CapiTainS Toolkit, Digital Editing and Data Reuse

Thibault Clérice

Les outils CapiTainS, l'édition numérique et l'exploitation des textes

CapiTainS toolkit, digital editing and data reuse

Résumé

L'édition numérique a pris une plus grande place dans les problématiques des chercheurs en sciences humaines. Or, les compétences nécessaires à leur valorisation sont souvent absentes ou coûteuses. Bien souvent, la production de telles éditions envisage l'interface majoritairement du point de vue *design* et de la validité scientifique. En faisant ainsi, ces projets perdent de vue la possibilité de réutiliser le texte par d'autres projets.

Issu de la collaboration de membres des équipes de *Perseids* et *Perseus* à Tufts et de la *Humboldt Chair of Digital Humanities* de l'université de Leipzig, le projet CapiTainS voudrait fournir une solution à ce problème. CapiTainS propose des solutions *open source*, réexploitables et extensibles visant à simplifier le développement et la valorisation d'éditions numériques. En fournissant cet ensemble d'outils, il permet aux chercheurs d'inscrire leur projet dans un réseau de données ouvertes et pleinement réutilisables. Par ailleurs, en offrant des logiciels simplifiant la réutilisation des données, CapiTainS cherche à réduire le coût nécessaire à l'exploitation des standards qu'il soutient.

Mots clefs : édition, standard, moissonnage, valorisation, réutilisation

Summary

Digital editing has become a major focus in Humanities' projects. Unfortunately, research laboratories often lack the skills or the time to digitally publish efficiently the outcome of their research. Most of the time, the emphasis in the production of digital editions is put on design and academic rigour. By doing so, these projects forget another audience: potential reusers of their data.

Collaboratively founded by members of the Perseids and Perseus staff at Tufts and of the Humboldt Chair of Digital Humanities at the university of Leipzig, the CapiTainS project wishes to be a solution. CapiTainS offers open source, reusable and extensible software solutions to simplify developing and promotion of digital editions. Using this toolkit, researchers can insert their project in a network of open and reusable data. Focusing also on potential data users, CapiTainS offers standard libraries that simplify communicating with API or local resources and mining them.

Introduction

> « *Harvesting document metadata has proved successful especially where documents are about a single topic and relatively short. Although these conditions hold for most scholarly articles in e-print archives, the primary data, for example, on which such articles are based are not as tractable »* David Smith[1]

1 David A. Smith, Anne Mahoney et Gregory Crane, « Integrating Harvesting into Digital Library Content », *Proceedings of the 2Nd ACM/IEEE-CS Joint Conference on Digital Libraries*, New York, NY, USA, ACM, 2002, (CDL '02), p. 183–184, <http://doi.acm.org/10.1145/544220.544257>.

« Le destin d'une technologie, indépendamment de son intérêt ou de sa qualité, tient aussi à son degré d'appropriation par les développeurs » Gautier Poupeau[2]

La multiplication des efforts de numérisation de textes des littératures classique, médiévale et moderne a durablement changé notre rapport à la production et à l'exploitation d'un écrit. D'une part, de nouvelles formes de recherches se sont formées. On analyse les textes en corpus, en masse. Pour les études médiévales par exemple, la stylométrie et la question de l'auctorialité dans les œuvres anonymes ont bénéficié de l'accès libre à ce type de ressources[3]. D'autre part, des chercheurs et des institutions de financement de la recherche ont conjointement cherché à obtenir une plus grande audience en mettant à disposition les fruits de la recherche au plus grand nombre. Le numérique a permis de réduire la difficulté d'accès aux savoirs pour les publics professionnels ou non, surtout grâce à l'*open data* et à l'*open access*.

Dans ce contexte, l'édition numérique des textes médiévaux connaît les mêmes problématiques que celle des textes classiques : le résultat doit être à la fois scientifiquement validé, archivable et standard, publiquement disponible dans des interfaces de lecture. Or, le développement d'interfaces et de ressources standards implique un coût important pour les équipes de recherche. CapiTainS est un projet *open source* cherchant à développer un ensemble de services, de modèles et d'interfaces de programmation pour la valorisation et l'exploitation d'éditions numériques et de leurs données. Fondé par Bridget Almas et moi-même en 2015, CapiTainS a été conçu comme un projet collaboratif, transdisciplinaire et désinstitutionnalisé[4]. En fournissant des réponses aux problématiques de modélisation, de communication et d'exploitation, CapiTainS réduit la difficulté de chacune de ces trois étapes pour des publics variés (éditeurs, développeurs, chercheurs, lecteurs) dans des contextes différents (hors-ligne, bibliothèque numérique, exposition virtuelle)[5].

Modéliser

L'accès numérique aux données pose très rapidement la question de la modélisation : comment peut-on transférer un savoir analogique dans espace numérique ? Pour répondre à cette question, il faut interroger la forme que le texte prend, à la fois dans ses réalisations intégrales (livres, manuscrits, inscriptions) et ses usages parcellaires (citation). Les études classiques en latin ou en grec ancien ont pratiqué une normalisation relative de ces systèmes au fil des siècles. On cite ainsi le chant 5 vers 1 de *L'Énéide* de Virgile « Verg. Aen. 5.1 ». Ce système implique pour sa réalisation numérique deux composantes : un texte hiérarchisé ou conçu comme tel, un modèle informatique permettant son interprétation et *a priori* son identification.

2 Gautier Poupeau, « La donnée : nouvelle perspective pour les bibliothèques », in Emmanuelle Bermès (éd.). *Vers de nouveaux catalogues*, éd. Emmanuelle Bermès, France, p. 159-171.

3 Quelques exemples : Jean-Baptiste Camps et Florian Cafiero, « Genealogical variant locations and simplified stemma: a test case », Brepols, 2012, p. 69-93, <https://halshs.archives-ouvertes.fr/halshs-01435633/document>. Mike Kestemont, Sara Moens et Jeroen Deploige, « Collaborative authorship in the twelfth century: A stylometric study of Hildegard of Bingen and Guibert of Gembloux », *Literary and Linguistic Computing*, vol. 30/2, juin 2015, p. 199-224, <DOI : 10.1093/llc/fqt063>.

4 Il a été financé à l'origine par la *Humboldt Chair of Digital Humanities* de Leipzig et le projet Perseids.

5 Quand cela a été possible, les exemples sont tirés du corpus en cours d'édition d'un travail de thèse. Ariane Pinche, *Édition nativement numérique des œuvres hagiographiques « Li Seint Confessor » de Wauchier de Denain d'après le manuscrit 412 de la Bibliothèque nationale de France,* université Lyon 3, en cours.

