Université de Lille

Mémoire de recherche

Rapport technique

Pour un nouveau roman, édition numérique

1er septembre 2023

Sous la direction de :
Mme Florence de Chalonge
M. Matthieu Marchal

Mével Adrien
Master 2 Lettres Modernes,
« Éditions numériques et imprimées de textes littéraires »

Année universitaire 2022-2023

1 Introduction au projet d'édition numérique

Publié en 1963, *Pour un nouveau roman*, est une œuvre composite dont l'unité tient à l'adéquation entre des prises de positions assumées et un style polémique. Loin d'être (seulement) un recueil de préconisations stylistiques, le recueil offre une vue globale sur les débats littéraires de son époque dont il est le reflet et le miroir. Le texte dans son ensemble constitue une véritable théorie littéraire séparant le bon grain et l'ivraie à la fois dans la tradition littéraire et dans la littérature contemporaine.

L'édition que nous réalisons entend proposer un contenu scientifique permettant de saisir la cohérence des théories esthétiques de Robbe-Grillet et un versant numérique permettant de proposer une lecture du recueil enrichie par des éléments d'analyses et de contextualisation.

Le présent document se veut un exposé des modalités techniques mises en œuvres, des difficultés que nous avons rencontrées, suivi des moyens mis en œuvre pour les surmonter.

Chaque outil utilisé fera l'objet d'une présentation succinte, après quoi leur apport au projet et les choix ayant mené à la réalisation de l'édition numérique de *Pour un nouveau roman* seront détaillés.

1.1 Le projet initial de l'édition numérique

Lors d'un précédent rendu nous nous donnions pour objectif de mettre en valeur les relations intertextuelles qui sous-tendaient l'œuvre (voir : 1).

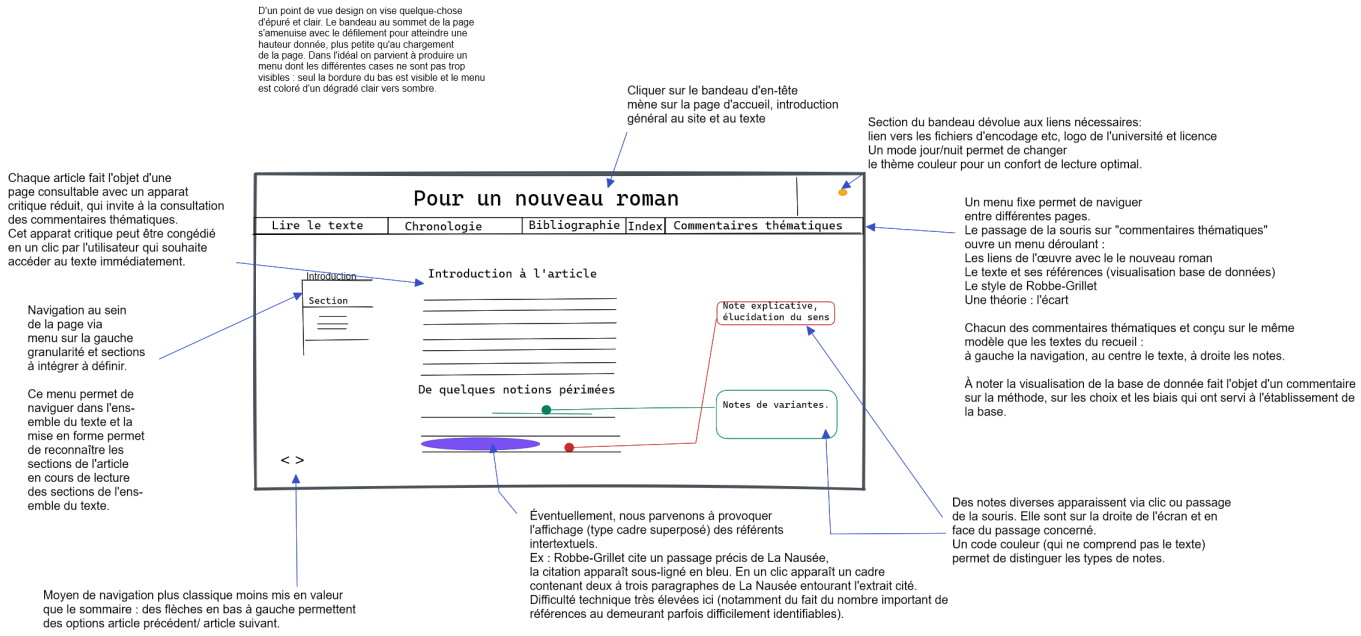

Figure 1 – Schéma du projet original

Nous opérons une première distinction entre « références » et « citations ». Lorsque p. 136 Robbe-Grillet écrit « *Mahu ou le Matériau*, ce titre est déjà un programme », il s'agit d'une référence ; mais lorsque au paragraphe précédent Robbe-Grillet écrit « "J'y pense, j'y pense [...]" », il s'agit d'une citation.

Un simple clic sur une citation (soit un passage extrait hors du texte de Robbe-Grillet) devait permettre de lire la citation en contexte (soit un ou deux paragraphes entourant le ou les passages cités). La proposition initiale envisageait déjà des limites aux citations à traiter ainsi mais les choix n'étaient pas encore faits. En effet, le nombre de citation à identifier et

le temps nécessaire à l'intégration de nombreux extraits relativement conséquents étaient les freins principaux à cette proposition, qui a, depuis été adaptée.

Dans l'état actuel, seules les citations issues des écrits de Sartre font l'objet d'un traitement proprement transtextuel. Nous privilégions cette partie du corpus car nous pensons insister dans le versant scientifique du projet sur les rapports qu'entretient la pensée de Robbe-Grillet dans *Pour un nouveau roman* avec celle de Sartre. Ainsi notre édition est-elle la plus proche possible de notre travail scientifique. Notons que nous ajoutons aux intertextes issus de Sartre, les citations issues des œuvres de Robert Pinget *Le Renard et la boussole* et *Mahu ou le matériau*. Cet ajout plutôt qu'un autre tient davantage à des raisons techniques que scientifiques : nous avons mené nos tests de faisabilité technique sur ce corpus.

— La diversité de ces œuvres dans lesquelles la digression perpétuelle fait office de narration permettent d'interroger les limites des extraits à intégrer. Afin d'intégrer le contexte d'une citation, il nous fallait déterminer quel était ce contexte ; un ou deux paragraphes ? Une phrase avant et après ? *Quid* alors des cas où les phrases elles-mêmes sont très longues voire déstructurées ? Nous optons pour un découpage dont nous espérons que la cohérence apparaîtra au lecteur ; plutôt que des bornes quantifiables en termes de distance textuelle, nous nous sommes efforcés d'intégrer des extraits tenus par une unité cohérente, que cette unité relève de l'épisode ou de la plus petite digression possible (dans le cas de Pinget).
— D'un point de vue technique, les citations issues des œuvres de Pinget nous permettent également d'identifier des citations pouvant renvoyer à plusieurs extraits d'une même œuvre, ce qui nous posait un problème technique que nous avons souhaité résoudre (voir : 5.1.2).
— Enfin Pinget nous a semblé être l'auteur pour lequel Robbe-Grillet réservait le plus d'éloge. La section « Un roman qui s'invente lui-même » est située en fin de l'« Anthologie » laissant entendre qu'il s'agit de l'auteur le plus « moderne ». Nous avions ainsi deux des pôles axiologiques mentionnés explicitement les plus éloignés.
— La version actuelle (juin 2023) ne contient pour l'instant que les citations de Robert Pinget afin de démontrer, d'une part la possibilité mais aussi l'intérêt d'un tel traitement.

Si l'objectif initial, et surtout, idéal, de ce travail était de traiter l'intégralité des citations, le temps nécessaire à un tel travail nous imposa de réduire la voilure. Ainsi, si toutes les références et citations feront l'objet d'un traitement minime : en HTMLet CSS (voir 4), nous programmons l'affichage d'une infobulle pour chacun de ces éléments lorsque la souris de l'utilisateur passe dessus, affichant : les statuts référentiels et axiologiques de la référence ou citation sur laquelle passe le curseur. Un code couleur permet au demeurant d'expliciter les valeurs (rouge, vert, bleu et noir pour les statuts axiologiques et des nuances allant du jaune au maron pour le statut référentiel).

Enfin, toujours dans un souci d'offrir la lecture la plus agréable possible, nous choisissons de ne pas produire de note explicative sur la plupart des références.

— Le contenu de ces notes nous paraîssait devoir être à la fois trop général et trop court pour avoir un quelconque intérêt scientifique.
— Malgré le temps conséquent consacré à concevoir et implémenter ces notes au contenu factuel, l'intérêt pour un lecteur néophyte qui a à sa disposition des encyclopédies en ligne nous paraissait très faible ; plus encore pour un lecteur spécialiste qui n'apprendrait sans doute rien, ou si peu de telles notes.
— Ces notes surchargeraient l'appareil critique de notre édition et, au fond, contreviendraient en partie au principe de l'édition : donner à lire, sans prendre par la main le lecteur, supposé capable de naviguer lui-même au sein de l'édition.

2 Encodage

Afin de proposer une édition numérique enrichie de *Pour un nouveau roman*, nous optons pour un encodage en XML-tei. Pensé courant octobre et novembre, l'encodage à proprement parler a commencé début mars.

2.1 Principes généraux

Le langage XML est un langage de balisage : l'encodeur entoure chacun des éléments qu'il juge devoir être encodé (l'ensemble du texte, un paragraphe, un mot, voire une lettre) de balises ouvrantes et fermantes éclairant la nature du passage qu'elles entourent (libre à l'encodeur d'encadrer un paragraphe en tant que paragraphe ou en tant que saut sémantique).

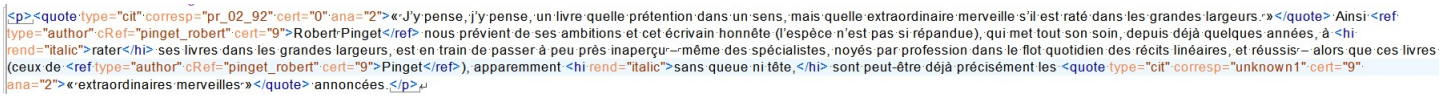

Figure 2 – Exemple d'encodage xml-tei

L'encodeur étant libre de choisir les éléments qu'il encode et comment il les encode, des conventions internationales ont émergé afin de permettre l'interopérabilité des corpus balisés. Le plus utilisé en science humaine demeure sans doute le standard de la *Text Encoding Initiative* (TEI) que nous employons pour la présente édition. Initialement pensé pour la restitution de sources anciennes (tels des manuscrits du XVI^{e}) cet encodage nous permet d'identifier des éléments structurels très fins et de mettre en correspondance différents extraits constituant un corpus (voir 2.2.2).

