record, "citeId")
41     title = etree.SubElement(record, "title")
42     added = etree.SubElement(record, "added")
43     modified = etree.SubElement(record, "modified")
44     workgroup = etree.SubElement(record, "workgroup")
45     detail = etree.SubElement(record, "detail")
46
47     # Ajout des nœuds relatifs à la balise record
48     exportXML.untaggleParse("C:/Users/guillaume/Desktop/Stages_INRA/Modélisation/Heurist/Import_données/Export_stutzmann_horse_28200473184555.xml")
49     database_text = str(etree.tostring(heurist.database, encoding='utf-8'))
50     database.set("id", exportXML.heurist.database["id"])
51     database.set("query", exportXML.heurist.database["query"])
52     database.set("depth", exportXML.heurist.database["depth"])
53     query.set("id", exportXML.heurist.query["id"])
54     query.set("db", exportXML.heurist.query["db"])
55     query.set("depth", exportXML.heurist.query["depth"])
56
57
58
59
60
61     # ajout de l'élément detail devant de l'élément id pour vérifier les informations relatives à l'entité en question : à tester.
62     # after ajout de l'élément id pour vérifier les informations relatives à l'entité en question : à tester.
63     # id/ detail
64     # detail.append(record)
65
66
67     print(etree.tostring(heurist, pretty_print=True))
68     tree = ET.ElementTree(heurist)
69     tree.write("ImportUseHorist.xml")
70
[Finished in 1.8s]

```

- 2) Celles générées automatiquement par Python, soit les identifiants donnés aux entités (ce qui correspond dans le document d'export xml au texte entre les balises <id> et <detail>).
- 3) Celles générées depuis le fichier csv qui contient les données renseignées et nettoyées relatives au projet de recherche.

3.2.2.1 27/04

Avant de coder, il faut déterminer quels sont les identifiants existants et ceux à générer, quels sont les attributs nécessaires à l'import, et quels sont les attributs générés par Heurist au moment de l'import. En voici un récapitulatif (provisoire) :

- Id existants :
 - Ceux des CSV (dans le cas de UseItem_Test_LL.csv, on trouve “H_ID Work 1”, “ID Horae 1”, “H_ID Work 2”, “ID Horae 2”, et ainsi de suite jusqu'à 4, car on a 4 niveaux hiérarchiques qui ont été déterminés au sein du livre d'heures)
 - Ceux de l'export au format xml des données dans Heurist (il s'agit de l'@id de l'entité doublé de l'@ConceptId dans l'élément <type>, qui peuvent par ailleurs se retrouver dans un élément <detail>)
- Id à générer automatiquement
 - Ceux qui se retrouveront dans le document d'import entre les balises <id> et <detail> par ricochet

- Le recId, présent dans l'URL dans la balise <citeAs>, qui se réfère à l'id de l'entité dans laquelle il se trouve
- Attributs nécessaires à l'import (/!\ doutes)
 - Ceux de l'élément <html>
 - Ceux de l'élément <query>
 - @name et @recordPointer dans <detail>
- Attributs générés par Heurist (/!\ doutes)
 - Ceux de l'élément <database>
 - Ceux de l'élément <record>
 - Ceux de l'élément <type>
 - Ceux de l'élément <workgroup>
 - @conceptId, @id, @basename dans <detail>
 - @termId, @termConceptId, @ParentTerm dans les <records> de type Relationship

Avant de coder, il est également important de bien comprendre comment fonctionnent les relations et les clés étrangères dans la base Heurist. La balise <detail> permet de faire des liens entre deux entités, tandis que la relation de séquence et de hiérarchie se fait grâce aux balises <detail> des relations typées. Il faut donc générer les clés étrangères au moment des relations (soit les éléments <detail> avec l'@IsRecordPointer pour une relation many;many ou bien l'élément <reversePointer> qui signale un lien direct pour une relation 1;many.).

Examen de l'id 2325 renvoyant au manuscrit 1185 de la Bibliothèque de l'Arsenal :

Les <reversePointer>, qui n'apparaissent jamais dans les entités de type <relationship> semble fonctionner comme des clés étrangères qui renvoient une entité vers une autre (vers Bibl et Web/Site par exemple) :

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hml>
<hml>
    <records>
        <record id="2" type="2">
            <detail conceptID="1125-118" name="Web page link" id="1138" basename="Web page link">http://www.sanoma.no/beyonduse/details.php?ssby=Alphabetic&amp;numscript=ParA&n1185</detail>
            <reversePointer id="1077" conceptID="0000-1877" basename="External link" name="External link">2316</reversePointer>
        </record>
        <record id="373" type="2">
            <detail conceptID="1125-118" name="Web page link" id="1138" basename="Web page link">http://www.sanoma.no/beyonduse/details.php?ssby=Alphabetic&amp;numscript=ParA&n1185</detail>
            <reversePointer id="1077" conceptID="0000-1877" basename="External link" name="External link">2316</reversePointer>
        </record>
        <record id="373" type="2">
            <detail conceptID="1125-101" name="Certain" id="227" basename="Locational certainty" termID="5342" termConceptID="1125-3340" ParentTerm="Certain">
                <@reversePointer id="1077" conceptID="0000-1877" basename="External link" name="External link">2316</reversePointer>
            </detail>
        </record>
        <record id="373" type="2">
            <detail conceptID="1125-118" name="Web page link" id="1138" basename="Web page link">http://www.sanoma.no/beyonduse/details.php?ssby=Alphabetic&amp;numscript=ParA&n1185</detail>
            <reversePointer id="1077" conceptID="0000-1877" basename="External link" name="External link">2316</reversePointer>
        </record>
        <record id="397" type="2">
            <detail conceptID="1125-118" name="Web page link" id="1138" basename="Web page link">http://www.sanoma.no/beyonduse/details.php?ssby=Alphabetic&amp;numscript=ParA&n1185</detail>
            <reversePointer id="1077" conceptID="0000-1877" basename="External link" name="External link">2316</reversePointer>
        </record>
    </records>
</hml>

```

La balise `<detail>` dans une entité de type relation peut aussi contenir un id présent ailleurs dans un `<reversePointer>` (elle contient alors l'`@reversePointer`), et elle fait donc un lien typé vers l'entité à laquelle elle fait référence dans cette même balise. En voici un exemple pour la relation "fortheUseof" toujours pour le même manuscrit :

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hml>
<hml>
    <records>
        <record id="666" type="1">
            <detail conceptID="2-1" name="Record relationship" type="id-1">
                <@reversePointer id="10215" conceptID="2-1" name="Target record pointer" name="Relationship type" termID="6437" termConceptID="0000-0437" ParentTerm="use">ForTheUseOf</@reversePointer>
                <@record id="10215" conceptID="2-1" name="Source record pointer" name="Relationship type" termID="6437" termConceptID="0000-0437" ParentTerm="use">ForTheUseOf</@record>
            </detail>
            <@record id="10215" conceptID="2-1" name="Source record pointer" name="Relationship type" termID="6437" termConceptID="0000-0437" ParentTerm="use">ForTheUseOf</@record>
        </record>
        <record id="666" type="1">
            <detail conceptID="2-1" name="Record relationship" type="id-1">
                <@reversePointer id="10215" conceptID="2-1" name="Target record pointer" name="Relationship type" termID="6437" termConceptID="0000-0437" ParentTerm="use">ForTheUseOf</@reversePointer>
                <@record id="10215" conceptID="2-1" name="Source record pointer" name="Relationship type" termID="6437" termConceptID="0000-0437" ParentTerm="use">ForTheUseOf</@record>
            </detail>
            <@record id="10215" conceptID="2-1" name="Source record pointer" name="Relationship type" termID="6437" termConceptID="0000-0437" ParentTerm="use">ForTheUseOf</@record>
        </record>
    </records>
</hml>

```

Reste à savoir si l'on peut considérer les id répertoriés dans la balise `<detail>` comme des cés étrangères...?(?)

Avancées dans le code Python :

Pour l'instant les `@xmlns:xsi` et `xsi:schemalocation` de la balise `<hml>` sont à rajouter à la main, les essais pour les implémenter automatiquement ont échoué :

Les éléments qui dépendent du parent <record> sont à interroger comme une liste depuis l'export xml :

```

1 import xml.etree.ElementTree as ET
2 from bs4 import BeautifulSoup
3 import untangle
4
5 exportXML = untangle.parse('/Users/gwenaellepelet/Desktop/Stagé_INRA/MémoireINRAE/BaseHeurist/Imports_données/Export_stuturm_horse_20200424152411.xml')
6 print(exportXML)
7 print(exportXML.records[0])
8 print(exportXML.records[0].records)
9 print(exportXML.records[0].records[0])
10 print(exportXML.records[0].records[0].records)
11 print(exportXML.records[0].records[0].records[0])
12 print(exportXML.records[0].records[0].records[0].records)
13
14 for record in exportXML.records[0].records:
15     print(record)
16     print(record.get('@id', default=None))
17
18 for db in exportXML.records[0].records[0].records:
19     print(db)
20     print(db['@id'])
21
22 for database in exportXML.records[0].records[0].records[0].records:
23     print(database)
24     print(database['@id'])
25
26 for database in exportXML.records[0].records[0].records[0].records[0].records:
27     print(database)
28     print(database['@id'])
29     print(database.get('@id'))
30
31 for db in exportXML.records[0].records[0].records[0].records[0].records[0].records:
32     print(db)
33     print(db['@id'])
34
35
Element <record> with attributes {'visibility': 'visible', 'username': 'logged_in_user', 'selected': 'no', 'depth': '0'}, children: [Element(name = id, attributes = {}, cdata = 40079), Element(name = type, attributes = {'@id': '92', 'conceptID': '0000-92'}, cdata = 1), Element(name = citeAs, attributes = {}), cdata = http://heurist-num-num.fr/ak3/heurist/?record=40079&database=horse], Element(name = title, attributes = {}), cdata = Agde (0) (2), Element(name = added, attributes = {}), cdata = 2028-04-21 16:34:47, Element(name = modified, attributes = {}), cdata = 2028-04-21 13:05:01, Element(name = workgroup, attributes = {'@id': '0'}, cdata = public), Element(name = row, attributes = {}), cdata = 2, Element(name = year, attributes = {}), cdata = 2, Element(name = detail, attributes = {'@conceptID': '0000-1191', 'name': 'Curus', 'id': '1191', 'basename': 'Curus', 'isRecordPointer': 'true'}, cdata = 1087), Element(name = detail, attributes = {'@conceptID': '0000-1191', 'name': 'Curus', 'id': '1191', 'basename': 'Curus', 'isRecordPointer': 'true', 'db': '1087'}, cdata = 1087), Element(name = detail, attributes = {'@conceptID': '0000-1191', 'name': 'Curus', 'id': '1191', 'basename': 'Curus', 'isRecordPointer': 'true', 'db': '402870'}, cdata = 402870), Element(name = detail, attributes = {'@conceptID': '0000-1191', 'name': 'Curus', 'id': '1191', 'basename': 'Curus', 'isRecordPointer': 'true', 'db': '402870'}, cdata = 402870), Element(name = detail, attributes = {'@conceptID': '0000-1191', 'name': 'Curus', 'id': '1191', 'basename': 'Curus', 'isRecordPointer': 'true', 'db': '402881'}, cdata = 402881), Element(name = detail, attributes = {'@conceptID': '0000-1191', 'name': 'Curus', 'id': '1191', 'basename': 'Curus', 'isRecordPointer': 'true'}), cdata = 402881]

```

Qu° de savoir comment paramétriser l'URL dans la balise <citeAs> avec le paramètre recId qui renvoie à l'id du même <record>, et le paramètre db qui renvoie au nom de la base dans <database>.

Code à approfondir.

3.2.2.2 28/04

Fonction pour créer autant de balise <record> qu'il y a d'enregistrements dans le CSV.

On peut récupérer des infos propres à la base depuis le document d'export en xml (comme les @id dans la balise <type>), mais ceux générés au moment de l'import ne sont pas à coder (cf valeurs "None" plus haut justement pour les @ générés au moment de l'import). Exemple avec l'@ConceptId :

```

OPEN FILES
x ImportUsaXML.py
x codeimporteur.py
x esaiCSV.py

68 #fonction pour générer automatiquement un id associé à un enregistrement
69 def record_id(numero_de_depart) :
70     """Calcule l'identifiant associé à l'enregistrement de la donnée (à partir de 600 000 ici)
71     spares numero de départ: numéro de départ associé à l'enregistrement
72     type numero_de_depart: int
73     returns: identifiant du record dans la balise id
74     """
75     records = []
76     with open ('/Users/gwenaellepatat/Desktop/Stage_INAH/MémoireORAE/BaseHeurist/Données/UselItem_Test_LL.csv', 'newline=\n') as csvfile:
77         spamreader = csv.reader(csvfile, delimiter=';', quotechar='\'')
78         for row in spamreader :
79             record_id = numero_de_depart + 1
80             id = str(record_id)
81             return id
82         id.text = record_id(599999)
83
84     #élément d'attribut de la balise type dépendant de la base Heurist : comment les implementer ?
85     type.set("id", exportXML.Hml.records.records[0].type["id"])
86     type.set("concept", exportXML.Hml.records.records[0].type["concept"])
87     type.set("conceptitive", exportXML.Hml.records.records[0].type["conceptive"])
88
89     File "/Users/gwenaellepatat/Desktop/ImportUsaXML.py", line 87, in <module>
90
91     #écriture de l'URL dans la balise citeAs
92     #Balise URL : https://heurist-huma-num.fr443/heurist/?recID=etree,final||/id/text||=Camp/0/etree,final||/ancestor::node/database/text()
93
94
95 Traceback (most recent call last):
96   File "/Users/gwenaellepatat/Desktop/ImportUsaXML.py", line 87, in <module>
97     type.set("concept", exportXML.Hml.records.records[0].type["concept"])
98   File "/Library/Python/2.7/site-packages/xml.etree.ElementTree.py", line 593, in setAttribut
99     self._setAttr(name, value, type)
100   File "/Library/Python/2.7/site-packages/xml.etree.ElementTree.py", line 1525, in _setAttribut
101     raise TypeError('Argument must be bytes or unicod
102     typeError: Argument must be bytes or unicod
103     [file] cmd: python -u "/Users/gwenaellepatat/Desktop/ImportUsaXML.py"
104     [dir]: /Users/gwenaellepatat/Desktop
105     [path]: /usr/bin/bin:/usr/sbin/sbin

```

Fonction pour générer automatiquement l'url de <citeAs> avec le paramètre recId et db résolue.

À retenir : Éviter les boucles dans les boucles, surtout avec ‘for row in spamreader’ car le spamreader est comme une liste sans mémoire qui pourrait corrompre les données au moment de l’import.

Id et type sont des fonctions en python, donc nommer ces éléments autrement.

Attention à la valeur attribuée à “quotechar”, bien regarder le csv -> ici les guillemets qui ne sont pas un séparateur de colonnes.

Toujours définir les fonctions au début du code et pas dans les boucles !

État du code :

```

OPEN FILES
x codeimporteur.py

1 # coding: utf-8
2
3 from bs4 import BeautifulSoup
4 import xml.etree.ElementTree as ET
5 from bs4 import BeautifulSoup
6 import untangle
7 from datetime import datetime
8 import os
9 import urllib
10
11 #écriture de l'URL dans la balise citeAs
12 def parent_URL_citeAsRecId(id):
13
14     #Génère automatiquement l'URL associé à l'enregistrement où la donnée dans laquelle elle se trouve
15     #parent enclôt le correspondant au record dans lequel on se trouve
16     type_recId = str
17     parent = parent.parent
18     parent_url = parent['url']
19     url_recId = parent_url + str(id)
20     return url_recId
21
22
23     parent = urllib.parse.urlencode({'recId': recId, 'db': db})
24     url_recId = "https://heurist-huma-num.fr443/heurist/"
25     url_recId += parent
26     return url_recId
27
28
29 #écriture des données depuis le document d'export XML de la base Heurist
30 exportXML = untangle.parse("/Users/gwenaellepatat/Desktop/Stage_INAH/MémoireORAE/BaseHeurist/Import_données/Export_stutmann_horse_20200426152411.xml")
31
32 #Initialisation
33 hal_etree=Element('hal')
34
35 #ouverture de la balise hal
36 database = Element('database')
37 query = database.SubElement(database, "query")
38 timestamp = etree.SubElement(query, "timestamp")
39 records = database.SubElement(database, "records")
40 records_count = database.SubElement(database, "recordsCount")
41 recordCount = etree.SubElement(query, "recordCount")
42
43 #Ajout d'élément de la balise record
44 #Initialiser le nombre de ligne record en fonction du nombre de données déclaré et insérée dans le CSV
45 with open ('/Users/gwenaellepatat/Desktop/Stage_INAH/MémoireORAE/BaseHeurist/Données/UselItem_Test_LL.csv', 'newline=\n') as csvfile:
46     domenech_heurist = csv.reader(csvfile, delimiter=';', quotechar='\'')
47
48     for id_record, row in enumerate(domenech_heurist, 600000):
49         id = id_record
50         record = etree.SubElement(records, "record")
51         record_id = etree.SubElement(record, "id")
52         id_node = str(id_record)
53         id_node.text = id_node
54         type_node = etree.SubElement(record, "type")
55         attributes_of_attributes de la balise type dépendent de la base Heurist

```

```

OPEN FILES
  codeimportHuri.py
  codemportheurist.py

  62     title = etree.SubElement(record, "title")
  63     title.text = str(row[0])
  64
  65     added = etree.SubElement(record, "added")
  66     added.text = str(datetime.now())
  67
  68     modified = etree.SubElement(record, "modified")
  69     modified.text = str(datetime.now())
  70
  71     workGroup = etree.SubElement(record, "workGroup")
  72     workGroup.text = "public"
  73
  74     detail = etree.SubElement(record, "detail")
  75
  76
  77
  78     # attribut même si il existe de nos envs la page d'accueil de la base Heurist,
  79     # il faut vers le schema XML sur certains où sont stockées les données : à faire à la main
  80     # mais pas dans ce script pour éviter d'avoir à faire à la main
  81     # Heurist set("value", exportXML("value"))
  82     # Heurist.set("value", "http://www.w3.org/2001/XMLSchema-instance")
  83     # Heurist.set("value", "http://www.w3.org/2001/XMLSchema-instance")
  84
  85     # Attributs et noms relatifs à la base de données :
  86     database.text = str(exportXML("val_database.cdata"))
  87     database.set("id", exportXML("val_database.id"))
  88
  89     # Attributs relatifs à l'option query :
  90     query.set("type", exportXML("val_query.type"))
  91     query.set("id", exportXML("val_query.id"))
  92     query.set("name", exportXML("val_query.name"))
  93     query.set("depth", exportXML("val_query.depth"))
  94
  95     # Attribut de l'élément record qui correspond au jour et à l'heure d'import de la base
  96     timestamp.text = str(datetime.now())
  97
  98     # Fait de l'utilisateur réalisant donc le calcul n'est pas nécessaire...
  99
  100    # Attribut de l'élément record généré par Heurist au moment de l'import
  101    # record.set("visibility", exportXML("val_record.visibility"))
  102    # record.set("visnote", exportXML("val_record.visnote"))
  103    # record.set("selected", exportXML("val_record.selected"))
  104    # record.set("use", exportXML("val_record.use"))
  105    # record.set("depth", exportXML("val_record.depth"))
  106    # record.set("parent", exportXML("val_record.parent"))
  107
  108    # Ajout de l'élément detail avant de faire que l'utilisateur renseigne les informations relatives à l'entité en question à modifier
  109    # record.append(detail)
  110    # detail.append(item)
  111    # item.append(attribute)
  112
  113
  114    print(etree.tostring(xml, xml_declaration=True, encoding="UTF-8", pretty_print=True))
  115
  116 tree.write("ImportUseHeurist.xml", encoding='utf8')

```

Reste à clarifier ce qui est à récupérer et comment dans les balises <detail> (clés étrangères et relations).

3.2.2.3 29/04 : définir le document cible = document d'import, pas d'export

On teste ce dont on a besoin pour réussir l'import :

Le document d'export est complexe, mais la structure pleine et complexe comprend des informations qui doivent être générées par Heurist

```

<record visibility="viewable" visnote="logged" in users" selected="no" depth="2">
<id>402881</id>
<type id="93" conceptID="0000-93">useItem</type>
<citeAs>https://heurist.huma-num.fr:443/heurist/?recID=402881&amp;db=stutzmann_horae</citeAs>
<title>Benedicta tu in mulieribus et benedictus fructus ventris tui Antiphon (useItem)</title>
<added>2020-04-21 16:39:56</added>
<modified>2020-04-21 16:42:08</modified>
<workgroup id="0">public</workgroup>
<detail conceptID="0000-1134" name="Work" id="1134" basename="title (work)" isRecordPointer="true">395935</detail>
<reversePointer id="1191" conceptID="0000-1191" basename="Cursus" name="Cursus">402874</reversePointer>
</record>

```

Ce qu'on veut importer, c'est un ou plusieurs [use]

- Un [record] lié, en général, à une entité existante [organisation]
- Une liste de [record] de type [useItem]
- Une liste de relations entre [useItem]

Document test : 3 enregistrements de chaque type

Usage

```

<record>
<id>IDauchoix</id>
<type id='voirquelestl'iddes[use]'>Use</type>
<title> ?

```

- + un élément qui pointe vers l'organisation : voir dans Agde.
 - + est-ce qu'on a besoin de reversePointer : c'est pas sûr ?
- </record>

Uselitem : par exemple un texte

```
<record>
<id>[auchoix]</id>
<type/> : cf. export
<title> : a priori, c'est la chaîne de caractère du XLSX
+ pointeur vers l'usage => clef étrangère
+ pointeur vers autre Uselitem => clef étrangère (contains / follows)
+ pointeur vers Work => lien direct
</record>
```

ForeignKey

```
<record>
<type> Record relationship
Pas de title
Nature de la clef (type de la relation)
Trois « detail » : deux objets pointés [uselitem] et une définition de relation
```



```
<record visibility="viewable" visnote="logged in users" selected="no" depth="0">
<id>402935</id>
<type id="1" conceptID="Z-1">Record relationship</type>
<citeAs>https://heurist.huma-num.fr:443/heurist/?recID=402935&amp;db=stutzmann_hora&lt;&gt;</citeAs>
<title>ImmediatelyFollows / Psalms &lt;-&gt; Hymn</title>
<added>2020-04-24 12:57:56</added>
<modified>2020-04-24 12:57:56</modified>
<workgroup id="6">laura.lebarbey@irht.cnrs.fr</workgroup>
<detail conceptID="2-7" name="Source record" id="7" basename="Source record pointer" isRecordPointer="true">402914</detail>
<detail conceptID="2-6" name="Relationship type" id="6" basename="Relationship type" termID="3266" termConceptID="2-3266" ParentTerm="Sequence">ImmediatelyFol
<detail conceptID="2-5" name="Target record" id="5" basename="Target record pointer" isRecordPointer="true">402911</detail>
</record>
```

Qu° des titres : attention aux masks implémentés dans Heurist, mais ils seront toujours modifiables après l'import. De plus, ils sont produits par la base au moment de l'enregistrement, donc il n'est pas utile de les générer depuis le document d'import des données.

Qu° du <resultCount> : ce à partir de quoi on exporte (4 niveaux dans Heurist).

Mise au point : Dans la base Heurist, quelques données sont implémentées selon le modèle souhaité, mais pas toutes. Le document d'export des données est l'application de notre format cible complet. Il faut donc identifier à l'intérieur les informations structurelles dont on a besoin pour l'import.

Premier template d'import pour le test :

```

<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://heuristnetwork.org" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://heuristnetwork.org/reference/scheme_hml.xsd">
<database>stutzmann_horae</database>
<records>
<!-- 3 essais de type "Use" : Agen, Aix, Amiens-->
<record>
<id>600000</id>
<type id="92">Use</type>
<title>Agen</title>
<detail conceptID="2-21" name="Organisation" id="21" basename="Organisation"
isRecordPointer="true">402892</detail>
</record>
<record>
<id>600001</id>
<type id="92">Use</type>
<title>Aix-en-Provence</title>
<detail conceptID="2-21" name="Organisation" id="21" basename="Organisation"
isRecordPointer="true">402960</detail>
</record>
<record>
<id>600002</id>
<type id="92">Use</type>
<title>Amiens</title>
<detail conceptID="2-21" name="Organisation" id="21" basename="Organisation"
isRecordPointer="true">10004</detail>
</record>
<!-- 6 essais de type uselitem pour pouvoir faire des relations de succession et de hiérarchie -->
<record>
<id>700000</id>
<type id="93">uselitem</type>
<title>Quem terra pontus sidera colunt adorant praedicant trinam regentem machinam claustrum Mariae
bajulat ; Cui luna sol et omnia deserviunt per tempora perfusa caeli gratia gestant puellae viscera ; Beata mater
munere cuius supernus artifex mundum pugillo continens ventris sub arca clausus est ; Beata caeli nuntio fecunda
sancto spiritu desideratus gentibus cuius per alvum fusus est ; Jesu tibi sit gloria qui natus es de virgine cum patre
et almo spiritu in sempiterna saecula ; Amen (Hymn)</title>
<detail conceptID="0000-9999" name="Use" id="9999" basename="Use"
isRecordPointer="true">600000</detail>
<detail conceptID="0000-1134" name="Work" id="1134" basename="title (work)">
<!-- 6 essais de type uselitem pour pouvoir faire des relations de succession et de hiérarchie -->
<record>
<id>700001</id>
<type id="93">uselitem</type>
<title>Domine, Dominus noster, quam admirabile est nomen tuum in universa terra ! quoniam elevata est
magnificentia tua super caelos. Ex ore infantium et lactentium perfecisti laudem propter inimicos tuos, ut destruas
inimicum et ultorem. Quoniam videbo caelos tuos, opera digitorum tuorum, lunam et stellas quae tu fundasti. Quid
est homo, quod memor es ejus ? aut filius hominis, quoniam visitas eum ? Minuisti eum paulominus ab angelis
; gloria et honore coronasti eum ; et constituisti eum super opera manuum tuarum. Omnia subjecisti sub pedibus
ejus, oves et boves universas, insuper et pecora campi, volucres caeli, et pisces maris qui perambulant semitas
maris. Domine, Dominus noster, quam admirabile est nomen tuum in universa terra ! (Psalm)</title>
<detail conceptID="0000-9999" name="Use" id="9999" basename="Use"
isRecordPointer="true">600001</detail>
<detail conceptID="0000-1134" name="Work" id="1134" basename="title (work)">
<!-- 6 essais de type uselitem pour pouvoir faire des relations de succession et de hiérarchie -->
<record>

```

```

<record>
  <id>700002</id>
  <type id="93">useItem</type>
  <title>O admirabile commercium creator generis humani animatum corpus sumens de virgine nasci dignatus est et procedens homo sine semine largitus est nobis suam deitatem (Antiphon)</title>
  <detail conceptID="0000-9999" name="Use" id="9999" basename="Use"
  isRecordPointer="true">600002</detail>
  <detail conceptID="0000-1134" name="Work" id="1134" basename="title (work)"
  isRecordPointer="true">399221</detail>
</record>
<record>
  <id>700003</id>
  <type id="93">useItem</type>
  <title>Matins</title>
  <detail conceptID="0000-9999" name="Use" id="9999" basename="Use"
  isRecordPointer="true">600000</detail>
  <detail conceptID="0000-1134" name="Work" id="1134" basename="title (work"
  isRecordPointer="true">394820</detail>
</record>
<record>
  <id>700004</id>
  <type id="93">useItem</type>
  <title>Benedicta tu in mulieribus et benedictus fructus ventris tui</title>
  <detail conceptID="0000-9999" name="Use" id="9999" basename="Use"
  isRecordPointer="true">600001</detail>
  <detail conceptID="0000-1134" name="Work" id="1134" basename="title (work"
  isRecordPointer="true">395935</detail>
</record>
<record>
  <id>700005</id>
  <type id="93">useItem</type>
  <title>Hours of the Virgin</title>
  <detail conceptID="0000-9999" name="Use" id="9999" basename="Use"
  isRecordPointer="true">600002</detail>
  <detail conceptID="0000-1134" name="Work" id="1134" basename="title (work"
  isRecordPointer="true">394805</detail>
</record>

<record>
  <!-- Matins : Contains : Quem terra pontus... -->
  <id>800001</id>
  <type id="1">Record</type>
  <detail conceptID="2-7" name="Source record" id="7" basename="Source record pointer"
  isRecordPointer="true">700003</detail>
  <detail conceptID="2-6" name="Relationship type" id="6" basename="Relationship type" termID="3266"
  termConceptID="2-3266" ParentTerm="Sequence">Contains</detail>
  <detail conceptID="2-5" name="Target record" id="5" basename="Target record pointer"
  isRecordPointer="true">700000</detail>
</record>
<record>
  <!-- Domine, Dominus noster, ... : ImmediatelyFollows : Benedicta tu... -->
  <id>800002</id>
  <type id="1">Record</type>
  <detail conceptID="2-7" name="Source record" id="7" basename="Source record pointer"
  isRecordPointer="true">700001</detail>
  <detail conceptID="2-6" name="Relationship type" id="6" basename="Relationship type" termID="3266"

```

```

termConceptID="2-3266"                                ParentTerm="Sequence">ImmediatelyFollows</detail>
    <detail conceptID="2-5" name="Target record" id="5" basename="Target record pointer"
isRecordPointer="true">700004</detail>
</record>
<record>
    <!-- Hours of the Virgin : Contains : O admirabile commercium... -->
    <id>800003</id>
    <type id="1">Record relationship</type>
    <detail conceptID="2-7" name="Source record" id="7" basename="Source record pointer"
isRecordPointer="true">700005</detail>
        <detail conceptID="2-6" name="Relationship type" id="6" basename="Relationship type" termID="3266"
termConceptID="2-3266"                                ParentTerm="Sequence">Contains</detail>
            <detail conceptID="2-5" name="Target record" id="5" basename="Target record pointer"
isRecordPointer="true">700002</detail>
        </record>
    </records>
</hml>

```

Les usages ont été choisis depuis le document UseItem Test LL.xlsx aux lignes suivantes :

- n°0167 pour Agen
- n°0334 pour Aix-en-Provence
- n°0553 pour Amiens

Les recordPointer vers des entités pre-existantes dans Heurist (déjà importées), notamment Work et Organisation, font donc référence aux identifiants déjà implémentés dans la base.

Doutes sur comment encoder la relation “Contains” ou “Follows” d’un “useItem” vers un autre, car le document d’export (Export_stutzmann_horae_20200424152411.xml) montre essentiellement les relations à l’œuvre avec un reversePointer vers le Cursus :

