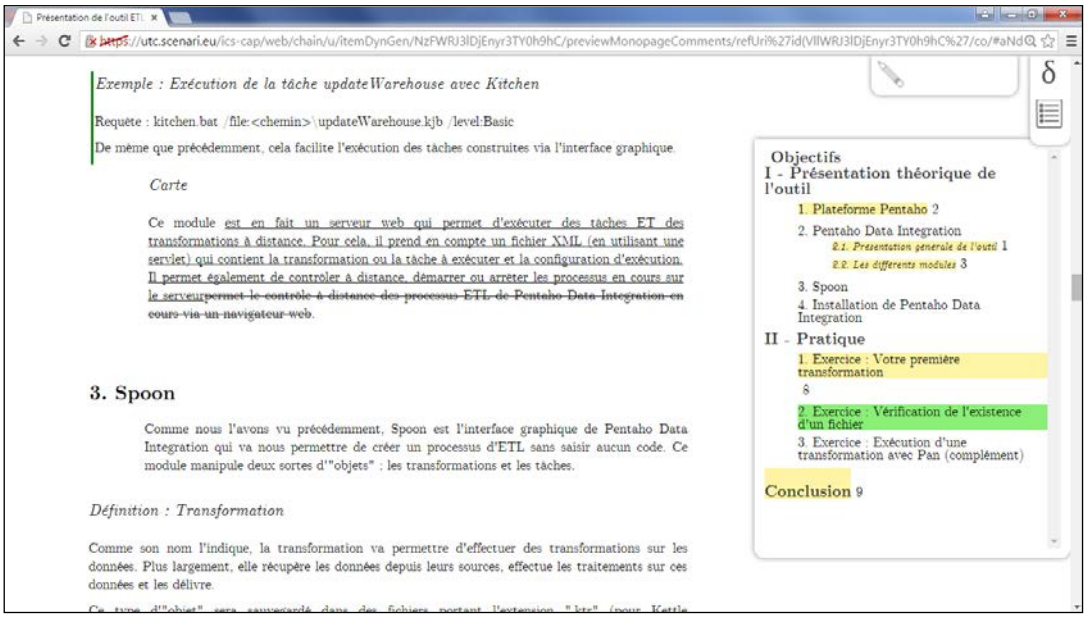
Conception de formes de relecture dans les chaînes éditoriales numériques

Léonard Dumas Milne Edwards

Par **Léonard DUMAS MILNE EDWARDS**

Conception de formes de relecture dans les chaînes éditoriales numériques

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 25 janvier 2016
Spécialité : Technologies de l'Information et des Systèmes : Unité de recherche Heudyasic (UMR-7253)

D2254

Thèse
Pour l'obtention du grade de

Docteur de l'Université de Technologie de Compiègne
Spécialité : Technologies de l'Information et des Systèmes

Conception de formes de relecture dans les chaînes éditoriales numériques

par
Léonard DUMAS MILNE EDWARDS

Soutenue le 25 janvier 2016 devant un jury composé de :

Mme Sylvie CALABRETTO	Professeur LIRIS-INSA Lyon *Rapporteur*
M. Jean CHARLET	Chargé de mission recherche (HDR) AP-HP & INSERM U1142 *Rapporteur*
M. Khaldoun ZREIK	Professeur Université Paris 8 *Examinateur*
M. Jean-Yves VION-DURY	Senior Scientist Xerox Research Centre Europe *Examinateur*
M. Serge BOUCHARDON	Professeur Université de Technologie de Compiègne *Examinateur*
M. Bruno BACHIMONT	Enseignant-Chercheur (HDR) Université de Technologie de Compiègne *Directeur*
M. Stéphane CROZAT	Enseignant-Chercheur Université de Technologie de Compiègne *Directeur*
M. Sylvain SPINELLI	Directeur Technique Kelis *Invité - encadrant entreprise*

Résumé

La production documentaire en contexte professionnel entraîne généralement un processus de révision dans lequel les documents doivent être relus avant validation et publication. Cette tâche importante fait face à de nouvelles difficultés avec le numérique. En effet, trois propriétés de l'écriture numérique sont problématiques : les documents évoluent très fréquemment et ne peuvent pas être relus entièrement à chaque version ; les interactions hypertextuelles rendent la tâche laborieuse, voire impossible ; la rééditorialisation documentaire augmente le nombre de formes documentaires à relire.

En tant que technologie d'écriture numérique avancée, les chaînes éditoriales XML sont un cadre pertinent pour l'étude de la relecture de documents numériques. Partant du constat que les formes documentaires qu'elles proposent, à savoir les formes génératrices (sources XML modifiables via un éditeur WYSIWYM) et les formes publiées (documents issus de la transformation des sources XML), font défaut à la relecture, nous envisageons la conception de formes documentaires dédiées à cette activité selon deux approches : la linéarisation, qui consiste à restaurer une certaine linéarité matérielle des contenus pour faciliter leur relecture exhaustive ; et la tabulation, qui vise à paralléliser, afin de mieux les comparer, les différents contextes de rééditorialisation d'un document.