Les textes comme objets hiérarchisés

En 1996, l'article « Refining our notion of what text really is: the problem of overlapping hierarchies » de A. Renear, E. Mylonas et D. Durand propose une modélisation des textes[6]. Cette modélisation, *Ordered Hierarchy of Content Objects* (OHCO), est une approche très semblable aux principes de la programmation orientée objet, des arbres et des graphes. Le texte y est analysable comme un ensemble d'objets hiérarchisés : il est un arbre, où chaque élément peut posséder un ou plusieurs voisins, un ou plusieurs enfants, mais un seul parent. Cette approche est très similaire à la manière dont les éditeurs de textes anciens fragmentent leur corpus. *L'Énéide* de Virgile est un ensemble de nœuds de données hiérarchisés suivant l'ordre « *Énéide* → Chant → Vers → Mot → Caractère » qui est généralement reconstitué sous la forme « Virg. Aen. VI.5 ».

Cette vision d'un texte hiérarchisé conditionne toute une vision de l'encodage des textes et est clairement utilisée par le standard TEI en édition textuelle numérique[7]. Ce paradigme instaure, via son langage d'implémentation XML[8], une hiérarchisation sémantique de l'information. Ceci se traduit par l'imbrication de divisions (*div*), de groupes de lignes (*lg*), de lignes (l) et de paragraphes (*p*) qui forment ensemble une hiérarchie de l'information textuelle qui se traduit par une mise en page particulière qui, elle-même, induit parfois des divisions sémantiques (chapitres, strophe, etc.). Cette structuration d'interface, de visualisation, peut être complétée par une structuration grammaticale ou par un apparat pour les éditions scientifiques, permettant de reconnaître les phrases, les lemmes et leurs variantes orthographiques. Ces structurations peuvent être complétées par un autre système (*milestone, pb*) qui vient alors seulement signifier l'existence d'un système concurrent à celui, plus canonique, du document[9]. Il s'agit alors de montrer des divisions thématiques ou documentaires.

L'article de Reiner, Mylonas et Durand soulève cependant un problème de taille : qu'en est-il des textes dont le contenu n'a pas vocation à être ordonné ou dont l'agencement n'est pas clair ou fixe, les inscriptions par exemple ? Visant un certain pragmatisme, la position de CapiTainS a été la suivante : il est important de fournir un mode de citation, d'atomisation du texte sans pour autant qu'il soit vu comme seul et unique. L'éditeur, en proposant une édition, crée de fait une hiérarchisation des données.

Dans cette perspective, un mode de citation est un moyen de transmission. Il possède des limites et fait l'objet d'une décision scientifique : le choix de CapiTainS est de pousser les éditeurs à proposer une hiérarchie de leurs textes qui fasse sens dans le cadre de leur projet afin que celui-ci soit partageable

---

6 Allen H. Renear, Elli Mylonas et David Durand, « Refining our notion of what text really is: The problem of overlapping hierarchies », 1993, <https://www.ideals.illinois.edu/handle/2142/9407>.

7 « A Gentle Introduction to XML - The TEI Guidelines », en ligne : <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/SG.html#SG152>. Consulté le 13 mai 2017, chap. 4. Complicating the Issue.

8 *Extensible Markup Language* (« langage de balisage extensible »). Le XML est un langage qui permet d'encoder des informations. Il est largement utilisé pour l'encodage de ressources textuelles de part son système de balise. Un paragraphe peut être noté « Texte ».

9 « <milestone> (borne) marque un point permettant de délimiter les sections d'un texte selon un autre système que les éléments de structure ; une balise de ce type marque une frontière. » « TEI element milestone », en ligne : <http://www.tei-c.org/release/doc/tei-p5-doc/fr/html/ref-milestone.html>. Consulté le 4 mai 2017.

Guidelines

C'est pourquoi les *guidelines* CapiTainS 2.0.0 se concentrent à rendre possible et simple l'explicitation d'un modèle de citation[10]. Historiquement, CapiTainS a été fondé pour produire des textes compatibles avec le standard *Canonical Text Services* (CTS) bien que l'on souhaite soutenir d'autres formes de modélisations. Ces *guidelines* se concentrent sur deux objectifs : expliciter et documenter des collections de textes, expliciter et rendre opérable des systèmes de citation.

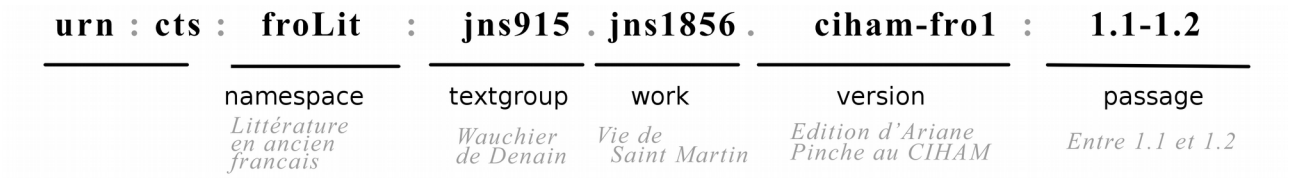

Figure 1 : Schéma des URN CTS

Le modèle CTS repose sur un système d'identifiant composé de cinq grands membres (cf. Figure 1). Ces URN comportent un domaine, un identifiant *textgroup*, un identifiant *work*, un identifiant *version* puis enfin un identifiant pour le *passage*. Ces identifiants, en dehors du dernier, reposent sur une hiérarchisation des éléments de métadonnées d'un texte :

- le domaine a été utilisé comme un outil de dénomination d'un corpus générique (*froLit* pour littérature ancien français) ou d'un corpus (*pdlrefwk* pour *Perseus Digital Library Reference Work* qui comprend des dictionnaires) ;
- le *textgroup*, bien que représentant souvent un auteur, identifie surtout un ensemble de textes qu'il fait sens de rassembler : les textes homériques, les livres de l'*Ancien Testament*, les œuvres de Cicéron, un ensemble d'inscriptions localisées géographiquement ;
- l'identifiant *work* représente le concept d'une œuvre. Bien que l'on cite souvent *L'Odyssée* d'Homère, nous ne précisons pas toujours quel manuscrit, quelle édition scientifique se cache derrière cette expression[11] ;
- l'identifiant *version* sert à identifier une édition numérique spécifique, une incarnation de l'œuvre.

Les normes CapiTainS proposent de répercuter cette norme dans un système de hiérarchie des fichiers et dossiers afin de faciliter la gestion de larges corpus[12]. Chaque niveau contient ensuite un fichier commun de métadonnées (cf. Figure 2). Afin d'éviter une redondance des données, il suffit de décrire une fois le *textgroup* ou l'œuvre conceptuelle pour l'ensemble de leurs réutilisations (cf. Code 1). Les descriptions des implémentations se font aussi à l'extérieur de l'édition en elle-même (cf. Code 2) pour éviter d'imposer un schéma à l'ensemble des éditeurs. Les métadonnées de base sont simples (*groupname* pour le *textgroup, title* pour le *work, label,*

10 « Capitains/Capitains.github.io: 2.0.0 », *Zenodo*, <DOI : 10.5281/zenodo.570516>.