```

<type id="93" conceptID="0000-93">useItem</type>
<citeAs>https://heurist.huma-num.fr:443/heurist/?recID=402909&db=stutzmann_horae</citeAs>
<title>Hours of the Virgin</title>
<added>2020-04-24 12:38:53</added>
<modified>2020-04-24 12:39:29</modified>
<workgroup id="0">public</workgroup>
<detail conceptID="0000-1134" name="Work" id="1134" basename="title (work)"
isRecordPointer="true">75966</detail>
<reversePointer id="1191" conceptID="0000-1191" basename="Cursus"
name="Cursus">402874</reversePointer>
</record>

```

Peut-être faire un renvoi vers les relations propres au useItem en question...à creuser.

Attention : chaque “useItem”, que ce soit dans une relation de hiérarchie ou de succession, est spécifique à un usage, donc bien tous les différencier selon l’usage auquel ils sont liés.

Test Import n°1 :

Message comme quoi la base de données ne contient pas tous les éléments structurels pour accueillir les données. Le message nous dit aussi que tout item importé crée un nouvel enregistrement, la base n'identifie pas ls doublons. On peut alors télécharger le tableau suivant :

elements_in_import

type	source id	ccode	name in source	target id	name in target
rectype			Record relationship		
rectype			Use		
rectype			uselitem		

L'import ne semble pas détecter dans notre document l'id de la source mais seulement son nom. Il manque aussi l'id de la cible et son nom. En revanche, à quoi correspond le "ccode" ??

Nouvel essai avec rajout de l'@ConceptId dans les balises <detail> et création d'un @id et d'un @conceptId pour la balise <detail> faisant référence à un usage.

Observations des résultats du test 1 :

The screenshot shows the HEURIST application interface. The top navigation bar includes Database, Verify, Import, Website, Management, Admin, FAIMS, Help, and a 'hosted by Huma-Num' logo. The left sidebar has sections for Design schema, Saved Filters (with 'My Filters (private)' and 'Recent changes'), and Database Managers (with 'HORAE (original)' and 'HORAE (working)'). The main area is titled 'Filtered Result' with a count of 4 items. One item is selected, highlighted in blue, with the message 'No data in the fields for this record'. The selected item's details include: Original ID: 0-00000, Created: 2020-04-29 15:13:53 (IT/13:53 local), Updated: 2020-04-29 15:13:53 (IT/13:53 local), Ownership: Database Managers - readable by anyone (public), and Rating: none.

La présentation n'est pas la bonne et les recordPointer n'apparaissent pas, mais on constate que l'on peut choisir les id associés à la donnée avec le champ "Original id".

This screenshot is similar to the previous one but shows a different filtered result set with 13 items. One item is selected, showing: Original ID: 0-70005, Created: 2020-04-28 15:13:53 (IT/13:53 local), Updated: 2020-04-28 15:13:53 (IT/13:53 local), Ownership: Database Managers - readable by anyone (public), and Rating: none.

This screenshot is identical to the previous ones, showing a filtered result set with 13 items. One item is selected, showing: Original ID: 0-70004, Created: 2020-04-29 15:13:53 (IT/13:53 local), Updated: 2020-04-29 15:13:53 (IT/13:53 local), Ownership: Database Managers - readable by anyone (public), and Rating: none.

Pour les uselitem, le champ Work n'apparaît pas, ni les relations “Contains”, mais Heurist crée de lui-même les <reversePointer> comme on peut le voir avec les liens vers le texte qui “immediatelyPrecedes”.

The screenshot shows the HEURIST software interface. The top navigation bar includes links for Database, Verify, Import, Website, Management, Admin, FAIMS, Help, and a 'hosted by Huma-Num' logo. The left sidebar contains sections for 'My Filters' (with a 'Recent changes' filter selected), 'Database Managers', 'HORAE', 'CORPUS', 'Work Level 1', and 'Faceted search'. The main area displays a 'Recent changes' list with 196 items. One item is highlighted, showing a 'Type 1: Record relationship' between Record ID 403032 and Record ID 403029. The details pane shows the source record (Record ID 403032), relationship type (Sequence 'ImmediatelyFollows'), target record (Record ID 403029), and original ID (0-600009). Below this, there's a 'LESS...' section with XML and HTML links, and a note about the record being readable by anyone (public).

Pour les record relationships, la présentation est plus semblable à ce qui est attendu dans le modèle, avec les champs “Source record”, “Relationship type” et “Target record”. Toutefois, le principal problème reste l’apparition du titre, du nom attribué aux données. Il faut donc revoir la construction des titres et de certains liens (Work, Use, Uselitem) dans le document d’import.

Tests à poursuivre.

30/04

Ce qui ressort du premier test d’import en xml, c’est un problème d’encodage sur le titre donné à l’instanciation de l’entité, ainsi que sur les liens directs entre les entités, soit les clés étrangères. Toutefois l’id et le type implémentés dans le template apparaissent bien.

/!\ Dès qu’on fait une transformation, vérifier que les données qu’on a prises en entrée se retrouvent en sortie.

Deuxième test avec :

- Retrait des @id et @conceptId dans la balise <detail>
- Application des masques implémentés dans Heurist pour voir si le problème vient de là
- Renvoie des uselitem aux relations qui les concernent dans la balise <detail>

Toujours même message d'erreur au moment de l'import.

Observation des résultats du Test 2 :

Record View Map-Timeline Map new Custom Reports Export
Network Diagram Crosstab

Record ID 403046 - no data has been entered in the fields used to construct the title ID:403046

Type: Received relationship

Original ID: 0-400002
accMat: 700001
3266
700004

LESS...

Created: XML HTML Right click to copy URL
Added: 2020-04-30 09:02:24 [11:02:24 local]
Updated: 2020-04-30 09:02:24 [11:02:24 local]
Ownership: Database Managers - readable by anyone (public)
Rating: none

Par rapport au Test 1, le type de la relation n'apparaît plus. Les liens sont indiqués comme "accMat" et ne sont plus typés comme on le souhaiterait. Cela signifie que la balise <title> n'a aucune incidence sur le nommage dans Heurist, et que l'@id dans la balise <detail> permet à la base de reconnaître les liens avec les autres entités.

Record View Map-Timeline Map new Custom Reports Export
Network Diagram Crosstab

Record ID 403044 - no data has been entered in the fields used to construct the title ID:403044

Type: uselink

Original ID: 0-700006
accMat: 600002
394908
800003

LESS...

Created: XML HTML Right click to copy URL
Added: 2020-04-30 09:02:23 [11:02:23 local]
Updated: 2020-04-30 09:02:24 [11:02:24 local]
Ownership: Database Managers - readable by anyone (public)
Rating: none

The screenshot shows the HEURIST software interface. At the top, there are navigation tabs: Database, Verify, Import, Website, Management, Admin, FAIMS, and Help. The main area is titled "Filtered Result" with a count of "n = 206". The results list shows multiple entries, each with a warning message: "**** No data in title fields for this record ****". On the right side, a detailed view for a specific record (ID 403038) is displayed, showing its original ID (0-000062), accMat (10024), and a "LESS..." section with XML and HTML links.

On peut noter la différence du message d'erreur quant au titre : pour les UseItem et les Record Relation, on lit un record suivi d'un numéro avec le message "*no data has been entered in the fields used to construct the title*", tandis que pour les Use on voit "***** No data in title fields for this record *****". Cela pourrait suggérer qu'il y a différentes manières d'encoder les titres dans l'import xml. Toutefois, à la différence du premier test, les liens vers les autres entités apparaissent mais avec le terme 'accMat'.

Troisième test avec rajout des @id dans balise <detail> (l'@conceptId est inutile car généré par la base).

Observation des résultats du Test 3 :

The screenshot shows the HEURIST software interface. At the top, there are navigation tabs: Database, Verify, Import, Website, Management, Admin, FAIMS, and Help. The main area is titled "Filtered Result" with a count of "n = 239". The results list shows multiple entries, each with a warning message: "**** No data in title fields for this record ****". On the right side, a detailed view for a specific record (ID 403068) is displayed, showing its original ID (0-000068), accMat (10024), and a "LESS..." section with XML and HTML links.

The screenshot shows the Heurist application interface. On the left, there is a sidebar with various filters and search options. The main area displays a "Filtered Result" list with 230 items. The first item in the list is highlighted. The details for this item are shown on the right, including its ID (403063), type (use), original ID (D-700000), and a note indicating no data has been entered in the title fields. Below this, there are sections for RELATED and LESS, each containing a list of related records.

This screenshot is identical to the one above, showing the same Heurist interface, filtered results, and record details for Record ID 403063.

Le problème sur les titres persistant : Faut-il rajouter un attribut à la balise pour qu'Heurist les intègre ? Faut-il les indiquer dans une autre balise ?

Par ailleurs, les relations sont typées correctement mais ce sont les liens vers les autres entités qui n'apparaissent pas : faut-il encoder les clés étrangères de manière spécifique ?

Potentiellement à voir avec Ian Johnson.

On peut parallèlement réfléchir à un import ne csv qui serait généré avec Python. Définition de la cible en cours, sachant qu'on s'appuie sur le même type de données que pour l'essai d'import en xml. En-tête suivante (**pour l'instant**) :

```
Id;Use;Office;Level 1;title level 1;Level 2;title level 2;Level 3;title level  
3;function;title;text;H_Id Work 1;Work Title 1;H_Id Work 2;Work Title 2;H_Id Work 3;Work  
Title 3;H_Id Work 4;Work Title 4;H_Id Organisation;Organisation;H_Id useItem
```

source;useItem source;H-Id Record Relationship;Record relationship;H_Id useItem
target;useItem target

04/05

Aide de Ian Johnson : il faut suivre le template hml suivant (Template_stutzmann_horae_20200429152715.html), soit garder les @conceptId, faire des renvoies avec comme forme d'identifiants "H-ID-nnnn" et @name dans <detail>. Le titre est à définir dans une balise <detail> : <detail conceptID="2-1" name="Title">TEXT</detail>.

Toutefois, le template est restrictif. Par exemple, pour les <useItem> :

```
<detail conceptID="0000-1012" name="Note">MEMO_TEXT</detail>  
  
<detail conceptID="0000-1134" name="Work" isRecordPointer="true">RECORD_REF</detail>  
  