Une partie des propositions faites dans ce mémoire a mené à la réalisation de prototypes ayant été expérimentés dans des situations d'usage des chaînes éditoriales Scenari en contexte pédagogique. Ces prototypes s'appuient sur des formes linéaires de relecture permettant notamment la comparaison de deux versions du document en se basant sur un algorithme de différentiel.

Mots-clés

Chaînes éditoriales XML, document fragmenté, relecture, différentiel, interactivité, rééditorialisation, polymorphisme.

Abstract

Documentary production in a professional context often involves a revising process in which documents need to be proofread before validation and publication. This important task faces new challenges when dealing with digital documents. Indeed, three features of digital writing are problematic: documents evolve very frequently and cannot be proofread each time as a whole; interactions provided by hypertexts make the task laborious or even impossible; document repurposing increases the views of content to proofread.

As an advanced digital writing technology, XML publishing chains are a relevant framework for studying proofreading of digital documents. Observing that the views of content proposed by publishing chains, namely the generative views (XML sources that can be modified through a WYSIWYM editor) and the published views (documents obtained by transformation of the XML sources), are not adapted for proofreading, we consider designing new views of content dedicated for this activity based on two approaches: linearization, which consists in restoring some material linearity among contents; and tabulation, which aims at parallelizing different repurposing contexts so that they can be better compared.

Part of the contribution presented here has led to the development of prototypes that have been experimented in the use of Scenari publishing chains in a pedagogical context. These prototypes rely on linear proofreading views allowing in particular the comparison between two versions of the document based on a diff algorithm.

Keywords

XML Publishing Chain, fragmented document, proofreading, diff, interactivity, repurposing, polymorphism.

Remerciements

Avant toute chose, je remercie vivement Stéphane Crozat et Bruno Bachimont, mes directeurs de thèse, ainsi que Sylvain Spinelli, directeur technique de Kelis, pour leur encadrement au cours de ces trois années. Stéphane, merci pour ta disponibilité et les conseils avisés que tu m'as donnés toujours avec enthousiasme. Bruno, merci pour la richesse de ton apport conceptuel qui m'a orienté dans le cheminement de ma recherche. Sylvain, merci pour ta clairvoyance et ton appui très précieux sur la technique et les usages. Merci aussi pour votre sympathie, votre bienveillance et votre humour qui ont fait de nos réunions des moments stimulants aussi bien humainement qu'intellectuellement pour moi.

J'adresse mes sincères remerciements à Sylvie Calabretto et Jean Charlet pour avoir accepté d'évaluer ce travail en tant que rapporteurs. Je remercie également les autres membres de ce jury de thèse, Jean-Yves Vion-Dury et Khaldoun Zreik pour les échanges que nous avons pu avoir à l'occasion de DocEng et de CIDE, et Serge Bouchardon dont j'ai suivi avec plaisir et intérêt les enseignements lors de mon cursus d'ingénieur.

Je suis très reconnaissant envers la société Kelis de m'avoir accompagné et fait confiance tout au long de la thèse. Merci à Xavier et Eric pour m'avoir partagé leurs expériences du métier, à Samuel pour son appui sur SCENARIbuilder, à Christelle pour son aide permanente, à Thibaut pour ses nombreux conseils, et à tous mes autres collègues. Mes remerciements vont également à Manuel Majada et tous les membres de l'unité ICS, Martine (un grand merci pour toute ton aide !), Stéphane P., Lionel, Dorine, Hamid, Sophie, Julie et Gabriel pour leur accueil, leur aide et leur sympathie. Merci aussi au laboratoire Heudiasyc et à l'équipe ICI pour leur accompagnement scientifique, ainsi qu'à l'École Doctorale de l'UTC. Enfin, je tiens à remercier Ludovic Gaillard et Unisciel de m'avoir donné l'opportunité de travailler sur le projet Faq2Sciences, l'IFCAM (et particulièrement Céline Bur) et l'UCANSS, clients de Kelis, de m'avoir permis d'illustrer mes travaux par leurs usages de Scenari, ainsi que Nicolas Salzmann du temps qu'il m'a accordé pour m'amener à voir ma thèse sous l'angle de l'analyse de la valeur.

Merci à tous les doctorants et docteurs que j'ai pu côtoyer, et particulièrement Thibaut, Antoine, Kevin et Rémy, ainsi qu'à mes amis de Rueil, de Paris, de l'UTC et d'ailleurs.

Pour finir, je remercie ma famille pour son entourage et son soutien inestimables depuis toutes ces années. Je pense énormément à ma mère, à la mémoire de qui je souhaite dédier mon travail.

Table des matières

Introduction 7
Chapitre 1 : Contexte métier 9
1.1 Chaînes éditoriales 9
1.2 Relecture 23
Chapitre 2 : Problématique 27
2.1 Le document numérique : entre instabilité, interactivité et rééditorialisation 28
2.2 Les chaînes éditoriales : une technologie de rééditorialisation documentaire 34
2.3 Vers la conception de formes de relecture 37
2.4 Cadre théorique 38
Chapitre 3 : État de l'art 43
3.1 Différentiel 44
3.2 Annotation 54
3.3 Correction automatique 58
Chapitre 4 : Propositions 66
4.1 Linéarisation 67
4.2 Tabulation 82
Chapitre 5 : Expérimentations 95
5.1 Contextes d'usage 95
5.2 Outils proposés 98
5.3 Retours d'usage 104
5.4 Évaluation 109
Conclusion 111
Bibliographie 113
Annexes 117

Introduction

Avec l'imprimerie s'est instaurée une « confiance dans la factualité des textes [et une] assurance dans leur établissement » (Cerquiglini, 1989, p. 17). Il n'en est pas de même pour les textes antérieurs à cette invention, la recopie manuscrite ayant immanquablement entraîné la profusion de variantes. C'est ainsi qu'a émergé la discipline de la philologie, dont le but est d'établir la version de référence d'un texte en prenant pour témoins les variantes qui en subsistent, à défaut d'avoir conservé l'original.