Lors du rapport d'étape soumis courant décembre 2022, nous avions l'intention de produire une ODD. C'est-à-dire un fichier permettant la génération d'un schéma et d'une documentation technique sur les choix d'encodage fait pour le présent travail (soit une spécialisation des éléments xml-TEI). Le temps nous a finalement manqué pour produire un tel travail qui nous a paru au demeurant en partie redondant avec le présent document.

2.2 Mise en œuvre

2.2.1 Un premier fichier d'encodage

Afin de pouvoir encoder le texte du *Pour un nouveau roman* nous nous sommes procuré une version numérique du texte en achetant la seule édition disponible de l'œuvre au format « .epub ». Nous savons que les fichiers d'un epub se présentent sous la forme d'une archive : il suffit de changer l'extension « .epub » en « .zip » pour pouvoir explorer son contenu et en extraire les fichiers contenant le texte.

Ces fichiers se présentent sous la forme de fichiers « .html », un fichier par chapitre du recueil. Grâce à un script perl nous récupérons l'ensemble du contenu textuel au sein d'un seul fichier .xml.

Le même script procède simultanément au nettoyage des fichiers d'origine (nous remplaçons les balises HTMLpar des balises xml-tei lorsque cela est pertinent et supprimons toutes les balises inutiles).

Ont été supprimés :

— les éléments vides servant à espacer le corps du texte (une indication des suppressions et des tailles des éléments supprimés est à chaque fois ajoutée en commentaire dans le fichier d'encodage)

— les éléments parasites qui redoublaient, entre autres, tous les éléments <p> sans ajouter d'information de mise en forme
— les éléments <b> et <a> muni d'un attribut @id marquant le début des chapitres

Ont été remplacés par les balises xml-tei jugées pertinentes :

— les éléments <a> munis d'un attribut @id signalant les débuts de page par des éléments <pb> munis d'un attribut @n
— les éléments <i>, au demeurant dépréciés selon les normes actuelles du web, par des éléments <hi> munis d'un attribut @rend avec pour valeur "italic"
— les éléments <p> marquant les paragraphes ont été conservés mais sans leur attribut @class de valeur "txt"
— les éléments <h1> marquant les titres ont été remplacés par des éléments <head>
— les éléments <h2> marquant les titres de sous-sections (par exemple « L'intrigue », sous-section de « De quelques notions périmées ») par des éléments <head> avec un attribut @type ayant pour valeur "subsection_head"
— les éléments <small> par des éléments <hi> munis d'un attribut @rend avec pour valeur "small-caps"
— les éléments <sup> par des éléments <hi> munis d'un attribut @rend avec pour valeur "exposant"
— les éléments <blockquote> par des éléments <cit>
— les éléments <p> et leurs attributs marquant la mise en forme du nom de l'auteur de la citation mise en exergue, par un élément <ref>

Notons que le nettoyage a été effectué en conservant, grâce à des commentaires, les traces de balises supprimées que l'on pourrait vouloir restaurer (tels les éléments utilisés dans les fichiers HTMLd'origine pour insérer du blanc dans le corps du texte).

Nous ajoutons des balises ouvrantes <quote> à chaque fois que le script de nettoyage rencontre le caractère "«" et fermante après le caractère "»", afin d'effectuer un premier repérage automatique des citations ou emprunts, qui seront ensuite complétés et corrigés à la main si besoin. Notons que le script de nettoyage ajoute également ces éléments en début et en fin de paragraphe du segment « Joë Bousquet le rêveur » où cela est nécessaire (lorsque Alain Robbe-Grillet cite plus d'un paragraphe il n'insère pas de guillemets, rendant le balisage automatique plus laborieux).

Nous ajoutons également des éléments marquant les sections du texte autour de chaque article du recueil ainsi qu'autour des passages identifiables à des sous-sections (telles les « notions périmées ») cette fois munis d'attributs @type ayant pour valeur "subsection".

Nous obtenons alors un fichier « .xml »valide qui n'est encore qu'une première étape pour l'encodage complet.

2.2.2 Vers un encodage XML-TEI en vue d'une édition numérique

Afin d'intégrer les références transtextuelles de *Pour un nouveau roman* à notre édition pour permettre l'édition enrichie que nous nous proposons de réaliser nous employons un encodage de type « corpus ».