11 Le concept CTS de *work* se rattache au concept homonyme dans la norme *Functional Requirements for Bibliographic Records (*FRBR).

12 Le corpus grec ancien de Perseus comporte en effet 1 423 textes. « PerseusDL/canonical-greekLit 0.0.58 », *Zenodo*, <DOI : 10.5281/zenodo.570421>.

description et *about* pour les implémentations), mais peuvent être complétées par des métadonnées RDF via le nœud *structured-metadata* pour être ensuite exploitées dans divers outils applicatifs.

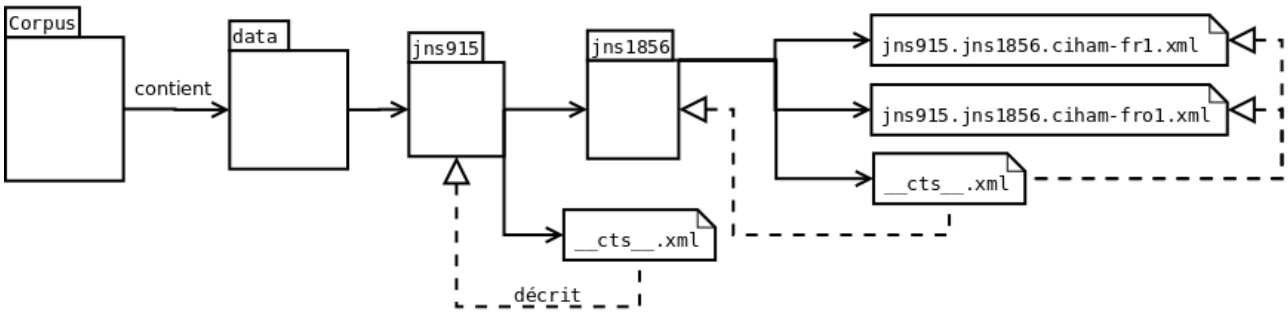

Figure 2 : Structure des dossiers et relations entre les fichiers. Ici, les deux fichiers (fr1 et fro1) sont une édition et une traduction

```
<textgroup urn="urn:cts:froLit:jns915"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:cpt="http://purl.org/capitains/ns/1.0#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:db="http://dbpedia.org/ontology/"
  xmlns="http://chs.harvard.edu/xmlns/cts">
  <groupname xml:lang="fre">Wauchier de Denain</groupname>
  <cpt:structured-metadata>
    <owl:sameAs>http://catalogue.bnf.fr/ark:/12148/cb12103422r</owl:sameAs>
    <foaf:gender>male</foaf:gender>
    <db:activeYearsStartYear>1200</db:activeYearsStartYear>
    <db:activeYearsEndYear>1250</db:activeYearsEndYear>
  </cpt:structured-metadata>
</textgroup>
```

Code 1 : Contenu d'un fichier de description de textgroup *(exemple : Wauchier de Denain)*

```
<work
   groupUrn="urn:cts:froLit:jns915" urn="urn:cts:froLit:jns915.jns1856"
   xmlns="http://chs.harvard.edu/xmlns/cts"
   xmlns:dct="http://purl.org/dc/terms/"
   xmlns:cpt="http://purl.org/capitains/ns/1.0#"
   >
   <title xml:lang="fre">Vie de saint Martin</title>
   <edition workUrn="urn:cts:froLit:jns915.jns1856" urn="urn:cts:froLit:jns915.jns1856.ciham-
fro1">
      <label xml:lang="fre">Vie de saint Martin</label>
      <description xml:lang="fre">édition du manuscrit :
         PARIS, Bibliothèque nationale de France, Manuscrits, fr. 00412
      </description>
      <cpt:structured-metadata>
         <dct:language>fro</dct:language>
         <dct:creator>http://orcid.org/0000-0002-7843-5050</dct:creator>
         <dct:creator>Ariane Pinche</dct:creator>
         <dct:isVersionOf>http://jonas.irht.cnrs.fr/manuscrit/45533</dct:isVersionOf>
         <dct:isVersionOf>http://gallica.bnf.fr/ark:/12148/btv1b84259980</dct:isVersionOf>
      </cpt:structured-metadata>
   </edition>
</work>
```

Code 2 :Contenu d'un fichier de description de work et de son édition

Le second objectif des *guidelines* est de permettre d'encoder un système de citation à l'intérieur d'un texte. Bien souvent, ce système de citation est implicite. La succession de div1, div2 puis de l pourraient indiquer un système de citation à trois niveaux. Nous avons décidé avec CapiTainS de rendre leur explicitation obligatoire à travers l'usage des simples outils fournis par le standard TEI, le XML et le XPath[13]. En effet, ce standard propose un ensemble de balise refsDecl qui ont pour objectif la « déclaration de système de référence »[14]. CapiTainS recommande ensuite la déclaration des niveaux de citation via l'utilisation de balises cRefPattern (cf. *Code 3*) qui associent des XPath (attribut replacementPattern) à un nom de niveau (attribut n) et à une expression régulière représentant le système de citation (attribut matchPattern). À travers cette explicitation du système de citation, de la langue du texte et de son identifiant, le texte est à la fois lisible pour l'éditeur et la machine, tant du point de vue de l'identification de l'objet que de ses sous-objets.

13 Ce langage permet de traverser l'arbre du fichier. Ainsi, si un texte est composé de représentant des chants et de <l> représentant des vers, l'accès à la ligne comportant le numéro n='1' dans le chant n='2' se fera /div[@n='2']/l[@n='1'].

14 Cette utilisation de ce tag pour cela a aussi été proposé dans Joel Kalvesmaki, « Canonical References in Electronic Texts: Rationale and Best Practices », vol. 8/2, 2014, <http://www.digitalhumanities.org/dhq/vol/8/2/000181/000181.html>.

```
<refsDecl n="CTS">
  <cRefPattern n="section" matchPattern="(\w+).(\w+)"
    replacementPattern="#xpath(/tei:TEI/tei:text/tei:body/tei:div[@n='$1']/tei:div[@n='$2'])">
    <p>Ce pointeur extrait les sections dont l'identifiant correspond à x.y (où x et y sont des ensembles de caractères alphanumériques)</p>
  </cRefPattern>
  <cRefPattern n="chapitre" matchPattern="(\w+)"
    replacementPattern="#xpath(/tei:TEI/tei:text/tei:body/tei:div[@n='$1'])">
    <p>Ce pointeur extrait les chapitres dont l'identifiant correspond à x (où x est des ensembles de caractères alphanumériques)</p>
  </cRefPattern>
</refsDecl>
```

Code 3: Exemple de déclaration refsDecl d'un système de citation CapiTainS

En proposant ces recommandations d'encodage et de séparation des objets, CapiTainS offre une couche supplémentaire de standardisation simple et légère pour l'édition de textes, sans pour autant imposer un format trop contraint. Ces règles de conduite permettent en effet d'agréer un grand nombre de projets tout en divisant les contraintes de catalogage, de citation et d'encodage « scientifique ».