La philologie est née avec la bibliothèque d'Alexandrie (288 avant JC) : les travaux d'édition menés par les grammairiens Zénodote, Aristophane de Byzance puis Aristarque de Samothrace ont notamment permis de fixer les premières versions de référence de l'œuvre d'Homère (fin du 8ème siècle avant JC), dont ils avaient hérité de nombreuses copies.

La méthode classiquement employée en philologie consiste à comparer plusieurs manuscrits en relevant les passages où ils divergent (*collation*), les différentes versions d'un même passage étant appelées *leçons*. La version de référence peut être établie par sélection du manuscrit considéré comme le meilleur représentant de l'original, ou bien par sélection, sur chaque passage, de la "meilleure leçon". Dans certains cas, les leçons non-fautives sont consignées dans l'*apparat critique* du texte édité.

Pour Cerquiglini (*ibid.*), la philologie (particulièrement celle du 19ème) a longtemps reposé sur une « pensée de la faute », qui néglige le fait qu'un scribe ait pu faire lui-même un travail d'édition critique à partir de plusieurs sources, ou bien une réécriture consciente et valable (typiquement, des variantes syntaxiques ont pu être utilisées pour certains mots, alors même que l'orthographe n'était pas totalement fixée). Cette philologie s'est d'ailleurs montrée peu efficace pour l'édition des textes médiévaux (*La Chanson de Roland*, *Le Lai de l'Ombre*, etc.), dont la glose et la paraphrase étaient constitutives. À la « pensée de la faute », Cerquiglini oppose ainsi l'« éloge de la variante ».

> En conclusion d'une contribution à la revue Genesis, Cerquiglini suggère que le numérique remet au jour la variance comme modèle d'écriture et appelle à une nouvelle philologie :
>
> « De l'*éloge de la variante* à une *apologie de l'hypertexte*. Par sa non-linéarité, sa faible hiérarchisation, sa forte connectivité ce dernier fournit à l'édition critique le cadre notionnel et technique qu'elle requiert afin de rendre compte des processus scripturaux protéiformes et variants. Rompant avec le standard de l'imprimé, démultipliant la trace écrite par l'image et par le son, cette philologie hypertextuelle redonne vie par ailleurs à des pratiques d'appropriation culturelle et de production du savoir que le livre, dans son essor magistral, avait fâcheusement reléguées dans l'ombre ; elles se rallument avec nos écrans. » (2010)

Les travaux présentés dans ce mémoire s'inscrivent dans le domaine de l'ingénierie documentaire et dans le contexte technologique des chaînes éditoriales numériques. Plus précisément, notre recherche aborde la problématique de la relecture menée au sein de la production documentaire instrumentée par une chaîne éditoriale, qui selon nous s'inscrit dans le cadre d'une philologie des documents numériques. Nos travaux ont été menés dans le cadre des activités de recherche et développement menées conjointement par la société Kelis, éditeur de la suite logicielle Scenari, et l'unité Ingénierie des Contenus et des Savoirs (UTC).

En reconfigurant le document dans sa dimension technique, le numérique a transformé nos manières d'écrire, de lire et plus largement de penser, comme l'avait envisagé Vannevar Bush (1945) dans son article « As we may think ». Devenu interactif, modifiable en permanence, rééditorialisable dans de nouveaux contextes, etc. le document numérique renouvelle en effet les conditions de production et d'accès aux savoirs, mais devient en même temps plus difficile à relire et à valider du fait de cette variance intrinsèque.

Cette problématique se manifeste en particulier au niveau des chaînes éditoriales, dont l'enjeu est de

proposer de nouvelles fonctions d'écriture tirant parti des spécificités du support numérique. Pour y parvenir, ces systèmes reposent d'une part sur la séparation entre une *forme génératrice* et des *formes publiées* (Crozat, 2012a), les secondes étant obtenues par différentes transformations de la première ; et d'autre part sur la fragmentation des documents au sein d'un graphe permettant d'instrumenter la rééditorialisation (Arribe, 2014). Nous soutenons que le numérique, à la base de notre problématique, permet aussi de la résoudre à travers le polymorphisme, soit le fait d'engendrer différentes formes documentaires à partir d'une même forme génératrice. L'enjeu de notre recherche est alors de proposer des *formes de relecture*. Ces formes ont une visée philologique dans le sens où elles doivent être des versions de référence permettant au relecteur de valider efficacement un document.