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model href="http://www.tei-c.org/release/xml/tei/custom/schema/relaxng/tei_all.rng" type="application/xml" schematypens="http://relaxng.org/ns/structure/1.0"?>
<?xml-model href="http://www.tei-c.org/release/xml/tei/custom/schema/relaxng/tei_all.rng" type="application/xml"
    schematypens="http://purl.oclc.org/dsdl/schematron"?>
<teiCorpus version="3.3.0" xmlns="http://www.tei-c.org/ns/1.0">
    <teiHeader>
    <TEI xml:id="">
        <teiHeader>
        <text>
    </TEI>
    <TEI xml:id="">
        <teiHeader>
        <text>
    </TEI>
    <TEI>
        <teiHeader>
            <fileDesc>
                <titleStmt>
                    <title>Title</title>
                </titleStmt>
                <publicationStmt>
                    <p>Publication Information</p>
                </publicationStmt>
                <sourceDesc>
                    <p>Information about the source</p>
                </sourceDesc>
            </fileDesc>
        </teiHeader>
```

Figure 3 – La structure du fichier d'encodage

Alors que la majorité des encodages TEI se contente d'un seul élément <TEI> contenant l'œuvre ou le manuscrit encodé nous employons un élément <corpus> qui contiendra plusieurs éléments <TEI> identifiés grâce à des attributs @xml:id. Une version vide du fichier XML a pour cela été produite. Cet XML vide converti au format texte brut est ensuite injecté par le script de fusion et de nettoyage des fichiers qui compose *Pour un nouveau roman*. Il contient :

- un élément <teiHeader> (sorte de carte d'identité du document ou du texte) pour l'ensemble du fichier, contenant des informations succintes sur le projet d'édition auquel est rattaché le fichier.
- et quelques éléments <TEI> accompagnés de <teiHeader>, vides pour les extraits des références transtextuelles.
- un élément <TEI> et <teiHeader> contenant les informations relatives à l'édition de *Pour un nouveau roman*. C'est cet élément <TEI> dans lequel sera injecté le texte nettoyé et pré-encodé par le script.

Si nous reproduisons l'intégralité de *Pour un nouveau roman* dans l'élément <text> qui lui correspond nous n'insérons dans les autres éléments <text> que les extraits qui nous intéressent : nous produisons bien une édition de *Pour un nouveau roman* inscrit dans un corpus plus vaste, pas l'édition d'un corpus dont *Pour un nouveau roman* ne serait qu'un élément. Extraits transtextuels et passages de *Pour un nouveau roman* correspondant sont ensuite liés via un jeu d'attributs @corresp et @xml:id.

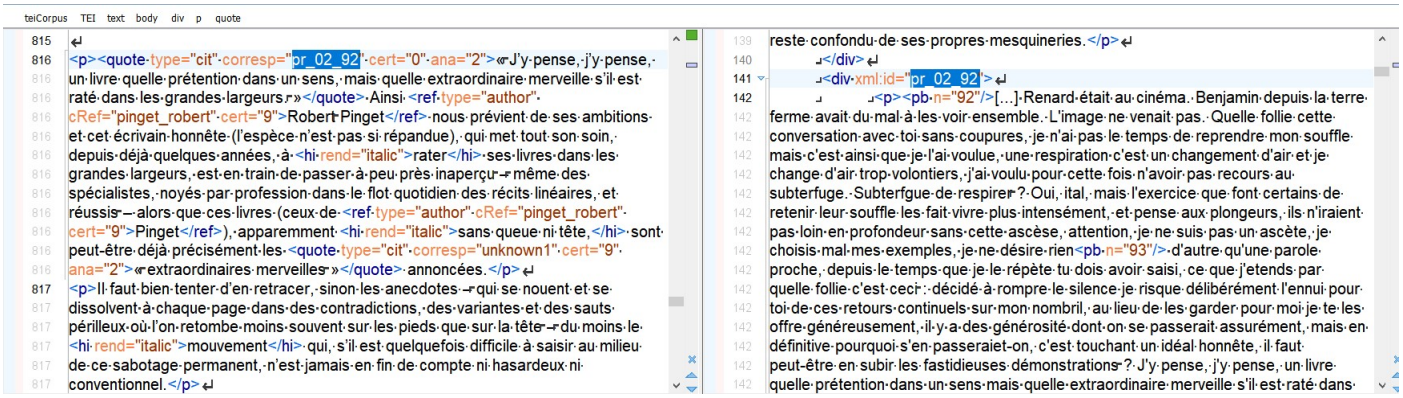

FIGURE 4 – Exemple de correspondance

2.2.3 Encodage sémantique automatisé

On désigne généralement par « encodage sémantique » l'ajout d'éléments XML identifiant des noms de personnages, des toponymes, des dates, des références, etc. Afin d'accélerer cette étape fastidieuse nous optons pour un premier encodage automatisé. Pour cela, nous ajoutons à notre script de nettoyage des lignes servant à repérer les noms d'auteurs et à les baliser selon le modèle suivant :

- un élément <ref> (référence)
- muni d'un attribut @type avec pour valeur "author"
- et d'un attribut @cRef (référence canonique) dont la valeur est constituée sur le modèle « nom_prenom ». Générée automatiquement par le script, cette valeur comprendra des majuscules et/ou des lettres accentuées (puisque Alain Robbe-Grillet écrit « Samuel Beckett » la valeur sera "Beckett_Samuel"), elle fera l'objet d'une correction grâce à l'emploi d'une feuille xsl.

Afin d'éviter le double balisage, nous balisons la dénomination complète (« Samuel Beckett ») en ajoutant un espace insécable entre le prénom et le nom, avant de baliser « Beckett » seul, s'il est précédé d'un espace sécable.

Puis grâce à une transformation XSL (voir 3)sommaire, nous générons l'affichage des éléments <quote>, <ref> et <hi> munis d'un attribut @rend="italic" afin de les identifier, corriger et compléter si besoin. Cette vérification se fait livre en main : la transformation renvoie également pour chaque élément identifié à la page à laquelle il apparaît. Cette étape est répétée plusieurs fois, ce qui nous permet de raffiner le balisage automatique à chaque passage.

Avant de passer à l'encodage manuel nous utilisons une autre transformation XSL afin de faciliter cette étape fastidieuse :

- Nous ajoutons des attributs aux éléments <quote> qui n'en ont pas, leurs valeurs doivent correspondre autant que possible aux éléments répertoriés dans notre base de données (voir 6.2.2).
  - @type correspondant à ttNature
  - @corresp qui correspond à l'@xml:id des extraits cités (encodé en <text>) et aux valeurs de ttIdent définies dans la base de données préfixé de "tt"
  - @cert qui correspond à mReferenceStatu, soit le statut de la référence (est-elle mentionnée, citée, etc.)
  - @ana correspondant au mAxiologicStatus (s'agit-il d'un blâme, d'un éloge, ou d'une mention indifférente ?).
- Nous modifions les valeurs des @xml:id des éléments qui entourent chacun des articles du texte, afin de les faire correspondre aux valeurs de notre base de donnée.
  - Par exemple, le chapitre « À quoi servent les théories » encodé par un n'aura

plus comme valeur de l'@xml:id "page006" mais "1". L'utilisation des pages comme identifiants nous paraît superflue, vu la présence d'éléments <pb> à chaque début de page.

— Nous retouchons les titres (de sections et de sous-sections) afin de corriger l'encodage d'origine qui encodait chacune des lignes d'un titre ou sous-titre ainsi que les dates dans deux éléments <head> différents. Les capitales sont également remplacées par des minuscules qui seront plus tard affichées en petites capitales.
— Les mentions des dates sont intégrées aux titres et encodées en tant que <date> avec un attribut @rend dont la valeur est "italic". Par ailleurs, nous en retirons les parenthèses qui provoquaient une erreur sans doute due au moteur d'expression régulière d'Oxygen[1], les parenthèses seront rétablies dans la version HTMLde l'édition.
— Tous les autres éléments et attributs déjà présents sont reproduits à l'identique. Par là nous nous assurons de pouvoir réappliquer la transformation sur notre fichier d'encodage manuel autant de fois que nécessaire. Nous nous contenterons de modifier le nom du fichier de sortie afin de ne pas écraser les précédentes itérations. Ainsi nous pouvons appliquer des corrections « en cours de route » sans perdre les ajouts manuels.

2.2.4 Encodage sémantique à la main

Certains contenus textuels ne peuvent être repérés automatiquement et nécessitent donc d'être encodés à la main.

En effet il convient d'attribuer les bonnes valeurs aux attributs, voire de rectifier l'encodage automatique qui ne peut, par exemple, distinguer entre la mention à la page 164 d'un titre « L'Année dernière à Marienbad » et la mention d'un élément diégétique de cette œuvre « se sont-ils vraiment rencontrés, aimés, l'année dernière à Marienbad ? » un peu plus loin.

Notons par ailleurs les difficultés à limiter le balisage. En effet l'on pourrait considérer certains éléments diégétiques, tels les allusions aux personnages des œuvres de Beckett ou les supposées réactions sus-citées que Alain Robbe-Grillet prête au public comme devant être encodées en tant que <quote> : citer un personnage d'une œuvre, n'est-ce pas citer l'œuvre ? ces réactions mêmes (re)constituées par Alain Robbe-Grillet ne sont-elles pas des éléments textuels mobilisés par Alain Robbe-Grillet à la manière de citation à réfuter ?

Pour régler ces difficultés nous nous appuyons sur les objectifs que nous nous sommes fixés au moment où nous avons élaboré ce projet d'édition numérique : nous souhaitons produire une édition qui expose les relations transtextuelles de *Pour un nouveau roman* pour le replacer dans son époque, ou plutôt pour donner les représentations que le texte produit de son époque. Dès lors, il nous paraît opportun de baliser les réactions du public décrites ou (re)constituées par Alain Robbe-Grillet afin de permettre de les comparer à la réalité de ces réactions. Au contraire les paraphrases (surtout si elles sont exactes) d'œuvres longuement commentées par Alain Robbe-Grillet, ne nous intéresse que modéremment. La source identifiée, ou plutôt, vérifiée, l'intérêt des passages cités n'ont que peu d'interêt par rapport au commentaire dans son ensemble.

De manière plus anecdotique, la recherche de chaînes de caractères contenant des apostrophes telles « L'Étranger, L'Immortelle » pose un problème épineux à résoudre car l'apostrophe est le signe utilisé par le moteur de recherche pour délimiter la chaîne. On écrit

— 'L'étranger',

[1]En expression régulière, les signes « () » servent à garder en mémoire les caractères qu'ils contiennent. Or, notre transformation recourait au moteur d'expression régulière inclus dans Oxygen pour intégrer les dates au titre. Ceci provoquait une erreur : la transformation considérait que les caractères « () » devait être interprétés comme des délimiteurs d'expression régulière et non comme le contenu à supprimer, ainsi les dates étaient bien supprimées mais pas les parenthèses les entourants.

— 'L'étranger',
— 'L""étranger' ;
le moteur renvoie une erreur. Après quelques essais nous abandonnons la correction de ce segment : pour moins d'une dizaine de mentions aisément identifiées à la main, chercher une solution trop longtemps ne présentait aucun intérêt.

L'encodage des sources citées se fait également à la main, chacun des textes du corpus est ajouté aux éléments <TEI> munis d'un attribut @xml:id servant à l'identifier sur le modèle : initialdel'auteur_numérodel'œuvre » ainsi *Mahu ou le matériau* sera désigné par « pr_01 » et *Le Renard et la boussole* par « pr_02 ». Chacun des extraits est ensuite inséré dans un élément ayant aussi un attribut @xml:id construit sur le modèle : iddel'œuvre_pagededébutdelacitation », ainsi le premier extrait de *Le Renard et la boussole* correspondra à « pr_02_09 ».

Nous employons des éléments <milestone/>, bornes, pour inscrire les points rhétoriques importants. Lors de la transformation XSL ces <milestone> seront transformés en élément <a/>, *anchor* ancre, ces éléments sans contenu seront invisibles au lecteur mais permettront de constituer un menu de navigation sur la gauche de l'écran en XSL (voir **??**). Leur attribut @id construit sur le modèle : « refutation2 » sera d'une part nécessaire à la bonne exécution du script (le lien hypertexte créé dans la navigation renvera vers cet identifiant unique à chaque ancre au sein du texte) et permettra de produire un nom compréhensible dans le menu ; « refutation2 » sera analysé par la transformation qui écrira dans le lien hypertexte « Deuxième réfutation ».

Les index des concepts adverses et des expressions privilégiées constituent des outils à l'usage des chercheurs mais également un point d'entré ludique dans le corpus.

Les expressions devant figurer dans ces index sont encodées en tant que <term/>, « terme (considéré technique) »au sein d'éléments « passage lié à une interprétation », munis d'attributs @type dont les valeurs « 0 » ou « 1 » orientent grâce à la transformation XSL, le mot vers l'index des concepts adverses ou des expressions privilégiées, respectivement. Les éléments englobe l"expression et son contexte permettant de l'expliciter (en effet la recension des adjectifs « vrai » ou « difficile » seuls serait de peu d'intérêt), l'expression encodée en <term> est ensuite mise en valeur via css, en rouge ou vert pour les notions adverses ou les expressions privilégiées respectivement.

Dans une version précédente soumise à évaluation nous nous proposions d'encoder ces éléments avec le couple <w/> « *word* » et <ab/> « *arbitrary segment* ». Cette solution a finalement était rejetée car elle n'était pas valide en xml-tei. Aussi, nous sommes-nous orientés vers des éléments plus spécifiques : employant des chercher/remplacer pour remplacer tous les élémnents <w/> déjà placés en élément <term/>.
Enfin, notre transformation XSL de balisage semi-automatique, légèrement modifiée nous permit de faire remonter l'attribut @type placé sur les éléments <w> sur les éléments .

Ces deux index prennent la forme de deux pages du site que le lecteur trouve dans un menu déroulant de la navigation en haut de page « Commentaires thématiques ». Après une courte introduction chacun des termes utilisés est listé avec une mention de page et en un clic sur le numéro de page, le lecteur peut être redirigé vers le passage du texte concerné. Si aucun des termes ne fait l'objet d'un commentaire spécifique (autre qu'à titre d'exemple), une introduction générale aux index est intégrée. Cette introduction sert à présenter cette part du travail et également à délivrer un commentaire sur l'aspect stylistique ici mis en valeur. Il ne nous a pas semblé opportun d'ajouter un commentaire pour chaque terme, d'une part car ces termes sont rarement en eux-mêmes des termes difficiles (« personnage », « intrigue » etc.) et d'autre part car un renvoi vers le passage du texte où le terme est employé nous semblait un outil bien plus intéressant autant pour le lecteur expert que pour le lecteur néophyte. Enfin, il nous semble que c'est en tant que système que ces expressions font sens.

3 Vers l'édition numérique : transformation XSL

Une fois l'encodage terminé, l'encodeur conçoit une transformation XSL, soit un fichier contenant des informations de traitement afin de passer d'un fichier XML peu lisible pour le lecteur à une édition numérique pour une lecture dans un navigateur web. En effet les feuilles de styles XSL permettent de conserver le fichier XML originel pour en créer d'autres de types divers, en l'occurence nous nous contentons de produire un site internet, soit des pages au format HTML. Notons que le langage de balisage HTMLest un dérivé de l'XML qui ne permet pas de structurer le contenu aussi finement que l'XML mais permet un affichage via navigateur web pour lecture.

3.1 Principes généraux

Les transformations XSL fonctionnent par *template*, patron, qui commande le traitement d'un ou de plusieurs éléments XML selon des restrictions diverses laissées au soin de l'auteur de la transformation. On peut par exemple transformer un élément <quote> muni d'un attribut @corresp en un lien hypertexte, qui, lié à des scripts (voir 5) permettra l'affichage de contenus supplémentaires.

3.2 Mise en œuvre

Pour tenter d'éviter une écriture redondante nous nous efforçons de produire des templates efficaces et réemployables. Par exemple pour constituer les pages HTMLde notre édition numérique il nous faut générer autant de « <header/> » (soit la section au sommet de la page) qu'il y a de pages. Aussi, nous contentons-nous de n'écrire qu'une seule fois le « <header/> » (et tous les éléments identiques sur toutes les pages) au sein d'un template nommé qui est ensuite appelé à chage génération de page avec des paramètres permettant de modifier quelques éléments essentiels qui doivent bien être uniques (tel le titre de la page).

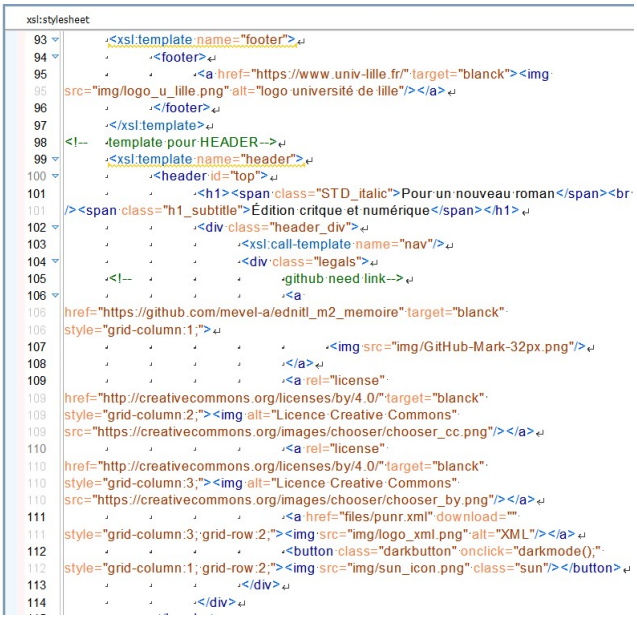

Figure 5 – Templates gérant le bas (footer) et le haut (header) des pages

Notons que la production des pages est générée par un template matchant la racine du document XML et appelant le template nommé « body » qui lui-même appelle les templates

« <header> », « <footer> », etc. adaptant le contenu de la page selon des paramètres déclarés au moment de l'appel du template.

FIGURE 6 – Extrait du template gérant la génération des pages du site

4 Résultat de la transformation : HTMLet css

Nos feuilles XSL ont transformé notre document XML difficilement lisible par un lecteur en plusieurs documents html, qui, liés à un fichier css, deviennent bien plus lisibles.

Architecture du web, le langage HTMLest, comme le langage XML dont il est un dérivé, un langage de balisage. Il s'agit ici encore de baliser des segments textuels en vu de les décrire mais contrairment à l'xml les balises utilisables en HTMLsont limitées afin d'être interprétables par un navigateur internet. Parmi ces balises on trouve :

— <p/>, un paragraphe,
— un segment de texte,
— <a/> une ancre, ou lien hypertexte.

Si le HTMLintervient au niveau sémantique, le CSS *Cascading Style Sheeet*, lui, sert à la mise en forme. C'est ce langage qui permet de transformer des segmentations sémantiques en véritable boîte sur la page ou de mettre en valeur (par un jeu de couleur par exemple) tel ou tel élément des pages.

4.1 Les notes de l'édition critique : création d'infobulles

Les interactions html/css permettent de générer des affichages utiles à notre édition numérique. Détailler avec précision les choix esthétiques et pratiques que nous avons été amené à faire n'aurait sans doute que peu de sens, cependant le travail effectué sur les infobulles mérite d'être examiné en détail, afin d'expliciter la manière dont le CSS agit sur le html.

Comme en xml-tei, nous disposons d'attributs spécifiques au HTMLpour caractériser nos éléments. Parmi eux l'attribut @class est d'une utilité particulière pour permettre les interactions entre HTMLet d'autres langages de programmation (dans le cas de notre travail CSS et Javascript). Ces attributs @class et leurs valeurs sont générés par notre transformation XSL et nous avons, dans le cas des infobulles un résultat qui ressemble à ceci :