Communiquer

L'encodage d'un texte n'est que l'étape préalable à leur accès, il faut aussi pouvoir les lire. Le développement d'interface occupe souvent la plus grande part des budgets de recherche en ce qu'il nécessite bien souvent le développement d'un ensemble de nouveaux outils pour visualiser le texte. Cet accès se fait souvent à travers des interfaces utilisateurs ignorant souvent les interfaces machines qui permettent alors de réutiliser des textes à travers des interfaces différentes. CapiTainS propose un ensemble de logiciels *open source* et libres qui permet aux producteurs de données de réduire le temps de développement pour la mise à disposition de textes dans des standards acceptés en laissant la possibilité de personnaliser l'apparence des interfaces utilisateurs.

API et Standards : mettre à disposition ses textes et métadonnées

Les interfaces machines, ou API (*Application Programming Interface*), permettent de communiquer avec un site ou un logiciel au niveau programmeur. Ces API, web ou logicielles, sont des points d'accès extérieurs aux données ou outils accessibles à l'intérieur des sites ou programmes. Pour un logiciel, il s'agit de donner accès à ses fonctionnalités à un logiciel tiers, pour un service web il s'agira d'offrir des points d'accès à ses données.

Le développement du web s'est orienté ces dernières années vers ce que certains ont appelé le web 3.0, celui des données connectées. Les données connectées permettent aux programmes de

réutiliser les informations présentes dans un site sans pour autant avoir à lire ce site comme un humain. Il est en effet peu probable qu'un algorithme d'analyse de texte soit sensible à la couleur d'une police, tandis que la structuration sémantique des données leur fournira un ensemble d'informations interprétables. Ce procédé, cette mise en API, est au centre du quotidien de chacun. Les applications Twitter, Google ou Facebook communiquent toutes uniquement à travers ces interfaces machines qui sont réalisées pour transmettre le minimum d'informations tout en les formatant de manière à les rendre compréhensibles et identifiables pour les développeurs qui devront les réutiliser. Ces APIs sont souvent accompagnées de documentations qui permettent aux ingénieurs de les utiliser. Aujourd'hui, chacun s'est habitué à utiliser les « Connexion via Google » ou « Se connecter via son institution » au niveau universitaire : il s'agit d'APIs d'identification qui permettent aux utilisateurs de réduire leur nombre d'identifiants.

Cette communication machine à machine et ses possibilités ont conduit les acteurs du monde numérique à lui chercher des formes de normalisation, de standardisation. Les éditeurs et producteurs de données, les institutions d'archives et bibliothèques en premier, se sont concentrés à produire des standards de mise à disposition de leurs catalogues via des interfaces web. Par exemple, l'OAI-PMH (*Open Archive Initiative — Protocol for Metadata Harvesting)* est un protocole standard d'interrogation (comment demander des informations à l'API) et de réponse (comment interpréter l'information) qui permet la communication de catalogues. Isidore, un logiciel développé au sein de la TGIR HumaNum, construit à partir de ce protocole entre autres une base de données des contenus scientifiques français et de leurs métadonnées disponibles[15]. C'est aussi ce protocole qu'implémentent des outils tels que WebOai d'HumaNum[16].

Cependant, OAI-PMH ne permet pas de résoudre l'ensemble des besoins des projets en sciences humaines, et n'a d'ailleurs pas été conçu pour cela. Si il offre un accès au catalogue, il ne résout pas la question de l'accès au texte en lui-même. Le protocole CTS répond à ce besoin. Il permet de naviguer parmi les références d'un texte (livre 1, livre 2, chapitre 1, etc.) et d'en retrouver le contenu via des requêtes standardisées. L'ensemble CapiTainS offre dans cette optique une application (Nautilus) qui permet la mise à disposition de contenus suivant les *guidelines* CapiTainS dans différents formats d'API[17]. La version 1.0.0 met à disposition une API CTS et un embryon d'API[18], tandis qu'il est prévu d'ajouter à ces API deux autres standards : une interface Sparql qui permettrait d'interroger la base de données de métadonnées à travers des requêtes libres, une interface OAI-PMH qui permettrait de compléter l'offre d'API et de cibler différents groupes d'utilisateurs. Ces deux interfaces sont d'ores et déjà implémentables grâce à un fonctionnement en triplet RDF du système de métadonnées derrière l'interface logicielle. Nautilus permet aux producteurs de données CapiTainS de mettre à disposition leurs textes pour la citation et la réutilisation via ses APIs.

15 « ISIDORE — Accès aux données et services numériques de SHS », en ligne : <https://www.rechercheisidore.fr/>. Consulté le 4 mai 2017.

16 « Catalogue OAI du consortium CAHIER », en ligne : <http://weboai.cahier.huma-num.fr/pmh?verb=ListSets>. Consulté le 4 mai 2017.

17 « Capitains/Nautilus: 1.0.0 », *Zenodo*, <DOI : 10.5281/zenodo.569847>.

18 Il est important de noter ici l'existence d'un projet encore en cours au moment de la rédaction de cet article à savoir le projet de standard Distributed Text Services (DTS) qui vise à simplifier et répondre aux problèmes soulevés par la communauté à propos du standard CTS (rapidité, standardisation, pagination, direction unique, identifiant, etc.) tout en étant retro-compatible avec celui-ci.

Ainsi, en utilisant les *guidelines* CapiTainS et Nautilus, il est possible d'offrir sans aucun coût de développement des interfaces CTS et DTS et à court terme des interfaces Sparql et OAI-PMH.

Interface(s) utilisateur(s)

CapiTainS offre la possibilité aux producteurs de données d'augmenter leur API d'une interface utilisateur. La logique est simple : en offrant un ensemble d'outils permettant à la fois la mise à disposition en API et la mise à disposition de textes en interface utilisateur, le travail de valorisation des données est facilité.