En nous appuyant sur un ensemble de cas d'usage dans les chaînes éditoriales (modèles documentaires, corpus...) mettant en jeu les différentes propriétés du numérique identifiées dans la problématique, nous proposons deux stratégies de conception de formes de relecture : la linéarisation et la tabulation. Sur le plan expérimental, nous avons mis en en œuvre la linéarisation dans deux contextes d'usage, et développé un outil de relecture basé sur un algorithme de différentiel développé par Kelis.

Organisation du mémoire

Ce mémoire est structuré en cinq chapitres.

Le premier chapitre s'attachera à présenter les chaînes éditoriales et la relecture, et à identifier les questions émanant de leur articulation.

Le second chapitre élargira la question de la relecture au document numérique dans son ensemble, à travers l'analyse de trois de ses propriétés : l'instabilité, l'interactivité et la rééditorialisation. Rapportant cette question aux différentes formes documentaires mises en jeu dans les chaînes éditoriales, nous envisagerons la nécessité de concevoir des formes de relecture. Nous présenterons ensuite les éléments théoriques concernant la notion de document et son évolution avec le support numérique. Il nous permettra de mettre en lien les propriétés abordées dans ce chapitre avec un ensemble de tropismes caractérisant le numérique.

Le troisième chapitre explorera l'état de l'art concernant les fonctions d'aide à la relecture qui répondent partiellement à notre problématique. Trois fonctions seront présentées : le différentiel, l'annotation et la correction automatique.

Le quatrième chapitre présentera la linéarisation et la tabulation, qui constituent nos propositions théoriques.

Le cinquième chapitre donnera une vue des expérimentations que nous avons menées. Il détaillera les contextes d'usage abordés, les prototypes développés ainsi que les retours d'usage. Nous décrirons ensuite le cadre épistémologique dans lequel nous positionnons nos travaux et dresserons finalement un bilan de notre recherche.

Chapitre 1

Contexte métier

1.1 Chaînes éditoriales 9
1.1.1 Exemple 10
1.1.2 Scenari 13
1.1.3 Opale 16
1.1.4 Topaze 21
1.2 Relecture 23
1.2.1 En révision professionnelle 23
1.2.2 Dans les chaînes éditoriales 25

Dans ce chapitre, nous commencerons par rappeler la définition des chaînes éditoriales, avant de détailler leur instrumentation dans la suite logicielle Scenari. Nous présenterons ensuite l'activité de relecture telle qu'elle est décrite en révision professionnelle, puis illustrerons la façon dont elle prend place dans les chaînes éditoriales à travers un exemple d'usage réel.

1.1 Chaînes éditoriales

« Une chaîne éditoriale est un procédé technologique et méthodologique consistant à réaliser un modèle de document, à assister les tâches de création du contenu et à automatiser la mise en forme. » (Crozat, 2007, p. 2).

Une première notion essentielle de cette définition est celle de *modèle de document*. Un modèle décrit la structuration du document indépendamment de son contenu. Il sert à la fois à guider l'auteur dans l'écriture du document et à contrôler la validité structurelle de ce dernier. On parle alors de document *structuré* (André *et al.*, 1989). Un document XML est un document structuré dont le modèle est spécifié par un schéma suivant le formalisme DTD (W3C), XML Schema (W3C) ou encore RelaxNG (OASIS).

La seconde notion est celle d'*automatisation de la mise en forme*. Dans une chaîne éditoriale, la mise en forme est effectuée par un algorithme, par exemple un ensemble de *templates* XSL (W3C), transformant le document XML dans un autre format dédié à la publication (HTML, PDF, etc.). Cette transformation se base sur des règles formalisées à partir de la connaissance du schéma.

En étant séparé de sa forme finale, un document peut légitimement prétendre à la publication multi-supports (Bachimont et Crozat, 2004). Pour cela, le modèle de document doit décrire uniquement la structuration *logique* du contenu, c'est-à-dire une structure faisant abstraction des différentes mises en forme possibles. Les éditeurs spécialisés dans la séparation fond/forme sont appelés éditeurs WYSIWYM (*What you see is what you mean*) (Power *et al.*, 1998 ; Van Deemter et Power, 2000), par opposition aux éditeurs WYSIWYG (*What you see is what you get*) traditionnellement utilisés dans les logiciels bureautiques (logique mono-support).

Une troisième notion sur laquelle insiste Crozat est celle de *réutilisation*, qui consiste à « référencer un fragment de contenu depuis plusieurs constructions documentaires » (Crozat, 2007, p. 50).

Approche industrielle

L'ingénierie industrielle est définie par Bachimont (2007) comme la standardisation de la production en vue de sa répétabilité. Elle se distingue de l'ingénierie artisanale, basée sur la production d'"œuvres uniques" à partir de techniques improvisées ou réinventées (*ibid.*).

Dans (Bachimont *et al.*, 2002), les auteurs abordent le problème de la massification de la production documentaire que rencontrent les organisations. Par exemple, un organisme de formation doit gérer la production des différentes supports pédagogiques (transparents de présentiel, livret apprenant aux formats web et papier...) pour chacune de ses offres.

