Médiévales

Langues, Textes, Histoire

73 | automne 2017

Le texte à l'épreuve du numérique

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

CapiTainS Toolkit, Digital Editing and Data Reuse

Thibault Clérice

Édition électronique

URL : <https://journals.openedition.org/medievales/8211>

DOI : 10.4000/medievales.8211

ISSN : 1777-5892

Éditeur

Presses universitaires de Vincennes

Édition imprimée

Date de publication : 15 décembre 2017

Pagination : 115-131

ISBN : 978-2-84292-801-8

ISSN : 0751-2708

Référence électronique

Thibault Clérice, « Les outils CapiTainS, l'édition numérique et l'exploitation des textes », *Médiévales* [En ligne], 73 | automne 2017, mis en ligne le 28 février 2019, consulté le 22 avril 2022. URL : <http://journals.openedition.org/medievales/8211> ; DOI : <https://doi.org/10.4000/medievales.8211>

Tous droits réservés

Thibault Clérice

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

La multiplication des efforts de numérisation et de publication en ligne de textes, issus notamment des littératures classique, médiévale et moderne, a aujourd'hui durablement changé notre rapport à la production et à l'exploitation d'un écrit. D'une part, certains types d'analyse (comme le recours à de très larges corpus) se sont multipliés. Pour les études médiévales par exemple, la stylométrie et la question de l'auctorialité dans les œuvres anonymes ont ainsi pu bénéficier d'un accès libre à ce type de ressources¹. D'autre part, chercheurs et institutions de financement de la recherche ont conjointement cherché à obtenir une audience plus importante en mettant à disposition du plus grand nombre les fruits de la recherche. Le numérique a pu ainsi permettre de réduire en partie la difficulté d'accès aux savoirs pour les publics, professionnels ou non, notamment quand les projets ont recours à l'*open data* et à l'*open access*.

Dans ce contexte, la publication comme l'édition critique numérique des textes médiévaux connaît les mêmes problématiques que celle des textes des autres époques : le résultat doit être à la fois scientifiquement validé, archivable et standardisé, tout en étant publiquement accessible *via* des interfaces de lecture. Or, le développement d'interfaces et de ressources standardisées implique un coût important pour les équipes de recherche. L'article qui va suivre a pour objectif de présenter, du point de vue de ses concepteurs et développeurs, un projet *open source* cherchant à développer

1. Quelques exemples : J.-B. CAMPS et F. CAFIERO, « Genealogical Variant Locations and Simplified Stemma : a Test Case », dans T. ANDREWS et C. MACÉ éd., *Analysis of Ancient and Medieval Texts and Manuscripts : Digital Approaches*, Turnhout, 2014, p. 69-93, <<https://halshs.archives-ouvertes.fr/halshs-01435633/document>> ; M. KESTEMONT, S. MOENS et J. DEPLOIGE, « Collaborative Authorship in the Twelfth Century : A stylometric Study of Hildegard of Bingen and Guibert of Gembloux », *Literary and Linguistic Computing*, 30/2 (2015), p. 199-224, <<http://doi.org/10.1093/lilc/fqt063>>.

un ensemble de services, de modèles et d'interfaces de programmation pour la valorisation et l'exploitation d'éditions numériques et de leurs données : CapiTainS. Fondé par Bridget Almas et moi-même en 2015, CapiTainS a été conçu comme un projet collaboratif, transdisciplinaire et désinstitutionnalisé². En fournissant des réponses aux problématiques de modélisation, de communication et d'exploitation des textes, CapiTainS réduit la difficulté de chacune de ces trois étapes pour des utilisateurs variés (éditeurs, développeurs, chercheurs, lecteurs), dans des contextes différents (hors ligne, en bibliothèque numérique, dans une exposition virtuelle)³.

Modéliser

Donner accès sous une forme numérique à des données textuelles pose très rapidement la question de la modélisation : comment peut-on transférer une ressource physique dans l'espace numérique ? Pour répondre à cette question, il faut interroger la structure que forme un texte, à la fois dans ses réalisations intégrales (livres, manuscrits, inscriptions) et ses usages parcellaires (la citation). Les études classiques en latin ou en grec ancien, fondées sur une longue et ancienne pratique philologique, posaient ainsi déjà la question de la structuration et ont proposé une normalisation relative, notamment des références, au fil des siècles. On cite ainsi le chant 5, vers 1, de l'*Énéide* de Virgile « Verg. *Aen.* 5.1 ». Il est aisé de transposer ce système sur le plan numérique : sa réalisation implique deux composantes : un texte hiérarchisé ou conçu comme tel, un modèle informatique permettant son interprétation et *a priori* son identification.