L'interface utilisateur pour les textes CapiTainS, Nemo, permet la réutilisation de corpus en connectant celle-ci à une instance Nautilus[19]. Nemo a été développé pour permettre son extension par des plugins, des thèmes graphiques, des feuilles XSLT, etc. L'application propose de base une navigation des collections (Collection → *TextGroup* → *Work* → Édition → Liste de références → Passage ou Premier passage) qui permettront à de nombreux projets de simplifier la phase de développement de l'outil applicatif pour se concentrer sur la personnalisation de l'outil et son design. Par ailleurs, via les métadonnées déclarées dans les fichiers de métadonnées, il est possible de fournir des interfaces multilingues proposant les noms de collection dans la langue du navigateur.

CapiTainS Nemo repose sur trois adresses dynamiques (ou *routes*) principales qui font appel à des identifiants (notés *{identifant}* ci-dessous) (cf. Figure 3) :

- la première permet de naviguer dans la hiérarchie du catalogue de textes (`/collections, /collections/{identifiant}`),
- la seconde liste l'ensemble des passages mis à disposition (`/text/{identifiant}/references`),
- la dernière affiche un passage (`/text/{identifiant}/passage/{identifiant}`).

19 « Capitains/flask-capitains-nemo: 1.0.0 », *Zenodo*, <DOI : 10.5281/zenodo.569863>.

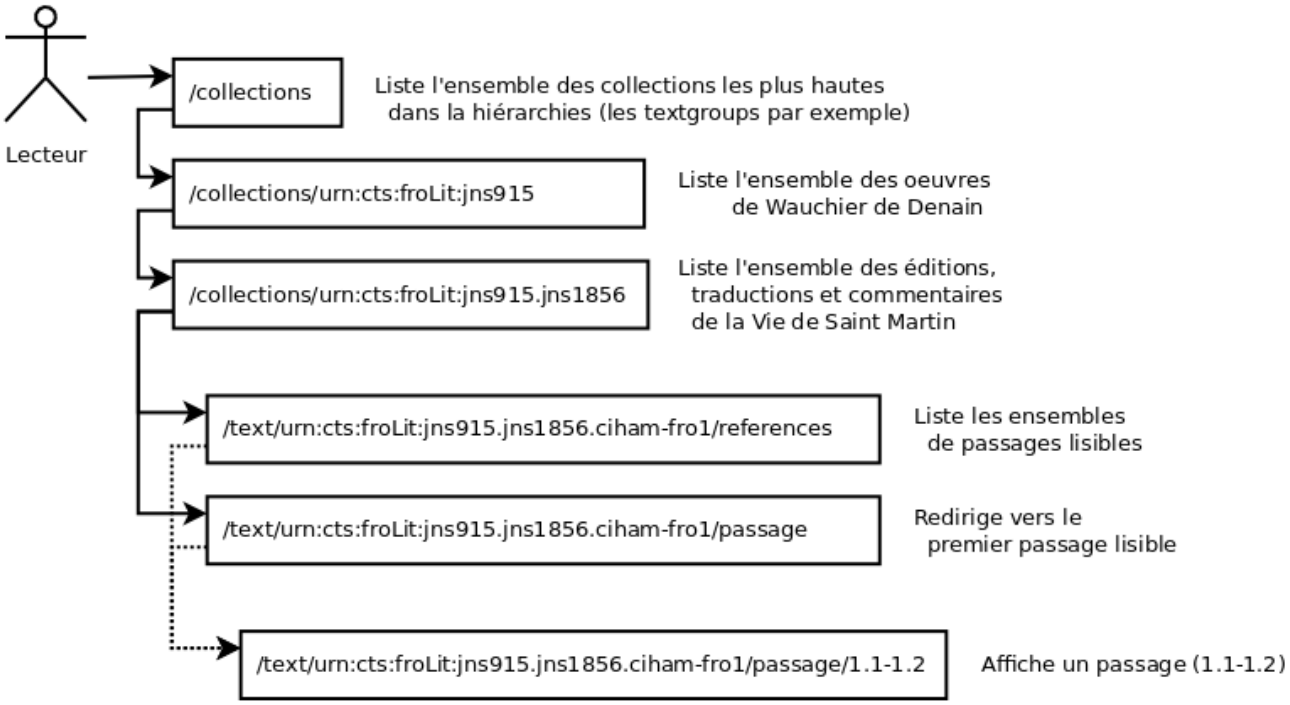

Figure 3 : Exemple de navigation sur Nemo

Celles-ci peuvent être augmentées, modifiées ou complétées par d'autres adresses via un système de *plugin*. Quant à la lecture des passages, il est possible d'établir des règles pour leur séquençage (la machine doit-elle pour chaque texte regrouper par poème ? Par nombre de paragraphes ? Par section ?) qui sont personnalisables à deux niveaux (niveau de l'édition spécifique, au niveau du corpus général). La version en ligne de la chaire Humboldt d'humanités numériques configure de cette manière la mise en disposition des textes en fonction des noms des niveaux de citation de chaque édition, traduction ou commentaire[20]. Si un texte est composé d'un système *book→ poem→ line*, le texte sera mis à disposition à raison de 1 poème à la fois dans Nemo, tandis qu'un système *book→ line* tel que l'œuvre *Odyssée* sera servie par ensemble de 30 lignes[21].

Avec Nautilus et Nemo, il est pensable de mutualiser des moyens afin de fournir des outils applicatifs supplémentaires tels que la recherche plein-texte ou la mise à disposition d'outils statistiques. En adoptant un socle minimal commun « API-navigation dans le catalogue-lecture » pour la mise à disposition de texte, il est possible de produire des interfaces et des outils plus rapidement en partageant leur base dans de multiples projets.

Exploiter

« *Texts now have a different purpose: they need to be much more than simply exchangeable via disk or email, they need to instantly respond to heterogenous needs* »,
Desmond Schmidt[22]

20 Disponible à l'adresse <http://cts.dh.uni-leipzig.de>.
21 Thibault Clérice, « OpenGreekAndLatin/cts_leipzig_ui », en ligne : <https://github.com/OpenGreekAndLatin/cts_leipzig_ui>. Consulté le 4 mai 2017.
22 Desmond Schmidt, « Towards an Interoperable Digital Scholarly Edition », *Journal of the Text Encoding Initiative*, novembre 2014, <DOI : 10.4000/jtei.979>.

Si CapiTainS simplifie la mise à disposition de ressources textuelles à travers des interfaces utilisateurs et machines, son objectif original était de faciliter la réutilisation et la production de ces données. En effet, bien que les *Linked Open Data* soient ouvertes et libres, elles peuvent s'avérer difficile d'accès pour les chercheurs et développeurs désireux de les réutiliser. C'est pourquoi certaines institutions telles que la *British Library* offrent leurs données sous la forme de fichiers CSV qui s'avèrent bien plus facilement manipulables que des triplets RDF. Le défi est donc de rendre simplement exploitables les données sans perdre leur standardisation.