```
<span class="ref">contenu sur lequel porte la note<span class="refinfo">la note elle-même</span></span> la suite du contenu
```

, soit deux éléments , segments textuels, le second muni d'une classe « refinfo » à l'intérieur du premier classé « ref » contient le contenu de la note qui sera en infobulle. L'affichage standard d'un tel code serait le suivant :

```
contenu sur lequel porte la notela note elle-même la suite du contenu
```

Or nous souhaitons que le second segment ne s'affiche que lorsque le curseur passe sur la souris. C'est ici qu'intervient le css.

```
    cursor: pointer;
}
.ref{
    color: #0f0;
    cursor: pointer;
    position:relative;

}
.ref .refinfo{
    display: none;
    text-indent: 0;

}
.ref:hover .refinfo{
    position: absolute;
    display: block;
    left: 20px;
    top: -30px;
    white-space: normal;
    border: 2px solid #18130e;
    border-radius: 5px;
    max-width: 600px;
    margin: 0px;
    padding: 6px;
    background-color: inherit;
    opacity: 0.96;
    filter: alpha(opacity=96); /*for IE8 et -*/
    text-align:justify;
    color: inherit;
}
```

FIGURE 7 – Le code CSS gérant les infobulles

4.1.1 Les notes liées aux références transtextuelles

Comme on peut le voir dans la figure 7, CSS emploie des *selector* auquel il attribue des propriétés. En l'occurence la *class* « ref » est sélectionnée ligne 299 et lui est attribuée une couleur (vert) ligne 300.

Nous intéresse davantage le selecteur (ou plutôt les sélecteurs additionnés pour restreindre les éléments ciblés) suivant « .ref .refinfo » qui se lit : l'élément classé « refinfo » contenu dans un élément classé « ref ». Ligne 304 lui est attribuée la propriété « display:none » qui empêche l'affichage de l'élément, le contenu de l'infobulle sera présent mais invisible. Il ne sera rendu visible que grâce à la propriété « display:block » attribuée aux éléments concernés par les sélecteurs « .ref:hover .refinfo », soit l'élément classé « refinfo » contenu dans un élément classé « ref » sur lequel l'utilisateur passe la souris (on parle de *pseudoclass* pour désigner le sélecteur « hover »). Les autres propriétés correspondent à des choix de designs pensés pour rendre l'infobulle lisible et pratique, ainsi les propriétés de positionnement « position:absolute;left:20px;top:-30px; » servent à ordonner le positionnement des infobulles selon une position absolue déterminée par rapport au dernier ancêtre positionné (en l'occurence l'ancêtre classé « ref » muni de la propriété « position:relative; »), le navigateur soustrait à ce point de référence 30 px depuis son sommet

(l'infobulle apparaît plus haut) et y ajoute 20 px depuis la gauche (l'infobulle est légérement décallée à droite).

Un tel résultat pourrait être atteint en Javascript mais l'execution d'un tel script serait légèrement plus lourde pour le navigateur et son écriture plus complexe que quelques propriétés CSS correctement agencées. Notons que les attributs @class ne sont pas seulement utilisés par le CSS mais également exploité par les scripts détaillés infra.

4.1.2 Notes critiques non liées aux références transtextuelles

Nous pourrions souhaiter ajouter des notes explicatives sur le corpus, ou simplement des notes type notes de bas de page au sein de nos commentaires.

S'il suffit pour les notes de nos commentaires d'être insérées directement au sein de des éléments HTMLapproprié pour se comporter comme des infobulles, l'ajout de note au sein du corpu nécessite un encodage particulier. Nous choisissons d'encoder le contenu de la note en tant qu'élément <note/> que nous insérons au sein d'un élément (sans attribut @type) qui contiendra la portion de texte concernée par l'annotation et l'annotation au sein de l'élément <note/>. Après quoi, notre transformation XSL vers HTMLproduit un élément classé « note » contenant l'élément classé « noteinfo » contenant le contenu de la note. Ces éléments sont classés différemment des notes liées aux références afin de permettre d'en distinguer la nature, mais leur fonctionnement est strictement identique (elles sont liées aux même propriétés CSS).

5 Une expérience de lecture : ajouts de scripts

5.1 Script pour la lecture

5.1.1 Javascript, présentation générale

Afin de permettre l'interactivité d'une page, nous employons un langage de programmation extrêmement courant : javascript. Ce langage de programmation permet la programmation de fonctions qui, selon l'élément sur lequel clique l'utilisateur, provoqueront tel ou tel comportement au sein de la page.

Une « fonction » est une suite d'instructions parfois conditionnées par des « paramètres », des informations extérieures à la fonction qui y sont injectées.

5.1.2 Afficher les citations

La première fonction que nous développons sert à afficher les extraits des autres œuvres cités ou mentionnés par Alain Robbe-Grillet. Nous choisissons de programmer un affichage que nous espérons élégant et pratique. En bas à droite de la page apparaît un encart contenant l'extrait cité, sa source et des informations quant à l'emploi de la citation (est-ce une « mention », est-ce un « blâme » ou un « éloge »?). L'encart est censé permettre de continuer à lire *Pour un nouveau roman* sans avoir à le refermer. Moins élégant peut-être qu'une version qui obscurcirait le reste de l'écran, nous pensons que permettre tel usage correspond davantage à ce que souhaiterait un lecteur effectif.

longue aiguille... » Il faut préciser : « La naissance d'un objet, j'ai remarqué, n'a pas lieu aujourd'hui, il y a du mouvement tout autour qui l'empêche de montrer la tête et demain tu t'avises qu'il existe. Par conséquent la meilleure explication des origines serait de commencer par des bruits de bouche et de glisser progressivement vers des paroles articulées jusqu'au moment où l'auditeur sans se poser aucune question participe à ton histoire. » C'est justement ce qui se produit ici ; peu à peu, au milieu des digressions et des faillites, une tache de couleur rousse s'impose dans le tableau (les héros font de la peinture dans *Le Renard,* comme ils écrivaient des romans dans *Mahu*). Cette masse d'abord indistincte prend bientôt la forme d'un renard, qui se dédouble en plusieurs personnages dont l'un n'est autre que David, le Juif errant. Ce renard invente un voyage, un voyage en Israël. Suit une espèce de reportage sur la vie dans les kibboutsim, interrompu çà et là par des évocations bibliques ou autres, des apparitions des pharaons et de sultans, des rencontres plus imprévues encore – celles par exemple de Don Quichotte et de son désert castillan. Le documentaire traîne en longueur, dans la chaleur de l'été palestinien. Le narrateur s'inquiète de cette lassitude qu'il constate chez le lecteur et chez lui-même, comme chez ses voyageurs : « Ça ne va pas, dit-il, seraient-ils déjà revenus ? » Pourtant Renard et David font encore la connaissance de Marie-Madeleine, dite « Mama », qui les aidera à se rembarquer, puis celle plus importante du Scribe accroupi : « Méfiez-vous... Ne lui parlez pas, il écrit tout. » En effet il note aussitôt lui-même sa propre rencontre... « Un fait, un fait entre des milliards, on leur dénie toute valeur dans la pratique, on les note, on les note c'est tout. »

Et le plus naturellement du monde, Renard, qui s'est depuis longtemps confondu avec le narrateur (nommé John Tintouin Porridge), se retrouve à Fantoine. Mahu et les autres lui reprochent de les avoir un instant quittés. Ne les aimait-il plus ? « Mais oui je vous aime, nous sommes liés pour la vie... me voici. » Au milieu des apéritifs et des déambulations vaseuses du retour, surgit mademoiselle Lorpailleur, la romancière ; elle demande à John où il voulait en venir. Il se rappelle

X
Statuts de la référence : mention ; éloge.

[51] [...] Une vraie conversation est avant tout futile et quant à la sonorité remplie d'hésitations et de sous-entendu, un vrai livre doit être le contraire dit-on. Quand je pense au Don Quichotte je sais que les inflexions de vois n'y sont pas traduites, mais l'art est si grand qu'aucune modulations n'échappe, aucune. Vive Cervantès. Nous laissions l'Espagne très loin derrière nous. Si j'avais su je serais retourné en Espagne. Mal connue. Jusqu'à Tolède seulement.

Le Renard et la bousssole, Pinget Robert, Paris, Les Éditions de Minuit, [1953] 2000.

X
Statuts de la référence : mention ; éloge.

[25] [...] Pourtant Renard me plaît, il me ressemble, je suis un rôdeur, je grapille à la sauvette, mon langage est incohérent, mais is la vérité est tordue et que j'ai décidé de lui laisser ma peau ? Il faut me pardonner mes singeries, rappelle-toi que don Quichotte avait un plat à barbe sur la tête, c'est ma sauvegarde.

Le Renard et la bousssole, Pinget Robert, Paris, Les Éditions de Minuit, [1953] 2000.

Figure 8 – Capture d'écran du résultat obtenu

Au moyen de notre transformation XSL, nous créons pour chacune des citations l'appel d'une fonction qui recevra en paramètre (selon la citation) l'identifiant du passage cité.

On remarque que l'on injecte en paramètre l'identifiant (@corresp) du passage cité, et les statuts référentiels et axiologiques de la référence (@cert et @ana, respectivement) ainsi qu'un dernier paramètre dont les valeurs possibles sont « 1 » ou « 0 » qui sert à orienter le comportement de la fonction nommée « displayExtract ».