En 1996, l'article « Refining our Notion of What Text Really Is : the Problem of Overlapping Hierarchies », de Allen H. Renear, Elli Mylonas et David Durand, propose de la même façon une modélisation des textes⁴. Cette modélisation, *Ordered Hierarchy of Content Objects* (OHCO), est une approche très semblable aux principes 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

2. Ce projet a été financé à l'origine par la Humboldt Chair of Digital Humanities de Leipzig dans le cadre du projet Open Greek and Latin et le projet Perseids à Tufts.

3. Quand cela a été possible, les exemples sont tirés de corpus en cours d'édition d'un travail de thèse.

4. A. RENEAR, E. MYLONAS et D. DURAND, « Refining our Notion of What Text Really Is : the Problem of Overlapping Hierarchies », 1993, <<https://www.ideals.illinois.edu/handle/2142/9407>>.

« *Énéide* → Chant → Vers → Mot → Caractère », qui est généralement reconstitué sous la forme « Virg. *Aen.* VI.5 ».

Cette vision hiérarchisée des textes conditionne toute une vision de leur encodage et est clairement utilisée par le standard TEI en publication et édition textuelle numériques⁵. Ce paradigme instaure, *via* son langage d'implémentation XML⁶, 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 ; cette dernière se traduit par une mise en page particulière qui, elle-même, induit parfois des divisions sémantiques (chapitres, strophe, etc.). Cette structuration peut être complétée par une structuration grammaticale ou par un appareil 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*, *ph*), système qui permet alors de marquer d'autres articulations que celles décrites plus haut, qui étaient celles canoniques du « document » tel qu'il est conçu en TEI⁷. Il s'agit alors de montrer des divisions thématiques ou documentaires.

L'article de Allen H. Renear, Elli Mylonas et David Durand soulève cependant un problème de taille : comment traiter les textes dont le contenu n'a pas vocation à être hiérarchisé 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 et donc, d'une certaine manière, d'atomisation du texte, sans pour autant que ce dernier soit vu comme le seul possible. 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 du texte. 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.

5. TEI Consortium, « 4. Complicating the Issue », *TEI P5 : Guidelines for Electronic Text Encoding and Interchange, Version 3.2.0.*, <<http://www.tei-c.org/release/doc/tei-p5-doc/en/html/SG.html#SG152>>.

6. *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 par son système de balise. Un paragraphe peut être noté « <p>Texte</p> ».

7. « <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 », <<http://www.tei-c.org/release/doc/tei-p5-doc/fr/html/ref-milestone.html>>).

C'est pourquoi les normes CapiTainS 2.0.0 ont pour principal objectif de rendre possible et simple l'explicitation d'un modèle de citation⁸. Historiquement, CapiTainS a été fondé pour produire des textes compatibles avec le modèle *Canonical Text Services* (CTS) bien que le projet souhaite soutenir d'autres formes de modélisation⁹. Ces normes se concentrent sur deux objectifs : expliciter et documenter des collections de textes, expliciter et rendre opérables des systèmes de citation.

Le modèle CTS (*Canonical Text Services*) repose sur un système d'identifiants composés de cinq grands éléments (fig. 1). Ces identifiants URN (*Uniform Resource Name*) 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 de catalogage d'un texte :

- le domaine a été utilisé comme outil de dénomination d'un corpus linguistique (*froLit* pour littérature en ancien français) ou d'une collection de textes déjà constituée (*pdlrefwk* pour *Perseus Digital Library Reference Work*, qui comprend des dictionnaires) ;

- le *textgroup*, bien que renvoyant souvent à un auteur, identifie surtout un ensemble de textes qu'il fait sens de rassembler : les textes de troubadours anonymes, les livres de l'*Ancien Testament*, les œuvres du cycle du Graal, un ensemble d'inscriptions localisées géographiquement ;

- l'identifiant *work* renvoie à l'œuvre, dans toutes ses réalisations. Bien que l'on cite souvent *Le Roman de la Rose*, nous ne précisons pas le plus souvent quel manuscrit, quelle édition scientifique se cachent derrière cette expression¹⁰ ;

- l'identifiant *version* sert à identifier une édition ou publication numérique spécifique.