Production et cycles de vie des données

Pour assurer un accueil favorable et se protéger des critiques les plus courantes, il est important que les données numériques sachent répondre à deux problématiques importantes :

- Leur « citabilité » universitaire : il doit être possible d'identifier la source, le moment de production et les producteurs d'une ressource au moment où celle-ci a été créée. C'est l'objectif d'outils tels que les *Digital Object Identifiers* (DOI) ;
- Leur conformité : les données doivent suivre, si on les veut compréhensibles, un ensemble de règles préalablement établies afin de s'assurer de leur qualité.

La première est bien souvent oubliée par les producteurs universitaires en cela qu'elle nécessite la production ou l'utilisation d'infrastructures qui sachent pérenniser l'information. Bien qu'elles existent, leur usage n'est pas encore rentré dans la pratique. En conséquence, on connaît un problème d'attribution et de valorisation du travail nativement numérique.

La seconde passe souvent via la validation, en XML tout du moins, en fonction de schémas tels que la TEI. Cependant, les vérificateurs de schéma traditionnels rencontrent des limites quand ils sont confrontés à l'exécution de code ou quand ils sont confrontés à des situations diverses qui nécessitent un calcul. C'est le cas des valeurs exécutables refsDecl des *guidelines* de CapiTainS. Pour vérifier l'unicité des nœuds de citations, un vérificateur de schémas devra calculer l'ensemble des nœuds et vérifier qu'il n'existe aucune séquence semblable.

En reprenant la notion de test et d'intégration continue à l'ingénierie logicielle, CapiTainS a développé les deux systèmes Hook[23] et HookTest[24] qui visent à répondre en partie à ces deux problèmes tout en s'appuyant sur d'autres infrastructures gratuites (Github et Travis) et une infrastructure du CERN, Zenodo, qui permet de créer une archive pérenne de ses données en y attribuant un DOI[25]. CapiTainS propose donc HookTest, qui, via le logiciel installé sur un poste personnel ou une configuration sur Github, permettra d'effectuer une batterie de tests : conformité Epidoc ou TEI, absence de répétition d'identifiant de passage, conformité des métadonnées, conformité des systèmes de citation, absence d'erreur de calcul, validité des identifiants des textes et de leur hiérarchie conceptuelle. Il confirme alors la compatibilité de chaque ressource avec les *guidelines* CapiTainS tout en conservant un ensemble d'informations permettant leur correction ou leur valorisation (nombre de mots par langue, de niveaux de citations, de nœuds de citation).

23 « Capitains/Hook: 1.0.0 », *Zenodo*, <DOI : 10.5281/zenodo.555931>.
24 « Capitains/HookTest: 1.0.0 », *Zenodo*, <DOI : 10.5281/zenodo.569841>.
25 « Zenodo, un entrepôt de données | Observatoire des technologies de l'IST », en ligne : <http://ist.blogs.inra.fr/technologies/2013/09/16/zenodo-un-entrepot-de-donnees/>. Consulté le 4 mai 2017.

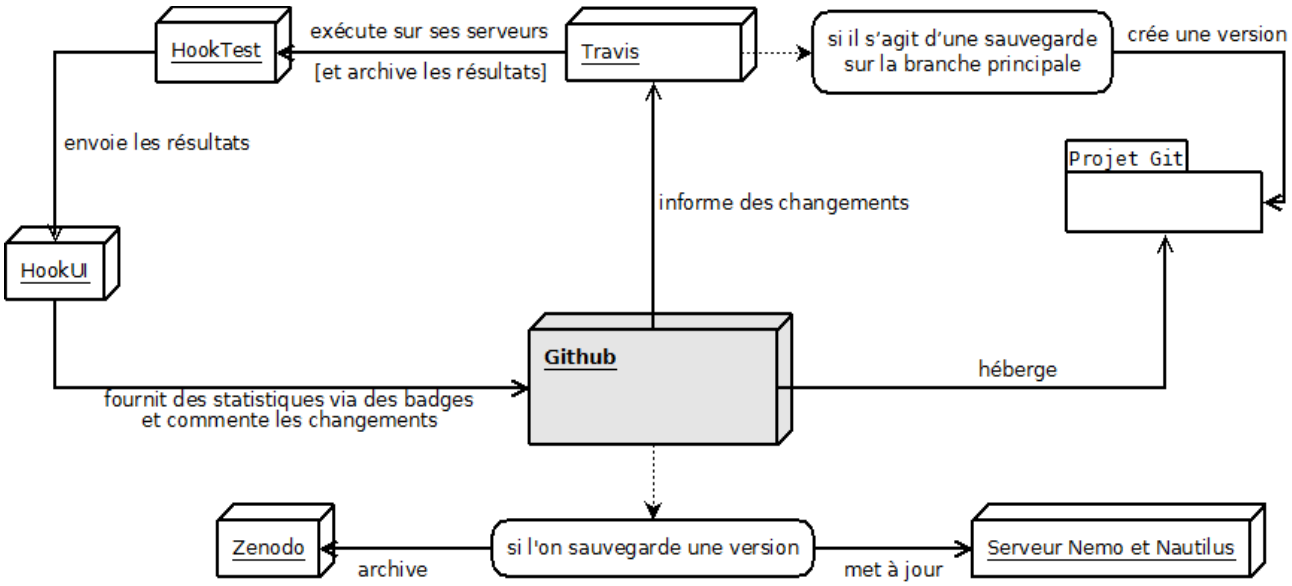

Figure 4: Chaîne de réaction proposée par CapiTainS lors de modifications de corpus (chaque interaction est optionnelle mais permet une réduction de la maintenance manuelle des services)

Dans un second temps, le résultat est utilisé pour produire une nouvelle version du corpus qui elle-même devient citable comme un instant T de celui-ci. Pour chacune de ces archives, un manifeste est produit : il contient la liste des fichiers qui se plient aux *guidelines* CapiTainS (cf. Figure 4). Ces corpus sont enregistrés sur le long terme via Zenodo qui produit un DOI pour chacune des archives obtenues. Au même instant, sur Github, une version du corpus sans données erronées est produite. Elle peut ensuite être incorporée presque instantanément aux corpus mis à dispositions en ligne dans les interfaces machines et utilisateurs.