Face à des outils auteur artisanaux (par exemple Dreamwaver pour l'édition WYSIWYG de pages web), « l'auteur doit simultanément concevoir son ingénierie pédagogique, formaliser ses savoirs, réaliser des choix ergonomiques et se préoccuper de l'esthétique de ses productions » (*ibid.*). Au contraire, les chaînes éditoriales s'inscrivent dans une approche industrielle. En effet, la production documentaire peut être homogénéisée sur l'ensemble des documents à produire, tant du point de vue de la structure que de la mise en forme (supports, charte graphique...). Déchargés des tâches de mise en forme, les auteurs peuvent se focaliser sur le contenu. Enfin, la réutilisabilité des contenus permet de mutualiser les mises à jour de plusieurs documents.

Ainsi, une chaîne éditoriale « permet de réduire les coûts de production et de maintenance des contenus, et de mieux contrôler leur qualité » (Crozat, 2007, p. 2).

1.1.1 Exemple

DITA (*Darwin Information Typing Architecture*) est un langage XML utilisé dans la rédaction de documents techniques. Soit la procédure DITA suivante (*respirationCirculaire.xml*) :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE task PUBLIC "-//OASIS//DTD DITA Task//EN"
"http://docs.oasis-open.org/dita/v1.1/OS/dtd/task.dtd">
<task>
    <title>La respiration circulaire</title>
<taskbody>
<steps>
  <step>
    <cmd>Commencez à souffler dans votre instrument.</cmd>
  </step>
  <step>
    <cmd>Quand vous arrivez à bout de souffle, gonflez légèrement les
joues tout en continuant à souffler, afin de constituer une réserve d'air.
</cmd>
  </step>
  <step>
    <cmd>En inspirant brièvement par le nez, expulsez l'air en vidant les
joues.</cmd>
  </step>
  <step>
    <cmd>Une fois les poumons à nouveau remplis, reprenez une expiration
normale.</cmd>
  </step>
</steps>
</taskbody>
</task>
```

Un schéma XML, défini par exemple dans le langage DTD, permet de contrôler la structuration d'un document XML. Soit la DTD suivante (*task.dtd*, simplifiée par rapport à DITA) :

```
<!ELEMENT task (title, taskbody)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT taskbody (context, steps, result?)>
<!ELEMENT context (#PCDATA)>
<!ELEMENT steps (step+)>
<!ELEMENT step (cmd, substeps*)>
<!ELEMENT cmd (#PCDATA | keyword)*>
<!ELEMENT keyword (#PCDATA)>
<!ELEMENT substeps (steps*)>
<!ELEMENT result (#PCDATA)>
```

La procédure précédente est valide par rapport à ce schéma. Notons que l'élément racine (*task*) n'est pas précisé dans la DTD mais au niveau de la déclaration du type de document (*DOCTYPE*) dans le fichier XML.

Les *templates* XSL suivants permettent de transformer la procédure au format HTML :

```xml
<xsl:template match="task">
  <h1><xsl:value-of select="title/text()"/></h1>
  <xsl:apply-templates select="taskbody"/>
</xsl:template>

<xsl:template match="taskbody">
  <xsl:apply-templates select="context"/>
  <ol>
    <xsl:for-each select="steps/step">
      <xsl:apply-templates select="."/>
    </xsl:for-each>
  </ol>
  <xsl:apply-templates select="result"/>
</xsl:template>

<xsl:template match="step">
  <li>
    <xsl:apply-templates select="cmd"/>
    <xsl:if test="substeps">
      <ol>
        <xsl:for-each select="substeps/step">
          <xsl:apply-templates select="."/>
        </xsl:for-each>
      </ol>
    </xsl:if>
  </li>
</xsl:template>
```

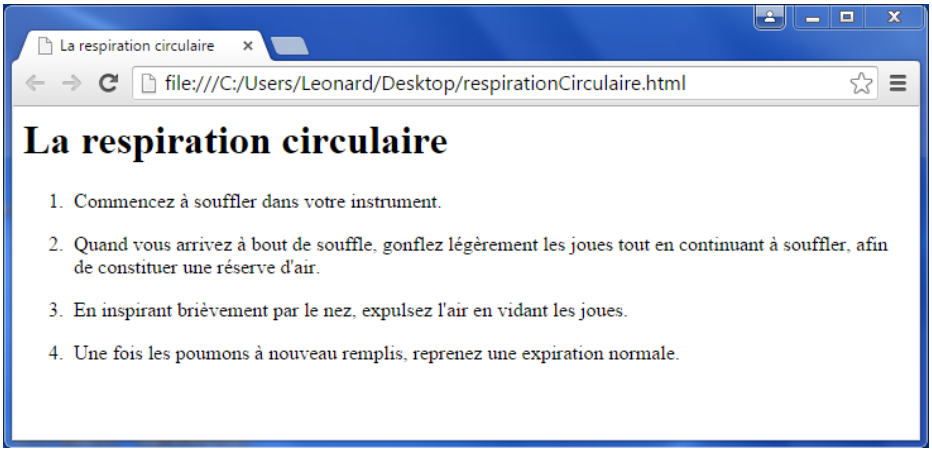


Figure 1 - respirationCirculaire.html.

DITA permet également de répertorier les procédures dans une *map* (*techniquesEtendues.map*) :

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map PUBLIC "-//OASIS//DTD DITA Map//EN"
 "http://docs.oasis-open.org/dita/dtd/map.dtd">
<map>
    <title>Les techniques de jeu étendues</title>
    <topicref href="pianoPrepare.xml"/>
    <topicref href="slapTongue.xml"/>
    <topicref href="respirationCirculaire.xml"/>
</map>
```