```
58
59  // FERMETURE DES EXTRAITS DÉJÀ OUVERTS
60  function closeExtract(){
61      let toHide=document.getElementsByClassName('extractDisplay');
62      for (var i = toHide.length-1; i >= 0; i--){
63          toHide[i].setAttribute('class','extractHide');
64      }
65      // Suppression des paragraphes d'annotation créés, pour éviter les doublons.
66      let toDelete=document.getElementsByClassName('deleteMe');
67      for (var i = toDelete.length-1; i >= 0; i--) {
68          toDelete[i].remove();
69      }
70  }
71  function displayExtract(corresp,mAxiologicStatus,mReferenceStatus,c){
72  // FERMETURE DES EXTRAITS DÉJÀ OUVERTS
73      if(c=='1'){closeExtract();}
74  // IDENTIFICATION DE L'EXTRAIT À AFFICHER
75      let toDisplay=document.getElementById(corresp);
76  // CONSTITUTION DES ÉLÊMENTS DE COMMENTAIRES, mention du statut axiologie et référentiel
77      // 1. Tester les valeurs de l'attributs, le traduire en toutes lettres et lui donner de la couleur.
78      var mAxiologicStatusTreated='';
79      var mReferenceStatusTreated='';
80      var axiSpan=document.createElement('span');
81      var refSpan=document.createElement('span');
82      if (mAxiologicStatus=='0') {
83          mAxiologicStatusTreated='blâme';
84          axiSpan.setAttribute('class','Axi0');
85      }
86      if (mAxiologicStatus=='1') {
87          mAxiologicStatusTreated='indifférent';
88          axiSpan.setAttribute('class','Axi1');
89      }
90      if (mAxiologicStatus=='2') {
91          mAxiologicStatusTreated='éloge';
92          axiSpan.setAttribute('class','Axi2');
93      }
94      if (mAxiologicStatus=='3') {
95          mAxiologicStatusTreated='ambiguë';
96          axiSpan.setAttribute('class','Axi3');
97      }
98      if (mReferenceStatus=='0') {
99          mReferenceStatusTreated='citation explicite';
100         refSpan.setAttribute('class','Ref0');
101     }
```

Figure 9 – Extrait du script gérant l'affichage des extraits cités

```
    if (mReferenceStatus=='3') {
        mReferenceStatusTreated='emprunt non déclaré fortement suggéré';
        refSpan.setAttribute('class','Ref3');
    }
    if (mReferenceStatus=='4') {
        mReferenceStatusTreated='emprunt ou mention non déclaré(e) non suggéré(e) reconstitué(e)';
        refSpan.setAttribute('class','Ref4');
    }
    let mReferenceStatusTreatedObj=document.createTextNode(mReferenceStatusTreated);
    let mAxiologicStatusTreatedObj=document.createTextNode(mAxiologicStatusTreated);
    let separator=document.createTextNode('<0xa0>; ');
    let point=document.createTextNode('.');

    refSpan.appendChild(mReferenceStatusTreatedObj);
    axiSpan.appendChild(mAxiologicStatusTreatedObj);

    // 2 Création des éléments html etc.
    let mQuoteStatus=document.createElement('p');
    mQuoteStatus.setAttribute('class','editor deleteMe');
    let mQuoteStatusTxt=document.createTextNode('Statuts de la référence<0xa0>: ');

    mQuoteStatus.appendChild(mQuoteStatusTxt);
    mQuoteStatus.appendChild(refSpan);
    mQuoteStatus.appendChild(separator);
    mQuoteStatus.appendChild(axiSpan);
    mQuoteStatus.appendChild(point);

    // 3 Déduction de la position à laquelle on ajoute les info
        // On insérera avant le frère du premier fils de la divTodisplay,
    // soit avant l'élément qui suit la croix pour fermer (premier fils de ToDisplay),
    // ie le 1er paragraphe (de l'extrait ou de commentaire de l'éditeur).
    let firstChild=toDisplay.firstChild;
    let theOneEvenbefore=firstChild.nextSibling;
    let theOnebefore=theOneEvenbefore.nextSibling;
    // 4 Apendage
    toDisplay.insertBefore(mQuoteStatus,theOnebefore);
// AFFICHAGE DE L'EXTRAIT CLIQUÉ
    toDisplay.setAttribute('class','extractDisplay');
}
```

Figure 10 – Extrait du script gérant l'affichage des extraits cités

En effet, cette fonction n'a qu'à rendre visible le passage concerné lorsque l'utilisateur clic sur une citation. Cependant les choses se compliquent dès lors que nous souhaitons afficher plus d'un extrait pour une même citation. En effet, la même citation par exemple « Don Quichotte » dans l'article « Un roman qui s'invente lui-même » peut renvoyer à plusieurs passages de *Le Renard et la boussole*. Or l'encodage xml-tei nous permet précisément d'insérer plusieurs valeurs à l'attribut @corresp séparées par un espace.

On aura donc : `<quote corresp="pr_02_25 pr_02_50">Don Quichotte</quote>` à prétraiter car notre script Javascript ne peut recevoir qu'une seule valeur en paramètre et ne sera pas capable d'interpréter une suite de valeurs comme telle. Dès lors nous avons opté pour un prétraitement en XSL (voir 11).

Nous souhaitons arriver au résultat suivant :