De cette manière, CapiTainS permet de :

- mettre à jour le serveur web, tant pour les lecteurs que pour les utilisateurs de l'API, en même temps que le corpus évolue, sans prendre de risque de sécurité et en réduisant un maximum l'intervention humaine via la mise à disposition de scripts de tests ou de mise à jour automatique[26] ;
- archiver et pérenniser chaque version estimée correcte par les gestionnaires de corpus ;
- rendre citable chaque version du corpus, facilitant la reproduction des travaux faits sur ceux-ci en utilisant le même corpus dans sa version spécifique ;
- simplifier et assurer la compatibilité de l'ensemble du silo de données : métadonnées, structure de citation. L'éditeur s'attache à assurer la qualité scientifique de l'édition sans s'inquiéter de l'impact potentiel de ses modifications, car la structure est vérifiée automatiquement ;
- en facilitant l'assurance qualité, la pérennisation, la citation et l'intégration rapide de changement sur des serveurs, le producteur de données réduit les risques d'incompatibilité

26 CapiTainS fournit un module qui permet de mettre en place rapidement un serveur et son application « Capitains/puppet-capitains », en ligne : <https://github.com/Capitains/puppet-capitains>. Consulté le 4 mai 2017..

ou de rupture des services de tiers. Si rupture il y avait, le tiers pourrait simplement héberger lui-même les données sous la version souhaitée.

Exploiter des ressources existantes

Originellement, CapiTainS a été fondé pour fournir des outils, non pas pour produire des données standardisées, mais pour les interpréter. Le premier outil, Sparrow, est une librairie javascript qui permet de simplifier l'interrogation d'API CTS du côté client[27]. Le second outil, MyCapytain, a été réalisé pour interroger ces mêmes APIs du côté client, puis pour servir et interpréter les *guidelines* CapiTainS.

Sparrow a été utilisé en priorité pour fournir des formulaires pour sélectionner des textes à éditer, mais aussi pour fournir des interfaces utilisateurs complètes telles que le projet *Journey of the Hero* (JOTH)[28]. Ce projet, mené par la directrice du projet Perseids, Marie-Claire Beaulieu, à Tufts, consiste en l'annotation de fiches de dictionnaires (identifiées par des URN CTS) en classe de mythologie. Il s'agit pour les étudiants de mettre en lumière les lieux, les personnages et les caractérisations en relation avec ces personnages à travers un ensemble d'annotations. JOTH est un exemple typique de la nécessité de mettre à disposition ses données textuelles. L'ensemble des textes sont chargés par Sparrow sans que le site ne les héberge lui-même. En sus du corpus « maître » contenant les vies de divinités et héros du dictionnaire de W.R. Smith[29], il existe un ensemble d'extraits de textes grecs et latins identifiés par les élèves comme des mentions ou des caractérisations qui sont aussi chargés sans être hébergés (cf. Illustration 1).

27 En programmation web, on distingue les notions de client et de serveur. Les calculs faits côté client permettent d'alléger la charge des serveurs, mais surtout rendent les interactions avec le site plus vivante : c'est le principe des outils de messageries qui communiquent et se rafraîchissent sans que l'utilisateur ne recharge la page.

28 « Perseids-project/journey-of-the-hero: Release 1.0.0 », *Zenodo*, <DOI : 10.5281/zenodo.288156>.