Par ailleurs, la procédure peut être référencée dans une autre map (*techniquesImpro.map*) :

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE map PUBLIC "-//OASIS//DTD DITA Map//EN"
  "http://docs.oasis-open.org/dita/dtd/map.dtd">
3 <map>
4     <title>Les techniques d'improvisation</title>
5     <topicref href="respirationCirculaire.xml"/>
6     <topicref href="musiqueModale.xml"/>
7     <topicref href="gammeParTons.xml"/>
8 </map>
```

À l'aide de deux transformations XSL différentes, ces *maps* pourraient être publiées au format HTML d'une part, sous la forme d'une liste de liens vers chacune des procédures ; et au format PDF d'autre part, dans une publication où toutes les procédures sont les unes à la suite des autres :

Figure 2 - techniquesEtendues.html.

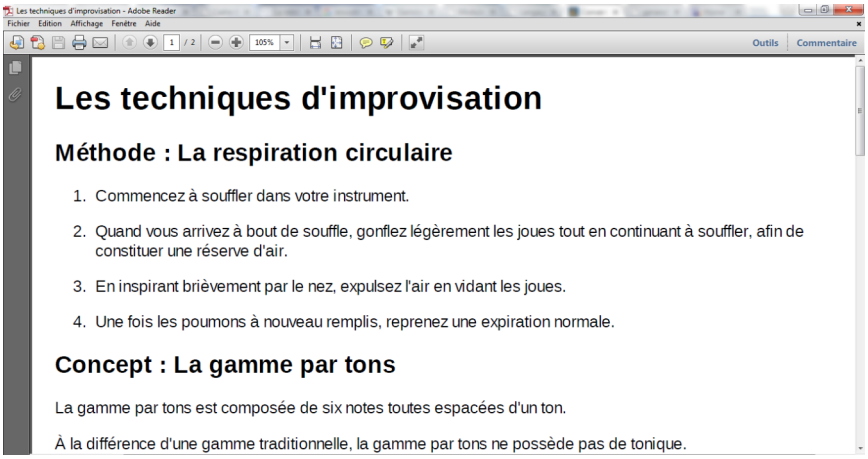


Figure 3 - techniquesImpro.pdf.

1.1.2 Scenari

La suite logicielle Scenari est constituée de deux environnements principaux :

- SCENARIchain pour l'édition suivant les principes d'une chaîne éditoriale (WYSIWYM, publication multi-supports, fragmentation, réutilisabilité, etc.) ;
- SCENARIbuilder pour la conception d'un modèle documentaire [p.119] visant à être installé et utilisé dans SCENARIchain.

Un modèle documentaire correspond à l'ensemble des ressources informatiques mobilisées par une chaîne éditoriale, à savoir :

- des schémas XML contrôlant la structure des fragments (fichiers XML) que l'auteur peut instancier ;
- des transformations XSL qui, appliquées aux fragments, permettent la publication de documents au format HTML, PDF, etc..

Édition et gestion