Les normes CapiTainS proposent de répercuter ces identifications dans un système de hiérarchie des fichiers et dossiers afin de faciliter la gestion de larges corpus¹¹. Chaque niveau contient ensuite un fichier commun de métadonnées (fig. 2). Afin d'éviter une redondance des données, il suffit de décrire une fois le *textgroup* ou l'œuvre pour l'ensemble de leurs réutilisations (fig. 3). Les descriptions des implémentations se font aussi à

8. T. CLÉRICE, M. MUNSON, et B. ALMAS, « CapiTainS Guidelines 2.0.0 », *Zenodo*<<http://doi.org/10.5281/zenodo.570516>>.

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

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

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

urn : cts : froLit : jns915 . jns1856 . ciham-fro1 : 1.1-1.2				
Namespace	Textgroup	Work	Version	Passage
<i>Littérature en ancien français</i>	<i>Wauchier de Denain</i>	<i>Vie de Saint Martin</i>	<i>Édition d'Ariane Pinche au CIHAM</i>	<i>Entre 1.1 et 1.2</i>

1. Schéma des URN CTS

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>
```

Source : 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 », C. Pierreville et B. Bureau dir., thèse de doctorat en cours.

3. Contenu d’un fichier de description de textgroup (exemple : Wauchier de Denain)

l'extérieur de l'édition en elle-même (fig. 4). Les métadonnées de base sont simples (*groupname* pour le *textgroup*, *title* pour le *work*, *label*, *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.

Le second objectif des normes 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* pourrait indiquer un système de citation à trois niveaux. Nous avons décidé avec CapiTainS de rendre leur explicitation obligatoire à travers l'usage des outils simples fournis par le standard TEI, le XML et le XPath¹². En effet, ce standard propose un ensemble de balises *refsDecl* qui ont pour objectif la « déclaration de système de référence¹³ ». CapiTainS recommande ensuite de déclarer les niveaux de citation *via* l'utilisation de balises *cRefPattern* (fig. 5), 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.

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 contraignant. Ces règles de conduite conviennent en effet à 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 son accès, il faut aussi pouvoir le lire. Le développement d'interfaces occupe souvent la plus grande part des budgets de recherche, parce 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 et permettant 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

12. Ce langage permet de traverser l'arbre du fichier. Ainsi, si un texte est composé de <div> 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'].

13. Une telle utilisation de ce tag a aussi été proposée dans J. KALVESMAKI, « Canonical References in Electronic Texts : Rationale and Best Practices », *Digital Humanities Quarterly*, vol. 8.2, 2014, <<http://www.digitalhumanities.org/dhq/vol/8/2/000181/000181.html>>.

```

<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>

```

4. Contenu d'un fichier de description de *work* et de son édition

Source : 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 », C. Pierreville et B. Bureau dir., thèse de doctorat en cours.

```

<refsDecl n="CTS">
  <cRefPattern n="section" matchPattern="(\\w+).(\\w+)"