<detail conceptID="0000-1188" name="Work (Cursus)">TEXT</detail>  
  
<detail conceptID="0000-1192" name="Sequence">TEXT</detail>
```

L'association vers l'entité Use, l'usage, n'existe pas. Il n'est pas possible de créer ce qui n'est pas déjà implémenté dans la base. Cela nous amène au problème des identifiants : ceux générés par le XML dans la balise <id> (qui se retrouvent dans le champs "Orginal-ID) sont automatiquement supplantés par ceux générés dans la base. Cela est problématique car il faut faire des exports pour ensuite pouvoir récupérer les identifiants puis faire les associations correctes entre les entités. Import fastidieux, même pour le format csv.

Se pose de plus la question de la taille du fichier d'import. La masse de données implique de réfléchir à la possibilité d'importer des fichiers lourds.

05/05

Évocation de la possibilité de fusionner les données au moment de l'import en fonction des identifiants générés par nous (et non pas par la base). Est-il plus sûr de faire la jointure entre les données après avoir généré des entités vides dans la base pour récupérer leurs H-ID ? **À suivre.**

06/05

Point général sur l'import du Cursus :

Import en deux temps :

- Import de toutes les entités

- Récupération des H-ID pour créer les relations (générée avec xslt ici)

En ce qui concerne la méréologie, chaque uselitem a deux relations :

- La section dans laquelle elle se trouve
- L'ordonnancement (ce qui précède ou ce qui suit)

Il est en effet nécessaire de simplifier le modèle pour l'import, tout en lui faisant refléter la complexité de la réalité décrite. Il faut ainsi d'abord vérifier que l'import fonctionne avant de créer la structure. Erreur de modélisation ici, car un cursus n'est pas relié à une organisation mais à une à plusieurs organisations, donc il peut y avoir plusieurs organisations pour un même usage. Il ne faut donc pas laisser "Office : Parvum" pour toutes les entités, mais il vaut mieux diviser un usage en "type" et "subtype", pour déterminer par exemple ce qui est de l'ordre du "Parvum" ou du "Magnum" dans l'usage concerné.

/!\ Plusieurs usages sont liés à une organisation mais le type d'office ne concerne pas tout le cursus.

Si l'incipit du uselitem n'est pas renseigné, le signaler dans la base par "NotIdentified".

In fine, trois thématiques principales abordées lors de cet exercice :

- Récupération des données héritées
- Sécurisation des données
- Contraintes à définir pour avoir des données propres

À noter qu'Excel reste un logiciel puissant pour gérer les données.

La contrainte sur la structuration des données doit permettre l'interopérabilité. Dans ce sens, il est important de repréter ce qui est générique et ce qui est spécifique. Il faut donc mettre la priorité sur le générique dans la mesure du possible, et trouver les solutions génériques pour tenir compte des nouveautés, soit identifier ce qui est nouveau et le placer sous la bonne étiquette générique.

L'essentiel est de comprendre les données et leur mise en place ; identifier les problèmes de modélisation et demander les solutions aux bonnes personnes. Dès que quelque chose semble étrange, le noter et le relever.

18/05

Import en CSV usage par usage (un CSV par usage), car de tout importer dans un même csv insère les uselitem dans le désordre. Contrairement à d'autres projets, l'ordre des données

n'est ici pas indifférent. Il faut en effet un ordre dans les recordPointer, ce qui est une demande rare, car le modèle est structurer en hiérarchie comme dans un arbre XML.

Attention aux titres des usages pour les retrouver une fois créés.

Il s'agit de faire les liens entre les usages et les useItem. On crée donc dans un premier temps les useItem avec les bons matchs sur les Works et les incipits, puis on importe dans un deuxième temps le cursus.

19/05

Continuation de l'import usage par usage.

Possible d'importer plusieurs cvs en même temps avec Heurist ? Cela créerait peut-être des problèmes sur les associations...

Certains usages sont associés à des Organisations avec un nom légèrement différent car ce sont des traductions.

Toutefois, le fait de créer des relations entre les useItem a désordonné la succession des textes au sein des Use. Le principal est de vérifier les données, les questions de cosmétiques viennent ensuite.

25/05

Correction des incipit directement dans Heurist, question de centralisation et de stabilisation des données.

01/07

Réunion avec Ian Johnson :

Bien de définir un ordre du jour avant de faire une réunion.

Question des objets en ordre décalé -> L'ordre des pièces doit refléter l'ordre de présence dans les manuscrits. En effet, chaque œuvre a un attribut de séquence pour ce qui précède et ce qui suit avec des renvois vers les œuvres concernées. L'ordre n'est donc ici pas déterminé selon l'id ou la date mais selon la position de la pièce.

La solution ponctuelle est de donner un attribut de séquence numérique au titre.

Pour la géolocalisation, il est souhaitable de pouvoir choisir entre entre le lieu de conservation des manuscrits et les usages via les records pointers et les relationships makers.

Pour cela, faire un ruleset, soit une série de règles pour ramener les lieux en question.

Version 6 en cours avec ajout de la possibilité de visualiser les données en tables.

On peut créer un site entièrement avec les custom reports.

Problème : si l'on découvre une pièce nouvelle, cela provoque un décalage dans la séquence, ce qui a des conséquences aussi sur les autres pièces. Chaque ajout d'une pièce implique la modification de deux autres. Le problème est le même pour les pièces à supprimer : besoin d'adapter le logiciel.

Question de la désambiguation à affiner, soit "expected but not attested", soit "identified".

Problème d'affichage pour la carte des diocèses : les shapefiles, format de fichier pour les systèmes d'informations géographiques, sont traités en latitude et longitude mais pas en projection Lambert. Pour lors, exporter les coordonnées et les noms en CSV et les importer comme des enregistrements individuels.

Pour vérifier l'intégrité des données, on peut faire un export de la base en xml et appliquer la fonction count() pour voir les id des œuvres qui pointent vers plus de 2 éléments -> erreur.

06/07

Import des données en XML:

Les données étant souvent imparfaites, il est permis de ne pas remplir toutes les balises <detail> du template au moment de l'import des données. Il est toutefois possible de vérifier leur intégrité sur Heurist.

Il n'est toutefois pas heureux de rajouter une balise <detail> qui n'apparaît pas dans le template, il est préférable de modéliser en amont en définissant bien les champs qui structurent les données, et les intégrer dans le template.

De plus, il faut que les records-id soient générés préalablement dans la base, donc faire l'import en 2 passes :

- Une première pour générer les id à partir des données brutes

- Une deuxième pour faire les liens entre les données avec la récupération des bons identifiants dans la base

4 Annotation et validation

Différences entre

- Savoir formalisé et apprentissage machine
- Implicite du savoir en humanités / explicite des humanités numériques
- Données / analyse des données / formalisation

04/05

Premiers pas avec Arkindex. Pistes d'améliorations/Remarques :

- Obliger de créer un nouvel élément pour changer de section, on ne peut modifier la section (soit le niveau) directement lors de la création de l'élément.
À NE PAS FAIRE, NE PAS TOUCHER AU NAME.
- On ne peut ajouter un niveau sur la même partie de texte sélectionnée, obliger de créer un rectangle à chaque section.
- Il manque un zoom avec une bonne résolution par page numérisée, manque de lisibilité.
- Le choix d'annoter uniquement la première ligne d'une section est-il pertinent quand la lettrine prend énormément de place, comme dans l'exemple ci-dessous ?



Par ailleurs, découverte d'un incipit d'hymne dans les Matines non-répertorié dans le manuel d'annotations :

The screenshot shows the Arkindex software interface. At the top, there are navigation links: Browse, Search, Import, and Corpora. Below that, it says 'HORAE | Interannotator agreement 2 / Volume AIREVILLE, Bibliothèque municipale, 015 (Reproduction intégrale) / Page 1'. On the left, there are buttons for 'RECTANGLE', 'POLYGON', 'SELECT', and 'RESET'. The main area is titled 'Polygon results' and shows a thumbnail of an open medieval manuscript. To the right of the thumbnail, there are four sections of data, each with a 'VIEW ELEMENT', 'UPDATE', and 'DELETE' button.

	Name	Type	Class
Section 0	Section	Hours of the Cross	
Section 1	Section	Matins	
Section 2	Section	Invocation	
Section 3	Section	Hymn	

At the bottom right, it says 'Version 0.12.1'.

Il s'agit de "Patris sapientia, veritas divina", une hymne attribuée à Egidio da Colonna (1245-1316). **À rajouter au manuel ? Oui mettre à jour. /!\ pas d'actualité, bien répertorié.**

05/05

Pb d'ergonomie sur lesquels Teklia travaille, notamment le déplacement du rectangle de sélection sur la page numérisée. Autres suggestions :

- Si l'on crée un élément après coup qui ne suit pas l'ordre du manuscrit, il serait intéressant de pouvoir le déplacer pour que les éléments suivent l'ordre du manuscrit, si cela est utile.
- Pour naviguer d'une page à l'autre plus rapidement, il pourrait être intéressant d'avoir un onglet "rechercher" pour accéder directement au numéro de page souhaité (à moins qu'il existe déjà mais que je ne l'ai pas vu).

Rappel de certains choix d'annotations :

- Pas de niveau 3 pour les heures abrégées de la Croix et de l'Esprit saint. Il faut donc annoter le véhicule d'invocation dans la bonne section de niveau 2, sachant que ces heures ne contiennent pas de laudes.
- Ne pas changer le nom de l'élément (laisser Section 0 etc.) car l'inscription du niveau des pièces se fera plus tard.
- Faire la distinction entre Heures de la Vierge / Heures de la Vierge (Avent) / Heures de la Vierge (Temporal). Elles dépendent de la partie de l'année liturgique. On peut

s'appuyer sur le texte et les rubriques pour les distinguer, mais les plus courantes sont les Heures de la Vierge.

- Les annotations de niveaux 3 ne concernent que les Heures de la Viege et l'Office des morts.
- Ne prendre que la première ligne de chaque pièce, même si elle est courte.

Recensement d'un nouvelle hymne : dans l'heure de Prime des Heures de la Croix, "Hora prime ductus est Iesus ad Pilatum".

06/05

Lien vers Tableau de bord Teklia :

https://redmine.teklia.com/login?back_url=https%3A%2F%2Fredmine.teklia.com%2Fproject%2Fhorae%2Fwiki%2FTableau_de_bord_annotations_IRHT

Résumé réunion avec équipe de Teklia :

Il a été question dans un premier temps de l'infrastructure. Le corpus inter-annotation a été préparé avec l'ensemble des **classes** à gérer (soit tous les contenus d'annotation, qu'il s'agisse des textes ou des niveaux hiérarchiques). Il y en a environ 4500. L'objectif est de pouvoir faire des recherches avancées dans les classes et la base de données, ce qui implique un gros travail de développement. À la différence de la plateforme Mirador, on doit pouvoir faire une recherche dans une liste.

La classe est un terme générique propre aux tâches du data mining, avec d'une part la classification et d'autre part le clustering. La classification consiste à déterminer le nombre de catégories à traiter a priori où chaque élément est associé à une classe. Quant au clustering, il fait référence à un tas d'objet dont on doit optimiser le regroupement.

Pour régler le problème de la sélection de la zone d'annotation (qui dépend de la résolution de l'écran), il faut faire des tests avec des résolutions différentes pour stocker les données en fonction. C'est bien la résolution de l'écran qui est en jeu dans l'enregistrement des coordonnées, et pas le zoom du navigateur. Le résultat final sera que l'alignement de la ligne du texte à la sélection lui correspond, la résolution de l'écran ne sera plus en jeu.

Il est rappelé que meilleure sera l'ergonomie, meilleure sera la qualité des données. Teklia souligne toutefois que développer une interface inter-annotateur en un mois et demi est trop rapide. Il faudrait donc pouvoir faire une interface test dans un premier temps pour vérifier la qualité du travail.

Il est ensuite question de segmentation et de reconnaissance du texte. Pour les pièces courtes et difficiles à identifier, qui font partie d'une production non-systématique, il faut construire la reconnaissance historique, tout en distinguant le variant "normal" du variant "pas normal".

Question des textes élémentaires. 4500 sont pour l'instant répertoriés, mais on en découvre sans cesse donc leur nombre n'est pas fixe.

Présentation du Machine learning en cours grâce au modèle UFCN pour la détection de ligne. Fonctionnement en deep learning (réseaux de neurones) avec insertion en image d'entrée des caractéristiques d'une ligne pour que la machine génère une image de prédiction au niveau pixel. Le modèle est entraîné avec le corpus d'Horae.

Problème sur la reconnaissance d'écriture : le système ne détecte pas le haut de la ligne mais plus bas, à corriger. La reconnaissance du texte se fait en effet par adaptation aux tailles de ligne. La reconnaissance prend donc en entrée des fenêtres rectangulaires. Il faut faire des tests selon le type de lignes, certaines peuvent être moins droites sur l'image, notamment à cause de la numérisation.

En ce qui concerne l'interface d'Arkindex, nous aurons une vision hiérarchique des annotations avec une arborescence dans laquelle il sera possible de naviguer.

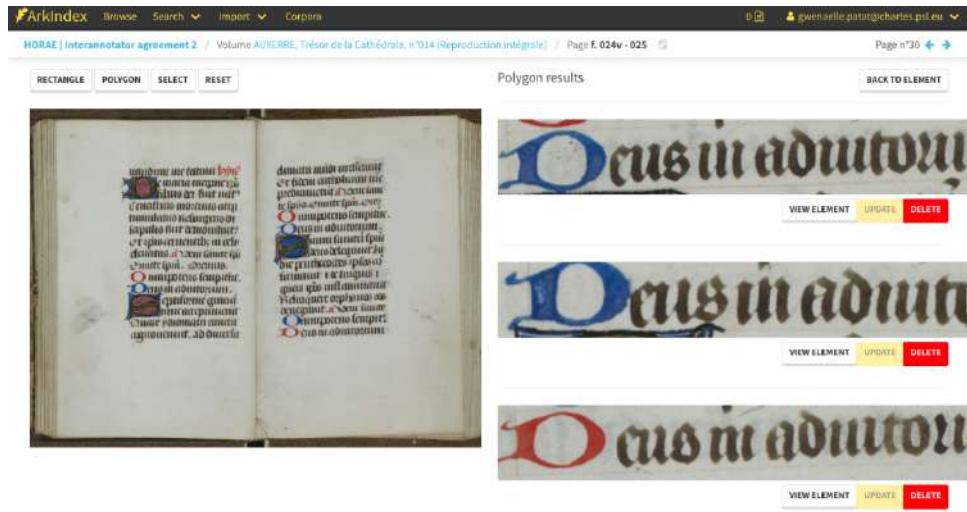
Le système de correction des coordonnées va être testé sur un échantillon de 50 manuscrits annotés. L'interface devrait être optimisée pour fin mai.

11/05

Pour suffrages et prières, créer des classes de type vide avec "Not Identified", mais nous n'avons pas encore accès aux classes de niveau 4 dans l'inter-annotateur.

Variantes repérées sur le manuscrit AUXERRE, Trésor de la Cathédrale, n°014 :

- Sexte précède Tierce dans les Heures de l'Esprit Saint aux f.24v-25



- Antienne des psaumes dans la partie Terce des Heures de la Vierge prenant la forme de "Tota pulchra" aux f.56v-57

Arkindex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume ALUERRE, Trésor de la Cathédrale, n°014 (Reproduction intégrale) / Page f.056v - 057

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT DELETE RESET

Tota pulat.

Name	Section 0
Type	Section
Class	Psalms

VIEW ELEMENT UPDATE DELETE

- Antienne des psaumes dans la partie Sexte des Heures de la Vierge prenant la forme de "In adore" aux f.60v-61

Arkindex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume ALUERRE, Trésor de la Cathédrale, n°014 (Reproduction intégrale) / Page f.060v - 061

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT DELETE RESET

In adore

Name	Section 0
Type	Section
Class	Psalms

CREATE

- Antienne des psaumes dans la partie None des Heures de la Vierge prenant la forme de "Ortus condusus" aux f.64v-65

Arkindex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume ALUERRE, Trésor de la Cathédrale, n°014 (Reproduction intégrale) / Page f.064v - 065

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT RESET

Ortus condusus

Name	Section 0
Type	Section
Class	Psalms

VIEW ELEMENT UPDATE DELETE

- Antiphonaire dédié à la Vierge en conclusion de l'Office des morts aux f.143v-144

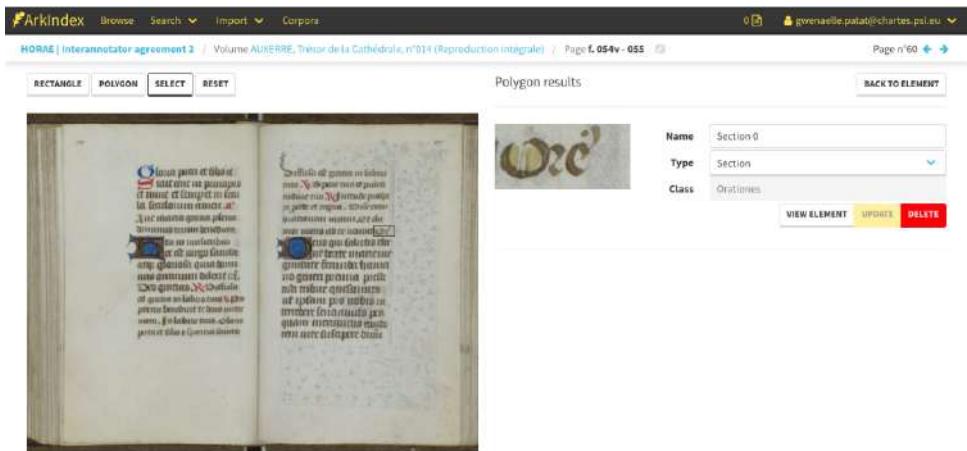
The screenshot shows a digital interface for manuscript transcription. At the top, there's a navigation bar with 'ArkIndex', 'Browse', 'Search', 'Import', 'Corpus', and a user email 'gwenaelle.piat@echartes.pdf.eu'. Below the navigation is a breadcrumb trail: 'HORAE | Interannotator agreement 2 / Volume AUXERRE, Trésor de la Cathédrale, n°014 (Reproduction intégrale) / Page f. 143v - 144'. On the left, there are buttons for 'RECTANGLE', 'POLYGON', 'SELECT', and 'RESET'. In the center, it says 'Polygon results' and 'BACK TO ELEMENT'. The main area displays a photograph of an open medieval manuscript. A red rectangle highlights a specific section of text on the right-hand page.

Doutes sur le manuscrit AUXERRE, Trésor de la Cathédrale, n°014 :

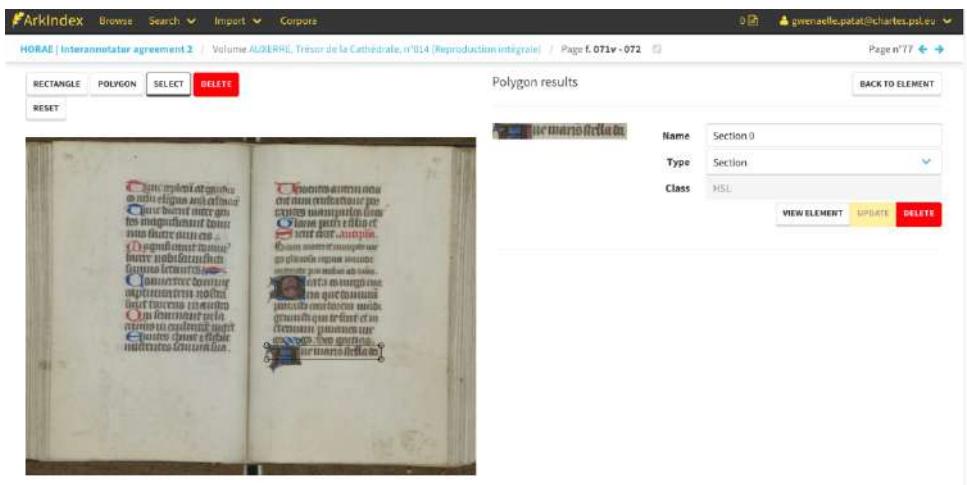
- Doutes sur le capitule de la leçon brève aux f.54v-55

This screenshot shows another page from the same manuscript. It features a photograph of the open book with text on both pages. Two distinct red rectangular boxes are overlaid on the text, one on each page, highlighting specific sections for closer examination.

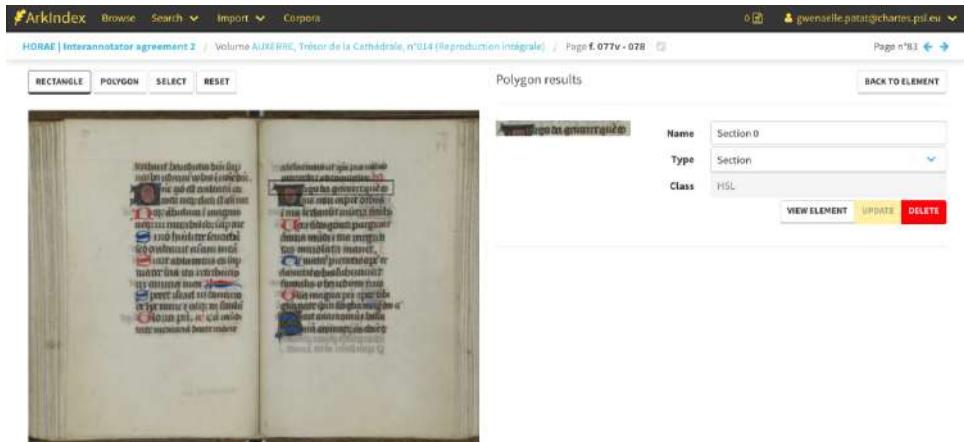
- Pas de section Preces notée pour l'ensemble des Heures de la Vierge



- Doutes sur le début de la section HSL dans la partie Vêpres des Heures de la Vierge aux f.71v-72



- Doutes sur le début de la section HSL dans la partie Complies des Heures de la Vierge aux f.77v-78



- Doutes sur le début des oraisons dans la partie Complies des Heures de la Vierge aux f.78v-79

The screenshot shows a digital manuscript viewer. On the left is a photograph of an open medieval manuscript. The right side features a 'Polygon results' interface. It lists two sections: 'Section 0' (Type: Section, Class: Canticle) and 'Section 1' (Type: Section, Class: Orationes). Each section has three buttons: 'VIEW ELEMENT', 'UPDATE', and 'DELETE'.

- Pas de section Preces et Orationes dans la partie Vêpre de l'Office des morts -> Oui (confirmation de Louis).

12/05

Si on peut voir les réclames (mot sous la dernière ligne d'une page annonçant ceux de la page suivante) sur les manuscrits numérisés, on ne peut faire de la collation (calcul du nombre de feuillets et de cahiers que comporte un volume), car cela demande une approche physique pour bien percevoir certains paramètres.

La liste des suffrages s'agrandit.

Particularités sur le manuscrit BEAUNE, Bibliothèque municipale, 054 (053) (Reproduction intégrale) par rapport au modèle défini dans le manuel d'annotation :

- Antienne du premier psaume de la section Laude dans les Heures de la Vierge qui correspond à "Sancta dei genitrix" au f.025v

HORAE | Interannotator agreement 2 / Volume BEAUNE, Bibliothèque municipale, 054 (053) (Reproduction intégrale) / Page f. 025v - 026 / Page n°45 ← →

RECTANGLE POLYGON SELECT RESET

Polygon results

Sicut erat adorandum

Name	Section 0
Type	Section
Class	Lauds

Sicut erat adorandum

Name	Section 1
Type	Section
Class	Invocation

Sicut erat adorandum

Name	Section 2
Type	Section
Class	Psalm

VIEW ELEMENT UPDATE DELETE

VIEW ELEMENT UPDATE DELETE

VIEW ELEMENT UPDATE DELETE

À noter que nous avons la même antienne au même endroit dans le manuscrit suivant BEAUNE, Bibliothèque municipale, 055 (054) au f.030v

HORAE | Interannotator agreement 2 / Volume BEAUNE, Bibliothèque municipale, 055 (054) (Reproduction intégrale) / Page f. 030v - 031 / Page n°37 ← →

RECTANGLE POLYGON SELECT RESET

Polygon results

Sicut erat adorandum

Name	Section 0
Type	Section
Class	Lauds

Sicut erat adorandum

Name	Section 1
Type	Section
Class	Invocation

Sicut erat adorandum

Name	Section 2
Type	Section
Class	Psalm

VIEW ELEMENT UPDATE DELETE

VIEW ELEMENT UPDATE DELETE

VIEW ELEMENT UPDATE DELETE

- Antienne du canticle dans la section Laude des Heures de la Vierge qui correspond à "Sancta Maria succurre miseris" f.033v

ArkIndex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume BEAUNE, Bibliothèque municipale, 054 (053) (Reproduction intégrale) / Page f. 033v - 034 / Page n°53 ← →

RECTANGLE POLYGON SELECT DELETE

Polygon results

Sicut erat adorandum

Name	Section 0
Type	Section
Class	Canticle

VIEW ELEMENT UPDATE DELETE

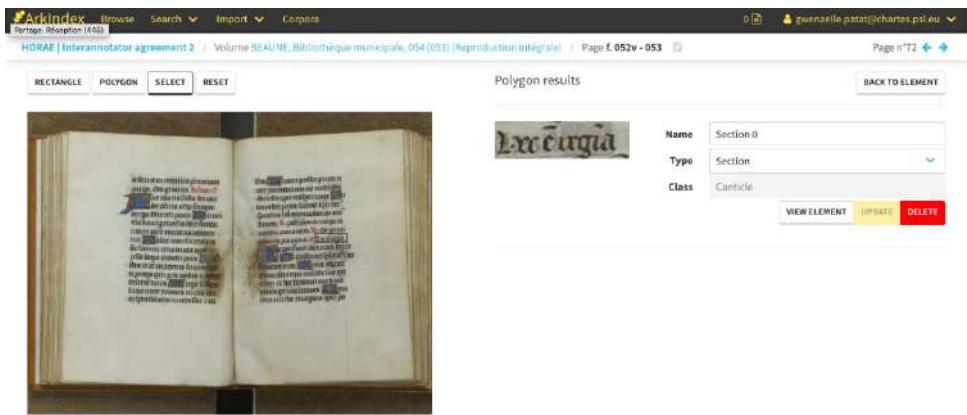
À noter que nous avons la même antienne au même endroit dans le manuscrit suivant BEAUNE, Bibliothèque municipale, 055 (054) au f.036v

The screenshot shows the ArkIndex interface for the manuscript BEAUNE, Bibliothèque municipale, 055 (054). The left side displays a photograph of an open medieval manuscript with two pages filled with dense Latin text in a Gothic script, decorated with several large, colorful initials. The right side shows a transcription of the text. At the top, there are search and navigation buttons: 'RECTANGLE', 'POLYGON', 'SELECT' (highlighted in red), and 'RESET'. Below these are two sections of transcriptions labeled 'Section 0' and 'Section 1'. Each section has fields for 'Name', 'Type', and 'Class'. For 'Section 0', the values are 'Section 0', 'Section', and 'HSL'. For 'Section 1', the values are 'Section 1', 'Section', and 'Canticle'. Each section also has 'VIEW ELEMENT', 'UPDATE', and 'DELETE' buttons.

- Antienne du premier psaume dans la section des Vêpres des Heures de la Vierge qui correspond à “Regali” au f.049v

The screenshot shows the ArkIndex interface for the manuscript BEAUNE, Bibliothèque municipale, 054 (053). The left side displays a photograph of an open medieval manuscript with two pages filled with dense Latin text in a Gothic script, decorated with several large, colorful initials. The right side shows a transcription of the text. At the top, there are search and navigation buttons: 'RECTANGLE', 'POLYGON', 'SELECT' (highlighted in red), and 'RESET'. Below these are three sections of transcriptions labeled 'Section 0', 'Section 1', and 'Section 2'. Each section has fields for 'Name', 'Type', and 'Class'. For 'Section 0', the values are 'Section 0', 'Section', and 'Vespers'. For 'Section 1', the values are 'Section 1', 'Section', and 'Invocation'. For 'Section 2', the values are 'Section 2', 'Section', and 'Psalms'. Each section also has 'VIEW ELEMENT', 'UPDATE', and 'DELETE' buttons. A large, stylized initial 'Regali' is visible at the bottom center of the transcription area.

- Antienne du canticle de la section Vêpres dans les Heures de la Vierge qui correspond à “regina” au f.052v



- Antienne du canticle de la section Complies des Heures de la Vierge qui correspond à "Salve regina" au f.057v

Doutes et questions :

Certains passages ont dans leur rubrique la mention "antiphon" avec le nom d'un saint. S'agit-il des suffrages ? Faut-il créer des classes vides pour ceux qui ne sont pas répertoriés ?

RECTANGLE POLYGON SELECT RESET

Polygon results

Les exemples ci-dessus se réfèrent à la Vierge Marie et saint Lazare à la fin de l'Office des morts dans deux manuscrits différents.

De même, la “prière pour les trespassés” dans l’Office des morts est-elle à annoter ?

ArkIndex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume BEAUNE, Bibliothèque municipale, 054 (053) [Reproduction intégrale] / Page f. 126v - 127

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT RESET

L'exemple ci-dessous correspond-il au suffrage de saint Lazare ?

ArkIndex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume BEAUNE, Bibliothèque municipale, 054 (053) [Reproduction intégrale] / Page f. 128v - 129

Polygon results BACK TO ELEMENT

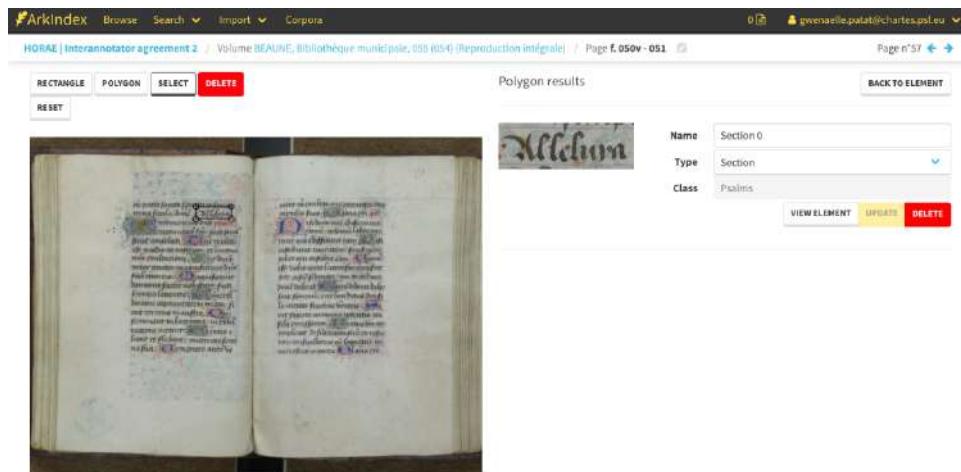
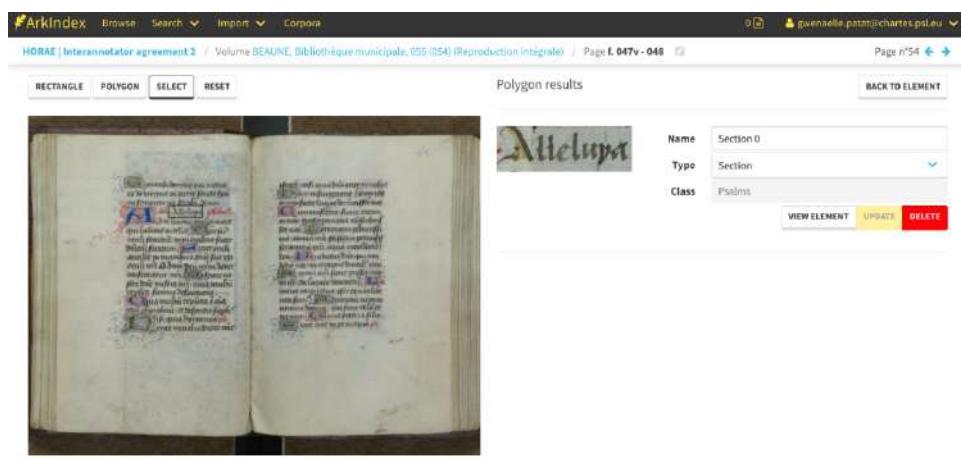
RECTANGLE POLYGON SELECT RESET

Name	Section 0
Type	Section
Class	Suffrages

Name	Section I
Type	Section
Class	None

Particularités

- Dans manuscrit BEAUNE, Bibliothèque municipale, 055 (054) (Reproduction intégrale), antienne du permier psaume de la section Sexte dans les Heures de la Vierge au f.047v qui correspond à “Alleluya”, tout comme l'antienne du permier psaume de la section None des mêmes Heures de la Vierge au f.050v :



- Dans manuscrit BEAUNE, Bibliothèque municipale, 055 (054) (Reproduction intégrale), antienne du permier psaume dans la section Vêpres des Heures de la Vierge qui correspond à “regali”

ArkIndex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume BEAUNE, Bibliothèque municipale, 055 (054) (Reproduction intégrale) / Page f. 052v - 053

Page n°59

RECTANGLE POLYGON SELECT RESET Polygon results BACK TO ELEMENT

	Name: Section 0
Type: Section	
Class: Vespers	
VIEW ELEMENT UPDATE DELETE	
	Name: Section 1
Type: Section	
Class: Invocation	
VIEW ELEMENT UPDATE DELETE	
	Name: Section 2
Type: Section	
Class: Psalms	
VIEW ELEMENT UPDATE DELETE	

- Dans manuscrit BEAUNE, Bibliothèque municipale, 055 (054) (Reproduction intégrale), suffrage à saint Lazare après Office des morts au f.121v

ArkIndex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume BEAUNE, Bibliothèque municipale, 055 (054) (Reproduction intégrale) / Page f. 121v - 122

Page n°128

RECTANGLE POLYGON SELECT RESET Polygon results BACK TO ELEMENT

	Name: Section 0
Type: Section	
Class: Suffrages	
VIEW ELEMENT UPDATE DELETE	
	Name: Section 1
Type: Section	
Class: test_class	
VIEW ELEMENT UPDATE DELETE	

- Dans manuscrit BEAUNE, Bibliothèque municipale, 055 (054) (Reproduction intégrale), suffrage à saint Claude après Office des morts au f.123v

ArkIndex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume BEAUNE, Bibliothèque municipale, 055 (054) (Reproduction intégrale) / Page f. 123v - 124

Page n°130

RECTANGLE POLYGON SELECT DELETE

Polygon results

BACK TO ELEMENT

Name: Section 0
Type: Section
Class: test_class

VIEW ELEMENT UPDATE DELETE

- Dans BnF. Bibliothèque de l'Arsenal. Ms-637 réserve, antienne du permier psaume de la section Sexte dans les Heures de la Vierge qui corespond à "In Odorem" à la p.57v

ArkIndex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume Bnf. Bibliothèque de l'Arsenal, Ms-637 réserve / Page 57v

Page n°130

RECTANGLE POLYGON SELECT DELETE

Polygon results

BACK TO ELEMENT

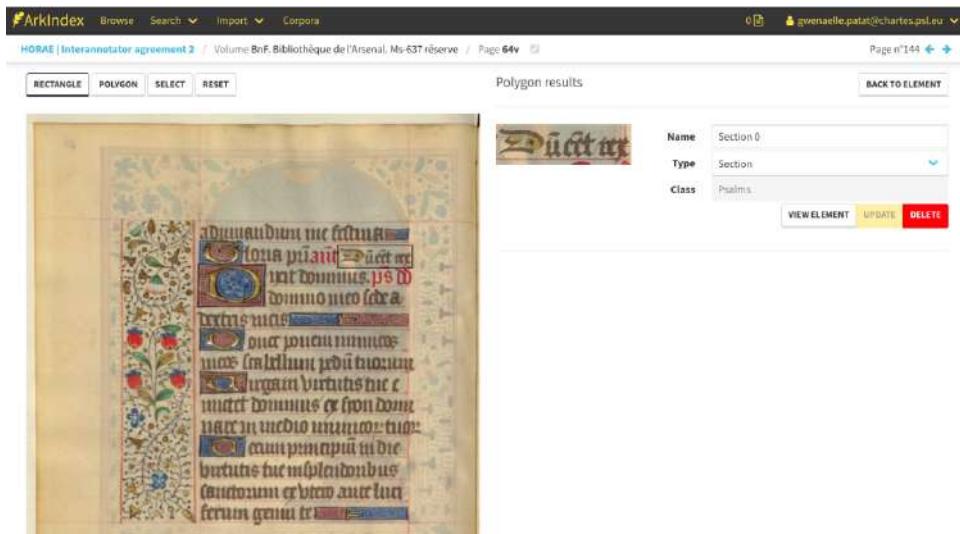
Name: Section 0
Type: Section
Class: Hymn

VIEW ELEMENT UPDATE DELETE

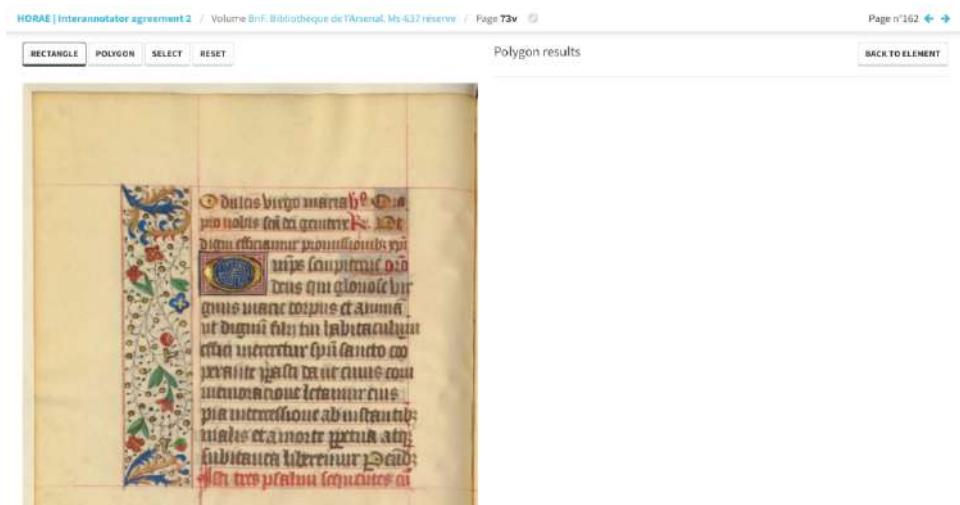
Name: Section I
Type: Section
Class: Psalms

VIEW ELEMENT UPDATE DELETE

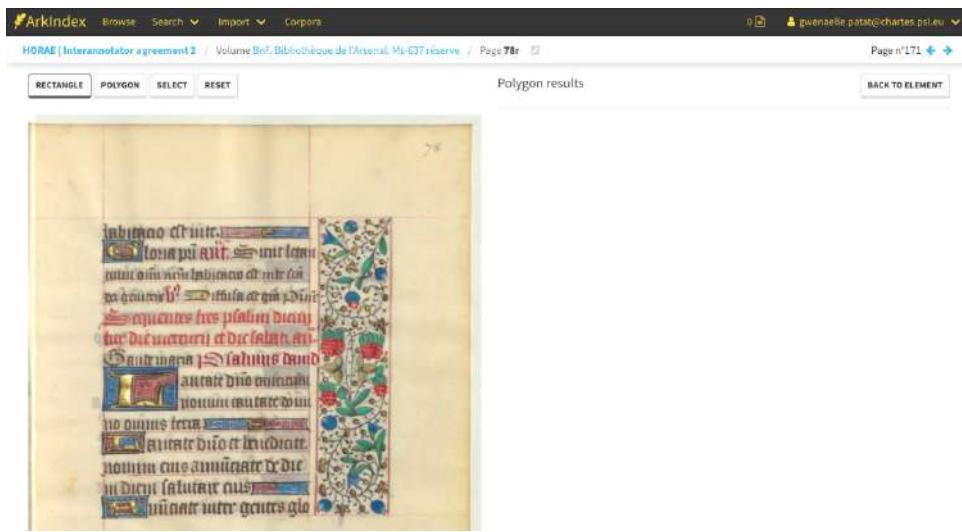
- Dans BnF. Bibliothèque de l'Arsenal. Ms-637 réserve, antienne du permier psaume de la section Vêpres des Heures de la Vierge qui correspond à "Ducent rex" p.64v



- Dans BnF. Bibliothèque de l'Arsenal. Ms-637 réserve, trois psaumes pour mardi et vendredi après les Heures de la Vierge à la p.73v



- Dans BnF. Bibliothèque de l'Arsenal. Ms-637 réserve, trois psaumes pour mercredi et samedi après les Heures de la Vierge à la p.78r



- La section Tierce arrive après la section Sexte dans le manuscrit BnF. Bibliothèque de l'Arsenal. Ms-651 réserve aux p.24v et 25r

ArkIndex Browse Search Import Corps gwenaelle.pata@charles.psl.eu Page n°171 BACK TO ELEMENT

HORAE | Interannotator agreement 2 / Volume BnF, Bibliothèque de l'Arsenal, Ms-651 réserve / Page 24v

Polygon results

RECTANGLE POLYGON SELECT RESET

SECTION Name: Section 0 Type: Section Class: Sexte

VIEW ELEMENT UPDATE DELETE

ArkIndex Browse Search Import Corps gwenaelle.pata@charles.psl.eu Page n°52 BACK TO ELEMENT

HORAE | Interannotator agreement 2 / Volume BnF, Bibliothèque de l'Arsenal, Ms-651 réserve / Page 24v

Polygon results

RECTANGLE POLYGON SELECT RESET

SECTION Name: Section 0 Type: Section Class: Sexte

VIEW ELEMENT UPDATE DELETE

ArkIndex Browse Search Import Corps gwenaelle.pata@charles.psl.eu Page n°53 BACK TO ELEMENT

HORAE | Interannotator agreement 2 / Volume BnF, Bibliothèque de l'Arsenal, Ms-651 réserve / Page 25r

Polygon results

RECTANGLE POLYGON SELECT DELETE

SECTION Name: Section 0 Type: Section Class: Tierce

VIEW ELEMENT UPDATE DELETE

The image shows three screenshots of the ArkIndex software interface for the BnF Ms-651 manuscript. The first screenshot shows folios 24v and 25r with the 'Polygon' tool selected. The second screenshot shows the 'Section 0' entry in the database with 'Sexte' assigned as the class. The third screenshot shows folio 25r with the 'Delete' button highlighted in red.

- Dans le manuscrit BnF. Bibliothèque de l'Arsenal. Ms-651 réserve, l'antienne du premier psaume dans la section Laudes des Heures de la Vierge correspond à "Dignare me" à la p.37v

The screenshot shows a digital interface for managing manuscript annotations. At the top, there's a header with 'ARKINDEX' and various navigation links like 'Browse', 'Search', 'Import', and 'Corpora'. Below the header, it says 'HORAE | Interannotator agreement 2 / Volume Bnf. Bibliothèque de l'Arsenal. Ms-651 réserve / Page 37v'. On the left, there are buttons for 'RECTANGLE', 'POLYGON', 'SELECT', and 'RESET'. The main area shows a photograph of a medieval manuscript page with Gothic script and colorful initials. A red polygon highlights the beginning of a psalm. To the right, a 'Polygon results' panel displays a table with one row. The row contains a small icon of a book, the text 'Dignare me.', and three input fields: 'Name' (set to 'Section 0'), 'Type' (set to 'Section'), and 'Class' (set to 'Psalms'). Below the table are buttons for 'VIEW ELEMENT', 'UPDATE', and 'DELETE'.

Doutes

Mettre des “test-class” pour les suffrages avec des saints non-repertoriés, et annoter prières.

- Dans manuscrit BEAUNE, Bibliothèque municipale, 055 (054) (Reproduction intégrale), trois oraisons sur la passion et la résurrection -> à annoter ? Si oui comment ?

This screenshot shows another instance of the ArkIndex annotation tool. The top bar includes 'ARKINDEX', 'Browse', 'Search', 'Import', 'Corpora', and a user email 'gwenaelle.pataud@charles.psl.eu'. The URL indicates 'HORAE | Interannotator agreement 2 / Volume BEAUNE, Bibliothèque municipale, 055 [054] (Reproduction intégrale) / Page f. 125v - 126'. The interface features 'RECTANGLE', 'POLYGON', 'SELECT', and 'RESET' buttons. The main image shows a double-page spread of a medieval manuscript with two columns of text per page. A large red polygon covers the entire left-hand page. The 'Polygon results' panel on the right shows a single row with a book icon, the text 'digne et misericordieus domine', and a dropdown menu for 'Section 0' under 'Name', 'Section' under 'Type', and 'Orations' under 'Class'. Buttons for 'VIEW ELEMENT', 'UPDATE', and 'DELETE' are at the bottom.

- Dans manuscrit BEAUNE, Bibliothèque municipale, 055 (054) (Reproduction intégrale), section inconnue au f.131r

ArkIndex Browse Search Import Corps

HORAE | Interannotator agreement 2 / Volume BEAUNE, Bibliothèque municipale, 055 (054) (Reproduction intégrale) / Page f.130v - 131

Polygon results

RECTANGLE POLYGON SELECT RESET

BACK TO ELEMENT

Name	Section 0
Type	Section
Class	Maria Magdalena

VIEW ELEMENT UPDATE DELETE

- Peut-on annoter le f.126v du le manuscrit BEAUNE, Bibliothèque municipale, 054 (053) (Reproduction intégrale) comme un suffrage aux trespassés ?

ArkIndex Browse Search Import Corps

HORAE | Interannotator agreement 2 / Volume BEAUNE, Bibliothèque municipale, 054 (053) (Reproduction intégrale) / Page f.126v - 127

Polygon results

RECTANGLE POLYGON SELECT RESET

BACK TO ELEMENT

Name	Section 0
Type	Section
Class	test_class

VIEW ELEMENT UPDATE DELETE

- Peut-on annoter cette partie comme une prière à la Vierge dans AUXERRE, Trésor de la Cathédrale, n°014 (Reproduction intégrale) au f.143v ?

ArkIndex Browse Search Import Corps

HORAE | Interannotator agreement 2 / Volume AUXERRE, Trésor de la Cathédrale, n°014 (Reproduction intégrale) / Page f.143v - 144

Polygon results

RECTANGLE POLYGON SELECT RESET

BACK TO ELEMENT

Name	Section 0
Type	Section
Class	Prayers

VIEW ELEMENT UPDATE DELETE

- Doute sur début de la leçon brève dans la section Tierce des Heures de la Vierge du manuscrit BnF. Bibliothèque de l'Arsenal. Ms-651 réserve à la p.54v



Polygon results

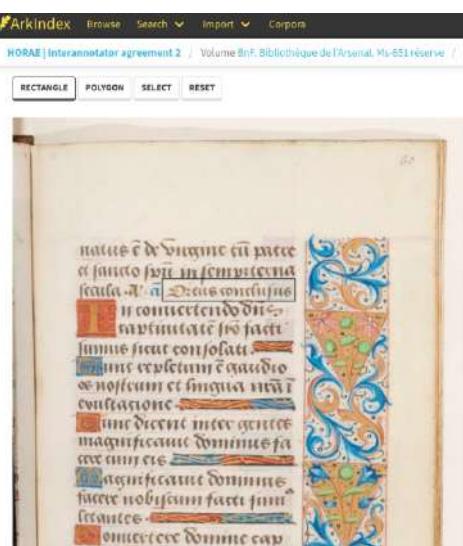
Name	Section 0
Type	Section
Class	Short lesson

View Element Update Delete

14/05

Particularités

- Antienne du premier psaume de la section None des Heures de la Vierge du manuscrit BnF. Bibliothèque de l'Arsenal. Ms-651 correspondant à "Ortus conclusus"



Polygon results

Name	Section 0
Type	Section
Class	Psalms

View Element Update Delete

- Antienne du canticle de la section Complies des Heures de la Vierge du manuscrit BnF. Bibliothèque de l'Arsenal. Ms-651 correspondant à "Sicut mirra"

Arkindex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume BnF Bibliothèque de l'Arsenal Ms-651 réservé / Page 72r

This polygon is too small

Polygon results

Sicut mirra.

Name	Section 0
Type	Section
Class	Canticle

VIEW ELEMENT UPDATE DELETE

The manuscript page features a large initial 'S' at the top, followed by dense Latin text in two columns. The text begins with 'I tua se clavis sit viscerca facta?'. The right margin contains decorative floral flourishes.

- Pas de Preces dans la section Complies des Heures de la Vierge du manuscrit BnF. Bibliothèque de l'Arsenal. Ms-651 correspondant à "Sicut mirra"
- Suffrage à la sainte Trinité après l'Office des morts dans le manuscrit BnF. Bibliothèque de l'Arsenal. Ms-651

RECTANGLE POLYGON SELECT RESET

Polygon results

Sicut mirra.

Name	Section 0
Type	Section
Class	Suffrages

Sicut sancta.

Name	Section 1
Type	Section
Class	Trinitas

VIEW ELEMENT UPDATE DELETE

Element created.

The page displays a large initial 'S' at the top left, followed by text in two columns. A portrait of a saint is visible on the left side. The right margin features intricate blue and gold floral patterns.

- Suffrage à l'archange saint Michel après la prière "O Intemerata" dans le manuscrit BnF. Bibliothèque de l'Arsenal. Ms-651

RECTANGLE POLYGON SELECT RESET

Polygon results

[BACK TO ELEMENT](#)

Et talen remittentiam in his
sua remittens precatis et se
leibus meis agere. Si vita
bus tuis beneplacenter sit et
grata. Si misericordia tua facias
sunt misericordiae cum sanctis et
electis suis misericordiam eius
quem conseruat semper eterna.
Amen. De sancto Michaeli.

Michael archangelus
Michael archangelus
te uocamus sem
misericordiam
uoxis dei et
tua nos defe
dere a uocella
te minime et
eternam ducere i

Michael archangelus
Michael archangelus
te uocamus sem
misericordiam
uoxis dei et
tua nos defe
dere a uocella
te minime et
eternam ducere i

Name: Section 0
Type: Section
Class: Suffrages
[VIEW ELEMENT](#) [UPDATE](#) [DELETE](#)

Name: Section 1
Type: Section
Class: Michael
[VIEW ELEMENT](#) [UPDATE](#) [DELETE](#)

- Suffrage à saint Claude dans le manuscrit BnF. Bibliothèque de l'Arsenal. Ms-651

RECTANGLE POLYGON SELECT RESET

Polygon results

[BACK TO ELEMENT](#)

bis facie incolare. Si dicam
efficiamur promiscimur vesti
Opus qui beatum in te
laum uocantes tuum
immaculatum deorashi in mea
lis tribus nobis quesumus
Si ene misericordia et precebas a
genuine misericordia letemur
Per dominum nostrum ihesum christum
ad sancto claudio. Amen.

Ode sola
Ode sola
tuum et
sobborc me
tuncum
sobborc
et suuccio
misericordia
tum reco

Name: Section 0
Type: Section
Class: test_class
[VIEW ELEMENT](#) [UPDATE](#) [DELETE](#)