```
<span onclick="displayExtract(pr_02_25_02);displayExtract(pr_02_25_50);>Don Quichotte</span>
```

, soit « lorsque l'utilisateur clique sur ce segment la fonction displayExtract est appelée deux fois sur deux extraits différents. Pour ce faire nous devons séparer les deux valeurs et répéter la valeur de l'attribut @onclick en ne changeant qu'un seul paramètre. Dans un premier template XSL nous testons la présence ou non d'un espace au sein de la valeur @corresp. S'il y a un espace une première partie de la valeur de l'attribut @onclick (soit un premier extrait) est gérée après quoi un autre template est appelé. Ce template nommé « correspAffect » reçoit en paramètre tout le contenu de l'attribut qui suit l'espace, il procède ensuite de même : gère le premier

extrait en créant l'appel de la fonction sur ce qui précède l'espace (soit le deuxième extrait), puis via un appel recursif va gérer les unes après les autres toutes les valeurs de l'attribut @corresp après l'espace. S'appelant lui-même le template boucle sur une partie toujours plus réduite de l'attribut @corresp qui correspondra à autant de paramètres ensuite envoyés dans le Javascript jusq'à ce qu'il n'y ait plus d'espace au sein du reste de @corresp, alors, le template produit un dernier appel à la fonction (s'il n'y a plus d'espace il reste l'identifiant d'un extrait) puis s'arrête.

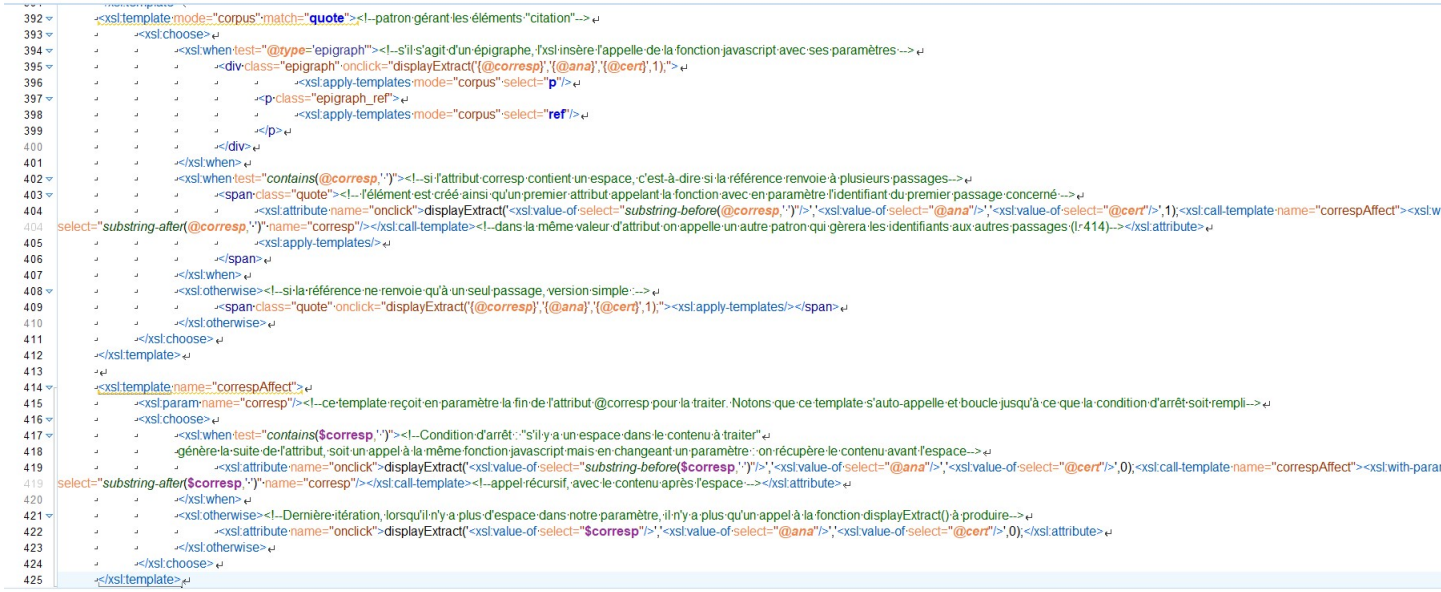

FIGURE 11 – Le template correspAffect qui sépare les correspondances multiples

Une fois tous les appels à la fonction générés, il convient de permettre à la fonction elle-même d'afficher effectivement ces extraits. En effet, afin de permettre au lecteur de refermer un encart contenant un extrait à volonté, nous avons créé une fonction qui repère les extraits affichés (s'il y en a) et les cache (on revient donc à la situation initiale). Cette fonction est attachée à un segment en haut des encarts des extraits où un « X » symbolise efficacement une croix. Cependant il paraît très probable que le lecteur ne ferme pas lui-même les encarts, nous l'encourageons d'ailleurs à le faire en lui permettant de continuer à lire le texte sans (trop) d'encombrement visuel. Aussi avons-nous décidé d'appeler la fonction refermant les encarts ouverts au sein de la fonction les affichant avant qu'elle ne les fasse apparaître. Lorsqu'un seul extrait est à afficher cela ne pose pas de problème : la fonction ferme les encarts affichées puis ouvre celui sur lequel le lecteur a clic. Cependant dans les cas où plusieurs extraits sont à afficher, la fonction les affichera puis les cachera de nouveau jusqu'au dernier appelé. Nous ajoutons donc un dernier paramètre à cette fonction : « c » dont les valeurs (« 0 » ou « 1 ») détermineront si la fonction doit, ou non, cacher les extraits précédemment affichés. Ainsi voit-on 11, lignes 419-422, l'ajout d'un '0' à la fin des appels générés s'ils ne sont pas les premiers (ou/et les seuls).

Si c'est bien Javascript qui produit l'affichage et lie plusieurs éléments HTMLgénérés via XSL depuis l'xml, c'est ici encore l'xsl qui nous permet de produire très efficacement des liens complexes entre différents langage de programmation.

5.1.3 Permettre des variantes d'affichage

Afin de permettre un confort de lecture accru, du moins personnalisable, deux boutons sont ajoutés. Le premier situé en haut à droite de la page permet de passer d'un thème clair à un

thème sombre. Le second n'est présent que sur les pages des articles de *Pour un nouveau roman* et permet d'afficher ou de masquer les numéros de page de l'édition utilisée pour le présent travail[2]. Ces deux boutons appellent des fonctions qui stockent le choix de l'utilisateur ce qui évite à l'utilisateur d'avoir à clicr de nouveau à chaque chargement de page. Notons que le choix de l'utilisateur est stocké parmi les fichiers temporaires du navigateur employé, dès lors il peut-être supprimé en vidant le cache du navigateur mais également en cliquant sur les boutons de manière à réinitialiser l'affichage, car le passage au thème clair et le masquage des numéros de page éliminent le fichier temporaire.

5.1.4 Afficher les variantes

5.2 Interactivité de la base de donnée

Afin de rendre la base de données lisible, plus aisément que dans un tableau nous employons l'outil GraphCommons[3].

Cet outil premet la production de graphique depuis un fichier de type .csv, (soit un document texte, interprétable comme un tableau par les applications de type tableur et assurant une interopérabilité très élevée notamment au sein de script perl ou javascript). Ainsi, nous exportons le contenu de la base de données au bon format puis retravaillons les données pour un meilleur affichage dans l'application. La base de données est d'abord exportée au format .csv, le tableau au fichier texte que l'on obtient a alors une colonne (les colonnes sont séparées au sein du texte par des virgules) par propriété des entitées (voir : 6.2.2) de la base afin d'être utilisable par l'application GraphCommons, les colonnes nécessitent d'être renommées et quelques valeurs ajustées. En effet, une colonne « weight » qui détermine la taille d'un lien est interprétée de manière croissante, or nous avions choisi de représenter l'intensité des liens de manière décroissante (« 0 » étant la citation et « la mention reconstituée »), pour un affichage approrpié nous modifions donc les valeurs pour les inverser et renommons les colonnes notamment grâce à des scripts perl rudimentaires.

6 Conception et réalisation d'une base de donnée

6.1 Principes généraux

On appelle base de données un mode de structuration de l'information qui permet de stocker un grand nombre d'informations sur un petit espace disque. Il existe plusieurs modèles de structuration de ces bases de données, mais pour le présent travail n'est employé que le modèle le plus courant : le modèle relationnel.

Dans une base de données relationnelle on ne stocke pas seulement des informations brutes telles « Robbe-Grillet, Une voie pour le roman futur » mais bien des informations mises en relations les unes avec les autres, chacune ayant une nature définie au sein de la base, on aurait donc plutôt : « L'auteur Robbe-Grillet a écrit l'article nommé "Une voie pour le roman futur". ».

Nous nous proposons d'illustrer via une base de données les liens que tisse chacun des textes constituant l'ensemble avec les publications antérieures et les référents (textuels ou autres) qui sous-tendent l'argumentation tout en rendant compte des différents thèmes abordés afin de donner une vue d'ensemble du recueil perçu comme un tissu de textes au sein d'un environnement dont il donne, de manière implicite et/ou explicite, une représentation.

[2]Alain Robbe-Grillet,*Pour un nouveau roman*, Paris, Éditions de Minuit, coll. « Double », [1963] 2011
[3]Voir : https://graphcommons.com/graphs/2ffc7c8c-3d1b-4814-966b-593b1c206f3c.

Les bases de données étant un mode pérenne de stockage et de partage des données, cet outil nous a semblé renforcer l'interopérabilité de notre travail. En effet, notre fichier XML étant un objet tourné vers l'édition, s'il est lui aussi pérenne et modifiable ; un ingénieur d'étude ou un chercheur au profil différent du nôtre pourrait préférer se servir de la base de donnée. Rapide, modifiable à souhait et pratique d'utilisation pour une alimentation continue au fil d'une lecture, la base de données relationnelle nous paraît un outil performant pour traiter et mettre en valeur les relations transtextuelles qui parcourent *Pour un nouveau roman* et l'intègrent au sein d'un corpus plus vaste. On pourrait s'imaginer qu'une fois la base établie et rendue accessible au public, des corrections ou des propositions émergent du lectorat. En effet si nous nous sommes efforcé de produire un travail rigoureux il paraît difficilement concevable qu'aucune erreur n'est était commise et aucune référence oubliée. On pourrait même aller jusqu'à imaginer une équipe de chercheurs, au profil plus axés humanités numériques que littérature seuls, se répartissent des sections du *Pour un nouveau roman* et entrent au fur et à mesure de leur lecture les références et citations, avant de se contrôler mutuellement.

6.2 Mise en œuvre

6.2.1 Modèle conceptuel

La première étape de constitution d'une base de données est l'élaboration d'un modèle conceptuel. Ce modèle conceptuel constitue une représentation sommaire d'une partie restreinte du monde, en l'occurrence une lecture donnée de *Pour un nouveau roman*. Les modèles conceptuels sont constitués :

- d'entités, les objets représentés (par exemple : les premières publications, les articles de *Pour un nouveau roman*, les références transtextuelles).
- chacun des objets d'une entité sont appelés « instances » (ainsi « Une voie pour le roman futur » est une instance de l'entité ARTICLES).
- d'attributs, les qualités de ces objets (par exemple : date de publication, page de début, nature de la référence).
- d'associations, les relations qui unissent les entités (par exemple : « correspond à », « mentionne »), elles peuvent également être munies d'attributs.
- chacune des associations est munie de cardinalités qui précisent le nombre minimal et maximal de fois où l'entité est impliquée dans l'association. Par exemple : l'entité « ARTICLE » peut ne pas mentionner d'entités de TRANSTEXTS et peut en mentionner un nombre virtuellement infini, la cardinalité de l'association « MENTION » à l'endroit de « ARTICLES » sera donc 0,n ; où 0 équivaut à la cardinalité minimale et n (plus d'une fois) à la cardinalité maximale.

Par convention on évite l'usage d'accents et d'espace et les noms d'entités et d'associations sont inscrits en majuscule, les entités sont représentées par un rectangle, les associations par un cercle (voir figure 12). Par ailleurs, parmi les attributs, notons la nécessité d'utiliser l'un des attributs comme clef primaire soulignée par convention, identifiant de chacun des objets.