  replacementPattern="#xpath(/tei:TEI/tei:text/tei:body/tei:div[@n='S1']/tei:div[@n='S2'])">
    <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='S1'])">
    <p>Ce pointeur extrait les chapitres dont l'identifiant correspond à x (où x est des
    ensembles de caractères alphanumériques)</p>
  </cRefPattern>

```

5. Exemple de déclaration refsDecl d'un système de citation CapiTainS

réduire le temps de développement pour la mise à disposition de textes dans des normes reconnues et partagées en laissant la possibilité de personnaliser l'apparence des interfaces utilisateurs.

Les interfaces machines tout d'abord, ou API (*Application Programming Interface*), permettent de communiquer avec un site ou un logiciel au niveau du 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 lui 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 conçues 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 standardisation. Les éditeurs et producteurs de données, en premier lieu les services d'archives et les bibliothèques, se sont concentrés sur la production de 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 entre catalogues. Isidore, un logiciel développé au sein de la TGIR Huma-Num, construit notamment, à partir de ce protocole, une base de données des contenus scientifiques français et de leurs métadonnées disponibles¹⁴. C'est aussi ce protocole qu'implémentent des outils tels que WebOai d'Huma-Num¹⁵.

Cependant, OAI-PMH ne permet pas de satisfaire l'ensemble des besoins des projets en sciences humaines, et n'a d'ailleurs pas été conçu pour cela. S'il offre un accès aux catalogues, 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.

14. « ISIDORE – Accès aux données et services numériques de SHS », <<https://www.rechercheisidore.fr/>>.

15. « Catalogue OAI du consortium CAHIER », <<http://weboai.cahier.huma-num.fr/pmh?verb=ListSets>>.

L'ensemble CapiTainS offre dans cette optique une application (Nautilus) qui permet la mise à disposition de contenus suivant les normes CapiTainS dans différents formats d'API¹⁶. La version 1.0.0 met à disposition une API CTS et un embryon d'API DTS (*Distributed Text Services*), tandis qu'il est prévu d'ajouter à ces API deux autres standards : une interface Sparql qui permettrait d'interroger les métadonnées RDF à travers des requêtes libres, ainsi qu'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.

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.

CapiTainS offre également la possibilité aux producteurs de données d'ajouter à leur API 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 Nautilus¹⁷. Nemo a été développé pour permettre son extension par des *plugins*, des thèmes graphiques, des feuilles XSLT, etc. L'application propose 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 collections dans la langue du navigateur.

CapiTainS Nemo repose sur trois adresses dynamiques (ou routes) principales qui font appel à des identifiants (notés *{identifiant}* ci-dessous) (fig. 6) :

- 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) ;

16. T. CLÉRICE et B. ALMAS, « Capitains/Nautilus 1.0.0 », *Zenodo*, <<http://doi.org/10.5281/zenodo.569847>>.

17. T. CLÉRICE, M. MUNSON, et B. ALMAS, « Capitains/flask-capitains-nemo 1.0.0 », *Zenodo*, <<http://doi.org/10.5281/zenodo.569863>>.

– la dernière affiche un passage (/text/{identifiant}/passage/{identifiant}).

6. 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 ?), 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 commentaire¹⁸. Si un texte est composé d'un système *book*→*poem*→*line*, le texte sera mis à disposition à raison d'un poème à la fois dans Nemo, tandis qu'un système d'un seul niveau *line* tel que l'*Eneas* sera présenté par ensemble de trente lignes¹⁹.

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 textes, il est possible de produire des interfaces et des outils plus rapidement en partageant leur base dans de multiples projets.

18. Disponible à l'adresse <<http://cts.dh.uni-leipzig.de>>.

19. T. CLÉRICE, « OpenGreekAndLatin/cts_leipzig_ui », <https://github.com/OpenGreekAndLatin/cts_leipzig_ui>.

Exploiter

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 difficiles 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.

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, pour être compréhensibles, un ensemble de règles préalablement établies afin de s'assurer de leur qualité.

Le premier point est bien souvent oublié par les producteurs universitaires, en cela qu'il 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.

Le second passe souvent *via* la validation des données, 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 (tel qu'un XPath) ou quand ils sont confrontés à des situations qui nécessitent un calcul. C'est par exemple le cas des valeurs exécutables refsDecl des normes 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 Hook²⁰ et HookTest²¹, qui visent à répondre en partie à ces deux problèmes tout en s'appuyant sur

20. T. CLÉRICE et B. ALMAS, « Capitains/Hook : 1.0.1 », *Zenodo*, <<http://doi.org/10.5281/zenodo.820883>>.

21. T. CLÉRICE, M. MUNSON, et B. ALMAS, « Capitains/HookTest : 1.1.5 », *Zenodo*, <<http://doi.org/10.5281/zenodo.997183>>.

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 DOI²². 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 normes de CapiTainS, tout en conservant un ensemble d'informations permettant leur correction ou leur valorisation (nombre de mots par langue présente dans le corpus, de niveaux de citations, de nœuds de citation).

7. 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 normes CapiTainS (fig. 7). 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 disposition en ligne dans les interfaces machines et utilisateurs.

22. « Zenodo, un entrepôt de données | Observatoire des technologies de l'IST », <<http://ist.blogs.inra.fr/technologies/2013/09/16/zenodo-un-entrepot-de-donnees/>>.

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 automatique²³ ;
- archiver et pérenniser chaque version estimée correcte par les gestionnaires de corpus ;
- rendre citable chaque version du corpus, facilitant la reproduction et la vérification 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 ou de rupture des services de tiers. En cas de rupture, le tiers pourrait simplement héberger lui-même les données sous la version souhaitée.

Originellement, CapiTainS a été fondé pour fournir des outils, c'est-à-dire 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 client²⁴. Le second outil, MyCapytain, a été réalisé pour interroger ces mêmes APIs du côté client, puis pour servir et interpréter les normes CapiTainS.

Sparrow a été utilisé en priorité pour fournir des formulaires visant à sélectionner des textes à éditer, mais aussi pour fournir des interfaces utilisateurs complètes, telles que le projet *Journey of the Hero* (JOTH)²⁵. 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

23. CapiTainS fournit un module qui permet de mettre en place rapidement un serveur et son application (« Capitains/puppet-capitains », <<https://github.com/Capitains/puppet-capitains>>).

24. 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.

25. B. ALMAS, M.-C. BEAULIEU et T. CLÉRICE, « Perseids-project/journey-of-the-hero : Release 1.0.0 », *Zenodo*, <<http://doi.org/10.5281/zenodo.288156>>.

textuelles. L'ensemble des textes sont chargés par Sparrow sans que le site ne les héberge lui-même. En sus du corpus « maître », qui contient les vies de divinités et héros du dictionnaire de W. R. Smith²⁶, il existe un ensemble d'extraits de textes grecs et latins, identifiés par les élèves comme des mentions ou des caractérisations, qui sont aussi accessibles sans être hébergés (fig. 8).

MyCapytain²⁷ est de son côté 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 normes. Elle est nécessaire et commune aux deux projets Nemo et Nautilus en ce qu'elle intègre les 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* (CLTK)²⁸ l'utilise pour convertir les corpus de Perseus et les autres corpus CapiTainS dans un format qui lui semble plus facile à intégrer à 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 Berti²⁹. Ce 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 lieux identifiés)³⁰ (fig. 9). MyCapytain est aussi écrit en python dans cet objectif : 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.

Ces deux exemples montrent comment, en fournissant les données, mais aussi en fournissant les outils permettant de les réutiliser, on peut

26. W. SMITH, *Dictionary of Greek and Roman Biography and Mythology*, Londres, 1867.

27. T. CLÉRICE, M. MUNSON, B. ALMAS, « Capitains/MyCapytain : 2.0.5 », *Zenodo*, <<http://doi.org/10.5281/zenodo.1038866>>.

28. K. P. JOHNSON, P. J. BURNS, T. KIRBY *et al.*, « Cltk : v0.1.41 », *Zenodo*, <<http://doi.org/10.5281/zenodo.6002>>.

29. Digital Classicist London Seminars, Monica Berti « Sunoikisis DC – An International Consortium of Digital Classics Programs », <<https://www.youtube.com/watch?v=zpBR0bb8gkx>>.

30. « Mromanello/SunoikisisDC_NER », https://github.com/mromanello/SunoikisisDC_NER.

Journey of the Hero

Dictionary of Greek and Roman Geography

By W.Smith

View on Perseus

Download Annotations: Places, Persons, Citations

Aἰτέμις(Ἀρτεμῖς), one of the great divinities of the Greeks. Her name is usually derived from ἀρτεμῖς, uninjured, healthy, vigorous; according to which she would be the goddess who is herself inviolate and vigorous, and also grants strength and health to others. (Plat. Cratyl. p. 406b. ; Strab. xiv. p.635; Lustath. ad Hom. pp. 32, 577, 1732.) According to the Homeric account and Hesiod (Hes. Th. 918) she was the daughter of Zeus and Leto, whence Aeschylus (Sept. 148) calls her Ἀηταιγένεια. She was the sister of Apollo, and born with him at the same time in the island of Delos. According to a tradition which Pausanias (8.37.3) found in Aeschylus, Artemis was a daughter of Demeter, and not of Leto, while according to an Egyptian story (Hdt. 2.156) she was the daughter of Dionysus and Isis, and Leto was only her nurse. But these and some other legends are only the results of the identification of the Greek Artemis with other local or foreign divinities. The place of her birth is for the same reason not the same in all traditions : some say that it was the grove of Ortygia near Ephesus (Tacit. Annal. 3.61; Schol. ad Pind. Nem. 1.1), others that it was Crete (Diod. 5.72), and others again, that she was the sister of Apollo, but born somewhat earlier, so that she was able to assist Leto in giving birth to Apollo. (Orph. Hymn. 34. 5; Spanheim, ad Callim. p. 476, &c.) In the description of the nature and character of this goddess, it is necessary to distinguish between the different points of view from which the Greeks regarded her, and also between the really Greek Artemis and certain foreign divinities, who for some resemblance or another were identified by the Greeks with their own Artemis,

Annotated by

Kevin Kapner, Lauren Savage

Network of People