- Prière à saint Gond dans le manuscrit BnF. Bibliothèque de l'Arsenal. Ms-651

Arkindex Browse Search Import Corpora

HORAE | Interannotator agreement 2 | Volume BnF. Bibliothèque de l'Arsenal. Ms-651 réservé / Page 124v

Polygon results

RECTANGLE POLYGON SELECT RESET

Page n°252 ← →

Polygon results

Name: Section 0
Type: Section
Class: Prayers

Sancte
Gond
tu as me
tenu.
Emmevi
en par la
fame etre.
faire cesse
la pestilence.
Sois en ville ou en ville.
Tu monstra ton autorite
Par ta grande legumolere
Defens nous de bau iſtue
Et de mortelle iniurie.
De buse tache epidomme
Fais que nous ayons affluer
De sancte et en consequencie.
Avec ton innumerabile misericorde.

VIEW ELEMENT UPDATE DELETE

- Suffrage à la Vierge dans le manuscrit BnF. Bibliothèque de l'Arsenal. Ms-651

HORAE | Interannotator agreement 2 | Volume BnF. Bibliothèque de l'Arsenal. Ms-651 réservé / Page 125r

Polygon results

RECTANGLE POLYGON SELECT RESET

Page n°253 ← →

Polygon results

Name: Section 0
Type: Section
Class: Suffrages

Sancte
Ana
Certo
dixit papa
Iohannes et ceteri
sacri capituli
Iustus et domini
imperiale est
eo quod eis
genuit. **S**ed dominus fuit ei
pericula multa. **A**nnonam
dilexit multum. **D**ominus ordo
Earque nobis clement
et misericordia patet. **D**e fidelis
maria magdalena domini
nostri sumus omnes ora diligendo
sicut etiam sumus. **R**emans ut et
mum na avidit tua misericordiam
semper tenet in te.

VIEW ELEMENT UPDATE DELETE

Name: Section 1
Type: Section
Class: test_class

VIEW ELEMENT UPDATE DELETE

Element created.

- Suffrage à sainte Geneviève dans le manuscrit BnF. Bibliothèque de l'Arsenal. Ms-651

HORAE | Interannotator agreement 2 / Volume Bnf: Bibliothèque de l'Arsenal: Ms.651 réservé / Page 127r

Polygon results

RECTANGLE POLYGON SELECT RESET BACK TO ELEMENT

Name: Section 0
Type: Section
Class: test_class
VIEW ELEMENT UPDATE DELETE

Element created.

- Pas de Preces dans la section Laude des Heures de la Vierge dans le manuscrit manuscrit CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale)
- Dans le manuscrit CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale), les Heures de la Vierge, après la section des Laudes, semblent être entrecoupées par le début des Heures de la Croix, puis le début des Heures du Saint-Esprit. S'ensuivent quelques suffrages, puis les Heures de la Vierge reprennent avec la section des Prime. **Pb dans la foliation, la collation ? Construction atypique ?**

ArkIndex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 [Reproduction intégrale] / Page E. 031v - 032

Polygon results

RECTANGLE POLYGON SELECT RESET BACK TO ELEMENT

Name: Section 0	Type: Section	Orations
Name: Section 1	Type: Section	Hours of the Cross
Name: Section 2	Type: Section	Matins

VIEW ELEMENT UPDATE DELETE

- Suffrages à saint Eloy et saint Thomas après le début des heures du saint Esprit dans le manuscrit CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale)

Arkindex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale) / Page f. 036v - 037

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT RESET

Section 0

- Name: Section 0
- Type: Section
- Class: Suffrages
- Buttons: VIEW ELEMENT, UPDATE, DELETE

Section 1

- Name: Section 1
- Type: Section
- Class: test_class
- Buttons: VIEW ELEMENT, UPDATE, DELETE

Section 2

- Name: Section 2
- Type: Section
- Class: test_class
- Buttons: VIEW ELEMENT, UPDATE, DELETE

- Antienne du premier psaume de la section Prime des Heures de la Vierge dans le manuscrit CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale) correspondant à "Beata Mater"

Arkindex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale) / Page f. 039v - 040

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT RESET

Section 0

- Name: Section 0
- Type: Section
- Class: Hymn
- Buttons: VIEW ELEMENT, UPDATE, DELETE

Section 1

- Name: Section 1
- Type: Section
- Class: Psalms
- Buttons: VIEW ELEMENT, UPDATE, DELETE

- La section Tierce des Heures de la Vierge du manuscrit CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale) n'a pas de Preces
- L'évangile de saint Jean dans le manuscrit CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale) est suivie de la prière Obsecro te et de suffrages
- Suffrage à la Vierge Marie dans le manuscrit CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale) ?

Arkindex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale) / Page f. 086v - 087

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT DELETE RESET

Section 0

Name: Section 0
Type: Section
Class: Catharina

Section 1

Name: Section 1
Type: Section
Class: Class...
CREATE

VIEW ELEMENT UPDATE DELETE

- Suffrage à saint Jacques dans le manuscrit CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale) ?

Arkindex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale) / Page f. 090v - 091

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT DELETE RESET

Section 0

Name: Section 0
Type: Section
Class: test_class

VIEW ELEMENT UPDATE DELETE

- Mention de saint Gilles dans Office des Morts du manuscrit CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale) ?

Arkindex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale) / Page f. 125v - 126

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT DELETE RESET

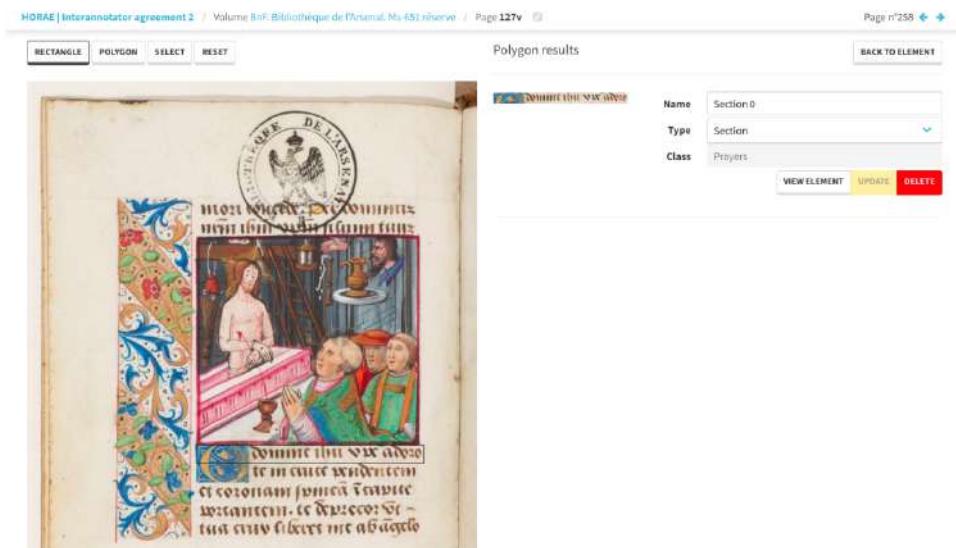
Section 0

Name: Section 0
Type: Section
Class: Dantes

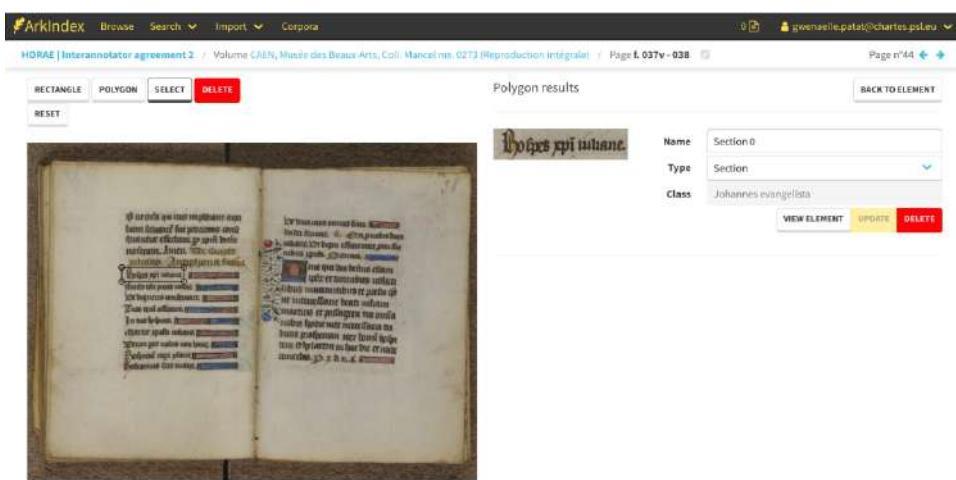
VIEW ELEMENT UPDATE DELETE

Doutes

- Prière à Jésus Christ dans le manuscrit BnF. Bibliothèque de l'Arsenal. Ms-651 ?



- Suffrage à saint Jean l'Évangéliste dans le manuscrit CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale) ?



- Deux suffrages à saint André dans le manuscrit CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale) ?

ArkIndex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale) / Page f. 087v - 088

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT RESET

Name: Section 0
Type: Section
Class: test_class

VIEW ELEMENT UPDATE DELETE

- Suffrages à deux saints dont je ne suis pas sûre de reconnaître les noms (saint Lagne et saint Philippe ?) dans le manuscrit CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale)

ArkIndex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume CAEN, Musée des Beaux-Arts, Coll. Mancel ms. 0273 (Reproduction intégrale) / Page f. 089v - 090

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT DELETE

RESET

Name: Section 0
Type: Section
Class: Marcus

VIEW ELEMENT UPDATE DELETE

Name: Section 1
Type: Section
Class: test_class

VIEW ELEMENT UPDATE DELETE

15/05

Particularités

- Dans Genève, Bibliothèque de Genève, Comites Latentes 38, prière pour l'élévation du Corps du Christ p. 37r après les psaumes pénitentiaux. **Référence au rituel de l'élévation pour le corps et le sang du Christ lors de l'eucharistie ?**

Arkindex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume Genève, Bibliothèque de Genève, Comites Latentes 38 / Page 37r

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT RESET

Sequuntur orationes deinde dicende in elevatione corporis Christi oitoy.

Ne verum corpus natum de maria Virgine. Vite passionis immolatum in cruce pro homine. cuius latum perforatum vero flavit sanguine. est nobis preustatum mortis in eximere. O clementis opere dulcis fili marie. Domine. oratio.

Alma sancta cura dei per quam salvi facti sunt mihi. seruos tuos redemisti. dum in cruce repridiasti. Vnde que de te manatur. a peccatis mundauit. quod pertinet primus homo. misericordius deponit. Sancta aura tu me mundi domino sanguine

VIEW ELEMENT UPDATE DELETE

- Prières pour les différents jours de la semaine (par ordre d'apparition) dans Genève, Bibliothèque de Genève, Comites Latentes 38. Celle du lundi suit immédiatement l'évangile de saint Jean :

Arkindex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume Genève, Bibliothèque de Genève, Comites Latentes 38 / Page 40r

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT RESET

In die dominica. oratio.

Domi domine os iustum ad laudandum et benedicendum sanctissimum et ineffabilem nomen tuum mundum; et omnes ab omnibus cantat omnes mondi vites et letantes et fac me semper obediens mandans et a te manu tua regnare. Et ante conspectum domini via letantes tunc audire mezar. **E**x domini uox in omnibus filium tuum qui regnabit et regnat in unitate spiritus sancti dominus. **A**men. omnia scilicet seculorum amen.

Manastas domine amio fortunatus nomen corpus manum et

VIEW ELEMENT UPDATE DELETE

Arkindex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume Genève, Bibliothèque de Genève, Comites Latentes 38 / Page 49r

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT RESET

In die lunae. oratio.

Eus omnipotens pater noster misericordia nostra et conforta nos omnibus diebus dote nre. Deus abraham de ipsa ac deus iacob miserere mei et misere cordi auctorizam meum sanctum michalem archangeli qui me protegat et defendat ab omnibus numeris misericordiis et misericordibus carnalibus et spiritualibus. Sancte michael archangeli per gratiam quam merenti habere deprecor te per unigenitum filium tuum omnipotentem dominum nostrum ihesum

VIEW ELEMENT UPDATE DELETE

ArkIndex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume Genève, Bibliothèque de Genève, Comites Latentes 38 / Page 53r

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT DELETE RESET

In die martis. ps.

Omne deus omnes pater
filius et spiritus sanctus
in manu. famulo tuo
victoriam contra omnes
unicos meos ut non possint in resistere
nece contra me aliquid ducere sed dirigat
virtus tua consilium eorum in bonum
Et tu deus sis formidatio mea et resum
unum meum et duxis defensionis mee
gratias dispensantur et confundatur
omnes adversarii mei. Benedic
qui es deus abraham deus ysaac deus
iacob deus omnius benedictus am. Li
feta me de omnibus irrationibus adiuva

Name: Section 0
Type: Section
Class: Prayers
VIEW ELEMENT UPDATE DELETE

ArkIndex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume Genève, Bibliothèque de Genève, Comites Latentes 38 / Page 55r

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT DELETE RESET

In die meritis. ps.

Edu moribus perire
domine intellige domine
meum. Intende uero
me meo rex meus et deus meus. Ne
ne in uiria armis me usq; in furore
tuo corporis me. Sciat madame
quoniam infelix sum sicut me de
quoniam contulata sunt offa mea.
Et anima mea turbata est ualde
sed tu domine usq; domiter
domine et aux animam meam salua
me fac propter misericordiam tuam. Edip
et in me et euandi me domine deus me

Name: Section 0
Type: Section
Class: Prayers
VIEW ELEMENT UPDATE DELETE

ArkIndex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume Genève, Bibliothèque de Genève, Comites Latentes 38 / Page 66r

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT RESET

In die iouis. ps.

Omne deus omnes rex
caelorum et terrae ubi
confiteor vobis peccatis
mea q; ego misericordus erector: per
auem nimis in existimationib; vobis
rebus oculis in operibus restituisse
in premiacione laeti ut in his
omnibus me reum esse confiteor.
Proterea non sum dumne- ego
peccator: scilicet oculis ad celum nec
wan filius natus q; in longinuam
removens me esse recognoscere. Propter
multitudinem peccatorum meorum q; ne
ego contra receptum tuorum feci.
videlicet in fidei tuae iniurias meas

Name: Section 0
Type: Section
Class: Class...
CREATE

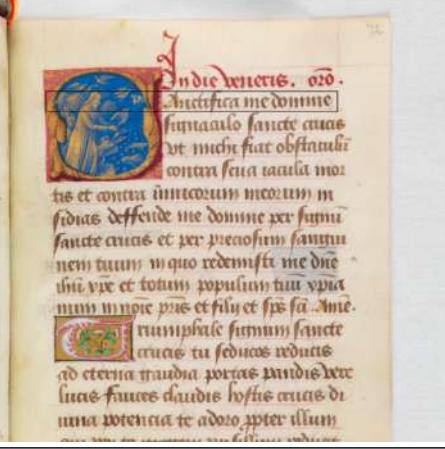
Horae | Interannotator agreement 2 | Volume Genève, Bibliothèque de Genève, Comites Latentes 38 | Page 72r

Polygon results

Section 0

Name: Section 0
Type: Section
Class: Prayers

VIEW ELEMENT **UPDATE** **DELETE**



*In die Veneris. ordo.
Auxilium me domine
sumculo sancte crucis
ut misericordia tua obstat
contum scia ualua mortis
et contra inimicorum meorum in
fidem defende me domine per signum
sancte crucis et per preciosum sanguinem
tuum in quo redemisti me domi
nus uix et totum populum tuum exia
num in nomine patris et filii et spiritus sancti. Ame.
Rumpiale sumum sancte
crucis tu sedes redue
ad eternam tristitia portas pundi dext
lucis fuitus diuidit hostis crucis di
uina potentia te adoro propter illum.*

Horae | Interannotator agreement 2 | Volume Genève, Bibliothèque de Genève, Comites Latentes 38 | Page 77r

Polygon results

Section 0

Name: Section 0
Type: Section
Class: Prayers

VIEW ELEMENT **UPDATE** **DELETE**



*In die sabbati. ordo.
Talat mater dolos
sa. uaria cence lacri
mosa. dum pendebat
filius. Iesus aran
gementem. contrastantem et dolere.
pertinuerunt gladii. Ego in tribus
et afflita. fuit illa benedicta. mar
yogenita. Quae interelat et dole
bat. et tremebat dum videbat. nat
uram inclinat. Iesus est homo q
non fleret. matrem vero si videret.
in tanto supplicio. Iesus non vo
test contumescere. priam matrem
contemplari. dolentem cui filio.*

- Prière aux anges dans Genève, Bibliothèque de Genève, Comites Latentes 38 p.114

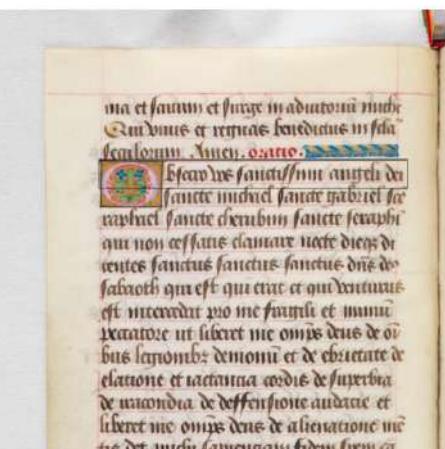
Horae | Interannotator agreement 2 | Volume Genève, Bibliothèque de Genève, Comites Latentes 38 | Page 52v

Polygon results

Section 0

Name: Section 0
Type: Section
Class: Prayers

VIEW ELEMENT **UPDATE** **DELETE**



*ma et sanum et surge in adiutorium nunc
Quoniam et regnas honoratus in se
scandolorum. Amen. oratio.
Eboracensis sancti iustini angeli dei
sancte iustine sancte gabrielis sa
raphaeli sancte clemens sancte scaphi
qui non cessans clamans nocte dies di
centes sanctus sanctus sanctus dñs de
saloeth qui est qui erat et qui venit
et intercedat pro me fragili et minimi
pectore ut liberet me omnis deus de o
bus legiombz demonum et de ebrietate de
clamone et taciturna modis de superbia
de iracundia de defensione audacie et
liberet me omnis deus de alienacione me
nis det mihi laeticiam fidem sicut et*

- Prière contre le péril et les ennemis dans Genève, Bibliothèque de Genève, Comites Latentes p.146

The image shows a page from a medieval manuscript. The text is written in two columns in a Gothic script. The top section is a prayer, and the bottom section begins with a large initial 'Q' followed by several names of saints.

Text (approximate transcription):

deus tuus duxi et parvum obedien
tiam et in omnibus operebone
vix in finem uite mee uiuam per
securitatem ut merear te diligere
seruire et obediere et tibi cum fias
tuis in eternum assistere et te
facie ad faciem videre et mihi
separari a te me permitte. Et ut
possim facias de vultu tuo faciu
tissimo. Et de eternablis
gloria tua in eternum in concede.
Quia in virtute perfecta viuis
et regnas deus per omnia secula
seculorum. Amen. Contre tout
peur et contre tous mes ennemis. Et p
mecum qui doit dire in pat' nez et in foiz. Aue.

Initial:

Qaput vpi frons hebre. cauli
Amoris. Imitatio salomonis.
libia danielis. cor dauid sunt miche
monach. dominus & contra les innocens.

- Oraisons à dire devant l'autel pour soi ou pour celui qu va entrer en bataille dans Genève, Bibliothèque de Genève, Comites Latentes p.69r

The image shows a page from a medieval manuscript. The text is written in two columns in a Gothic script. The top section is a prayer, and the bottom section begins with a large initial 'Q' followed by several names of saints.

Text (approximate transcription):

meus invisibilis corporales et incorpo
rales deus meus pone illos ut rotam
et flammam ante faciem venturam
immaculam gabrielis raphaelis ias
pat melchior bastus iohannes
lucas marcus et matheus custode
me ab omni malo amen. Ihesus
autem transiens per medium u
lorum ibat. Oh dor dire ces oraison
enfin deuine laurel poe son pour ce
lui qui doit entrer en bataille et aux
qui se doit recommander a dieuz alas
Quelcunqur uulcunqur lequel pour
pater et filius et spiritus so
nus es in sancta trinitate unus de
respite de sed maestatis san et co
gita de nobis misera deus auem

- Prières aux apôtres puis aux saints innocents dans Genève, Bibliothèque de Genève, Comites Latentes p.84r

De apostolis. Oratio.

Incussum ap̄lū domini nū
ihsu vīt̄ beatissime petre
pauli andrea iacobe iohannes thoma
iacobe philippe bertholomee matthee
francis thadée matthia barnabae luce
et mattei cum omnibz discipulis q̄s
domini nū ihsu vīt̄ orate pro me
ut abstrahatis me dō mīndō sciat
abstrahit vs et donet michi sc̄i
vestrum suu rei in extremo agmine
vō. Amen. **De sanctis martyribz.**

mult̄ martyres innocentes q̄
sequuntur dominum q̄tauz
erit qui cantant p̄imiae deo et
agni et in ore nra nō est mendaciu
qui sine misericordia ante thronū
agni. obsecrāt̄ us beatissimum et glo
riosissimum ut intercedatis pro me
corrip̄to et insecuri. domine ihsu v̄p̄c

Name	Section 0
Type	Section
Class	Prayers
VIEW ELEMENT UPDATE DELETE	

Name	Section 1
Type	Section
Class	Prayers
VIEW ELEMENT UPDATE DELETE	

- Prières aux martyrs et aux confesseurs dans Genève, Bibliothèque de Genève, Comites Latentes p.84V

pro corum intercessione insecuri me. **A.**

De omnibz martyribz
martyres gloriosi qui fortis
quintus in extrema passione in
terred pro me in domino et misericordia
ressum in vita scalo confidem vici
erit p̄t̄ nra et deum p̄metemam
de eisdem reatu agere in obz dieb;
vite me. obsecrāt̄ te dñe ihsu v̄p̄c ut
per corum intercessionem insecuris
me et dimittit in oīa vita mea A.

De confessoribz

anctissimi confessores qui
vix quiescere nū vix et vixisse hic
mundum in malum possum per
eum qui nec idem habebantur

Name	Section 0
Type	Section
Class	Prayers
VIEW ELEMENT UPDATE DELETE	

Name	Section 1
Type	Section
Class	Prayers
VIEW ELEMENT UPDATE DELETE	

- Prières aux vierges et à tous les saints dans Genève, Bibliothèque de Genève, Comites Latentes p.85r

HORAE | Interannotator agreement 2 / Volume Genève, Bibliothèque de Genève, Comité Latentes 38 / Page B5r

Polygon results

RECTANGLE POLYGON SELECT RESET

BACK TO ELEMENT

Name	Section 0
Type	Section
Class	Prayers

VIEW ELEMENT UPDATE DELETE

Name	Section 1
Type	Section
Class	Prayers

VIEW ELEMENT UPDATE DELETE

- Prière “Stabat mater dolorosa” dans Suisse, Zürich, Zentralbibliothek, Rheinau 169 p.30r

HORAE | Interannotator agreement 2 / Volume Suisse, Zurich, Zentralbibliothek, Rheinau 169 / Page 30r

Polygon results

RECTANGLE POLYGON SELECT RESET

BACK TO ELEMENT

Name	Section 0
Type	Section
Class	Prayers

VIEW ELEMENT UPDATE DELETE

- Psaume et antienne non-répétoriés dans la section prime des Heures de la Vierge dans Suisse, Zürich, Zentralbibliothek, Rheinau 169 à la p.62r, correspondant à “Laudate dominum” et “ecce te pulchra”

ArkIndex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume Suisse, Zürich, Zentralbibliothek, Rheinau 169 / Page 62r

Polygon results BACK TO ELEMENT

RECTANGLE POLYGON SELECT RESET

Corde pulchra

mater misericordie tu
nos ab iuste protege et
boga misericordia

Oratibi domine
quoniam tu es de misericordia
tu patet et sanato spu
tu in tempore nostra
cita amen. Et coram pulchra

Unde domine p
Quoniam omnes gen
tes laudent cum omni
populi.

Quoniam tu misericordia
mecum habes nos misericordia

Name: Section 0
Type: Section
Class: Psalms
VIEW ELEMENT UPDATE DELETE

- Pas de Preces dans la section Tierce des Heures de la Vierge dans Suisse, Zürich, Zentralbibliothek, Rheinau 169
- Prière avec lettrine manquante à la page 174v dans Suisse, Zürich, Zentralbibliothek, Rheinau 169

ArkIndex Browse Search Import Corpora

HORAE | Interannotator agreement 2 / Volume Suisse, Zürich, Zentralbibliothek, Rheinau 169 / Page 174v

Polygon results BACK TO ELEMENT

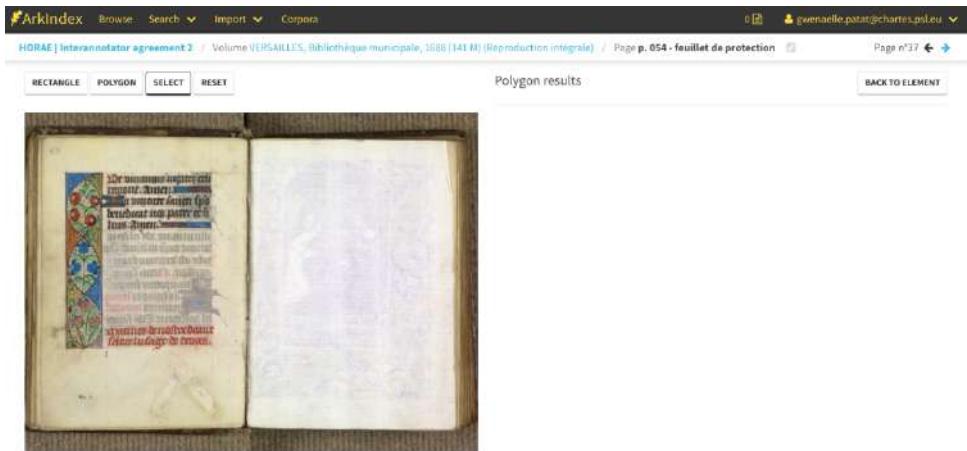
RECTANGLE POLYGON SELECT DELETE RESET

Aude maria rego

Quoniam maria rego
nuntias beatis solis
interventu que gabrielis
arriugeti dicas credi
didi domine rego deuini et
dominem graniti i post
patrum rego nuncolata
permaneant. Gabriele
arriugetum fatus di
muntis te esse affatum.
utrum tu de ipm fatus
nomens impregnatum
crucesat indeus infelix
qui dicit ipm ex iosephi

Name: Section 0
Type: Section
Class: Prayers
VIEW ELEMENT UPDATE DELETE

- Dans VERSAILLES, Bibliothèque municipale, 1688 (141 M) (Reproduction intégrale), Heures de la Vierge précisée selon l'usage de Troyes à la p.054



- Dans VERSAILLES, Bibliothèque municipale, 1688 (141 M) (Reproduction intégrale), pas de preces et d'orationes dans la section Vêpres de l'Office des morts

Doutes

- Dans VERSAILLES, Bibliothèque municipale, 1688 (141 M) (Reproduction intégrale) p.230, section avec en rubrique “les quatre salves” et une prière “Salve regina” : à encoder comme une prière ?

HORAE | Interannotator agreement 2 / Volume VERSAILLES, Bibliothèque municipale, 1688 (141 M) (Reproduction intégrale) / Page p. 164 - 165

Polygon results BACK TO ELEMENT

Name	Type	Class	Actions
Section 0	Section	Preces	VIEW ELEMENT UPDATE DELETE
Dñe exaudi ora	Text		
Section 1	Section	Orationes	VIEW ELEMENT UPDATE DELETE
Dre?	Text		
Section 2	Section	Prayers	VIEW ELEMENT UPDATE DELETE
Salve regina mis	Text		

18/05

Ajout de nouvelles classes (cf. Suffrages, Passion selon saint Jean, etc.), mais certaines ne sont pas encore auto-complétées il me semble (cf. Lazarus). Vérification du corpus inter-annotateur terminé.

26/05

Utilisation de “Kaldi Horae Recognizer” pour transcription automatique (technique de machine learning).

10/06

Correction des annotations des calendriers avec comme type “text_line” et comme classe “Calendar”. Cf. Guide Teklia avec nouveaux types, nouvelles classes et nouvelle interface : https://gitlab.com/mlbonhomme/annotation_horae.

11/06

Annotation des éléments de la structure d'une page pour réentraîner le modèle avec comme types :

- Pages
- Miniatures
- Zones de texte
- Lignes de texte
- Initiales
- Éléments décoratifs

16/06

Continuation de la correction du volume 557 dans to_correct_1.

Peut-on ajouter la class ‘Calendar’ dans les types ‘text_line’ et ‘page’ ?

Souvent KL en début de page de calendrier : abréviation de “Kalendae”/”Calendes”, pour le premier jour de chaque mois.

The screenshot shows a digital interface for transcribing medieval manuscripts. On the left, a sidebar titled "Page France_Paris_Bibliothèque ..." lists "Text line 0" through "Text line 23". Each line number is preceded by a small green circle with a white icon. The main area displays a double-page spread of a medieval manuscript. The left page is mostly blank, while the right page contains dense handwritten text in two columns. A prominent green rectangular stamp is visible on the right page, containing text and a circular seal. The top of the interface has a navigation bar with the URL "HORAE | DLA Annotations / Volume 557 / Page France_Paris_Bibliothèque nationale de France_fr. 01874_6" and buttons for "Page n°153", "Display", and "Actions".

Réunion avec Teklia

Point sur ce qui a été fait : Annotations segments et sections de niveau 1-2-3 par IRHT (corpus interannotateurs + un corpus pour le Machine Learning). Résultats copiés dans Annotations du corpus HORAE 790. Affichage de la classe comme nom de l'élément du côté Teklia. Arrivée du zoom et interface de transcription dans Arkindex. Correction des id -> les liens entre les indexs de classes dans Arkindex et dans Heurist ont été faits à partir des id d'Heurist. Problèmes de performance car trop de requêtes sur la base de données donc installation d'un système en clusters (partitionnement des données sur plusieurs serveurs) pour obtenir plus de rapidité dans le temps de réponse.

Du côté IRHT, les annotations interannotateurs ont permis de produire une vérité terrain. **Une fois encodé en TEI, le catalogue de Leroquais permettra aussi de produire une vérité terrain sur les grandes structures des livres d'heures.** Aussi complément des annotations sur le calendrier pour affiner l'apprentissage machine au niveau des lignes.

Point sur ce qui va être fait : Ordonnancement des lignes sur la page. Les pages sont prises e, compte mais pas le nombre de colonnes, d'autant plus qu'on croise rarement une répartition en colonnes dans le corpus. Pour améliorer la performance du corpus ML (machine learning), ajouter des transcriptions sur des text lines d'un corpus large.

Stratégies d'évolution : Utiliser les psaumes des livres d'heures et ajouter une transcription, même imparfaite. Utiliser des algorithmes d'alignement pour la répartition sur la page. Faire un corpus d'environ 200 images de psaumes sur HORAE et aligner avec Transkribus. /!\ le manifeste IIIF peut s'importer en une seule fois avec Transkribus mais perte de l'id. De plus, les serveurs IIIF ne renvoient pas la taille d'image souhaitée donc décalage avec la détection de ligne.

En ce qui concerne la création du modèle linguistique, faire apprendre plusieurs fois la Bible comme des textes différents car beaucoup de références bibliques dans livres d'heures. Idée d'intégrer un dictionnaire latin -> si l'on risque de trop normaliser alors qu'il existe bcp d'abréviations, fournir un texte sans langue est absurde. Utilisation du **post-processing** pour faire détecter les anomalies à la machine. Il ne s'agit pas d'améliorer la transcription du texte mais sa détection.

Pour la segmentation, même si le texte est prioritaire, il peut être intéressant d'avoir les éléments visuels qui structurent la page. On peut essayer d'avoir texte + images sur ¼ du corpus pour mettre en avant la spécificité des livres d'heures.

Ajout des données de segmentation de manière incrémentale, par palier, afin que chaque valeur ajoutée apporte une amélioration sans dysfonctionnement.

Identifier ensuite les termes de référence pour le plagiat. Travail de similarités entre les phrases.

Bilan : Chez Teklia, mettre en ligne la nouvelle interface avec la transcription possible et la classe `not_identified`. Traiter en HTR les manuscrits, créer les niveaux 1-2-3 sur les 10 manuscrits annotés en niveau 4. Si des images avec des reliures sont indiquées, c'est pour que le modèle ne détecte pas de lignes sur ce type d'images.

Côté IRHT, préparation des corpus d'alignement. Comparaison des différents corpus interannotateurs pour extraire les différences et produire de la vérité terrain. Vérifier le réalignement.

Annotation des sections dans une page : rajout des classes `marges_miniatures`, `marges_ornement` et `mages_texte`.

22/06

Retour sur les calendriers inclus dans la classification des pages (avec les reliures, le ff. Blancs, etc.). Si l'on annote ligne par ligne la structure tabulaire des calendriers, on perd l'intelligence des données, car l'on ne se donne pas les moyens de créer un texte avec les structures stables suivantes :

- Nombre d'or (chiffre servant à déterminer les phases de la lune en calculant l'épacte)
- La lettre dominicale (Un système de représentation des jours de la semaine consiste à leur attribuer une lettre au lieu des traditionnelles appellations lundi, mardi, etc. Ces lettres sont attribuées pour une année donnée. Pour une année donnée, on fait correspondre successivement chacune des sept premières lettres (A, B, C, D, E, F et G) à chacun des jours de l'année, en commençant par A pour le 1er janvier, puis en répétant le cycle tous les sept jours.)
- Le n° d'ordre dans les mois
- Les indications liturgiques -> **à creuser**
- Les indications accessoires -> **à creuser**

Il faudrait donc traiter différemment les calendriers en créant une fonctionnalité d'annotation nouvelle, soit un tableau à 4 colonnes (**pour les informations de structuration?**) puis diviser les lignes. Pour éviter les fusions de lignes par la machine, les séparer de quelques pixels par un frame. On peut alors comparer la vérité terrain établie par **Aaron Macks**.

23/06

Lien vers une base de données relationnelle de Aaron Macks avec un corpus de calendriers de livres d'heures : <http://www.cokldb.org/>. Nom du projet : The *Corpus Kalendarium* (CoKL). Intéressant pour voir ce que l'on peut faire pour avoir une visualisation des données ->

possible de rechercher des saints par date et par nom et de voir dans quels livres d'heures ils sont mentionnés ; de comparer les calendriers de divers manuscrits en mettant en regard les usages, les langues, la foliation, les transcriptions de calendriers, etc.

20/07

Poursuite de l'établissement d'une vérité terrain pour les psaumes pénitentiels pour améliorer la reconnaissance de la division des psaumes : cf. Ajout de 801p. sur Transkribus, manuscrit probablement du XVe car format de pages allongé et iconographie de David en prière ou regardant Béthsabée au bain.

23/07

Réunion avec Teklia :

- Import d'images dans Transkribus (psaumes pénitentiels) pour augmenter la vérité terrain en HTR et faire un corpus de réentraînement en plus de l'*Obsecro te*. Avec Kaldi Horae, calcul du taux d'erreurs sur 247 l. d'*Obsecro te* et environ 40-48% de vocabulaire commun pour les 10 manuscrits utilisés (corpus interannotateur ?). Pour l'alignement sur Transkribus, faire une liste des pages mal-alignées.
- Il existe parfois des différences entre le texte imposé et le texte d'origine. Ce n'est pas une transcription parfaite, il s'agit d'intégrer du bruit à partir de la **transcription de Jacob**.
- Possible de faire une estimation des mots coupés en fin de ligne : si les lignes sont globalmeent courtes dans les livres d'heures (moins de 7 mots par l. en moyenne), il reste un manuscrit de luxe pour les gens non-spécialistes de la lecture, donc les scribes évitent si possible de couper les mots. Fusionner les fins et les débuts de l. pour améliorer la reconnaissance de vocabulaire.
- Amélioration de l'HTR sur 50 manuscrits :
 - Correction des annotations d'analyse de pages (lignes de texte, calendriers, reliure et images sans texte)
 - Classes de pages à valider
 - Baisse du temps de traitement car téléchargement des images en plus basse résolution, mais cela détériore la performance de segmentation des lignes. Système sensible à l'algorithme de sous-résolution, donc il faut améliorer le modèle avec cet algorithme.
 - Réduction du temps d'écriture dans arkindex car mise ne place d'un cluster (regroupement) postgresql. Cela implique la mise à jour des données à partir d'une certaine fréquence. Plusieurs serveurs traitent les requêtes.
- Important de détecter les doubles et les simples pages.

- 45 à 50% du contenu de la Bible est présent dans les 10 livres d'heures analysés. Sachant que 1/3 du livre d'heures est composé de psaumes.
- Question des couleurs : entraîner le modèle sur les rubriques avec les classes “red”, “blue”, “gold” et “green”. Analyser la variation de pixels au sein d'un même manuscrit quand il n'est pas en noir et blanc.
- Question des calendriers avec annotation tabulaire.

28/07

Pour Transkribus, signaler quand les lignes ne sont pas assez hautes mais ne pas les corriger systématiquement et aligner le texte tout de même. Identifier les forces et les faiblesses de la machine pour gérer les projets, trouver des solutions pour pallier les problèmes.

Le logiciel Transkribus peut être sensible à la couleur de l'encre et mal distinguer les lignes si cette couleur est semblable à celle du parchemin.

[4.1.1 Rubriques](#)

[4.1.2 Niveaux 1-2-3](#)

[4.1.3 Type de pages](#)

[4.1.4 Type de zone](#)

[4.1.5 Textes](#)

5 Ergonomie avec Oxygen XML

5.1 Principes

Trois façons de contraindre Oxygen et d'aider l'utilisateur :

- Css (feuille de style en cascade)
- Framework (infrastructure logicielle)
- cc_config : permet notamment de régler les menus déroulants (configuration électronique)

La CSS est la plus légère, mais pas optimale pour des opérations complexes allant chercher des informations directement dans des documents extérieurs. En effet, la CSS fonctionne en permanence, donc le logiciel devient très lent. Le ralentissement est dû principalement au nombre d'éléments vérifiés et non à la taille du document de référence extérieur (parcé en xml, consommant peu d'énergie).

Le framework permet de définir des actions enchaînées les unes aux autres.

Attention, les syntaxes de CSS et framework se ressemblent beaucoup, notamment pour « string-join » et les appels à des documents, mais la gestion des guillemets simples et doubles n'est pas toujours la même.

28/05

Réunion au sujet de l'ergonomie de la saisie d'une bibliographie:

Mieux vaut faire une numérotation automatique, surtout en cas de rajouts?

<filiation> peut être utile pour englober plusieurs <msItem>.

Utilisation de CSS, actions Oxygen et config pour simplifier l'action utilisateur (restreindre les valeurs d'un attribut par exemple, appeler l'auteur dont on a besoin en auto-complétion avec liste défilante par ordre alphabétique, etc.).

La gestion de projet, c'est coordonner les tâches de l'équipe. Méthode agile : diviser les problèmes et y revenir au moment voulu. Prioriser les tâches.

29/05

Cf. Telota (*The Electronic Life of the Academy*) à l'Université des Sciences de Berlin et le projet *ediarum*. Il s'agit d'encoder les manuscrits en XML-TEI via un mode plus convivial, élaboré grâce à la CSS. La base de données XML est en ligne, sur existdb, ce qui simplifie la collaboration. L'index est centralisé dans la base de données et peut être mis à jour. Le site met à disposition l'inventaire des données actuelles. Il est créé à partir d'existdb, de XQuery et de XSLT.

La CSS et le framework Oxygen, dans lequel on peut créer des actions enchaînées, se complètent.

5.2 CSS

Voir aussi les thread du forum oxygen, notamment

- <https://www.oxygenxml.com/forum/topic19849.html>

Idée de créer une fonction qui permette, à la saisie d'une donnée, de créer un menu déroulant de références provenant d'un document extérieur. Attention à l'utilisation de guillemets et d'apostrophes autour de la variable et du chemin de fichier. Pour qu'une boîte de dialogue édite plusieurs entrées, faire une condition avec la fonction *position()*. Amélioration de Oxygen 21.1 pour augmenter les possibilités de traiter les variables et de les insérer.

- <https://www.oxygenxml.com/forum/topic19894.html>

Syntaxe avec string_join() en CSS : comment faire pour que les éléments du menu-déroulant soit inséré avec les fonctions concat() ou normalize-space() ? Attention, les combo boxes éditables ne peuvent rendre que les valeurs définies dans la CSS, sinon l'utilisateur pourrait saisir de nouvelles données ajoutées au modèle. Travailler avec des valeurs définies dans des menus déroulants éditables est risqué. Il vaut mieux dans ce cas utiliser des combos-boxes non-éditables. Il est toutefois possible avec les frameworks d'implémenter des formulaires de contrôle personnalisés.

5.2.1 Liste déroulante fermée

Documentation sur variables : <https://www.oxygenxml.com/doc/versions/22.1/ug-editor/topics/editor-variables.html>

Documentation sur oxy_combobox : <https://www.oxygenxml.com/doc/versions/22.0/ug-authorEclipse/topics/combo-box-editor.html>

Par exemple :