Dans SCENARIchain, la production documentaire est organisée en ateliers. Chaque atelier instancie un modèle documentaire et dispose de fonctionnalités d'édition (via un éditeur WYSIWYM, spécialisé en fonction du modèle) et de gestion (arbre de classement, recherche, réseau de fragments...). SCENARIchain peut être utilisé en local ou bien en mode client-serveur. Depuis la version 4 de Scenari, un mode de stockage des fragments s'appuyant sur une base de données permet notamment d'instrumenter des fonctions collaboratives, entre autres :

- la gestion du cycle de vie des fragments ("brouillon", "à valider", "validé"...) ;
- l'ajout de *calques* (Arribe, 2014) aux ateliers afin de permettre de modifier des fragments de façon temporaire (modifications vouées à être réintégrées à l'atelier d'origine) ou définitive (dérivation des fragments pour un autre contexte de diffusion) ;
- l'historique des modifications d'un fragment.

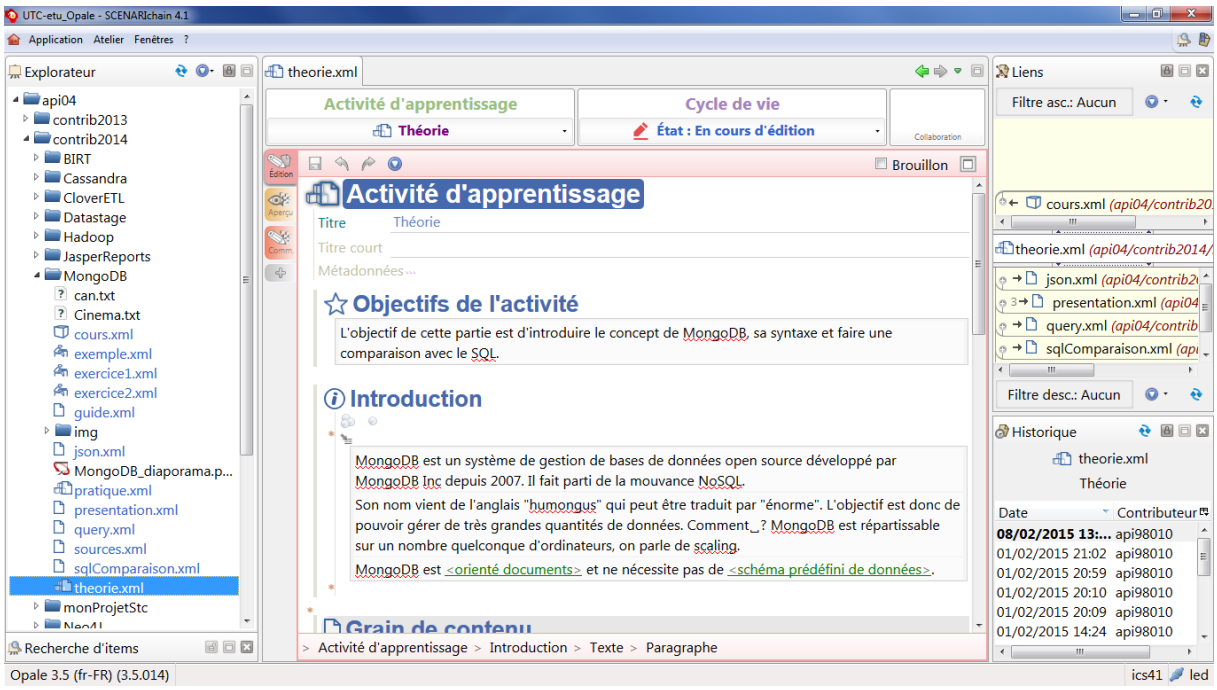


Figure 4 - Écran d'un atelier dans SCENARIchain.

L'écran ci-dessus est composé de trois zones principales :

- à gauche, l'arbre de classement ;
- au centre, l'éditeur WYSIWYM d'un fragment, avec la gestion de son cycle de vie en haut ;
- à droite, les liens ascendants et descendants (en haut) ainsi que l'historique (en bas) de ce fragment.

Publication et prévisualisation

Dans un modèle documentaire, les transformations sont définies au niveau de certains types de fragment appelés *racines de publication*. Un document publié est un fichier ou ensemble de fichiers, prêt à être diffusé sur un serveur FTP, envoyé par mail, etc.. La transformation peut également être exécutée dynamiquement (pendant le processus d'écriture par exemple), afin de prévisualiser le document dans une mise en forme proche du résultat de la publication. À travers un système de