```
graph TD
    Demeter -- Sister --> Artemis
    Dionysus -- Father --> Artemis
    Leto -- Mother --> Artemis
    Apollo -- Brother --> Artemis
    Zeus -- Father --> Artemis
```

Attestations

urn:cts:greekLit:tlg0085.tlg004.perseus-grc2.128-148,
σύ τ', ὃ Διονέης φιλόμαχον κράτος βυσίπολος
γενοῖ, ἰαλλὰς, ὃ θ' ἔκτος ποντομέδων ἀναίχθη βλάβη
Ἡοσιδάων μαχοντὲς ἐπιδόσαν φόβον, ἐπιδόσαν δίδου, σύ τ',
Ἄρης, οὐδ', φεῖ, πόλιν ἐπὶ νόμιμον Κάδρου φιλόδον κήδεσσι τ'
ἐναργῶς, καὶ Κύπρις, ἥ τ' εἰ γένους προμήτωρ, Διεύσιν· σέθεν
γὰρ ἐξ αἱματογενόναμεν· λατοῖσι σε θεοκλῆταις αὐτοῖσι
πελαζόμεσθα, καὶ σύ, Λύκει' ἄναξ, Λυκείος γενοῦστροπῇ δαΐα
σάνων ἀντίτας, σύ τ', ὃ Λατογένει-α κοῖρα, τόδων εὐτοκάου
Ἀρτεμι φίλα.

urn:cts:greekLit:tlg0525.tlg001.perseus-grc2.1.1.1,
τῆς ἡμετέρας τῆς Ἑλληνικῆς κατὰ νήσους τὰς Κυκλάδας καὶ
πέλαγος τὸ Αἰγαῖον ἄρκυ Σόνιον πρόκειται γῆς τῆς Ἀττικῆς·

8. Fiche d'Artemis dans *Journey of the Hero*

Source : K. Kapner, L. Savage, « Artemis », dans B. Almas, M-C. Beaulieu, T. Clerice éd., *Journey of the Hero*, <http://joth.perseids.org/joth.html#book/urn:cts:pdirefwk:vial88890045.003.perseus-eng1/read/urn:cts:pdirefwk:vial88890045.003.perseus-eng1:A.artemis_1>