```
scriptNote{
    display: block;
    font-weight: bold;
    content: "Ecriture : " oxy_combobox(edit, "@script",
        editable, false,
        values, "caroline, littera textualis, littera textualis meridionalis, littera semitextualis,
        littera cursiva, littera hybrida, littera semihybrida, humanistique, humanistique cursive, hors
        système", labels, "caroline, littera textualis, littera textualis meridionalis, littera semitextualis,
        littera cursiva, littera hybrida, littera semihybrida, humanistique, humanistique cursive, hors
        système", width, 150px)
        oxy_combobox(edit, "@rend", editable, false, values, "formata, libraria, currens",
        width, 150px);
}
```

5.2.2 Recherche d'information dans le document

Exemple : utilisation de <listPlace> en <sourceDesc> pour l'index des noms de lieux

Documentation pour oxy_popup : <https://www.oxygenxml.com/doc/versions/22.1/ug-editor/topics/pop-up-editor.html>

CSS

```

p > placeName:after {
    content: " index"
    oxy_popup(edit, '@ref',
    values,
    oxy_xpath('string-join(//listPlace/place/@xml:id, ",")'),
    selectionMode, single,
    labels,
    oxy_xpath('string-join(//listPlace/place/placeName[1], ",")')
)
}

```

XML

```

<p><placeName ref="d6e10983">Aisey</placeName></p>

```

Vue CSS

```

▷ Aisey index Aisey-sur-Seine ◁

```

Liste référentielle à partir de l'index initial

```

▷▷ Aisey index Aisey-sur-Seine ◁

```

Aignay-le-Duc
Aisey-sur-Seine
Albon
Alise-Sainte-Reine
Alneto

5.2.3 Recherche d'information

```

bibl[sameAs]{
display:block;
content:oxy_xpath(
oxy_concat(
"doc('${cfdu}/biblio-saint-omer.xml')//listBibl/biblStruct[contains(@xml:id, '',
attr(sameAs,''))]/normalize-space(string())"
)
);
}

bibl:before{
content: "ID"

```

```

oxy_combobox(edit, '@sameAs',
values,
oxy_xpath(
  oxy_concat(
    'string-join(doc("", oxy_url(${cfdu}/biblio-saint-omer.xml'),
")//listBibl/biblStruct[contains(@xml:id, substring-after("",attr(sameAs),"","#))]/@xml:id,
",
"))
),
editable, true,
labels,
oxy_xpath(
  oxy_concat(
    'string-join(doc("", oxy_url(${cfdu}/biblio-saint-omer.xml'),
")//listBibl/biblStruct[contains(@xml:id, substring-after("",attr(sameAs),"","#))]/@xml:id,
",
")
))
)
}
}

```

Les deux fonctions ont les effets suivants :

- si <bibl> a un @sameAs, elle cherche dans le document **biblio-saint-omer.xml** le contenu du <biblStruct>
- <bibl> est précédé par une boîte de choix qui permet d'éditer l'attribut @sameAs et qui donne la liste de tous les identifiants @xml:id de la biblio (toujours le même fichier **biblio-saint-omer.xml**)

NOTA : les fonctions utilisent « contains » et « substring-after » (pour gérer #) au lieu de l'égalité, cela peut avoir des incidences négatives. En effet, utiliser les fonctions contains() et substring-after() ne permet pas de récupérer précisément une valeur. Ils sélectionnent toutes les publications avec un identifiant qui peut être suivi ou précédé de n'importe quelle chaîne de caractère. En Xpath et en CSS, la condition d'égalité peut être indiqué avec “=” pour une égalité stricte, “~=” pour que la chaîne de caractère contient tel mot, etc. Cf. https://www.w3schools.com/cssref/css_selectors.asp.

Comme ces deux affichages sont permanents, le logiciel devient très lent, mais ça marche.

La modification de ce matin (28/05/2020, ajout de #Leclercq1957 en @sameAs

▽ ▷ 132ra-132rb ▷ ***Epistola 186*** ▷ ***Ad Symonem filium castellani de Oisy. xciii*** ▷ ***C***
 ▷ ***et omnes qui ad uos pertinent bene ualere*** ▷
ID #Leclercq1957 **VIII, p. 8-9**

ou, ici, sans le # (cf. la CSS fautive), on affiche le contenu du <biblStruct> dans le fichier de bibliographie

► 134va-139r ► Epistola 190. Ad Innocentium de erroribus Abaelardi ► Ad eundum contra quedam capitula illa heresum. cccviii

▼ Salutatio ▼ Amanissimmo patri et domino Innocentio summo pontifici, frater B. Clareullus uocatus abbas, modicum id quo est... -... ▾

▼ Textus ▾ Oportet ad uestrum referri apostolatum pericula.... ▾ Hec capitula partim in libro theologie, partim in libro sententiarum magistri Petri, partim in libro cuius titulus est Scito te ipsum reperta sunt. ▾ bibl

ID Ledercq1957 ▾ Sancti Bernardi opera SBO JeanLeclercq HenriRochais Rome 1957VIII, p. 17-40.

Cette recension des capitula, deux fois révisée, est moins proche de l'originale que la version plus ancienne, présente dans La Haye BR 70 H (voir la notice), autre manuscrit de Clairmarais (Ms. p. 105-106).

b12ab_12Rue ► Bernardus ► Epistola 190. Ad Gildonem abbatem Armentorium ► Inquit apostola dominus Bernardus abbatu[m] ad Milanum

La boîte permet de rechercher les identifiants



Documentation oxy_url : <https://www.oxygenxml.com/doc/versions/22.1/ug-editor/topics/dg-url-function.html>

Documentation pour oxy-button : <https://www.oxygenxml.com/doc/versions/22.1/ug-editor/topics/button-editor.html>

5.2.4 Interface

Nota : On améliorera la présentation. : on peut afficher les identifiants ou n'importe quoi d'autre, cf. paramètres de la fonction et string-join

5.2.5 Actions

Documentation actions pour un framework :

<https://www.oxygenxml.com/doc/versions/22.1/ug-editor/topics/dg-create-custom-actions.html>

La CSS peut appeler des actions définies dans le « framework ». Ici, on appelle les actions « inserTEI_msItem' » et « inserTEI_watermark » .

```

4527 </action>
4528 <action>
4529   <field name="id">
4530     <string>inserTET_msItem</String>
4531   </field>
4532   <field name="name">
4533     <string>msItem</String>
4534   </field>
4535   <field name="description">
4536     <string>Ajout d'un texte</String>
4537   </field>
4538   <field name="largeIconPath">
4539     <string>${framework}/icons/list-add.png</String>
4540   </field>
4541   <field name="smallIconPath">
4542     <string>${framework}/icons/list-add.png</String>
4543   </field>
4544   <field name="accessKey">
4545     <string></String>
4546   </field>
4547   <field name="accelerator">
4548     <null/>
4549   </field>
4550   <field name="actionModes">
4551     <actionMode-array>
4552       <actionMode>
4553         <field name="xpathCondition">
4554           <string>/</String>
4555         </field>
4556         <field name="argValues">
4557           <serializableOrderedMap>
4558             <entry>
4559               <String>fragment</String>
4560               <String>&lt;msitem xmlns="http://www.tei-c.org/ns/1.0">
4561                 &lt;locus>&lt;/locus>
4562               </String>
4563             </entry>
4564             <entry>
4565               <String>insertLocation</String>
4566               <string>.</String>
4567             </entry>
4568           </serializableOrderedMap>
4569         </field>
4570       </actionMode>
4571     </actionMode-array>
4572   </field>
4573   <field name="enabledInReadOnlyContext">
4574     <boolean>false</Boolean>
4575   </field>
4576 </action>
4577 <action>
4578   <field name="id">
4579     <string>insertTET_watermark</String>
4580   </field>
4581   <field name="name">
4582     <string>liste</String>
4583   </field>
4584   <field name="description">
4585     <string>Ajout d'un filigrane</String>
4586   </field>
4587   <field name="largeIconPath">
4588     <string>${framework}/icons/list-add.png</String>
4589   </field>
4590   <field name="smallIconPath">
4591     <string>${framework}/icons/list-add.png</String>
4592   </field>
4593   <field name="accessKey">
4594     <string></String>
4595   </field>
4596   <field name="actionModes">
4597     <actionMode-array>
4598       <actionMode>
4599         <field name="xpathCondition">

```

```

4496 </field>
4497 <field name="actionModes">
4498   <actionMode>array</actionMode>
4499     <field name="xpathCondition">
4500       <String></String>
4501     </field>
4502     <field name="argValues">
4503       <serializableOrderedMap>
4504         <entry>
4505           <String>fragment</String>
4506           <String>&lt;watermark xmlns="http://www.tei-c.org/ns/1.0"&gt;&lt;locus&gt;&lt;dimensions>&lt;idno/&gt;&lt;/dimensions&gt;&lt;/watermark></String>
4507         </entry>
4508         <entry>
4509           <String>insertLocation</String>
4510           <String>.</String>
4511         </entry>
4512         <entry>
4513           <String>insertPosition</String>
4514           <String>After</String>
4515         </entry>
4516       </serializableOrderedMap>
4517     </field>
4518     <field name="operationID">
4519       <String>ro.sync.ecss.extensions.common.operations.InsertFragmentOperation</String>
4520     </field>
4521   </actionMode>
4522 </actionMode>array</field>
4523 </actionMode>array</field>
4524 <field name="enabledInReadOnlyContext">
4525   <Boolean>false</Boolean>
4526 </field>
4527 </action>
4528

```

Les deux premières mises en forme appellent la même fonction (ajouter un msItem), mais à deux endroits différents, soit pour ajouter un <msItem> autonome, soit pour ajouter un <msItem> à l'intérieur d'un <msItem>

Syntaxe :

```

summary:after, msItem:last-of-type:after{
  display:block;
  color:#84bff9;
  font-size:x-small;
  content:"Ajouter un texte" oxy_button(transparent, true,
  action, oxy_action(
    id, 'inserTEI_msItem'));
}

msContents > msItem > *:last-child:after{
  display:inline;
  color:#84bff9;
  font-size:x-small;
  content:
  "Ajouter une sous-partie" oxy_button(transparent, true,
  action, oxy_action(
    id, 'inserTEI_msItem'));
}

supportDesc > support > material:after{
  content:"Décrire les filigranes" oxy_button(transparent, true,
  action, oxy_action(
    id, 'inserTEI_watermark'));
  color:#84bff9;
  font-size:x-small;
}

```

}

Documentation vers oxy_button : <https://www.oxygenxml.com/doc/versions/22.1/ug-editor/topics/button-editor.html>

Documentation vers oxy_action : <https://www.oxygenxml.com/doc/versions/22.0/ug-editor/topics/dg-action-function.html>

5.3 Framework et actions

Lire aussi : <https://www.oxygenxml.com/forum/topic12855.html>

Les actions peuvent être personnalisées soit dans le *Document Type Configuration Dialog Box* d’Oxygen, soit dans un fichier externe qui définit de nouvelles actions. L’intérêt de cette dernière méthode est, d’une part, de partager, copier ou réutiliser une action individuelle pour de multiples projets ou frameworks, d’autre part, de faciliter le développement et de tester la configuration de l’action jusqu’à atteindre le résultat souhaité.

5.3.1 Créer des actions

Google et l’aide d’Oxygen sont vos amis

5.3.2 Syntaxe (une fois créé)

5.4 Mise en oeuvre

5.4.1 Vérifier réactivité du logiciel

Changer la recherche extérieure pour favoriser un import unique de la biblio et une recherche à l’intérieur du document.

Pour un import unique de la bibliographie avec la recherche extérieure, récupérer tous les <bibl> enlevant le @sameAs. Pour faire de même avec une recherche interne au document, ajouter un <xi:include> :

```
<sourceDesc>
  <p>
    <xi:include href="biblio-saint-omer.xml" xmlns:xi="http://www.w3.org/2001/XInclude">
      <xi:fallback>biblio-saint-omer.xml appears to be missing</xi:fallback>
    </xi:include>
    <listPlace>
      <place xml:id="d6e10977">
```

Il n'est ainsi plus besoin de mettre le chemin de fichier. Cette solution est plus rapide car la recherche se fait directement à l'intérieur du document.

Comment associer une feuille CSS aux actions définies dans le framework ? Ou un framework à un XML ?

02/06

On peut invoquer une action référencée par son identifiant, si elle est bien définie dans le Document Type associé, ou bien on peut l'invoquer directement dans la CSS grâce à la fonction oxy_action() avec les @actionID ou @action.

Pour que la CSS puisse reconnaître un nouveau framework, le définir et le déclarer dans les paramètres d'Oxygen.

!\\ Ce que l'on appelle de l'automatisation est souvent dans les faits de la semi-automatisation, car la machine a besoin de préparation de données qui demande un certain temps, tout d'heures de programmation et d'optimisation de paramètres.

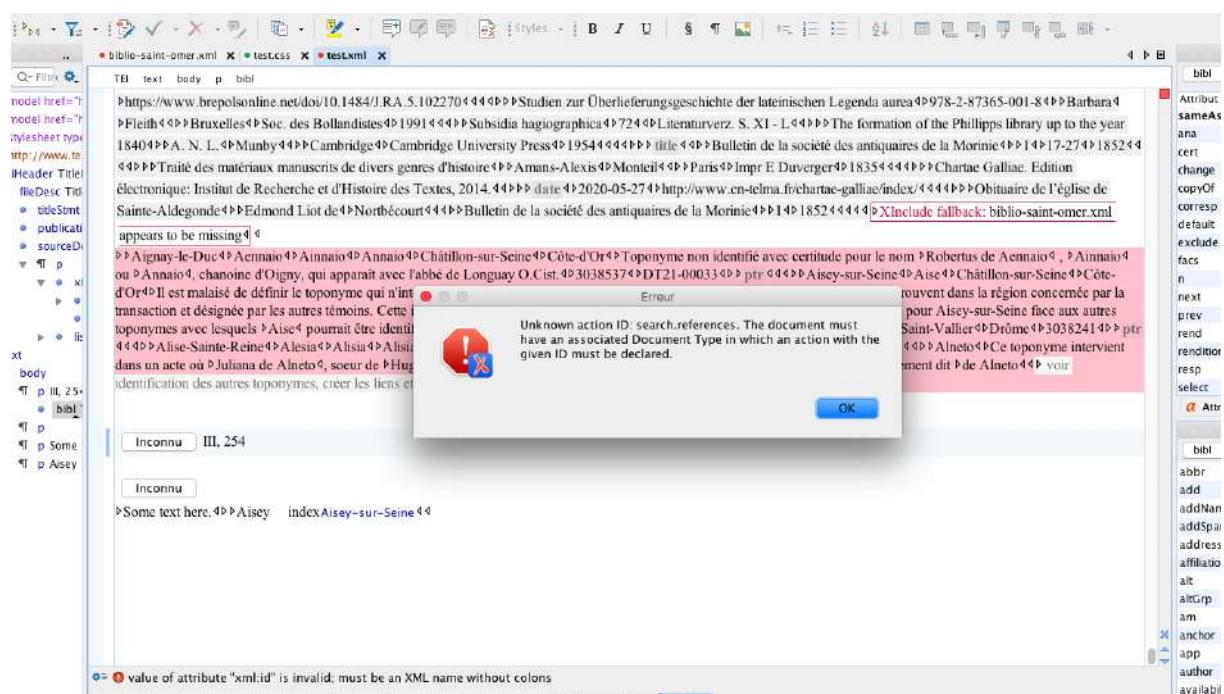
5.4.2 CSS+action 1

CSS

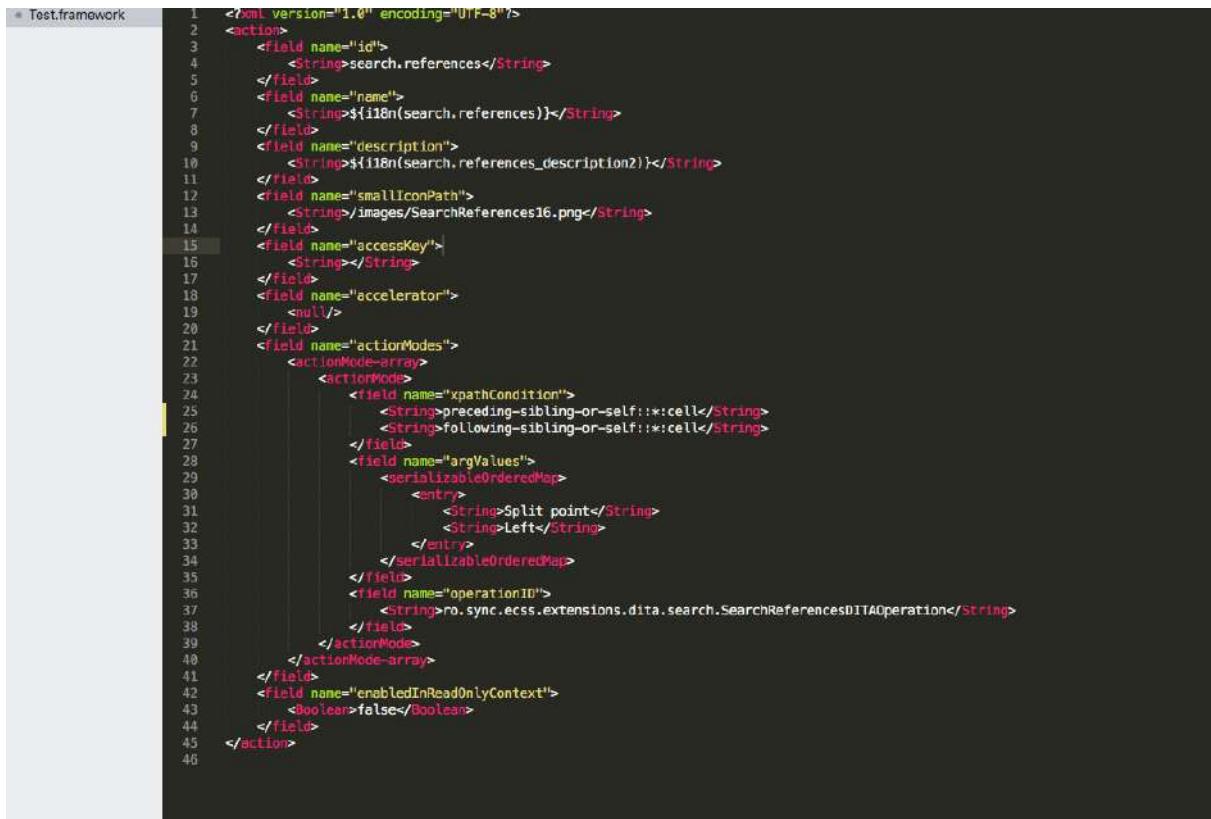
- affiche l'identifiant contenu en @sameAs
- affiche un bouton « Voir référence bibliographique complète »

```
<sourceDesc>
  <p>
    <xi:include href="biblio-saint-omer.xml" xmlns:xi="http://www.w3.org/2001/XInclude">
      <xi:fallback>biblio-saint-omer.xml appears to be missing</xi:fallback>
    </xi:include>
    <listPlace>
      <place xml:id="d6e10977">
```

Tentatives de créer un bouton en mode auteur, dans un premier temps directement dans le CSS selon une action déjà définie, mais l'xml ne l'a pas reconnue :

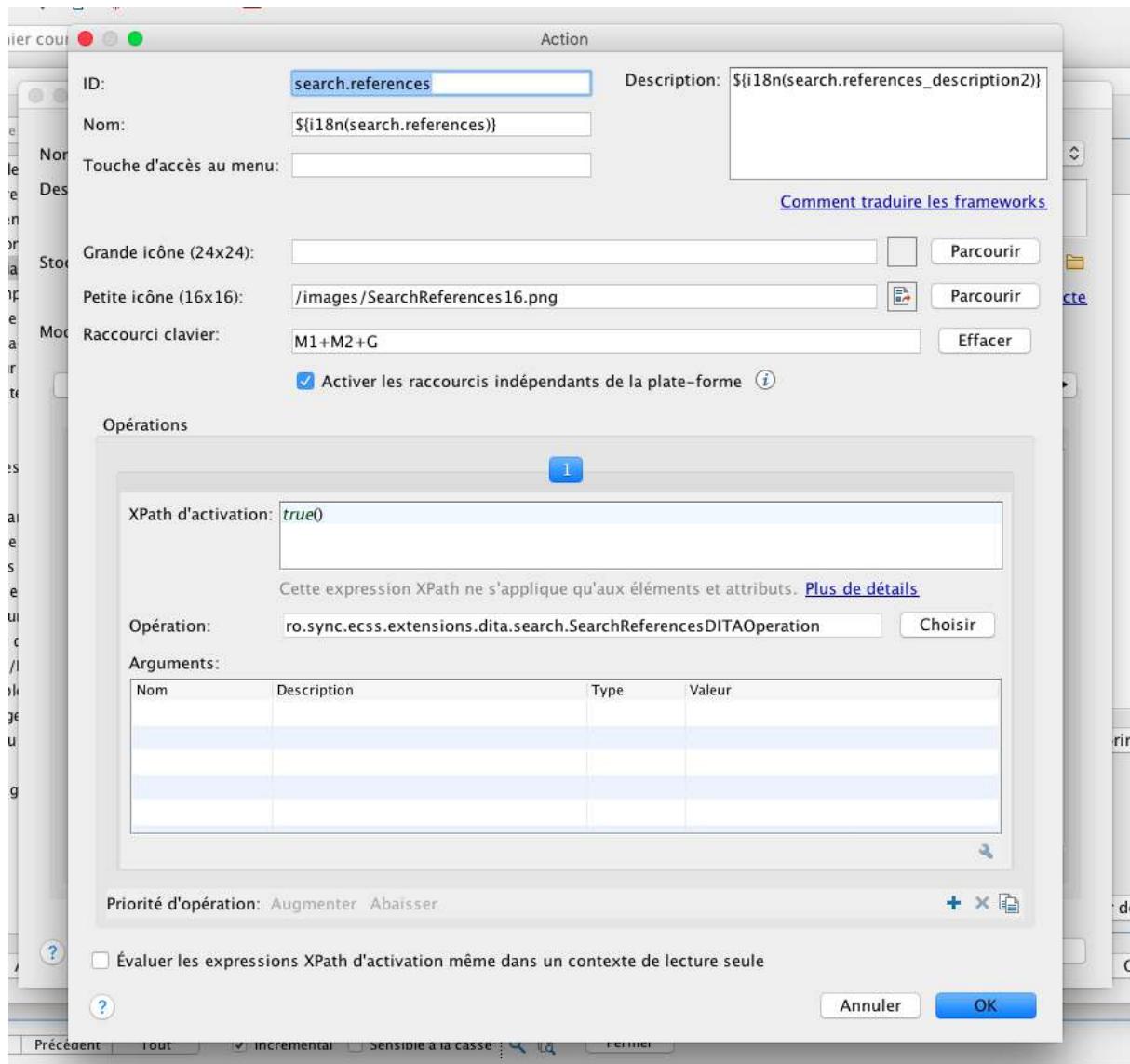


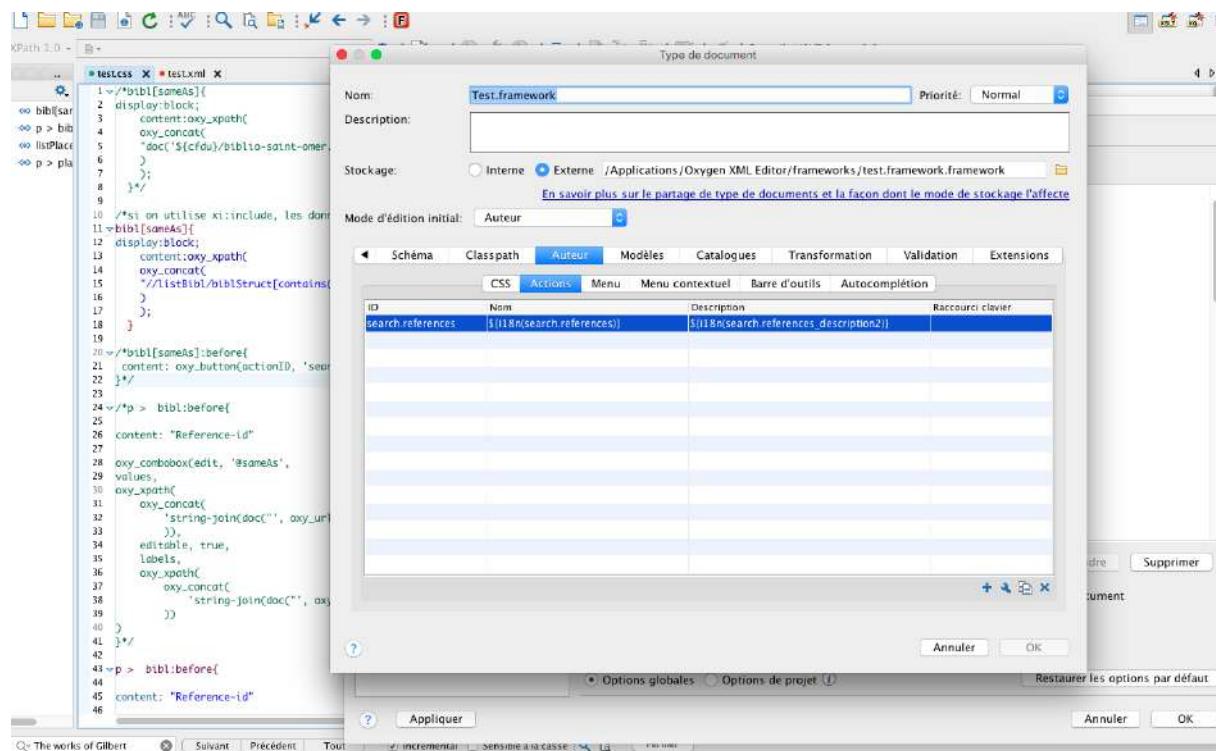
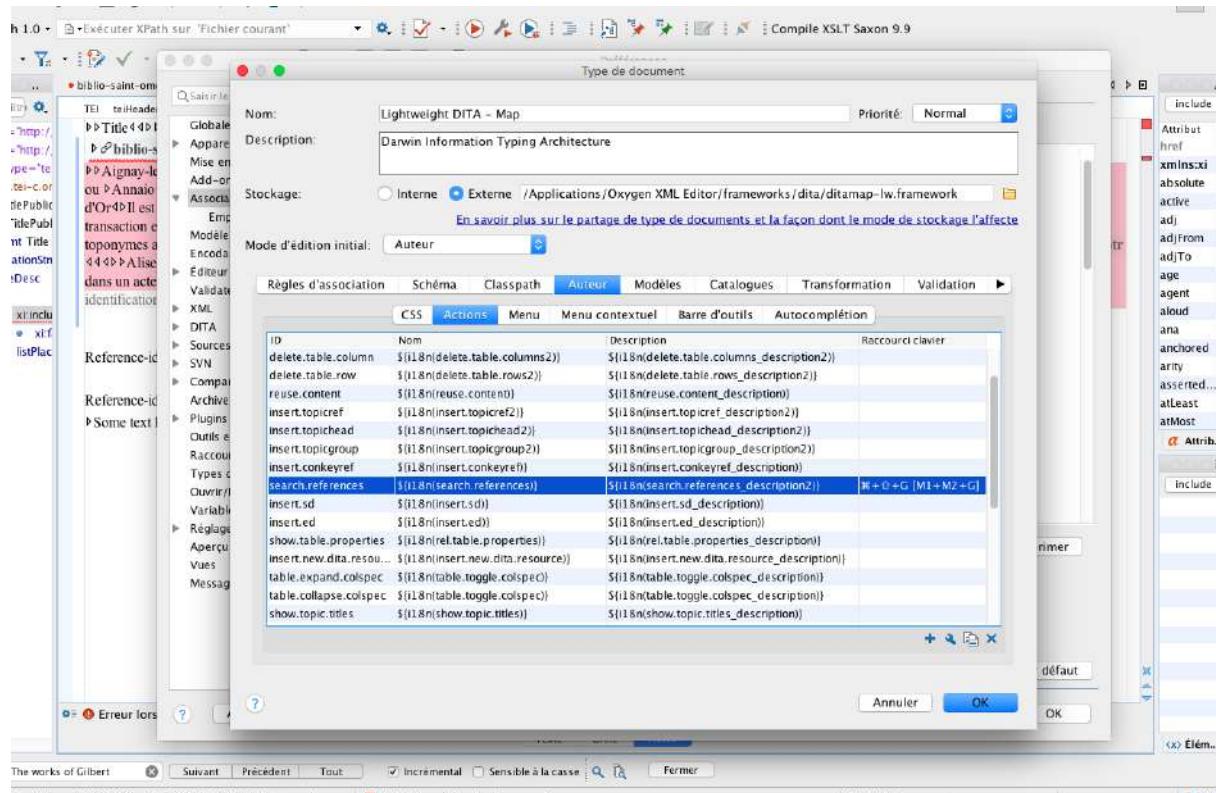
Donc deuxième tentative de créer l'action directement dans les préférences d'Oxygen :

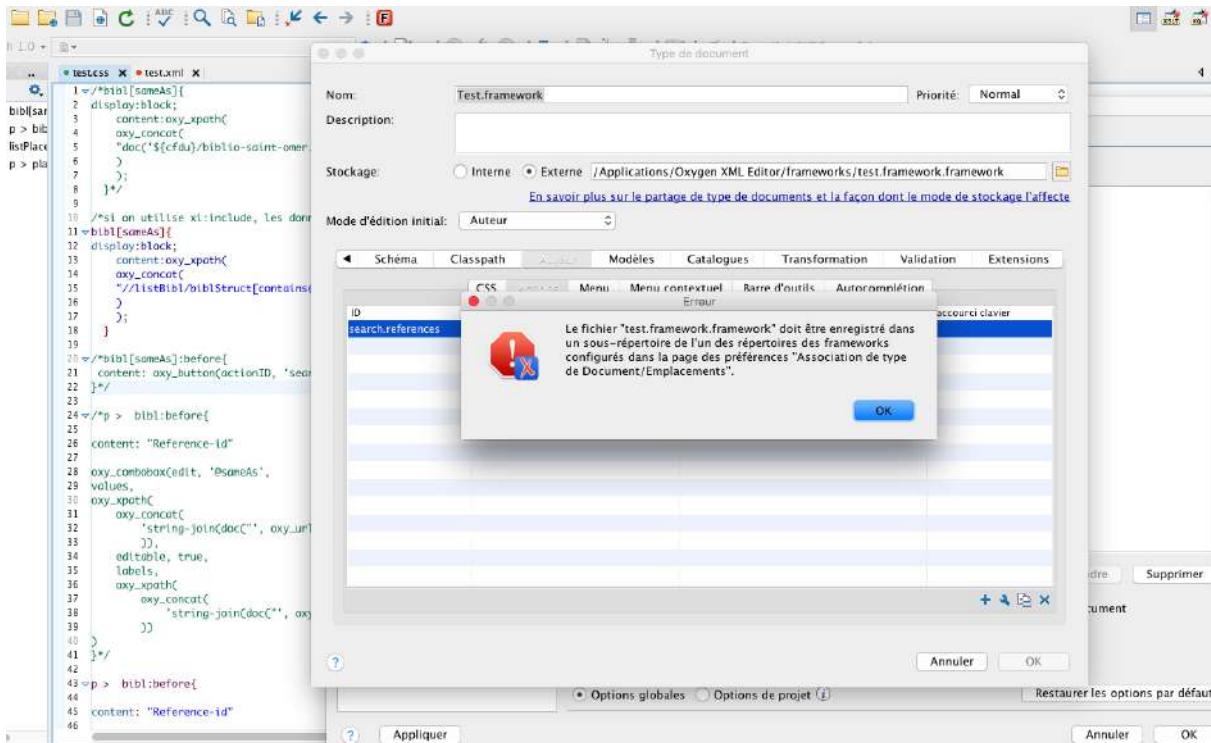


The screenshot shows a code editor window for an XML file named 'Test.framework'. The XML code defines an 'action' element with various fields. Notable fields include 'name' set to 'search.references', 'description' set to a localized string, 'smallIconPath' pointing to a PNG file, and 'actionModes' which contains an array of 'actionMode' elements. Each 'actionMode' has an 'xpathCondition' field and an 'argValues' field containing a 'serializableOrderedMap' with entries for 'Split point' and 'Left'. The 'operationID' field is set to 'ro.sync.ecss.extensions.dita.search.SearchReferencesDITAOperation'. The 'enabledInReadOnlyContext' field is set to 'false'. The code is numbered from 1 to 46.