29 William Smith, *Dictionary of Greek and Roman Biography and Mythology*, 1867, 1 232 p.

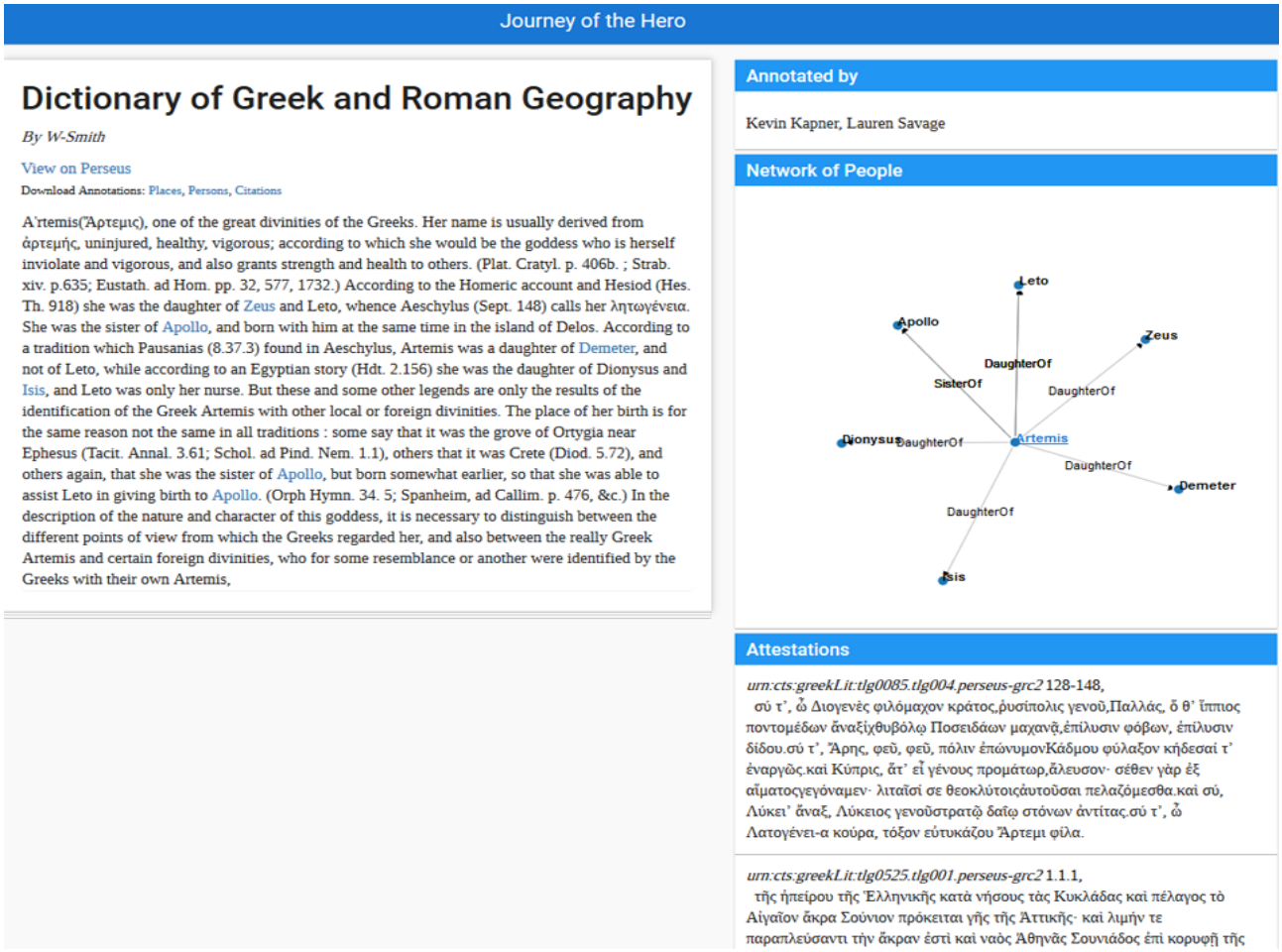

Illustration 1 : Fiche d'Artemis du *Journey of the Hero*

MyCapytain[30] est une librairie logicielle qui n'a pour vocation que de simplifier et d'éviter de réécrire l'ensemble des démarches d'interprétations des résultats d'API ou de textes suivant les *guidelines*. Elle est nécessaire et commune aux deux projets Nemo et Nautilus en ce qu'elle intègre leurs fonctions primaires nécessaires à leur bon fonctionnement : lecture, navigation, interrogation, etc. MyCapytain a aussi pour objectif de simplifier, en dehors de projets orientés application web, la communication avec ces ressources, qu'elles soient locales (fichiers sur son ordinateur) ou distantes (APIs CTS hébergées par d'autres projets). Il est utilisé par différents outils externes aux institutions fondatrices : par exemple, le projet *Classical Language ToolKit*[31](CLTK) l'utilise pour convertir les corpus de Perseus et les autres corpus CapiTainS dans un format qui lui semble plus facile à intégrer avec d'autres ressources.

Mais il peut aussi être utilisé dans d'autres cadres, y compris d'enseignement. Ce fut le cas dans l'un des séminaires *Sunoikisis Digital Classicist,* un webinaire, dirigé par Monica Berti[32]. *C*e séminaire a pour ambition d'enseigner les bases des techniques numériques d'exploitation ou de production des données. Lors de leur intervention, Matteo Romanello et Francisco Mambrini ont utilisé MyCapytain comme outil de récupération des textes pour ensuite appliquer à ceux-ci de la reconnaissance d'entités nommées (noms propres ou mots représentant des lieux, des personnes, des

30 « Capitains/MyCapytain: 2.0.0 », *Zenodo*, <DOI : 10.5281/zenodo.569838>.
31 Kyle P. Johnson, Patrick J. Burns, Tyler Kirby *et al.*, « Cltk: v0.1.41 », *Zenodo*, <DOI : 10.5281/zenodo.6002>.
32 Digital Classicist London Seminars, « Monica Berti (Leipzig), "Sunoikisis DC - An International Consortium of Digital Classics Programs" », <https://www.youtube.com/watch?v=zpBR0bb8gxk>.

lieux identifiés)[33] (cf. Code 4). MyCapytain est écrit en python dans cet objectif aussi : python est un langage très utilisé dans le traitement automatique du langage et dans le traitement de données en général, tout en ayant un niveau nécessaire de compréhension assez bas pour les débutants.

```
from MyCapytain.resolvers.cts.api import HttpCtsResolver
from MyCapytain.retrievers.cts5 import HttpCtsRetriever
from MyCapytain.common.constants import Mimetypes, XPATH_NAMESPACES

# On crée un outil de communication avec une API CTS
resolver = HttpCtsResolver(HttpCtsRetriever("http://cts.dh.uni-leipzig.de/api/cts"))
# On veut demander le passage du livre 1 de la Guerre des Gaules
# Soit http://cts.dh.uni-leipzig.de/api/cts?
request=GetPassage&urn=urn:cts:latinLit:phi0448.phi001.perseus-lat2:1
passage = resolver.getTextualNode("urn:cts:latinLit:phi0448.phi001.perseus-lat2", "1")
# On l'affiche en plein texte
print(passage.export(Mimetypes.PLAINTEXT))
# COMMENTARIUS PRIMUS Gallia est omnis divisa in partes tres, quarum unam incolunt Belgae,
aliam Aquitani, tertiam qui ipsorum lingua Celtae, nostra Galli appellantur. Hi omnes lingua, institutis,
legibus inter se differunt. Gallos ab Aquitanis Garumna flumen, a Belgis Matrona et Sequana dividit.
Horum omnium fortissimi sunt Belgae, propterea quod a cultu atque humanitate provinciae longissime
absunt, minimeque ad eos mercatores saepe commeant atque ea quae ad effeminandos animos pertinent
important, pr...
```

Code 4 : Code nécessaire pour récupérer le premier livre de la Guerre des Gaules *en texte, alors que l'original est à distance et en XML. Il ne faut que trois lignes de codes et trois lignes d'import pour réutiliser des données libres et standards.*

Ces deux exemples montrent comment, en fournissant les données, mais aussi en fournissant les outils permettant de les réutiliser, on peut faciliter la fédération d'une communauté autour de celles-ci. Il semble souvent compliqué pour les développeurs et les chercheurs en sciences humaines d'interagir avec ces données, ce qui les amène souvent à demander des versions « brutes » ou simplifiées telles que des fichiers CSV ou des fichiers textes suivant le type de données. En fournissant les outils nécessaires à leur exploitation, CapiTainS tend à supprimer cette barrière dans le développement de nouveaux outils applicatifs.

Conclusion

CapiTainS vise à faciliter à la fois la production de silos de données, sans imposer de réel modèle scientifique, et à simplifier leur exploitation par les producteurs autant que les utilisateurs. En tant qu'écosystème, il permet :

- de mettre en place des plates-formes de lecture pour les utilisateurs ;
- de mettre en place des stratégies de mises à disposition des données dans des formes et dans des objectifs différents, qu'il s'agisse de métadonnées ou de données textuelles ;
- de penser à la fois à la production et à la réexploitation interne et externe des données, pour éviter que ces standards ne deviennent des technologies de niche ;

33 « Mromanello/SunoikisisDC_NER », en ligne : <https://github.com/mromanello/SunoikisisDC_NER>. Consulté le 4 mai 2017.

- de produire des communautés de production de savoirs, mais aussi de production technique, à la manière d'un CMS via son système de *plugins*. Ainsi, des outils comme Plokamos[34] peuvent être théoriquement installés par tout autre projet utilisant Nemo[35].

Le projet CapiTainS n'est pas révolutionnaire et n'a pas vocation à le devenir. Cependant, il offre la possibilité aux études philologiques et historiques de centraliser les efforts sur la production de données en prenant en compte les impératifs numériques nouveaux. Il ne s'agit plus de simplement produire des données interéchangeables, mais de les rendre un peu plus interopérables, de les partager et enfin de prendre en compte les utilisateurs en fournissant les outils nécessaires à leur réutilisation.

34 Développé par Perseids pour faire des annotations standardisées et les visualiser dans Nemo tout en les enregistrant dans des triple-stores.

35 Frederik Baumgardt, *Nemo Plokamos Plugin*, <https://github.com/perseids-project/nemo_plokamos_plugin>.