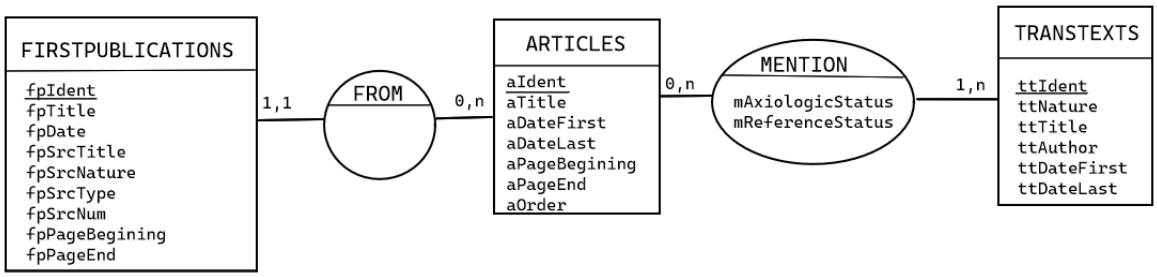

Figure 12 – Modèle conceptuel

Afin de permettre une implémentation aisée et rigoureuse nous préfixons nos attributs avec la (ou les) première(s) lettre(s) de l'entité ou de l'association à laquelle ils renvoient ; dans les cas où une association commence par la même lettre qu'une autre entité nous lui substituons les initiales des deux entités mises en relation.

Chacun des articles de *Pour un nouveau roman* constitue une instance de l'entité ARTICLES définie par un identifiant (aIdent), un titre (aTitle), une ou deux dates (aDateFirst et aDateLast) déclarée(s) par Alain Robbe-Grillet, leur place dans le recueil (aOrder) et leur étendue incarnée par deux attributs aPageBegining et aPageEnd correspondant respectivement à la première et à la dernière page de l'article.

La deuxième entité FIRSTPUBLICATIONS est liée par une association FROM à ARTICLES. Ses attributs préfixés « fp » caractérisent l'instance constituée par la première publication, ceux préfixés « fpSrc » décrivent la source de cette première publication, soit le journal ou la revue dont elle est issue. Créer une nouvelle entité pour ces sources ne nous a pas paru nécessaire car ces sources ne nous intéressent qu'en ce qu'elles induisent une tonalité (polémique, scientifique, savante, profane) ou une réception particulière aux premières publications.

Intitulée TRANSTEXTS, la quatrième et dernière entité est constituée de toutes les œuvres, auteurs ou concepts (identifiés comme étant de seconde main) mentionnées par Alain Robbe-Grillet. La nature diverse (« caricature bien connue » ou simplement « Heidegger ») des instances de cette entité explique le foisonnement d'attributs qui seront, selon les cas, sans valeur ou bel et bien mobilisés.

L'association MENTION illustre les liens qu'entretiennent les ARTICLES avec les TRANSTEXTS, les attributs mAxiologicStatus et mReferenceStatus caractérisent le lien que *Pour un nouveau roman* entretient avec telle ou telle référence. Si les valeurs possibles de mAxiologicStatus sont relativement restreintes (« eloge », « blame », « ambigue »), les valeurs de mReferenceStatus sont plus difficiles à caractériser simplement. En effet si dans certains cas Alain Robbe-Grillet cite une œuvre de manière explicite en donnant auteur et titre, il s'épargne bien souvent de donner des références précises ; alors nous faut-il être en mesure de caractériser toutes les nuances de l'implicite : l'auteur est-il cité sans être nommé ? l'emprunt manifeste est-il désigné comme un emprunt d'une source à son tour déclarée ou non ? etc. Aussi optons-nous pour un système similaire à celui mis en œuvre pour l'attribut asImportance. Si nous nous sommes efforcé d'établir un système rigoureux et adapté au texte, telles catégories ne se défont jamais tout à fait d'une appréciation subjective (voir 6.3).

6.2.2 Modèle relationnel

La deuxième étape de la constitution d'une base de données est la conversion du modèle conceptuel au modèle relationnel qui correspond à une représentation schématique de la manière dont les données seront inscrites dans la base. Le point crucial de cette conversion est la gestion des associations. Entités et associations sont remplacées par des relations ou *tables* qui, selon

les cas, illustrent des relations de dépendances ou non entre elles.

En effet, lorsqu'une seule des entités liées par une association à une autre a une cardinalité maximale de « 1 », cela signifie qu'elle n'a pas d'existence indépendante de l'autre entité. Alors l'association ne devient pas une relation mais n'est plus présente dans le modèle relationnel que par la présence d'une « clef secondaire » dans la relation dépendante de l'autre, cette clef secondaire a la même valeur que la clef primaire de l'instance cible. Au contraire, lorsque les deux entités sont reliées par une association dont les cardinalités maximales sont « n », alors l'association devient une relation contenant deux clefs secondaires : les clefs primaires des deux instances liées par l'association. Ce qui donne pour notre modèle :

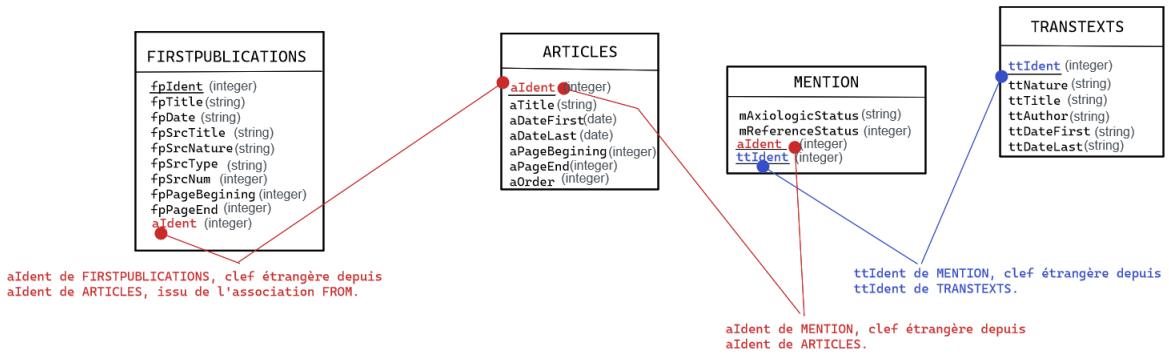

Figure 13 – Modèle relationnel

L'association FROM reliant les entités FIRSTPUBLICATIONS et ARTICLES disparaît dans le modèle relationnel car la cardinalité maximal de FIRSTPUBLICATIONS a pour valeur « 1 », laquelle est donc dépendante de ARTICLE dont la cardinalité maximale a pour valeur « n » (un article peut être une compilation ou une réécriture de plusieurs publications premières mais les articles originaux ne correspondent jamais qu'à un seul article du recueil final). Dès lors les instances FIRSTPUBLICATIONS contiennent désormais une clef secondaire qui correspond à la clef primaire d'une instance de ARTICLES.

L'association ABOUT devient une table car les deux entités qu'elle relie ont pour cardinalité maximale « n » (un même SUBJECT peut être traité par plusieurs ARTICLES et chaque ARTICLES peut traiter de plusieurs SUBJECT). ABOUT est dans le modèle relationnel une relation avec pour clef primaire deux clefs secondaires, l'une correspondant à la clef primaire de ARTICLES, l'autre correspondant à la clef primaire de SUBJECTS.

L'association MENTION devient une table car les deux entités qu'elle relie ont pour cardinalité maximale « n » (un même ARTICLES peut faire référence à plusieurs TRANSTEXTS et chaque TRANSTEXTS peut être mentionné par plusieurs ARTICLES). MENTION est dans le modèle relationnel une relation avec pour clef primaire deux clefs secondaires, l'une correspondant à la clef primaire de ARTICLES, l'autre correspondant à la clef primaire de TRANSTEXTS.

6.2.3 Implémentation

Lors de l'implémentation, nous nous connectons à un serveur local (soit un serveur hébergé sur notre propre machine) via un logiciel dédié et rentrons à la main ou grâce à des scripts les données qui prennent dans l'interface de l'application l'apparence de tableaux (on retrouve notre modèle relationnel). Les scripts servant à la création de la base n'ont en eux-mêmes que peu d'intérêt : on envoie littéralement des chaînes de caractères dans un ordre donné, dans un langage qui semble d'un anglais délesté d'une bonne part de sa syntaxe.

En décembre 2022, nous avons produit une première version de cette base de données relationnelle dans le cadre de l'évaluation du cours de base données animé par Mme Delphine Tribout. Le modèle soumis alors à évaluation nécessitait quatre entités : la base incluait une entité « SUBJECTS », un recensement des domaines abstraits dont traitaient les ARTICLES. Afin d'être le plus pertinent possible nous nous proposions de produire des valeurs les plus précises possibles pour l'attribut sDomain (« théorie Litteraire XX^{e} » plutôt que « litterature »).

Cette entité a depuis été retirée du modèle car elle nous semblait avoir peu d'intérêt tant le choix des domaines à affubler à tel ou tel article était d'une part redondant (les mêmes domaines étaient attribués à tous les articles), d'autre part le fruit d'une appréciation personnelle parfois difficile à objectiver. Si un outil est toujours le produit d'une recherche particulière et, dès lors, le résultat d'une lecture donnée, le découpage des domaines traités par les articles du recueil nous paraissait au mieux d'un intérêt limité (« À quoi servent les théories » traite de théorie littéraire du XX^{e} et de philosophie), au pire, difficilement défendable. Par exemple, nous avions réuni l'ensemble des filiations du nouveau roman tels « Faulkner » ou « Kafka » généralement mentionnés ensemble au sein du domaine « histoire littéraire internationale synchronique » plutôt que de les séparer dans des catégories par siècle et/ou pays car Alain Robbe-Grillet ne fait pas une histoire de la littérature anglaise ou tchèque mais inscrit ses références dans une histoire littéraire internationale ; on aurait pu également considérer que ces références, puisqu'elles s'inscrivent dans une volonté de décrire une filiation au nouveau roman, devraient être rattachées au domaine « théorie littéraire XX^{e} ». De manière générale, il nous semblait que l'attribution de domaine aux références, effectuée en fonction de notre lecture du recueil, induisait trop de choix problématiques pour être pleinement satisfaisante.

6.3 Mode d'établissement des valeurs des attributs

6.3.1 aOrder, conception de la structure du recueil

Il convient de noter une particularité dans la structure du recueil qui a nécessité un choix de notre part : cinq articles sont présentés dans le recueil comme des sous-sections d'un chapitre « Éléments d'une anthologie moderne », dès lors il eût pu paraître nécessaire de prévoir des valeurs de aOrder sur le modèle 5.1, 5.2 etc. dénotant sections et sous-sections, cependant dans la mesure où l'article enchâssant les cinq critiques littéraires constituant l'ensemble est ajouté *a posteriori* il nous a paru préférable de le considérer comme un article à part entière détaché de ses sous-articles qui n'entretiennent aucun lien explicite si ce n'est leur introduction, sorte de propos général ayant une fonction de seuil, ce choix nous paraît d'autant plus déterminant que l'on note l'absence de conclusion achevant de constituer l'ensemble.

De même si la table des matières de *Pour un nouveau roman* présente des sous-sections « personnage », « intrigue », « engagement » de l'article « De quelques notions périmées », ces sous-sections sont bien moins marquées dans le texte et nous semblent constituer davantage des paragraphes titrés issus d'articles fortement réécrits pour s'intégrer comme un tout homogène.

6.3.2 Valeurs de mReferenceStatus