```
<?xml version="1.0" encoding="UTF-8"?>
<action>
    <field name="id">
        <string>search.references</String>
    </field>
    <field name="name">
        <String>${i18n(search.references)}</String>
    </field>
    <field name="description">
        <String>${i18n(search.references_description2)}</String>
    </field>
    <field name="smallIconPath">
        <String>/images/SearchReferences16.png</String>
    </field>
    <field name="accessKey">
        <String></String>
    </field>
    <field name="accelerator">
        <null/>
    </field>
    <field name="actionModes">
        <actionMode-array>
            <actionMode>
                <field name="xpathCondition">
                    <String>preceding-sibling-or-self::*:cell</String>
                <String>following-sibling-or-self::*:cell</String>
                </field>
                <field name="argValues">
                    <serializableOrderedMap>
                        <entry>
                            <String>Split point</String>
                            <String>Left</String>
                        </entry>
                    </serializableOrderedMap>
                </field>
                <field name="operationID">
                    <String>ro.sync.ecss.extensions.dita.search.SearchReferencesDITAOperation</String>
                </field>
            </actionMode>
        </actionMode-array>
    </field>
    <field name="enabledInReadOnlyContext">
        <Boolean>false</Boolean>
    </field>
</action>
```







Mais échec de l'enregistrement : dans quel répertoire ?

Association d'un bouton à une action à approfondir.

03/06

Dans la pratique, mieux vaut modifier un framework qui existe déjà plutôt que d'en créer un. L'espace de stockage correspond à un emplacement déclaré où se trouve le dossier avec le framework.

Il faut avant tout déclarer l'environnement dans les Préférences, Association de Type de Document et Emplacement. On peut ensuite modifier et appliquer l'action.

Certaines actions sont spécifiques au format DITA (Darwin Information Typing Architecture), d'autres au format TEI, donc utiliser celles disponibles au format TEI.

Tests d'actions définies dans le framework TEI-MsDesc, mais attention, la plupart des actions insèrent de nouvelles balises dans le document, d'où le fait que l'on peut se demander s'il faudrait modifier l'action ajout d'un paragraphe dans laquelle on ajoute une balise <bibl> avec l'attribut @sameAs dans laquelle il y aura la référence bibliographique complète.

Il faut que le bouton appraise après la boîte de dialogue Référence-id, comme ce qui suit :

Il faut utiliser l'attribut `@sameAs` pour faire apparaître une boîte de dialogue avec la référence complète. Utiliser le paramètre `$ask` dans la boîte de dialogue pour afficher et faire afficher la référence entière comme une confirmation de `@sameAs` : le faire directement dans la CSS ou dans le framework (cf.action `InserBiblio`).

Différents tests sur l'action `InserBiblio` du framework TEI-MsDesc pour que le bouton permette d'afficher la référence :

Screenshot of the Oxygen XML Editor interface showing two error dialogs related to XSLT processing.

The top window displays an XSLT transformation from `biblio-saint-omer.xml` to `test_GP.css`. The transformation includes a template rule for `TEI text body p bibl` which handles the `sameAs` attribute. The error message is:

```

Une erreur s'est produite lors de l'expansion de la variable
éiteur 'ask', à la ligne 4, colonne 77. Veuillez vérifier la
syntaxe de la variable éiteur : '$ask'.
'Recherche de référence bibliographique',
generic,
'string-join(/listBib/biblStruct[contains(@xml:id,
substring-after(".", attr(@sameAs)), "#")]/@xml:id, ".")'
'.

```

The bottom window shows a similar setup, but with a different error message:

```

Une erreur s'est produite lors de l'expansion de la variable
éiteur 'ask', à la ligne 5, colonne 3. Veuillez vérifier la syntaxe
de la variable éiteur : '$ask'.
'Recherche de référence bibliographique',
generic,
'xpath_eval(@sameAs)'
'.

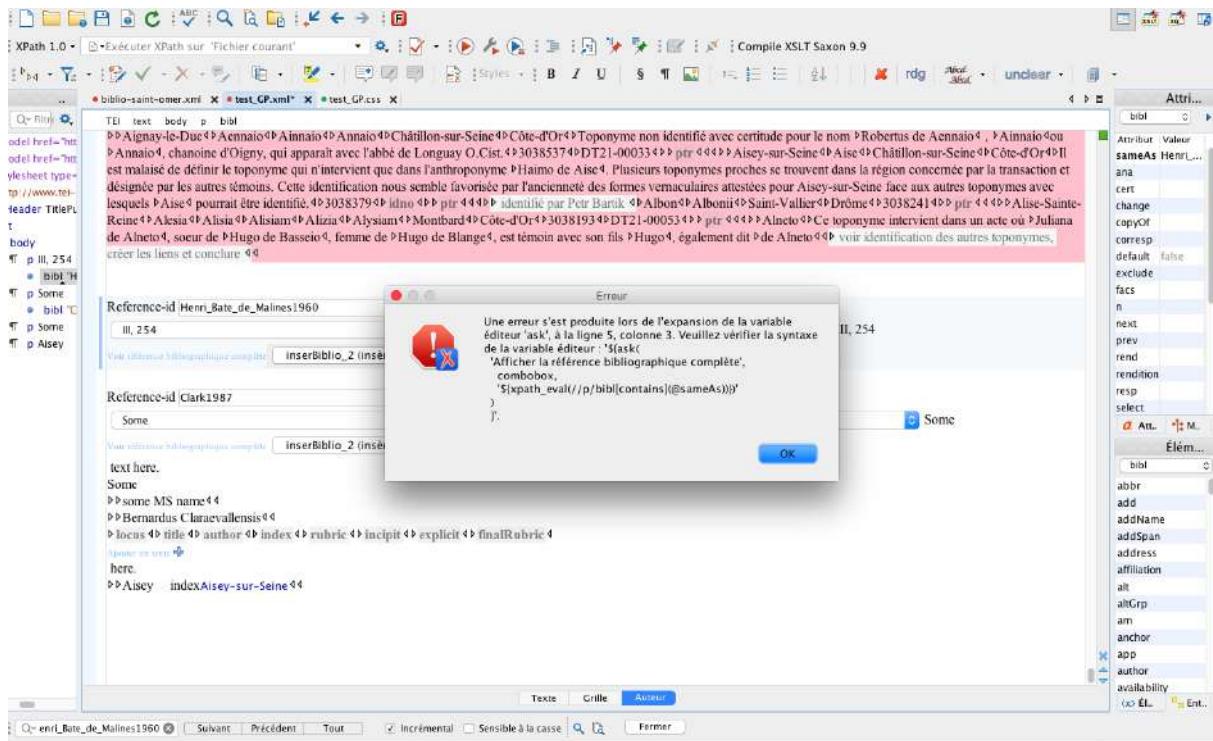
```

Screenshot of the Oxygen XML Editor showing two identical error dialogs. The top dialog is for the file 'test_GP.xml' and the bottom one is for 'test_GP.css'. Both dialogs are reporting the same error:

```

Une erreur s'est produite lors de l'expansion de la variable
éditeur 'ask', à la ligne 5, colonne 3. Veuillez vérifier la syntaxe
de la variable éditeur : '$ask'
'Recherche de référence bibliographique',
generic,
$xpath_eval{//listBib/bibStruct[contains(@xml:id,",
@sameAs,$)]}'
```

The error message is displayed in a red box at the top of the editor interface. The code causing the error is highlighted in the main text area.



La boîte de dialogue de la Référence-bibliographique est bien reliée à celle de Référence-id, mais toujours le problème sur l'action du bouton -> **jouer avec les variables \$ask et \$answer pour qu'elles renvoient aux bonnes boîtes de dialogues ? Le faire dans les paramétrages de l'action (framework) ou dans la CSS ?**

04/06

CSS pour rendre la valeur de l'attribut @sameAs cliquable et refermable, donc manière d'afficher puis de masquer une information :

```
p > bibl:before{
content: "id: "
oxy_popup(      edit, '@sameAs',
values,      oxy_xpath(
oxy_concat(          'string-join(//listBibl/biblStruct[@xml:id =
",attr(sameAs),""]/@xml:id, ",")'
)),
labels,      oxy_xpath(
oxy_concat(          'string-join(//listBibl/biblStruct[@xml:id =
",attr(sameAs),""]/title, ",")'
resultSeparator, "|",
columns, 10,
selectionMode, "multiple",
color, "blue",
```

```
    fontInherit, true);  
    font-size:15px;  
}
```

CSS pour créer un bouton ouvrant un pop-up affichant l'identifiant contenu dans le même attribut @sameAs grâce à l'action display_biblStruct et au paramètre \${ask('Hello world!', generic, '\${xpath_eval("./@sameAs)}')} :

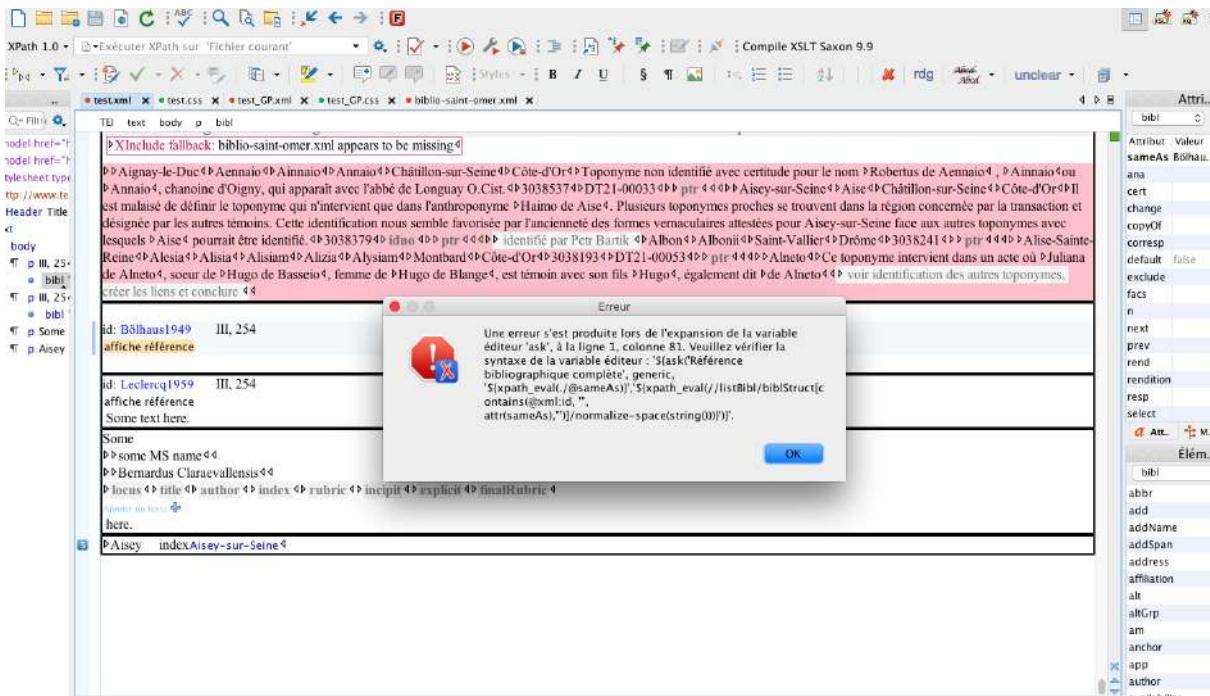
```
bibl[sameAs]:after{  
display:block;  
color:#84bff9;  
font-size:x-small;  
content: oxy_button(transparent, true, action,  
oxy_action(      id, 'display_biblStruct'));
```

Pour afficher la valeur entière (la référence bibliographique complète) et garder l'attribut @sameAs, faire une boîte de dialogue avec en parallèle une valeur et un label, donc il faut dans un premier temps afficher le contenu dans un dialogue “generic” puis régler l'accord value/label. Dans l'idéal, faire un xpath_eval à l'intérieur d'un xpath_eval, l'un pour vérifier que l'@sameAs de <bibl> est équivalent à l'@xml:id de <biblStruct>

/!\ Les guillemets sont inversés entre la CSS et le paramétrage de l'action.

Essai 1 :

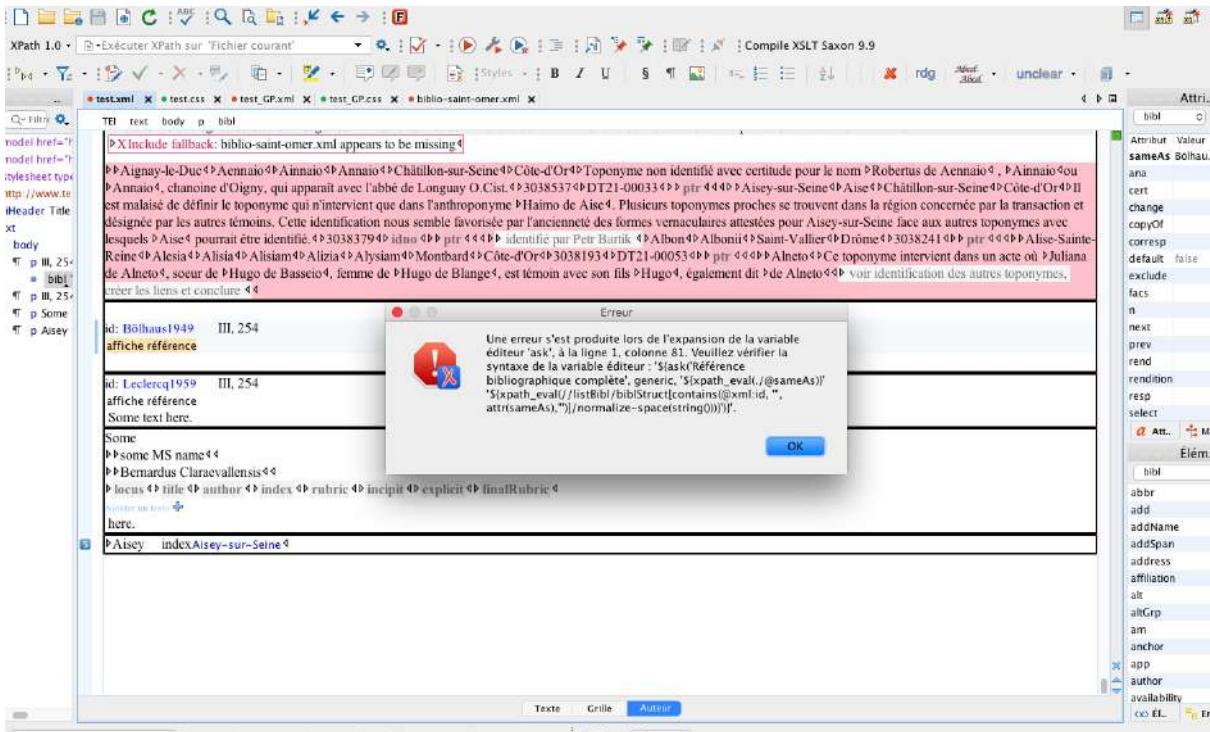
Dans les paramétrages de l'action display_biblStruct, insertion de deux \$xpath_eval dans \$ask séparés par une virgule.



Résultat : erreur de syntaxe.

Essai 2 :

Dans les paramétrages de l'action display_biblStruct, insertion de deux \$xpath_eval dans \$ask mais sans virgule.



Résultat : erreur de syntaxe.

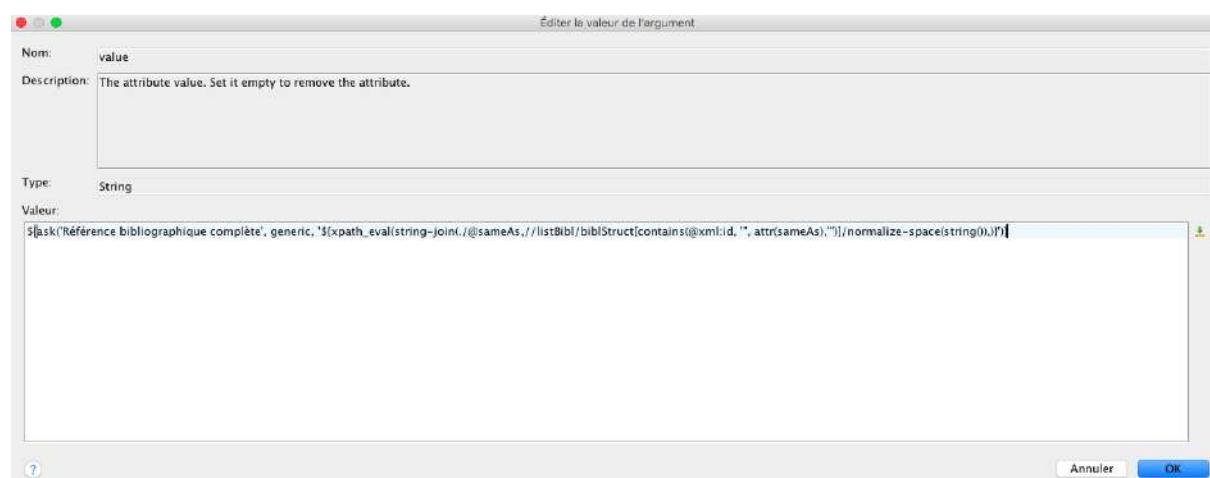
Essai 3 :

Dans les paramétrages de l'action display_biblStruct, insertion d'un seul \$xpath_eval dans \$ask, celui menant vers le contenu de biblStruct.

Résultat : Message d'erreur.

Essai 4 :

Dans les paramétrages de l'action display_biblStruct, insertion de deux \$xpath_eval dans \$ask réunis par la fonction string-join().



Résultat : Pas de messages d'erreurs mais rien ne s'affiche dans la boîte de dialogue.

Essai 5 :

Retouche de la CSS -> création d'une fenêtre pop-up pour afficher la référence complète définie dans l'action display_biblStruct, sachant que le pop-up est imbriqué dans oxy-button.

```
bibl[sameAs]:after{
  display:block;
  color:#84bff9;
  font-size:x-small;
  content: oxy_button(transparent, true,
  action, oxy_action(
    id, 'display_biblStruct')
  oxy_popup(
    edit, '#text',
    values,
    oxy_xpath(
      oxy_concat(
```

```

'string-join(//listBibl/biblStruct[@xml:id = "",attr(sameAs),"]//title, ",")'
)
oxy_concat(
'string-join(//listBibl/biblStruct[@xml:id = "",attr(sameAs),"]//author, ",")'
)
),
labels,
oxy_xpath(
oxy_concat(
'string-join(//listBibl/biblStruct[@xml:id = "",attr(sameAs),"]//title, ",")'
)
oxy_concat(
'string-join(//listBibl/biblStruct[@xml:id = "",attr(sameAs),"]//author, ",")'
)
),
resultSeparator, "|",
columns, 10,
selectionMode, "multiple",
color, "blue",
fontInherit, true));
font-size:15px
}

```

Résultat : Pas de messages d'erreurs mais rien ne s'affiche dans la boîte de dialogue.

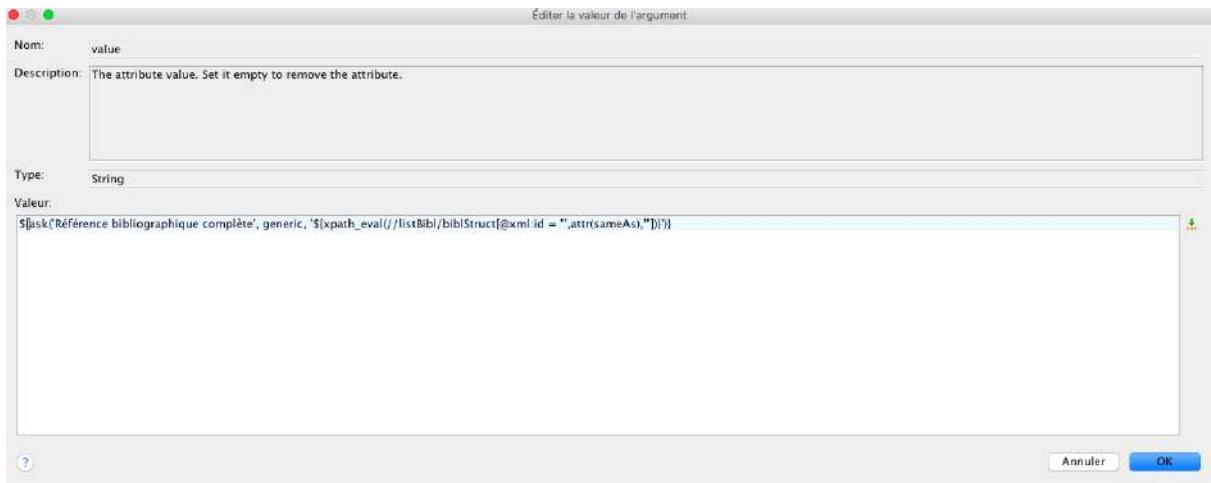
Essai 6 :

Dans les paramétrages de l'action display_biblStruct, insertion d'un \$xpath_eval vers le contenu de <biblStruct> dans \$ask avec la fonction string-join().

Résultat : Pas de messages d'erreurs mais rien ne s'affiche dans la boîte de dialogue.

Essai 7 :

Dans les paramétrages de l'action display_biblStruct, insertion d'un \$xpath_eval vers le contenu de <biblStruct> dans \$ask dont l'@xml:id doit correspondre à l'identifiant dans l'@ sameAs de <bibl>.



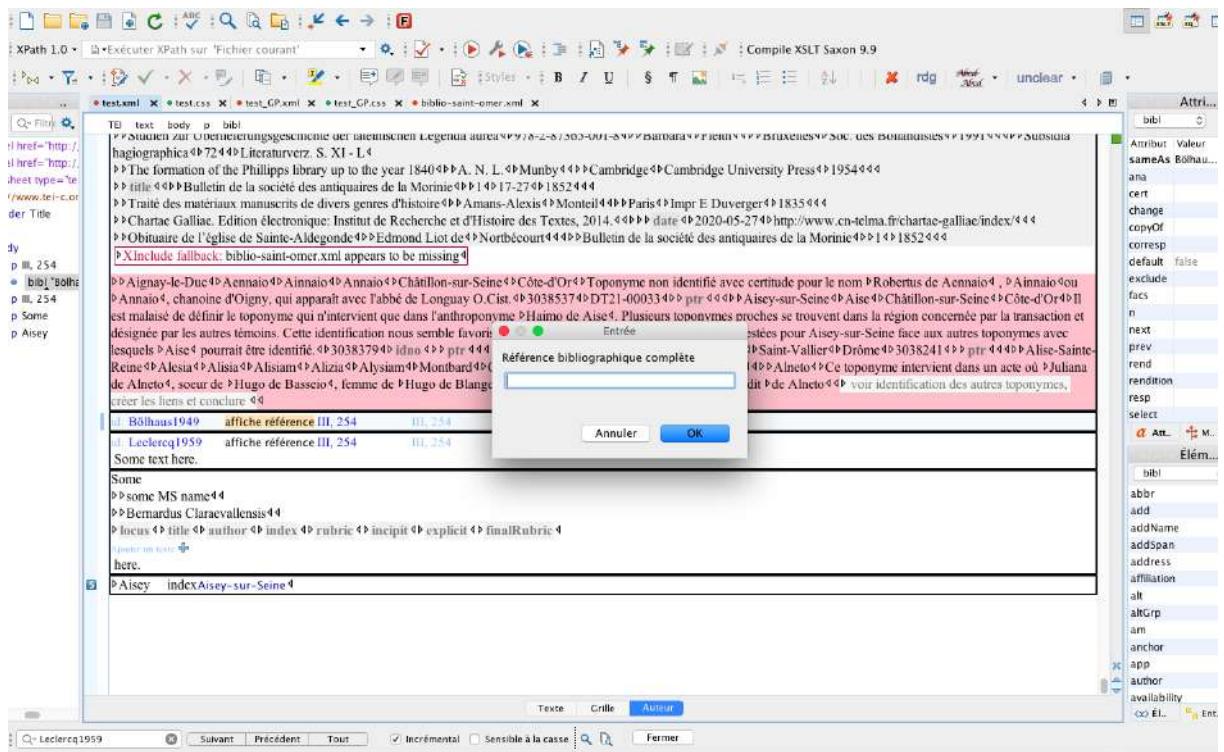
Résultat : Cela affiche la valeur de l'@ sameAs mais pas le texte entre les balises.

Essai 8 :

Ajout du chemin de fichier et de /text() dans l>xpath_eval.



Résultat : Pas de messages d'erreurs mais rien ne s'affiche dans la boîte de dialogue.

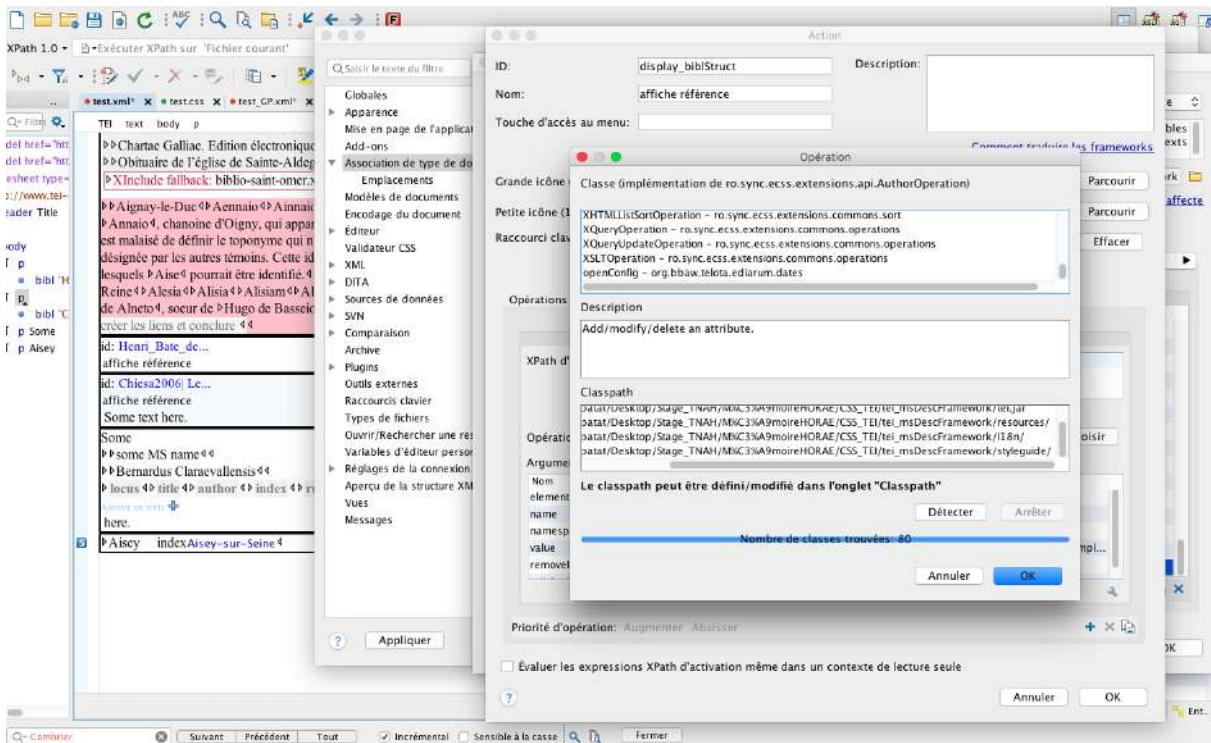


Essai 9 :

Tentative du même Xpath sans le chemin de fichier.

Résultat : Pas de messages d'erreurs mais rien ne s'affiche dans la boîte de dialogue.

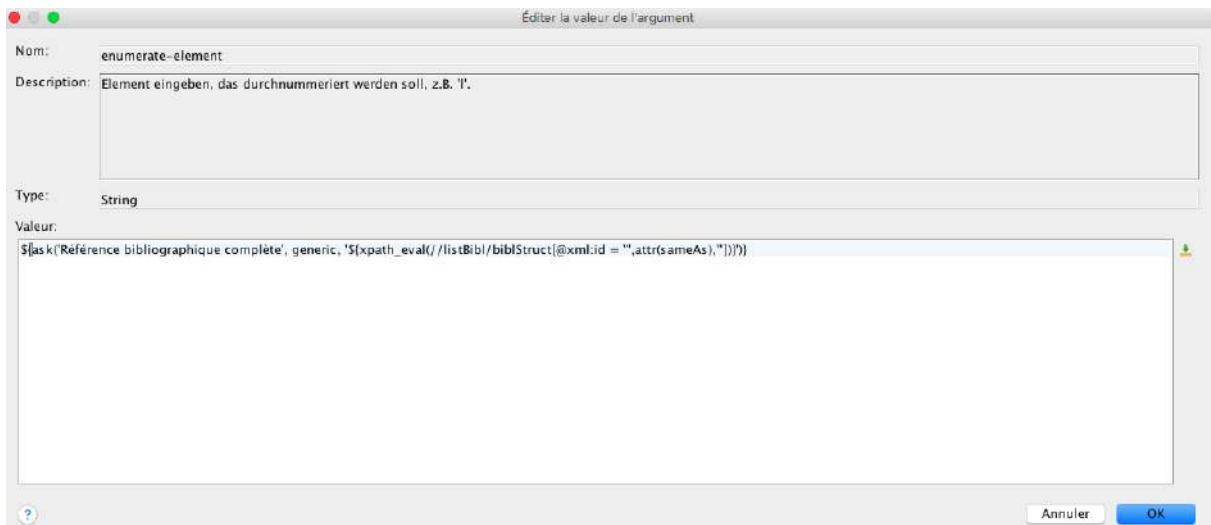
Conclusion : La classe de la fonction `display_biblStruct`, `ChangeAttributeOperation`, permet d'afficher la valeur d'un attribut, mais pas le texte entre des balises. Trouver un moyen de le faire. Changer la classe ?



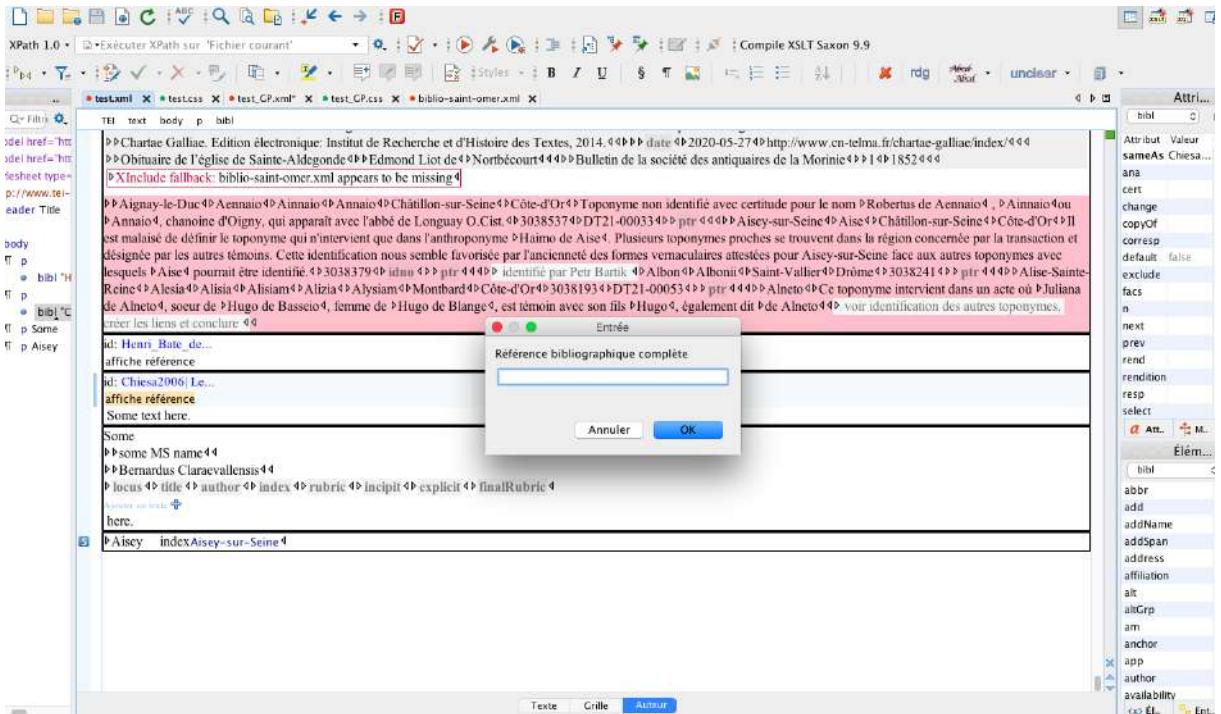
Cf. `DisplayModeConstants` avec `Display_Mode_Block_Tags` mais peut-être pas compatible pour le format tei car pas proposée dans les paramètres d'Oxygen. Recherche dans le dossier `framework_ms_Desc` mais pas trouvée.

Essai 10 :

Choix de la classe `EnumerateElementsOperation`.

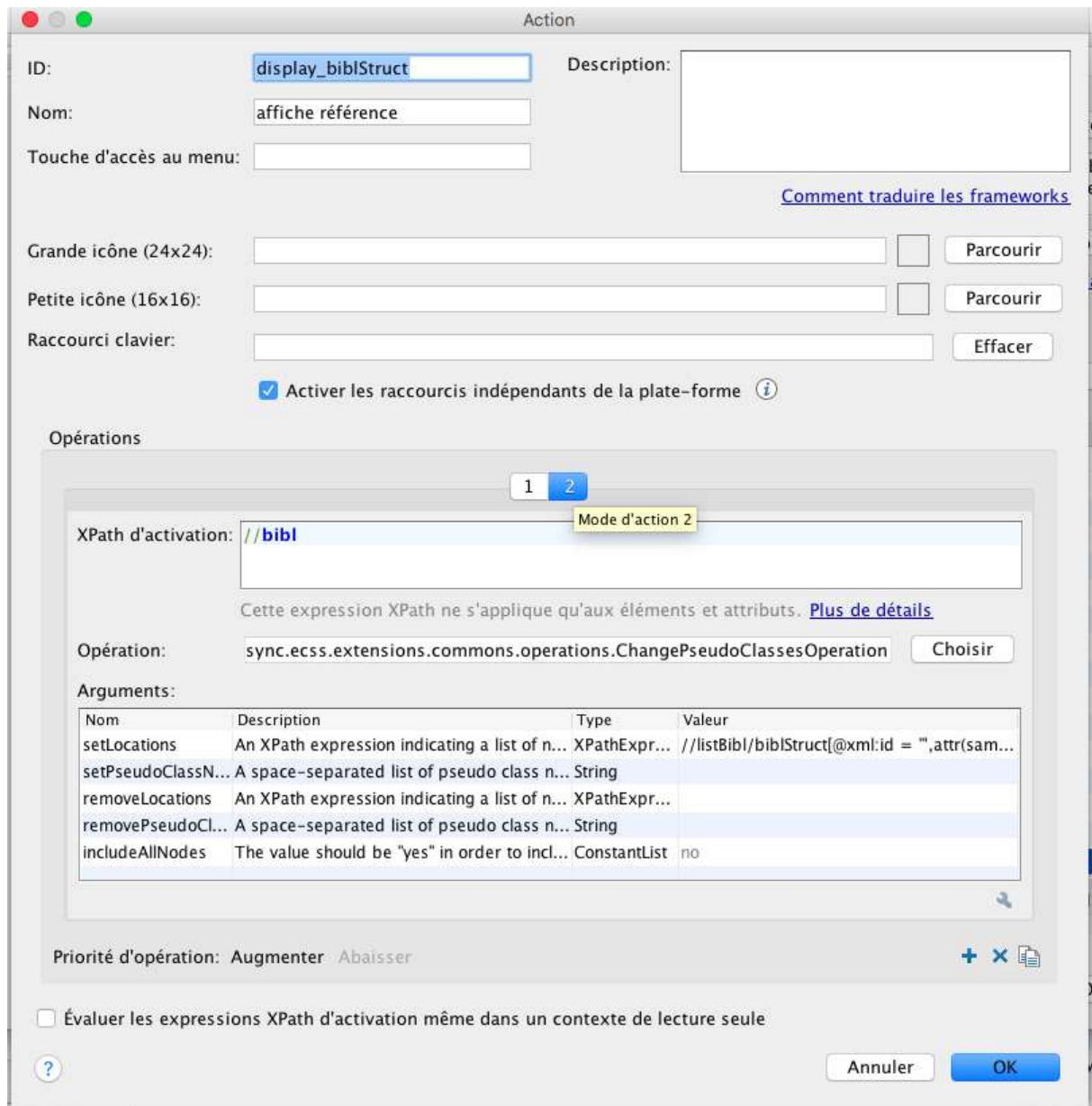


Résultat : Pas de messages d'erreurs mais rien ne s'affiche dans la boîte de dialogue.



Essai 11 :

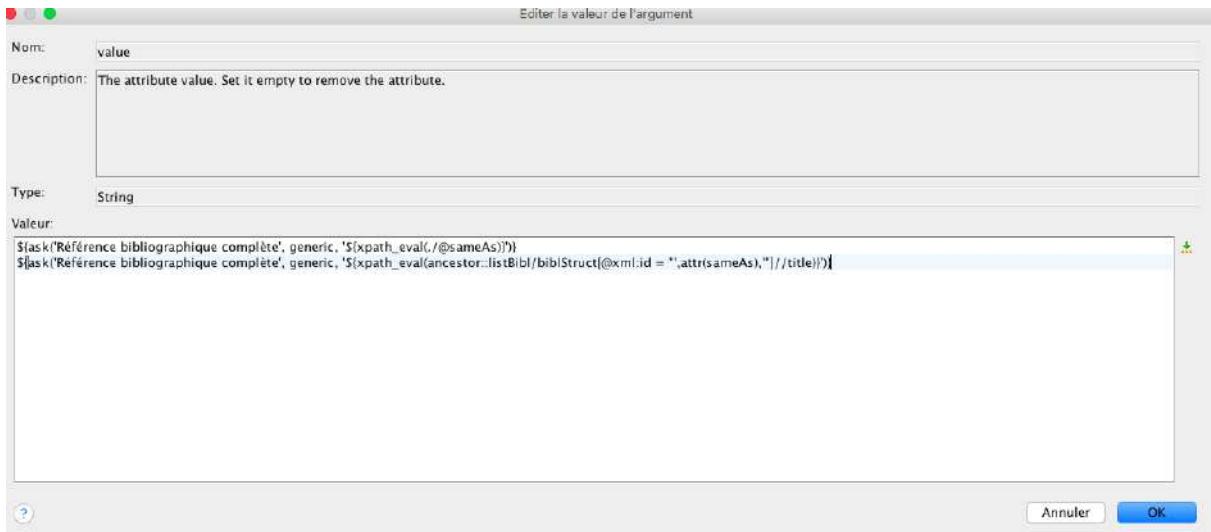
Tentative avec classe ReloadContentOperation en plus de celle ChangeAttributeOperation mais la classe n'est pas adaptée à l'action souhaitée.



Observations : La boîte de dialogue n'affiche plus seulement l'id sélectionné mais un autre après un `|`. L'action `display_biblStruct` est paramétrée au niveau de `<bibl>` : est-ce pour cela que le contenu de `<biblStruct>` n'est pas affiché ?

Essai 12 :

Insertion de plusieurs `xpath_eval` grâce à plusieurs `$sak`.



Résultat : Pas de messages d'erreurs mais rien ne s'affiche dans la boîte de dialogue.

Essai 13 :

Modifications des values dans oxy_popup et tentatives d'en insérer plusieurs à suivre suivant la documentation suivante : <https://www.oxygenxml.com/doc/versions/18/ug-author/topics/pop-up-editor.html>