```
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 comment atque ea quae ad
effeminandos animos pertinent important, pr...
```

9. 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 standardisées.

129

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 peut les amener 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.

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 mise à disposition des données dans des formes et pour des objectifs différents, qu'il s'agisse de métadonnées ou de données textuelles ;
- de penser à la fois la production et la réexploitation interne et externe des données, pour éviter que ces standards ne deviennent des technologies de niche ;
- 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 Plokamos³¹ peuvent être théoriquement installés par tout autre projet utilisant Nemo³².

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 nouveaux impératifs numériques. Il ne s'agit plus simplement de produire des données interchangeables, 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.

Thibault Clérice – École nationale des chartes, Centre Jean Mabillon (EA 3624), PSL University

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

32. F. BAUMGARDT, *Nemo Plokamos Plugin*, <https://github.com/perseids-project/nemo_plokamos_plugin>.

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

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 proposer des solutions *open source*, réexploitables et extensibles visant à simplifier le développement et la valorisation de publications et éditions numériques. Cet ensemble d'outils permet aux chercheurs d'inscrire leur projet dans un réseau de données ouvertes et pleinement réutilisables, réduisant ainsi le coût nécessaire à la ré-exploitation de données produites.

Édition numérique – CapiTainS – humanités numériques.

CapiTainS Toolkit, Digital Editing and Data Reuse

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 aims to offer open source, reusable and extensible software solutions to simplify the development and promotion of digital editions and publications. This toolkit enables researchers to insert their project into a network of open and reusable data, and helps reduce costs in mining them.

CapiTainS – Digital Editing – Digital Humanities.