Afin de modéliser de manière efficace et rigoureuse le statut des références, nous avons opté pour un système d'entiers inversement proportionnels au degré d'explicite des références dans le texte d'Alain Robbe-Grillet.

- Valeur **0, explicite** : citation, du moins segment présenté dans le texte comme telle dont la source (auteur ou œuvre) est mentionnée.
- Valeur **1, mention** : l'entité est mentionnée sans être citée. Il peut s'agir d'une glose interprétée (où l'interprétation de Robbe-Grillet est explicite).

— Valeur **2, mention ambiguë** : cette valeur est réservée presque exclusivement à des entités collectives mentionnées sans nécessairement que les signifiés (les auteurs désignés par « les critiques traditionnels ») soient identifiables. Pareille identification étant difficile voire impossible : on constate qu'il s'agit bien souvent d'un procédé rhétorique visant à discréditer sans les nommer des adversaires réels ou imaginaires.
— Valeur **3, emprunt non déclaré fortement suggéré** : réservée aux cas où Alain Robbe-Grillet emprunte un concept, cite ou glose une référence dont il ne donnera pas la source mais dont la paternité est suffisamment présente à l'esprit de ses lecteurs ou suffisamment appuyée par lui pour être inférée. Ainsi lorsque Robbe-Grillet disserte sur « Le petit détail qui fait vrai » à la page 176, le lecteur compétent reconnaît sans mal la conception que défendait Barthes du style de Balzac régulièrement mobilisée par Alain Robbe-Grillet.
— Valeur **4, emprunt ou mention non déclaré(e) non suggéré(e) reconstitué(e)** : réservée pour des emprunts que nous identifions sans qu'ils ne soient signalés par l'auteur. Ainsi p. 69 lorsque Alain Robbe-Grillet cite des lieux propices à la poésie romantique y glissant « vallon » nous identifions Lamartine. Enfin notons que dans ces cas comme dans les cas précédents lorsque la valeur de l'un des attributs est reconstituée par l'éditeur nous les insérons entre crochets, pour les repérer et corriger aisément si besoin mais également par honnêteté intellectuelle si pareil travail devait être amené à intégrer un travail de recherche plus vaste sur *Pour un nouveau roman*.

Faisant toutes deux l'objet d'une interprétation de l'éditeur, les valeurs « 3 » et « 4 » peuvent sembler proches. C'est en effet le cas, elles dépendent fortement de l'éditeur qui peut reconnaître des références non produites par Robbe-Grillet ou au contraire ne pas en reconnaître. Il nous a néanmoins semblé nécessaire de différencier la valeur « 4 » de « 3 » car « 4 » marque un degré d'intervention plus élevé qui pourrait relevé de la surinterprétation : si nous proposons de lire une référence à Lamartine dans l'emploi du terme « vallon » p. 69, il convient de remettre cette proposition à sa juste place. Les indices pour identifier la références sont maigres : le contexte traite d'un style anthropomorphique induisant fortement la poésie romantique sans qu'elle ne soit explicitement citée. Là où la valeur « 3 » aurait pour modèle une expression du type « un certain poète romantique ayant écrit un certain poème à propos d'un vallon » ; le lecteur interprète et risque de se tromper mais il est bien sûr que le texte suggère un auteur.

Si pareil projet devait bénéficier d'une équipe plutôt que d'un seul encodeur/éditeur, nous serions tentés de faire de la catégorie « 4 » une catégorie temporaire dont chacun des membres serait destiné à être évalué pour être passé en « 3 » ou supprimé. Nous pensons que cette catégorie « 4 » marque plus encore que les autres la subjectivité de l'éditeur et qu'il serait nécessaire ici d'avoir recourt à une forme de collégialité ou de collaboration pour faire un sort aux références identifiées comme relevant de cette catégorie.

6.3.3 Valeurs de mAxiologicStatus

Pour délimiter les valeurs de axiologicStatus nous partons de deux polarités premières, le blâme et l'éloge constituant le moteur des théories de Alain Robbe-Grillet et le cœur de sa rhétorique, auxquels nous adjoignons deux autres statuts axiologique : l'ambiguïté et l'indifférence.

S'il est aisé de reconnaître que la référence Balzac (ses œuvres ou le concept empreint des conceptions de Barthes) fait l'objet d'un blâme il est plus difficile de juger le statut d'un référent comme Stendhal qui n'est pas mentionné pour lui-même mais comme argument servant à critiquer « un jeune écrivain contemporain qui écrirait comme Stendhal ». Nous avons choisi d'utiliser les valeurs « blame » et « eloge » de manière indifférente lorsque la référence est critiquée ou vantée de manière explicite ou sollicité comme raison d'une critique ou d'un éloge portant sur une tierce référence.

Le statut « ambiguë » sert lorsque Alain Robbe-Grillet exprime de manière explicite une

difficulté à rejeter ou inclure tout à fait une référence comme symptomatique de la modernité (objet d'éloge) ou de la tradition (objet de blâme). Cette valeur sert également dans les cas où Alain Robbe-Grillet se sert d'un argument proprement sartrien pour critiquer Sartre (parfois désigné de manière implicite (valeur 3 de mReferenceStatus) par une formule telle « les engagés » ou pléthore de synonymes désignant ce que l'on qualifierait encore aujourd'hui de « stalinien »). Cette valeur marque donc également l'habilité rhétorique, d'aucuns diraient la « mauvaise foi » d'Alain Robbe-Grillet. Notons cependant que le pastiche à valeur de charge au sein de « Nouveau roman, homme nouveau » n'est pas considéré comme ambiguë car l'emprunt à Sartre ici ne sert qu'à renforcer encore une opposition frontale à ses thèses.

Enfin la valeur « indifférent » sert à désigner une mention qui n'est pas mobilisée dans l'argumentation semblant avoir une valeur plus neutre de comparaison dénuée du moindre jugement de valeur sur le référent lui-même. En effet dans les premières pages du recueil Alain Robbe-Grillet s'attaque à un « dictionnaire encyclopédique de notre temps » pour la définition que celui-ci propose de Schönberg. Si dans ce cas, la référence au musicien n'est pas tout à fait neutre (le choix de ce compositeur intellectuel supposé hermétique rappelle le nouveau roman), il est difficile de rattacher le référent « Schönberg » au système axiologique de l'essai ; c'est bien le dictionnaire encyclopédique qui fait l'objet d'une charge. Et si l'on devine une sympathie pour le musicien de la part de Robbe-Grillet, cette sympathie est inférée par le lecteur sans être partie prenante de l'argumentaire.

6.3.4 Les valeurs datées de TRANSTEXTS et FIRSTPUBLICATIONS en *string*

Dans le cas des dates de TRANSTEXTS constituées selon les cas d'une seule date ttDateFirst (de publication) ou de deux ttDateFirst et ttDateLast (naissance et mort d'un auteur ou première publication et traduction française antérieure à la publication de l'article au sein duquel la référence est mobilisée). Lors de l'implémentation des données relevant des dates un problèmes mineurs s'est posé à nous. Les données rentrées doivent être rattachées à un *domain* soit un type de valeur au sein d'une liste étendue mais fermée contenant entre autre : date, entier, décimal etc. Or SQL pose une limite au *domain* date : il ne permet pas d'enregistrer des dates antérieures à 1900. Nous avons dû opter pour le domaine *string* car certaines dates étaient antérieures à 1900.

De même dans le cas des FIRSTPUBLICATIONS, nous avons choisi dans le modèle relationnel d'utiliser le domaine *string* pour l'attribut fpDate car selon la nature de la source (quotidien, mensuel, annuel) la valeur de l'attribut changera de structure (10 octobre 1957, été 1963) ne permettant pas de l'inscrire comme une date (à moins de créer des dates arbitraires, ce qui n'aurait à nos yeux pas de sens).

6.3.5 Valeurs de l'attribut ttTitle de TRANSTEXTS

Au sein de TRANSTEXTS nous réunissons des instances de nature diverse (les discriminer est le rôle de l'attribut ttNature) pour lesquels le choix de l'attribut ttTitle peut sembler étrange : nous nous trouvons parfois à employer cet attribut pour inscrire le nom d'un concept dans la base. En effet si la dénomination de « titre » peut paraître saugrenue pour des instances de nature « concept » elle nous paraissait moins arbitraire que n'aurait pu l'être « nom » pour des œuvres, au demeurant l'emploi de « nom » ne nous satisfaisait pas non plus pour désigner les concepts. Par ailleurs considérer un concept comme un titre avait l'avantage de pouvoir plus aisément lui attribuer un auteur, voire appliquer des mises à jour ultérieures sur la base dans le cas où l'usage particulier que Robbe-Grillet fait de tel ou tel concept pourrait être rattaché à une œuvre particulière.

Notons que dans les cas où nous reconstituons une référence non mentionnée et peu sous-entendue (valeur « 4 » de mReferenceStatus) ou complétons pour paraphraser des propos allusifs de Robbe-Grillet, nous ajoutons des crochets droits à la valeur afin de noter notre intervention. Dans ces cas nous nous sommes efforcé de suivre une dénomination canonique non ambiguë.

7 Conclusion

Tout au long de notre travail nous nous sommes efforcé d'employer la technique au service de questions scientifiques. Et si, bien souvent, la technique interroge le choix scientifique, nous nous sommes évertué à ne pas confondre questions et difficultés techniques.

Sans doute, tout n'y est pas, sans doute chercher à identifier et produire toutes les citations en contexte relève de l'impossible : Alain Robbe-Grillet peut s'être trompé, l'édition avoir disparu, une citation en appelle une autre, etc. Pareil projet cependant se mène non en ayant à cœur de couvrir l'ouvrage entier (du moins sans s'en faire l'illusion) et toute son époque, mais bien en sachant que l'on ne propose qu'une lecture parmi d'autres, qui n'épuise pas l'œuvre.