```

bibl[sameAs]:after{
    display:block;
    color:#84bff9;
    font-size:x-small;
    content: oxy_button(transparent, true,
    action, oxy_action(
        id, 'display_biblStruct')
    oxy_popup(
        edit, '#text',
        values,
        oxy_xpath(
            oxy_concat(
                'string-join(//listBibl/biblStruct[@xml:id = "", attr(sameAs), "]//title, ",")'
            )
            oxy_concat(
                'string-join(//listBibl/biblStruct[@xml:id = "", attr(sameAs), "]//author, ",")'
            )
            ),
        labels,
        oxy_xpath(
            oxy_concat(
                'string-join(//listBibl/biblStruct[@xml:id = "", attr(sameAs), "]//title, ",")'
            )
            oxy_concat(
                'string-join(//listBibl/biblStruct[@xml:id = "", attr(sameAs), "]//author, ",")'
            )
            ),
        resultSeparator, "|",
        columns, 10,
    
```

```

selectionMode, "multiple",
color, "blue",
fontInherit, true));
font-size:15px
}

```

Résultats : Erreurs de syntaxe ou aucun affichage.

Essai 14 :

Valeur suivante dans un des arguments de l'action display_biblStruct : \${ask('Référence bibliographique complète', generic, '\${xpath_eval(doc("\${cfdu}/biblio-saint-omer.xml")//listBibl/biblStruct[contains(@xml:id="",attr(@sameAs,"")]/normalize-space(string-join(descendant::*, ", ")))}')}

Résultat : pas de message d'erreurs mais la box ne renvoie rien.

Conclusion : Peut-être essayer une oxy_action_list pour renvoyer à la fois l'@ sameAs et la référence bibliographique complète.

05/06

Essai 1 :

Valeur suivante dans un des arguments de l'action display_biblStruct :

```

${ask
('Référence bibliographique complète',
generic,
'${xpath_eval(doc("${cfdu}/biblio-saint-omer.xml")//listBibl/biblStruct[@xml:id="",attr(@sameAs,"")]/text())}'}

```

!\\ On met le double slash avant text() car ce n'est pas un enfant direct de <biblStruct>.

Résultat : pas de message d'erreurs mais la box ne renvoie rien.

Essai 2 :

Dans la CSS, attention au string-join() des oxy_concat() : si un titre contient une virgule, il sera coupé si le séparateur est une virgule. Donc on peut remplacer la virgule par une autre délimiteur comme ci-dessous :

```

bibl[sameAs]:after{
    display:block;
    color:#84bff9;
    font-size:x-small;
    content:oxy_button(transparent,
                        true,
                        oxy_action(
                            'display_biblStruct')
                        '#text',
                        id,
                        oxy_popup(
                            edit,
                            values,
                            oxy_xpath(
                                oxy_concat(
                                    'string-join("//listBibl/biblStruct[@xml:id
                                )
                                oxy_concat(
                                    'string-join("//listBibl/biblStruct[@xml:id
                                )
                                ),
                                ),
                            labels,
                            oxy_xpath(
                                oxy_concat(
                                    'string-join("//listBibl/biblStruct[@xml:id
                                )
                                oxy_concat(
                                    'string-join("//listBibl/biblStruct[@xml:id
                                )
                                ),
                                ),
                                resultSeparator,
                                columns,
                                selectionMode,
                                color,
                                fontInherit,
                                font-size:15px
                            }

```

Résultat : pas de message d'erreurs mais la box ne renvoie rien.

Essai 3 :

Modification de la CSS pour obtenir le titre correspondant dans la pop-up :

```

bibl[sameAs]:after{
    display:block;
    color:#84bff9;
    font-size:x-small;
    content:oxy_button(transparent,
                        true,
                        oxy_action(
                            'display_biblStruct')
                        '#text',
                        id,
                        oxy_popup(
                            edit,
                            values,
                            oxy_xpath(
                                oxy_concat(

```

```

'//listBibl/biblStruct[@xml:id
)           =      "",attr(sameAs),""]//title/normalize-space(string())
),
),
labels,
oxy_xpath(
  oxy_concat(
    '//listBibl/biblStruct[@xml:id
)
),
resultSeparator,
columns,
selectionMode,
color,
fontInherit,
font-size:15px
),
"|,
30,
"multiple",
"blue",
true));
}

```

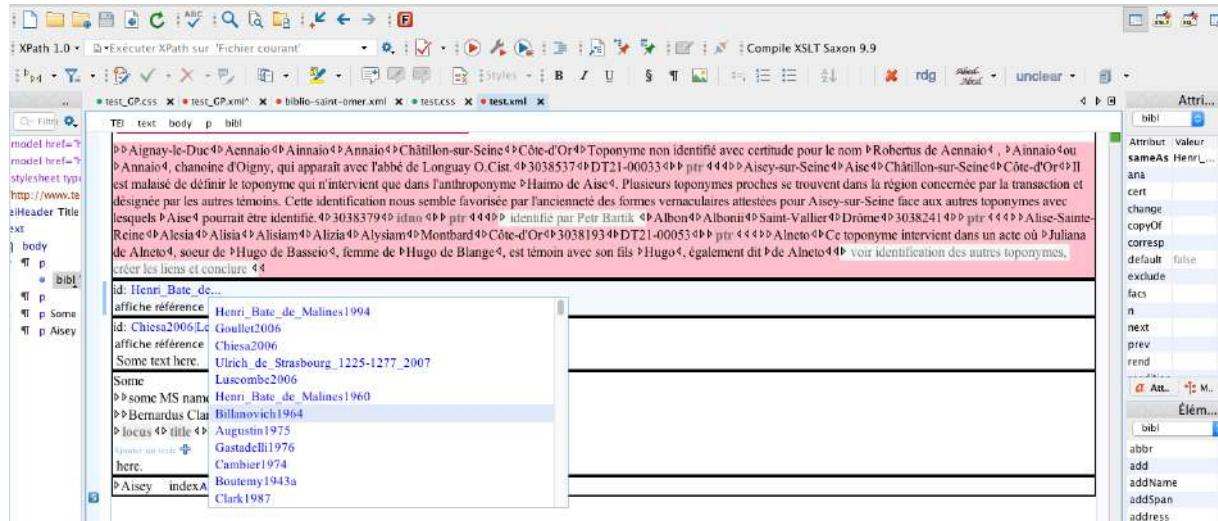
Puis modification du paramètre \$ask dans le paramétrage de l'action display_biblStruct où on enlève la condition d'égalité dans l>xpath-eval() (**Demander dans Forum Oxygen si elle est autorisée dans ce contexte -> Oui avec xpath_eval() imbriqués, toujours mettre un xpath_eval() devant un chemin xpath dans les paramétrages de l'action**) :

```

${ask('Référence bibliographique complète',
generic,
'${xpath_eval("//listBibl/biblStruct//text())"})'}).

```

Résultat original : Peu importe la référence id choisie, apparition fugace dans la pop-up du premier titre de biblio_saint_omer.xml ("Speculum divinorum et quorundam naturalium. Parts VI-VII, On the unity of intellect, On the platonic doctrine of the ideas") :



The screenshot shows two instances of a modal dialog box in the Oxygen XML Editor. The dialog is titled "Référence bibliographique complète" and contains the text "Speculum divinorum et quorundam natura". The background shows an XML editor interface with various tabs and toolbars.

Conclusion: La condition d'égalité avec la référence id choisie n'est pas établie.

Essai 4 :

Dans paramétrage de l'action, remplacement de la condition d'égalité dans le paramètre \$xpath-eval() par la fonction contains() et current-node, et définition du chemin vers le titre, comme dans la CSS :

```

${ask('Référence bibliographique complète',
generic,
'$xpath_eval(//listBibl/biblStruct[contains(., current-node()/@sameAs)]/@xml:id//title)')}

```

Résultat : pas de message d'erreurs mais la box ne renvoie rien.

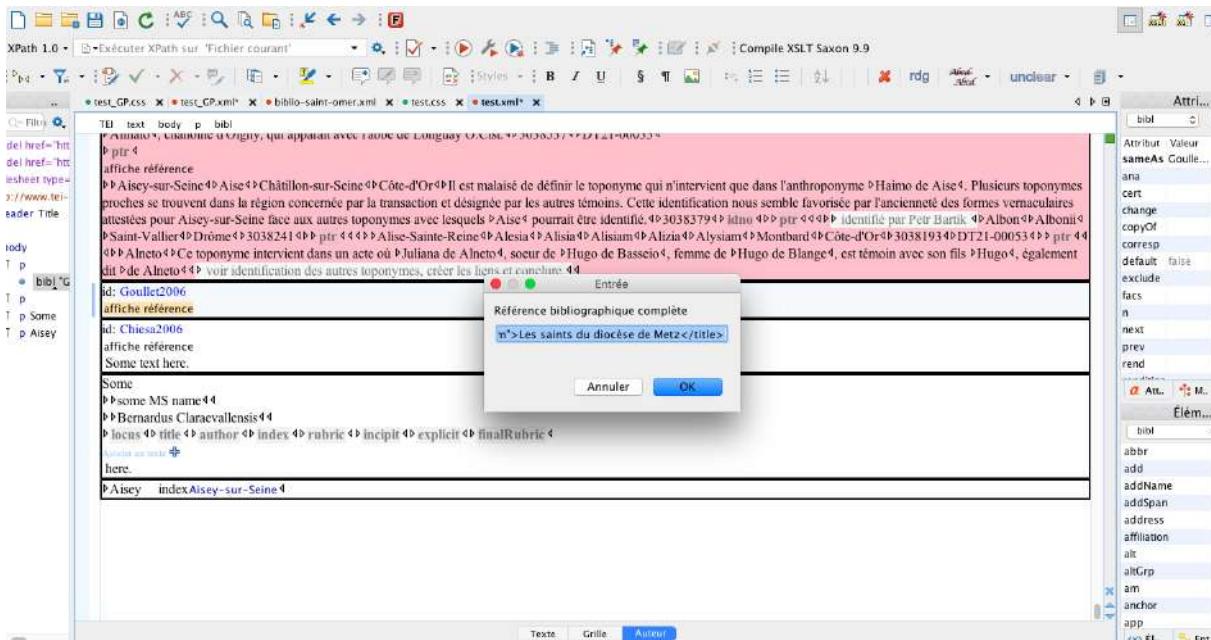
Solution

Faire une condition d'égalité dans le paramétrage de l'action grâce à une variable \$xpath_eval() imbriquée dans une même variable \$xpath_eval(), sous la forme suivante :

```
 ${ask('Référence bibliographique complète',
generic,
'$${xpath_eval("//biblStruct[@xml:id=\"$${xpath_eval(./@sameAs)}\"]//title})}')
```

Le chemin vers l'information que l'on veut obtenir doit être le même que celui défini dans la pop-up de la CSS :

```
bibl[sameAs]:after{
    display:block;
    color:#84bff9;
    font-size:x-small;
    content: oxy_button(transparent, true,
    action, oxy_action(
        id, 'display_biblStruct')
    oxy_popup(
        edit,
        values,
        oxy_xpath(
            oxy_concat(
                '//listBibl/biblStruct[@xml:id
            )),
        labels,
        oxy_xpath(
            oxy_concat(
                '//listBibl/biblStruct[@xml:id
            )
        ),
        resultSeparator,
        columns,
        selectionMode,
        color,
        fontInherit,
        font-size:15px;
    ) = "",attr(sameAs),"]//title"
    ) = "",attr(sameAs),"]//title"
),
    "|,
    10,
    "multiple",
    "blue",
    true));
}
```

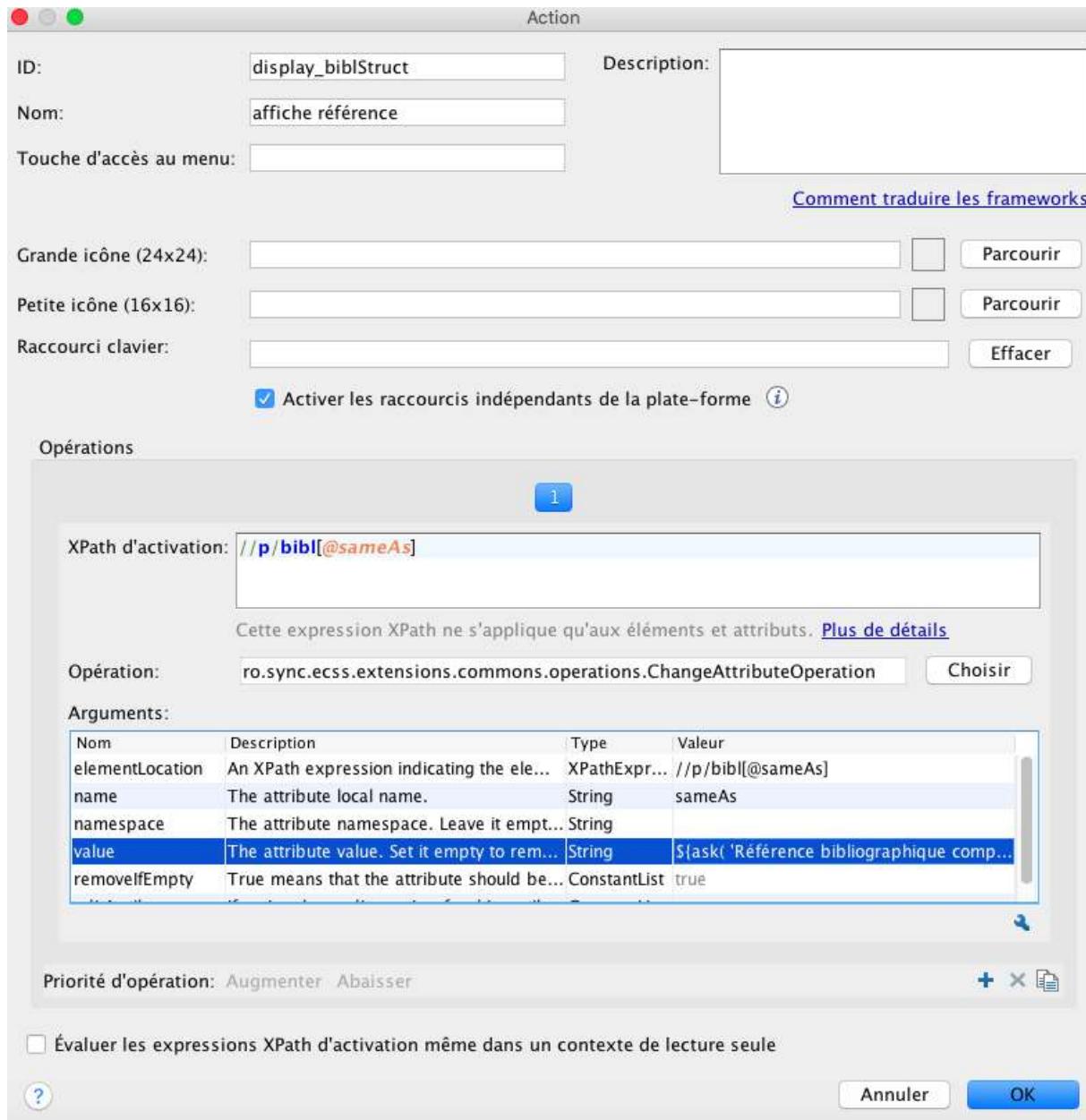


Bien préciser aussi dans paramétrage de l'action l'xpath d'activation vers un `<bibl>` enfant directe de `<p>` et avec l'`@sameAs`, car d'autres `<bibl>` sont descendants de `<p>` et cela peut créer des confusions :

```

<sourceDesc>
  <p>
    <xi:include href="biblio-saint-omer.xml" xmlns:xi="http://www.w3.org/2001/XInclude">
      <xi:fallback>biblio-saint-omer.xml appears to be missing</xi:fallback>
    </xi:include>
    <listPlace>
      <place
        <placeName type="preferred" xml:lang="fre" cert="low">Aignay-le-Duc</placeName>
        <placeName ref="#d1e131210">Aennaio</placeName>
        <placeName ref="#d1e159495">Ainnaio</placeName>
        <placeName ref="#d1e131116">Annaio</placeName>
        <district type="canton">Châtillon-sur-Seine</district>
        <district type="département">Côte-d'Or</district>
        <note>Toponyme non identifié avec certitude pour le nom <q>Robertus de
          Aennaio</q>, <q>Ainnaio</q> ou <q>Annaio</q>, chanoine d'Oigny, qui
          apparaît avec l'abbé de Longuay O.Cist.</note>
        <idno type="GeoNames">3038537</idno>
        <idno type="Dico-Topo">DT21-00033</idno>
        <bibl><ptr target="https://fr.wikipedia.org/wiki/Aignay-le-Duc"/></bibl>
      </place>
    </listPlace>
  </p>

```



Toutefois **attention** : Les données sont modifiables dans la boîte de dialogue, donc créer une combobox non éditable qui a pour valeur (real_value) l'@xml:id et pour label la référence bibliographique (rendered_value). Cf. <https://www.oxygenxml.com/doc/versions/22.1/ug-editor/topics/editor-variables.html>. /!\ au traitement de ":" et ";" et des guillemets eux-mêmes. Le système doit fonctionner si on a plusieurs valeurs.

Essai 1

```
$ { ask
('Référence bibliographique complète',
combobox,
```

```
('@xml:id':'${xpath_eval("//biblStruct[@xml:id='${xpath_eval(./@sameAs)}'"]//title)}')
```

Échec

Essai 2 :

```
 ${ask(  
  'Référence bibliographique complète',  
  combobox,  
  ('${xpath_eval(  
    //biblStruct[@xml:id='${xpath_eval(./@sameAs)}']:  
    '${xpath_eval(${xpath_eval(//biblStruct[@xml:id= attr(sameAs,''])//title)})};  
    '${xpath_eval(${xpath_eval(//biblStruct[@xml:id= attr(sameAs,''])//author)})},  
    'default')})
```

=> erreur de syntaxe : enlever la condition d'égalité avec les guillemets simples.

Essai 3 :

```
 ${ask(  
  'Référence bibliographique complète', combobox,  
  ('${xpath_eval(  
    //biblStruct[@xml:id='${xpath_eval(./@sameAs)}']:  
    '${xpath_eval(${xpath_eval(//biblStruct//title)})};  
    '${xpath_eval(  
      //biblStruct[@xml:id='${xpath_eval(./@sameAs)}']:  
      '${xpath_eval(${xpath_eval(//biblStruct//author)})},  
      'default')})
```

=> erreur de syntaxe : cf. Modèle suivant :

```
 ${ask('message',  
   combobox,  
   ('real_value1':'rendered_value1';...;'real_valueN':'rendered_valueN'),  
   'default')}
```

Soit 4 arguments et deux parenthèses s'englobant. Oublie d'une parenthèse + erreur probablement au niveau du default donc bonne version :

```

${ask(
'Reférence bibliographique complète',
combobox,
('${xpath_eval(
//biblStruct[@xml:id="${xpath_eval(./@sameAs)}"]/@xml:id}':
 '${xpath_eval(//biblStruct//title)}';
`${xpath_eval(
//biblStruct[@xml:id="${xpath_eval(./@sameAs)}"]/@xml:id }':
 '${xpath_eval(//biblStruct//author)}',
`${xpath_eval(//biblStruct//title)})}

```

=> Toujours faute de syntaxe

```

${ask( 'Référence bibliographique complète',
combobox,
('//*[@xml:id="${xpath_eval(./@sameAs)}"]/@xml:id':
'//biblStruct//title';
'//*[@xml:id="${xpath_eval(./@sameAs)}"]/@xml:id':
'//biblStruct//author',
'//biblStruct//title')}

```

Échec

Solution pour afficher tous les titres et tous les auteurs des <biblStruct> :

```

${ask(
'Reférence bibliographique complète',
combobox,
('${xpath_eval(//*[@xml:id="${xpath_eval(./@sameAs)}"]/@xml:id}':
 '${xpath_eval(//biblStruct//title)}';
`${xpath_eval(//*[@xml:id="${xpath_eval(./@sameAs)}"]/@xml:id}':
 '${xpath_eval(//biblStruct//author)}',
`${xpath_eval(//biblStruct//title)})}

```

Solution pour afficher titre(s) et auteur(s) des <biblStruct> appelés :

```

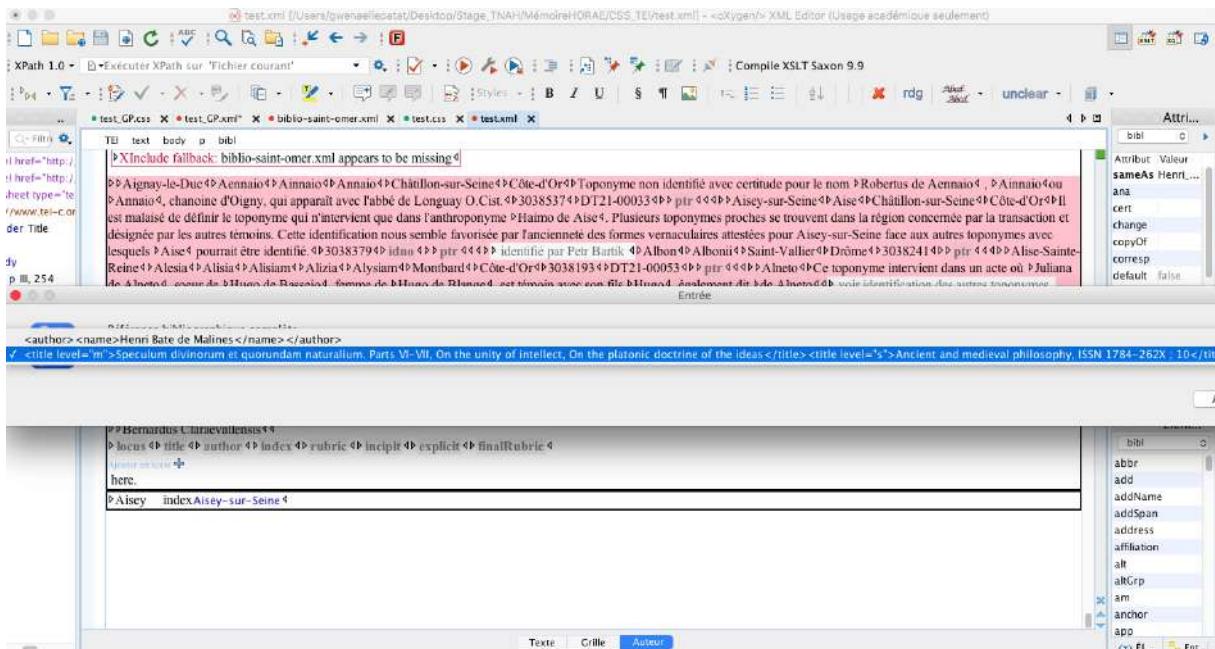
${ask(
'Reférence bibliographique complète',
combobox,
('${xpath_eval(//*[@xml:id="${xpath_eval(./@sameAs)}"]/@xml:id}':
 '${xpath_eval(//*[@xml:id="${xpath_eval(./@sameAs)}"]//title)}';

```

```
'${xpath_eval("//biblStruct[@xml:id='${xpath_eval('./@sameAs)}'"]/@xml:id)}':
'${xpath_eval("//biblStruct[@xml:id='${xpath_eval('./@sameAs)}'']//author')},
'${xpath_eval("//biblStruct[@xml:id='${xpath_eval('./@sameAs)}'']//title')}')
```

La CSS qui va avec :

```
bibl[sameAs]{
    display:block;
    color:#84bff9;
    font-size:x-small;
    content:oxy_button(transparent, true,
    action, oxy_action(
    'display_biblStruct')
    id, '#text',
    oxy_combobox(edit, false,
    editable,
    values,
    oxy_xpath(
        oxy_concat(
            '//listBibl/biblStruct[@xml:id
        ) = """",
        attr(sameAs, "")]/title'
        ),
        ),
    labels,
    oxy_xpath(
        oxy_concat(
            '//listBibl/biblStruct[@xml:id
        ) = """",
        attr(sameAs, "")]/title'
        ),
        ),
    columns, 10,
    color, "blue",
    fontInherit, true
    )
);
font-size:15px;
}
```



Pour obtenir la **référence complète**, paramétrage de l'action suivant :

```

${ask(
'Reférence bibliographique complète',
combobox,
('${xpath_eval("//biblStruct[@xml:id='${xpath_eval('./@sameAs)}"]/@xml:id}':
 '${xpath_eval("//biblStruct[@xml:id='${xpath_eval('./@sameAs)}']//text()')}'),
 '${xpath_eval("//biblStruct [@xml:id='${xpath_eval('./@sameAs)}']//title')}')

```

CSS suivante :

```

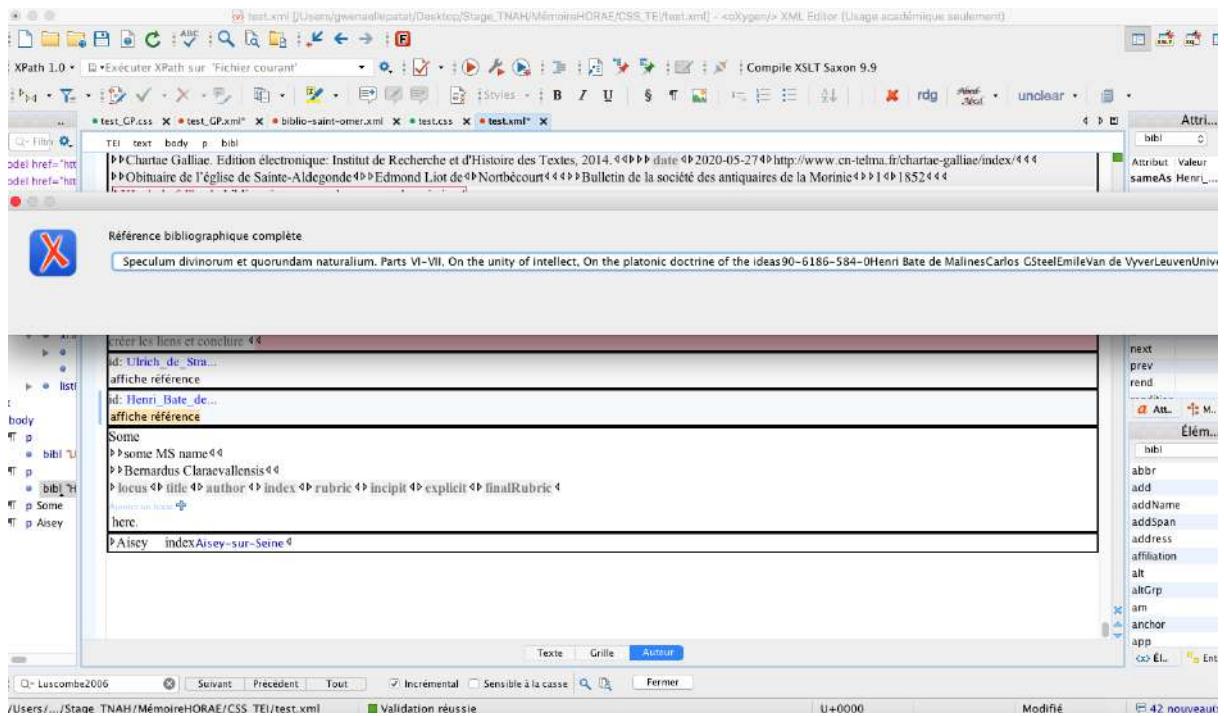
bibl[sameAs]:after{
  display:block;
  color:#84bff9;
  font-size:x-small;
  content:oxy_button(transparent,
    true,
    oxy_action(
      'display_biblStruct')
    '#text',
    false,
    action,
    id,
    oxy_combobox(
      edit,
      editable,
      values,
      oxy_xpath(
        oxy_concat(
          '//listBibl/biblStruct[@xml:id
            )
            ),
            labels,
            oxy_xpath(
              oxy_concat(
                //listBibl/biblStruct[@xml:id
                  = """",
                  attr(sameAs),"]//text()'=
                  """",
                  attr(sameAs),"]//text()'=

```

```

        )
        ),
color,
fontInherit,
)
);
font-size:15px;
}

```



Mais ça ne fonctionne que pour une référence sélectionnée.

Pour plusieurs @sameAs sélectionnés :

```

${task(
'Reférence bibliographique complète',
combobox,
('${xpath_eval("//biblStruct[@xml:id='${xpath_eval('./@sameAs)}'"]/@xml:id}')':
 '${xpath_eval("//biblStruct[@xml:id='${xpath_eval('./@sameAs)}'"]//text())}';
 '${xpath_eval("//biblStruct[@xml:id='${xpath_eval('./@sameAs)}'']/@xml:id}')':
 '${xpath_eval("//biblStruct[@xml:id='${xpath_eval('./@sameAs)}'"]//text())'),
 '${xpath_eval("//biblStruct[@xml:id='${xpath_eval('./@sameAs)}'']//title)'}

```

=> Non car ça affiche 2 fois le même sameAs qu'on ne peut changer. À creuser.

Pour le confort visuel : améliorer la gestion des espaces et faire que la boîte de dialogue s'adapte à la taille du contenu.

08/06

Fonction simplifiée affichant uniquement le texte et pas les balises :

```
 ${ask(  
'Référence           bibliographique           complète',  
combobox,  
('{xpath_eval("./@sameAs)":${xpath_eval("//biblStruct[@xml:id='${xpath_eval("./@sameAs)}']/stri  
ng-join(./text(),",","))}'),  
'${xpath_eval("./@sameAs)}')}
```

Le premier `xpath_eval()` n'a pas besoin de transférer l'`@sameAs` en `@xml:id` pour le récupérer. Il n'est possible d'avoir qu'une valeur affichée pur une valeur réelle. Faire attention aux sauts de ligne et aux espaces. Si la combobox choisit ligne par ligne, il est possible de concaténer pour afficher les informations de manière plus fine (ex. : "Date : ...").

Paramétrages de l'action `inserBiblio_3` :

L'objectif est d'afficher les références bibliographiques peu importe le nombre de références sélectionnées, sachant q'une ligne correspond à une référence dans la `combobox()`.

```
'Référence bibliographique complète',  
combobox,  
('{xpath_eval("//biblStruct[contains(.,  
"${xpath_eval("./@sameAs)}")]/@xml:id}':${xpath_eval("//biblStruct[contains(.,  
"${xpath_eval("./@sameAs)}")]/string-join(./text(),",",))}'),  
'default2')}
```

Résultat : Affiche "default2" pour une ou plusieurs références sélectionnées.

Rappel de la syntaxe :

```
 ${ask('message',  
combobox,  
'real_value1':  
'rendered_value1';...;  
'real_valueN':  
'rendered_valueN'),  
'default')}
```

Essai 1 :

```
 ${ask(  
'Référence           bibliographique           complète',  
combobox,  
('{xpath_eval("//biblStruct[contains(., "${xpath_eval("./@sameAs)}")]/@xml:id}':  
"${xpath_eval("//biblStruct[contains(., "${xpath_eval("./@sameAs)}")]/string-join(./text(),",",))}'),
```

```
'default')}
```

Résultat : Affiche ‘default’ pour une ou plusieurs références sélectionnées.

Essai 2 :

```
 ${ask(  
 'Référence bibliographique complète',  
 combobox,  
 ('${xpath_eval("//biblStruct[contains(., "${xpath_eval('./@sameAs)}")]/@xml:id}'):  
 '${xpath_eval("//biblStruct[contains(., "${xpath_eval('./@sameAs)}"]/[string-join(./text(), ", ")]}'),  
 '${xpath_eval('./@sameAs)}'  
)}
```

Résultat : Cela affiche la valeur de l’@sameAs sélectionné. Si plusieurs références sont sélectionnées, elles s’affichent à la suite séparées par un “ | ”. Jouer sur l’affichage de la combobox dans la CSS ?

Essai 3

```
 ${ask(  
 'Référence bibliographique complète',  
 combobox,  
 ('${xpath_eval("//biblStruct[contains(., "${xpath_eval('./@sameAs)}")]/@xml:id}'):  
 '${xpath_eval("//biblStruct[contains(., "${xpath_eval('./@sameAs)}"]/[string-join(./text(), ", ")]}'),  
 '${xpath_eval("//biblStruct /string-join(./text(), ", ")}'  
)}
```

Résultat : /!\ Cela affiche tout ce qui est contenu dans les <biblStruct> et c'est très lent. Affiner ce qui est dans le paramètre ‘default’.

Essai 4

```
 ${ask(  
 'Référence bibliographique complète',  
 combobox,  
 ('${xpath_eval("//biblStruct[contains(., "${xpath_eval('./@sameAs)}")]/@xml:id}'):  
 '${xpath_eval("//biblStruct[contains(., "${xpath_eval('./@sameAs)}"]/[string-join(./text(), ", ")]}'),  
 '${xpath_eval("//biblStruct[contains(., "${xpath_eval('./@sameAs)}"] /string-join(./text(), ", ")}'  
)}
```

Résultat : Cela n'affiche rien. Problème sur la fonction contains(). Il faut faire le lien entre l’@sameAs de <bibl> et l’@xml:id de <biblStruct>.

Essai 5

```
 ${ask(  
 'Référence bibliographique complète',  
 combobox,  
 ('${xpath_eval("//biblStruct[contains(., "${xpath_eval("//bibl[@sameAs]/@sameAs}")})]/@xml:id}'):
```

```

'${xpath_eval("//biblStruct[contains(.,      "${xpath_eval("//bibl[@sameAs]/@sameAs)}"])/string-
join(./text(), ", ")})'),
'default'
)}

```

On précise dans l'xpath_eval() dans la fonction contains() que l'@sameAs est contenu dans la balise <bibl>.

Résultat : Cela affiche 'default'.

Essai 6

```

${ask(
'Référence          bibliographique          complète',
combobox,
('${xpath_eval("//biblStruct[contains(., "${xpath_eval("//bibl[@sameAs]/@sameAs)}"])/@xml:id}':
'${xpath_eval("//biblStruct[contains(.,      "${xpath_eval("//bibl[@sameAs]/@sameAs)}"])/string-
join(./text(), ", ")})'),
'${xpath_eval("//biblStruct[contains(.,      "${xpath_eval("//bibl[@sameAs]/@sameAs)}"])/string-
join(./text(), ", ")})'
)}

```

Résultat : Cela n'affiche rien.

Essai 7

```

${ask(
'Référence          bibliographique          complète',
combobox,
('${xpath_eval("//biblStruct/@xml:id[contains(., ${xpath_eval("//bibl[@sameAs]/@sameAs)}")]
}':
'${xpath_eval("//biblStruct/@xml:id[contains(., ${xpath_eval("//bibl[@sameAs]/@sameAs)}")
/string-join(./text(),
",",
"))})'),
'${xpath_eval("//biblStruct/@xml:id[contains(., ${xpath_eval("//bibl[@sameAs]/@sameAs)}")
/string-join(./text(),
",",
"))})'
)}

```

Résultat : Cela n'affiche rien.

Essai 8

Partir dans l'xpath_eval() du paramètre real_value de <bibl> et pas de <biblStruct>, car dans l'arborescence xml le bouton actionnant les paramètres se trouve à cet endroit.

```

${ask(
'Référence          bibliographique          complète',
)

```

```

combobox,
('${xpath_eval("//bibl[@sameAs]/@sameAs[contains(., ${xpath_eval("//biblStruct/@xml:id})")]
)})':
`${xpath_eval("//biblStruct/@xml:id[contains(., ${xpath_eval("//bibl[@sameAs]/@sameAs})"]
/string-join(./text(),
",
"))
)`},
`${xpath_eval("//biblStruct/@xml:id[contains(., ${xpath_eval("//bibl[@sameAs]/@sameAs})"]
/string-join(./text(),
",
"))
)`}'
)

```

Résultat : Combobox toujours vide.

Essai 9

Changement du contenu de la fonction contains()

```

${ask(
'Référence bibliographique complète',
combobox,
('${xpath_eval("//biblStruct [contains(@xml:id, "", attr(sameAs,""))]')':
`${xpath_eval("//biblStruct[contains(@xml:id, "", attr(sameAs,""))]/string-join(./text(), ", "
)`},
`${xpath_eval("//biblStruct/@xml:id[contains(@xml:id, "", attr(sameAs,""))]/string-
join(./text(),
",
"))
}`'
)

```

Résultat : Combobox toujours vide.

Essai 10

Inversion des guillemets dans la fonction contains()

```

${ask(
'Référence bibliographique complète',
combobox,
('${xpath_eval("//biblStruct [contains(@xml:id, "", attr(sameAs,""))]')':
`${xpath_eval("//biblStruct[contains(@xml:id, "", attr(sameAs,""))]/string-join(./text(), ", "
)`},
`${xpath_eval("//biblStruct/@xml:id[contains(@xml:id, "", attr(sameAs,""))]/string-
join(./text(),
",
"))
}`'
)

```

Résultat : Combobox toujours vide.

Essai 11 :

Essayer xpath_eval() avec une CSS développant les mêmes chemins XPath.

CSS :

```
bibl[sameAs]:after{
    display:block;
    color:#84bff9;
    font-size:x-small;
    content:oxy_button(transparent,
        true,
        oxy_action(
            'inserBiblio_3')
    );
    action,
    id,
    oxy_combobox(
        edit,
        editable,
        values,
        oxy_xpath(
            oxy_concat(
                '//listBibl/biblStruct[contains(@xml:id,
                    ''),
                '',
                attr(sameAs, "")]//text()'
            )
        ),
        labels,
        oxy_xpath(
            oxy_concat(
                '//listBibl/biblStruct[contains(@xml:id,
                    ''),
                    '',
                    attr(sameAs, "")]//text()'
            )
        ),
        color,
        fontInherit,
    );
    font-size:15px;
}
```

Action dans les paramètres de l'action inserBiblio_3 :

```
 ${ask(
    'Référence bibliographique complète',
    combobox,
    ('${xpath_eval("//biblStruct/@xml:id[contains(., "${xpath_eval("//bibl[@sameAs]/@sameAs}")]"})':
    '${xpath_eval("//biblStruct/@xml:id[contains(., "${xpath_eval("//bibl[@sameAs]/@sameAs}")]" /string-join(./text(),
        '',
        ")))}'),
    '${xpath_eval("//biblStruct/@xml:id[contains(., "${xpath_eval("//bibl[@sameAs]/@sameAs}")]" /string-join(./text(),
        '',
        ")))}'
)}
```

Résultat : Combobox toujours vide.

Essai 12

```
 ${task(
'Référence bibliographique complète',
'combobox,
('{$xpath_eval("./@sameAs")}':
'${xpath_eval("//biblStruct[contains("${xpath_eval(@xml:id)}","${xpath_eval("//bibl[@sameA
s]/@sameAs)})"]}/string-join(./text(),""),
'${xpath_eval("//biblStruct[contains("${xpath_eval(@xml:id)}","${xpath_eval("//bibl[@sameA
s]/@sameAs)})"]}/string-join(./text(),""))
)}
```

Ajout du paramètre onChange dans la CSS

```
bibl[sameAs]:after{
  display:block;
  color:#84bff9;
  font-size:x-small;
  content: oxy_button(transparent, true, action, oxy_action(
    id,
    oxy_combobox(
      edit,
      editable,
      values,
      oxy_xpath(
        oxy_concat(
          '//listBibl/biblStruct[contains(@xml:id,
        '')",
        attr(sameAs,"")]///text()''
      )
    ),
    labels,
    oxy_xpath(
      oxy_concat(
        '//listBibl/biblStruct[contains(@xml:id,
        '')",
        attr(sameAs,"")]///text()''
      )
    ),
    color,
    fontInherit,
    onChange,
    id,
  );
  font-size:15px;
}
```

Résultat : Combobox toujours vide.

Essai 13

```

${ask(
'Référence bibliographique complète',
combobox,
('${xpath_eval(@@sameAs)}':
`${xpath_eval(ancestor::biblStruct[contains("${xpath_eval(@xml:id)}","${xpath_eval(@@sameAs)}"]]/string-join(./text(),
",",
"${xpath_eval(@@sameAs)}')`}

```

Résultat : Combobox vide, cela n'affiche même pas la valeur par défaut (@sameAs).

Essai 14 :

```

${ask(
'Référence bibliographique complète',
combobox,
('${xpath_eval(@@sameAs)}':
`${xpath_eval("//biblStruct[contains("${xpath_eval(@xml:id)}","${xpath_eval(@@sameAs)}"]]/string-
-join(//biblStruct/text(),
",
")),
`${xpath_eval(@@sameAs)}')

```

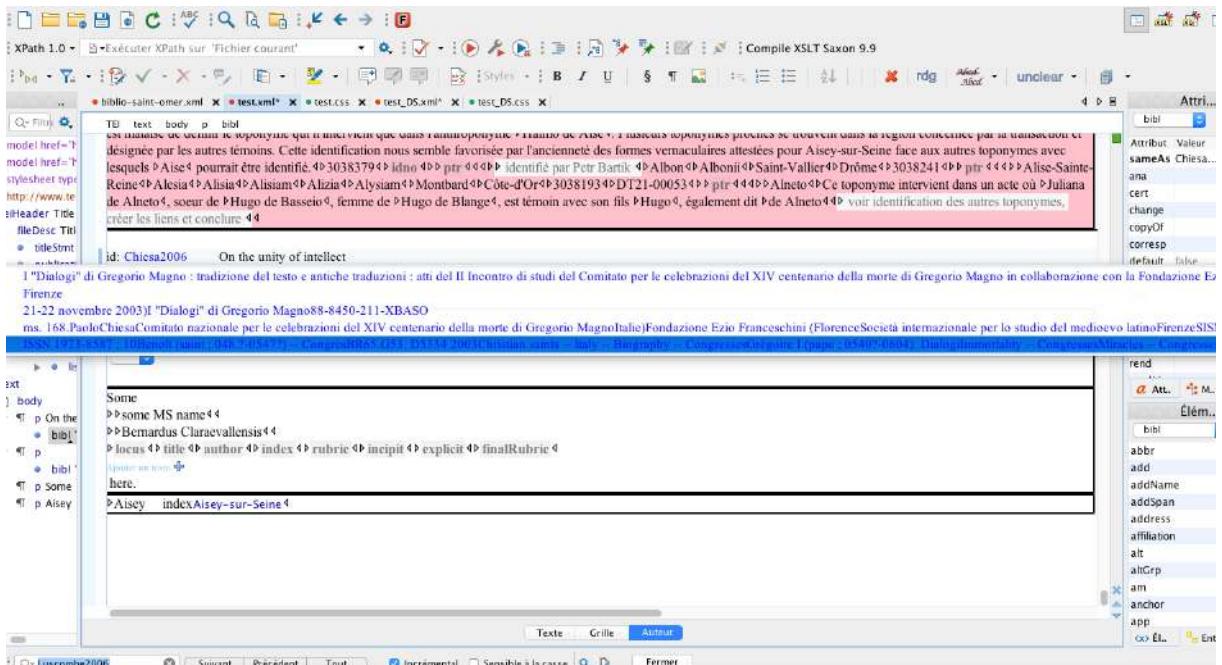
CSS sans oxy-button :

```

bibl[sameAs]:after{
  display:block;
  color:#84bff9;
  font-size:x-small;
  content:
    edit,
    editable,
    values,
    oxy_xpath(
      oxy_concat(
        '//listBibl/biblStruct[contains(@xml:id,
          '',
          attr(sameAs,""))]/text()'
      )
    ),
    labels,
    oxy_xpath(
      oxy_concat(
        '//listBibl/biblStruct[contains(@xml:id,
          '',
          attr(sameAs,""))]/text()'
      )
    ),
    color,
    fontInherit,
    onChange,
    id,
  );
  font-size:15px;
}
oxy_combobox(
  '#text',
  false,
)

```

Résultat : La référence complète s'affiche avec l'action linserBiblio_3.



Mais cela ne marche pas quand plusieurs références sont sélectionnées : inclure la possibilité dans l'@sameAs de contenir deux xml:id avec le séparateur "|".

Essai 15

```

${ask(
'Référence                                bibliographique                               complète',
combobox,
('${xpath_eval(./@sameAs)}': '${xpath_eval("//biblStruct[contains(@xml:id,
"${xpath_eval(./@sameAs)}")]/string-join(//text(),
",",
"${xpath_eval(./@sameAs)})}'})

```

Résultat : Ça ne marche que pour une seule référence sélectionnée.

Essai 16

Tentative d'inclure la fonction substring-after() dans contains() pour signaler la possibilité d'un séparateur entre deux id dans l'@sameAs. Risque que ça ne prenne en compte que les références multiples.

```

${ask(
'Référence                                bibliographique                               complète',
combobox,
('${xpath_eval(./@sameAs)}':
'${xpath_eval("//biblStruct[contains(@xml:id, substring-after("${xpath_eval(./@sameAs)}", " | "))]/string-join(//text(),
",",
"${xpath_eval(./@sameAs)})}'})

```

Résultat : Cela fonctionne pour une référence, et lorsque deux sont sélectionnées, l'action ne retourne que la référence bibliographique de l'id après le séparateur “|” dans @sameAs, donc ajouter la fonction substring-before() pour récupérer aussi l'id avant le séparateur.

Essai 17

```

${ask(
'Référence                                bibliographique                      complète',
combobox,
('${xpath_eval(./@sameAs)}':
`${xpath_eval("//biblStruct[contains(@xml:id, substring-after("${xpath_eval(./@sameAs)}", "|"))]/string-join(./text(), " "))}`;
`${xpath_eval(./@sameAs)}':
`${xpath_eval("//biblStruct[contains(@xml:id,      substring-before("${xpath_eval(./@sameAs)}",
" |"))]/string-join(./text(), ",           ")))}`),
`${xpath_eval(./@sameAs)})'}

```

Résultat : Cela fonctionne pour une référence, et lorsque deux sont sélectionnées, l'action ne retourne que la référence bibliographique de l'id avant le séparateur “|” dans @sameAs, donc ajouter différemment la fonction substring-before() pour récupérer aussi l'id avant le séparateur.

Essai 18

```

${ask(
'Référence                                bibliographique                      complète',
combobox,
('${xpath_eval(./@sameAs)}':
`${xpath_eval("//biblStruct[contains(@xml:id,  "${xpath_eval(./@sameAs)}")]/string-join(./text(),  " ",
"))}`;
`${xpath_eval(./@sameAs)}':
`${xpath_eval("//biblStruct[contains(@xml:id,      substring-after("${xpath_eval(./@sameAs)}",
" |"))]/string-join(./text(), ", ))}`),
`${xpath_eval(./@sameAs)}':
`${xpath_eval("//biblStruct[contains(@xml:id,      substring-before("${xpath_eval(./@sameAs)}",
" |"))]/string-join(./text(), ",           ")))}`),
`${xpath_eval(./@sameAs)})'}

```

Dans la CSS :

```

bibl[sameAs]:after{
  display:block;
  color:#84bff9;
  font-size:x-small;
  content:oxy_button(transparent, true, action, oxy_action('inserBiblio_3')
    id, oxy_combobox(edit, editable, values,
    oxy_xpath(

```

```

        oxy_concat(
        '//listBibl/biblStruct[contains(@xml:id,
        '//listBibl/biblStruct[contains(@xml:id,
        '//listBibl/biblStruct[contains(@xml:id,
        )
        ),
        labels,
        oxy_xpath(
            oxy_concat(
                '//listBibl/biblStruct[contains(@xml:id,
                '//listBibl/biblStruct[contains(@xml:id,
                '//listBibl/biblStruct[contains(@xml:id,
                )
                ),
                color,
                fontInherit,
                onChange,
                id,
                )
                );
        font-size:15px;
    }

```

Résultat : Cela fonctionne pour une seule référence sélectionnée, mais lorsqu'il y'en a plusieurs, la combobox affiche la référence avant le séparateur.

Essai 19 :

```

${ask(
'Référence                                bibliographique                      complète',
combobox,
('${xpath_eval(./@sameAs}')':
 '${xpath_eval("//biblStruct[contains(@xml:id,           substring-after("${xpath_eval(./@sameAs})",
" | ")]/string-join(./text(), ", ")}';
`${xpath_eval(./@sameAs}')':
 `${xpath_eval("//biblStruct[contains(@xml:id,           substring-before("${xpath_eval(./@sameAs})",
" | ")]/string-join(./text(),
",           ")))}),
`${xpath_eval(./@sameAs}')'}

```

Résultat : Cela fonctionne pour une seule référence sélectionnée, mais lorsqu'il y'en a plusieurs, la combobox affiche la référence après le séparateur seulement.

09/06

Conseils : Prédéfinir une direction à prendre avant de coder , le pourquoi et le comment ; Modifier un élément à chaque étape pour comprendre ce qui ne fonctionne pas.

Solution : Parmétrage d'un ensemble d'actions se complétant.

- InserBiblio_2 permet de saisir un mot-clef qui appelle une référence bibliographique

```

<action>
    <field name="id">
        <String>inserBiblio_2</String>
    </field>
    <field name="name">
        <String>inserBiblio_2 (insère @sameAs)</String>
    </field>
    <field name="description">
        <String>Demande chaîne de caractères</String>
    </field>
    <field name="largeIconPath">
        <String></String>
    </field>
    <field name="smallIconPath">
        <String></String>
    </field>
    <field name="accessKey">
        <String></String>
    </field>
    <field name="accelerator">
        <null/>
    </field>
    <field name="actionModes">
        <actionMode-array>
            <actionMode>
                <field name="xpathCondition">
                    <String></String>
                </field>
                <field name="argValues">
                    <serializableOrderedMap>
                        <entry>
                            <String>editAttribute</String>
                            <String>true</String>
                        </entry>
                        <entry>
                            <String>name</String>
                            <String>sameAs</String>
                        </entry>
                        <entry>
                            <String>namespace</String>
                            <String></String>
                        </entry>
                        <entry>
                            <String>value</String>
                            <String>${ask('Saisir      un      mot', generic,
                                '${selection}')}</String>
                        </entry>
                    </serializableOrderedMap>
                </field>
                <field name="operationID">

```

```

<String>ro.sync.ecss.extensions.commons.operations.ChangeAttributeOperation</String>
    </field>
    </actionMode>
    </actionMode-array>
</field>
<field name="enabledInReadOnlyContext">
    <Boolean>false</Boolean>
</field>
</action>

```

- InserBiblio_3 permet de lister des références dans un @sameAs séparés par le signe ‘€’

```

<action>
    <field name="id">
        <String>inserBiblio_3</String>
    </field>
    <field name="name">
        <String>inserBiblio_3 (liste références)</String>
    </field>
    <field name="description">
        <String>Liste références pertinentes</String>
    </field>
    <field name="largeIconPath">
        <String></String>
    </field>
    <field name="smallIconPath">
        <String></String>
    </field>
    <field name="accessKey">
        <String></String>
    </field>
    <field name="accelerator">
        <null/>
    </field>
    <field name="actionModes">
        <actionMode-array>
            <actionMode>
                <field name="xpathCondition">
                    <String></String>
                </field>
                <field name="argValues">
                    <serializableOrderedMap>
                        <entry>
                            <String>name</String>
                            <String>sameAs</String>
                        </entry>
                        <entry>
                            <String>value</String>

```

```

                <String>${xpath_eval("//biblStruct[contains(., ${xpath_eval('./@sameAs)})]/concat('€', @xml:id))}</String>
            </entry>
        </serializableOrderedMap>
    </field>
    <field name="operationID">

<String>ro.sync.ecss.extensions.common.operations.ChangeAttributeOperation</String>
    </field>
    </actionMode>
    </actionMode-array>
</field>
<field name="enabledInReadOnlyContext">
    <Boolean>false</Boolean>
</field>
</action>
```

- `Display_biblStruct` permet d'afficher la référence bibliographique d'un seul id dans l'`@sameAs`

```

<action>
    <field name="id">
        <String>display_biblStruct</String>
    </field>
    <field name="name">
        <String>affiche référence</String>
    </field>
    <field name="description">
        <String></String>
    </field>
    <field name="largeIconPath">
        <String></String>
    </field>
    <field name="smallIconPath">
        <String></String>
    </field>
    <field name="accessKey">
        <String></String>
    </field>
    <field name="accelerator">
        <null/>
    </field>
    <field name="actionModes">
        <actionMode-array>
            <actionMode>
                <field name="xpathCondition">
                    <String>//bibl</String>
                </field>
                <field name="argValues">
                    <serializableOrderedMap>
```

```

        <entry>
            <String>name</String>
            <String>sameAs
        </String>
        </entry>
        <entry>
            <String>value</String>
            <String>${ask(
'Référence bibliographique complète',
combobox,
('${xpath_eval(.):@sameAs}':'${xpath_eval("//biblStruct[@xml:id='${xpath_eval(.):@sameAs}']/stri
ng-join(./text(), ", ")}'),
`${xpath_eval(.):@sameAs})}'</String>
            </entry>
            </serializableOrderedMap>
        </field>
        <field name="operationID">

```

```

<String>ro.sync.ecss.extensions.commons.operations.ChangeAttributeOperation</String>
        </field>
        </actionMode>
        </actionMode-array>
    </field>
    <field name="enabledInReadOnlyContext">
        <Boolean>false</Boolean>
    </field>
</action>

```

- InserBiblio_3a permet de rechercher toutes les références correspondantes inscrites dans l'@sameAs. La fonction contains() dans une syntaxe xslt via un xsl:for-each permet de boucler sur les @xml:id comprenant des caractéristiques communes avec ce qui est inscrit dans l'@sameAs.

```

<action>
    <field name="id">
        <String>inserBiblio_3a</String>
    </field>
    <field name="name">
        <String>retrieveRefs</String>
    </field>
    <field name="description">
        <String>cherche toutes références correspondant à
@sameAs</String>
    </field>
    <field name="largeIconPath">
        <String>${framework}/icons/list-add.png</String>
    </field>
    <field name="smallIconPath">
        <String>${framework}/icons/list-add.png</String>
    </field>
    <field name="accessKey">
```

```

        <String></String>
    </field>
    <field name="accelerator">
        <null/>
    </field>
    <field name="actionModes">
        <actionMode-array>
            <actionMode>
                <field name="xpathCondition">
                    <String></String>
                </field>
                <field name="argValues">
                    <serializableOrderedMap>
                        <entry>
                            <String>script</String>
                            <String>&lt;?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 xmlns:saxon="http://saxon.sf.net/"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:tei="http://www.tei-c.org/ns/1.0"
 exclude-result-prefixes="xs saxon" version="2.0">

<xsl:param name="currentElementLocation"/>

<xsl:template match="/">
    &lt;!--
        sourceLocation parameter was set to /* to have a larger context.
        We can pinpoint the element that contained the caret
        using $currentElementLocation

        select="saxon:eval(saxon:expression($currentElementLocation))"
-->
    &lt;xsl:variable name="sameAs" select="@sameAs"/>
    &lt;xsl:element name="bibl" namespace="http://www.tei-c.org/ns/1.0">
        &lt;xsl:attribute name="sameAs">
            &lt;xsl:for-each      select="doc(${cfdu}/biblio-saint-omer.xml)//biblStruct[contains(., $sameAs)]">
                &lt;xsl:text>&lt;/xsl:text>
                &lt;xsl:value-of select="@xml:id"/>
                &lt;xsl:text>:&lt;/xsl:text>
                &lt;xsl:value-of select="string-join(text(), ', ')"/>
                &lt;xsl:text>';&lt;/xsl:text>
            &lt;/xsl:for-each>
        &lt;/xsl:attribute>
        &lt;xsl:value-of select=". "/>
        &lt;xsl:text> (&lt;/xsl:text>&lt;xsl:value-of select="$sameAs"/>&lt;/xsl:text>)&lt;/xsl:text>
    &lt;/xsl:element>
&lt;/xsl:template>
```

```

<xsl:stylesheet></String>
    </entry>
    <entry>
        <String>sourceLocation</String>
        <String></String>
    </entry>
    </serializableOrderedMap>
</field>
<field name="operationID">

<String>ro.sync.ecss.extensions.common.operations.XSLTOperation</String>
    </field>
    </actionMode>
    </actionMode-array>
</field>
<field name="enabledInReadOnlyContext">
    <Boolean>false</Boolean>
</field>
</action>

```

! l' xsl:template match="*" sert à partir de n'importe quel élément n'importe où
xsl:template match="/" sert à partir de l'élément le plus englobant du document, donc cela marche pour un document qui a des éléments
xsl:template match="/" matche seulement la racine du document, donc cela fonctionne aussi pour un document vide
Il ne vaut mieux pas parler "d'élément racine" mais de "nœud racine", car c'est l'ancêtre de tous les autres nœuds dans un document.

- InserBiblio_4 permet de proposer un choix de références selon l'@sameAs, soit toutes celles qui comportent un mot-clé en commun

```

<action>
    <field name="id">
        <String>inserBiblio_4</String>
    </field>
    <field name="name">
        <String>inserBiblio_4 (choix de références)</String>
    </field>
    <field name="description">
        <String>Liste références pertinentes</String>
    </field>
    <field name="largeIconPath">
        <String></String>
    </field>
    <field name="smallIconPath">
        <String></String>
    </field>
    <field name="accessKey">
        <String></String>
    </field>

```

```

</field>
<field name="accelerator">
    <null/>
</field>
<field name="actionModes">
    <actionMode-array>
        <actionMode>
            <field name="xpathCondition">
                <String></String>
            </field>
            <field name="argValues">
                <serializableOrderedMap>
                    <entry>
                        <String>name</String>
                        <String>sameAs</String>
                    </entry>
                    <entry>
                        <String>value</String>
                        <String>${ask(
'Référence bibliographique complète',
combobox,
(
'${xpath_eval(tokenize(@sameAs,           "€")[1])}':${xpath_eval("//biblStruct[@xml:id
"${xpath_eval(tokenize(@sameAs, "€")[1])}"]/string-join(./text(), ", "))}';
'${xpath_eval(tokenize(@sameAs,           "€")[2])}':${xpath_eval("//biblStruct[@xml:id
"${xpath_eval(tokenize(@sameAs, "€")[2])}"]/string-join(./text(), ", "))}';
'${xpath_eval(tokenize(@sameAs,           "€")[3])}':${xpath_eval("//biblStruct[@xml:id
"${xpath_eval(tokenize(@sameAs, "€")[3])}"]/string-join(./text(), ", "))}';
'${xpath_eval(tokenize(@sameAs,           "€")[4])}':${xpath_eval("//biblStruct[@xml:id
"${xpath_eval(tokenize(@sameAs, "€")[4])}"]/string-join(./text(), ", "))}';
'${xpath_eval(tokenize(@sameAs,           "€")[5])}':${xpath_eval("//biblStruct[@xml:id
"${xpath_eval(tokenize(@sameAs, "€")[5])}"]/string-join(./text(), ", "))}';
'${xpath_eval(tokenize(@sameAs,           "€")[6])}':${xpath_eval("//biblStruct[@xml:id
"${xpath_eval(tokenize(@sameAs, "€")[6])}"]/string-join(./text(), ", "))};
),
'autre'
)}</String>
            </entry>
        </serializableOrderedMap>
    </field>
    <field name="operationID">

<String>ro.sync.ecss.extensions.common.operations.ChangeAttributeOperation</String>
    </field>
    </actionMode>
    </actionMode-array>
</field>
<field name="enabledInReadOnlyContext">
    <Boolean>false</Boolean>
</field>

```

```
</action>
</action-array>
</field>
```

Si l'on peut jouer avec la CSS, le paramétrage directement dans le framework l'allège. Elle sert alors essentiellement à appeler l'action via une fonction oxy-action().

On peut ainsi faire des listes de références selon le type de document, etc.. Cela est mieux pour faire des recherches à la fois plus rapides et plus lisibles, mais aussi pour produire des données plus nombreuses de meilleure qualité.

5.4.3 CSS+Action2

CSS

- si pas de @sameAs, bouton « rechercher référence »
- boîte de dialogue
- importer référence

6 Exploitation d'informations : usages liturgiques

6.1 Réseau des usages

- 1) Reproduire réseau : Cf. poster

6.2 Histoire des manuscrits

6.3

6.4