commentaires lié aux sources XML, il est possible de faire annoter (par un relecteur typiquement) les contenus depuis la prévisualisation, et de retrouver ces annotations dans les fragments au niveau de l'éditeur (notons que les annotations sont expurgées des publications classiques) :

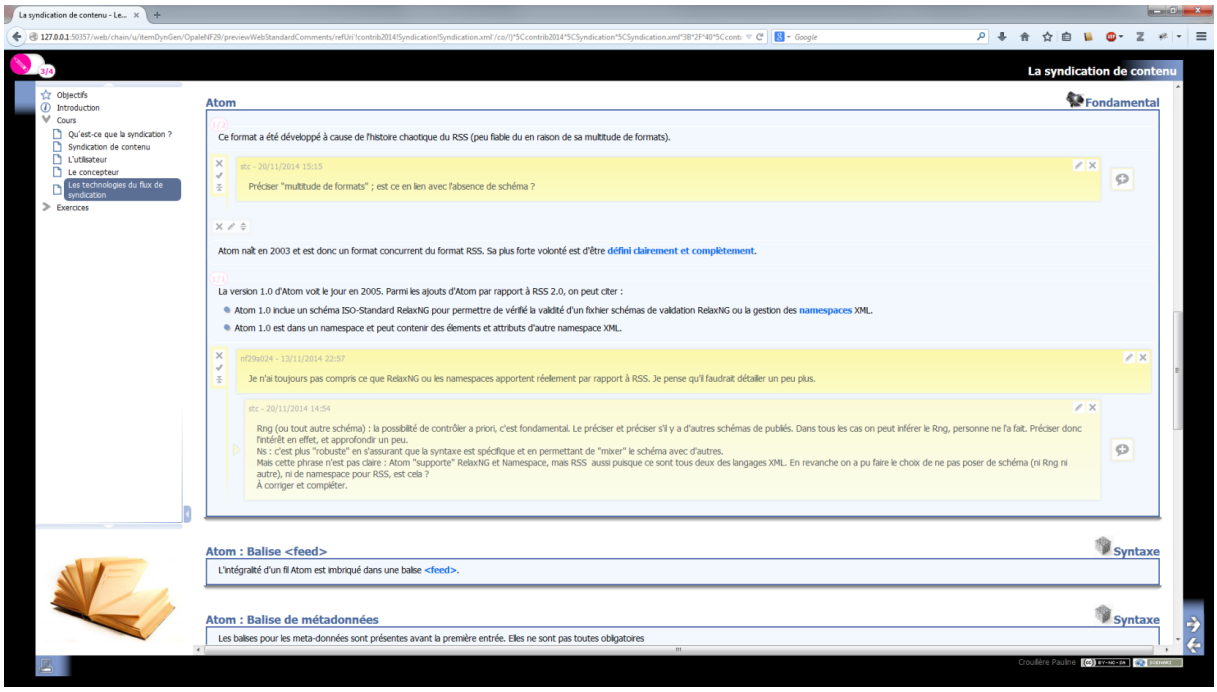


Figure 5 - Prévisualisation avec commentaires.

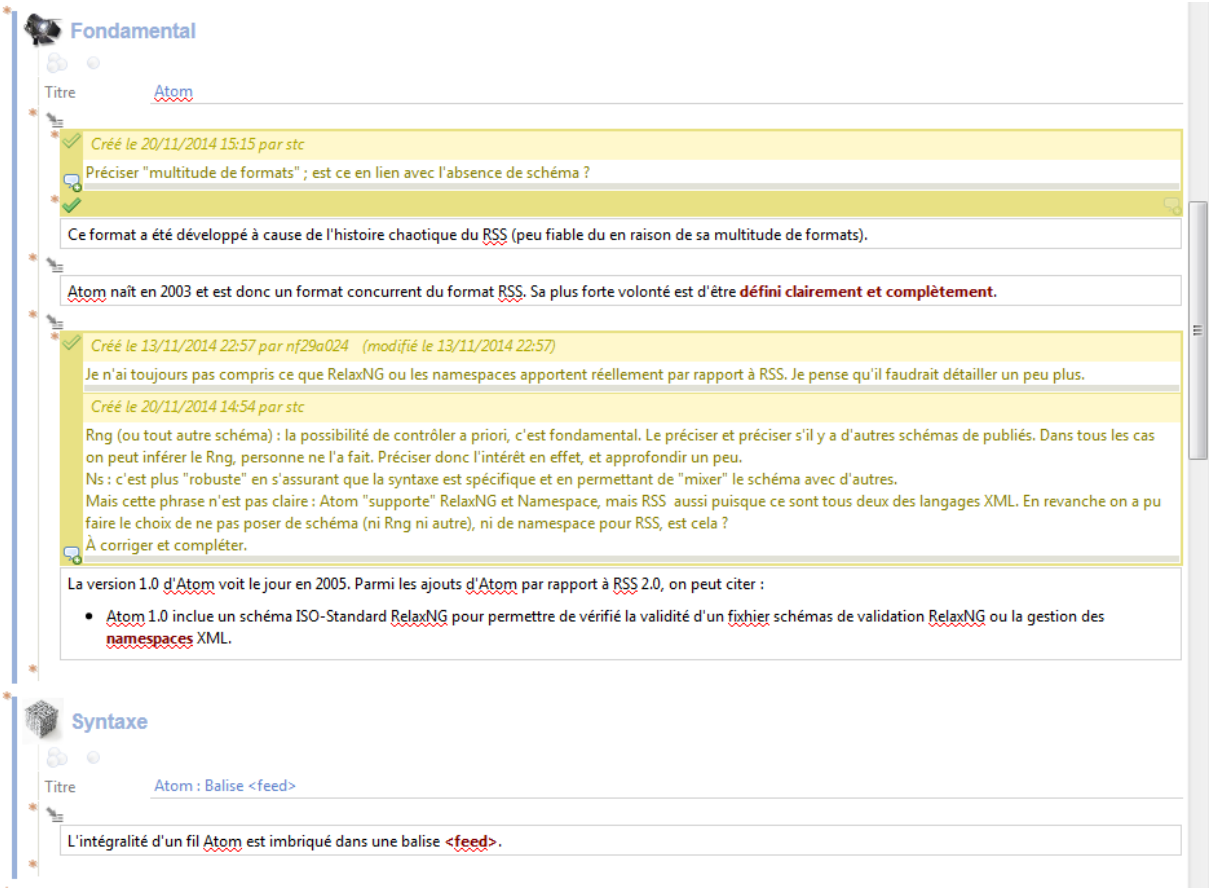


Figure 6 - Commentaires dans l'éditeur.

Dans la suite de cette section, nous présenterons les modèles documentaires Opale et Topaze, qui

seront mobilisés comme exemples tout au long de ce mémoire.

1.1.3 Opale

Opale (Gonzales-Aguilar *et al.*, 2012) est un modèle dédié à la production de documents académiques. Ces documents, appelés *modules*, suivent une structure linéaire à travers un plan hiérarchique. Ils peuvent être publiés aux formats HTML pour le web (classique ou diaporama) et ODT ou PDF (avec l'extension OpaleGenPdf) pour le support papier.

Grains de contenu

Les grains sont composés de *balises pédagogiques* (information, définition, exemple, remarque, conseil...) permettant à l'auteur de préciser l'intentionnalité associée au contenu.

Figure 7 - Grain de contenu avec balises pédagogiques.

Exercices

Les *exercices auto-évalués* (QCU, QCM, textes à trous...) sont des questionnaires interactifs auxquels le lecteur peut répondre via l'IHM de la publication web (cases à cocher, champs de saisie...). L'auteur doit renseigner les solutions et explications à afficher lors de la correction (dans le cas d'un QCU ou QCM, on peut avoir une explication par réponse en plus d'une explication globale).

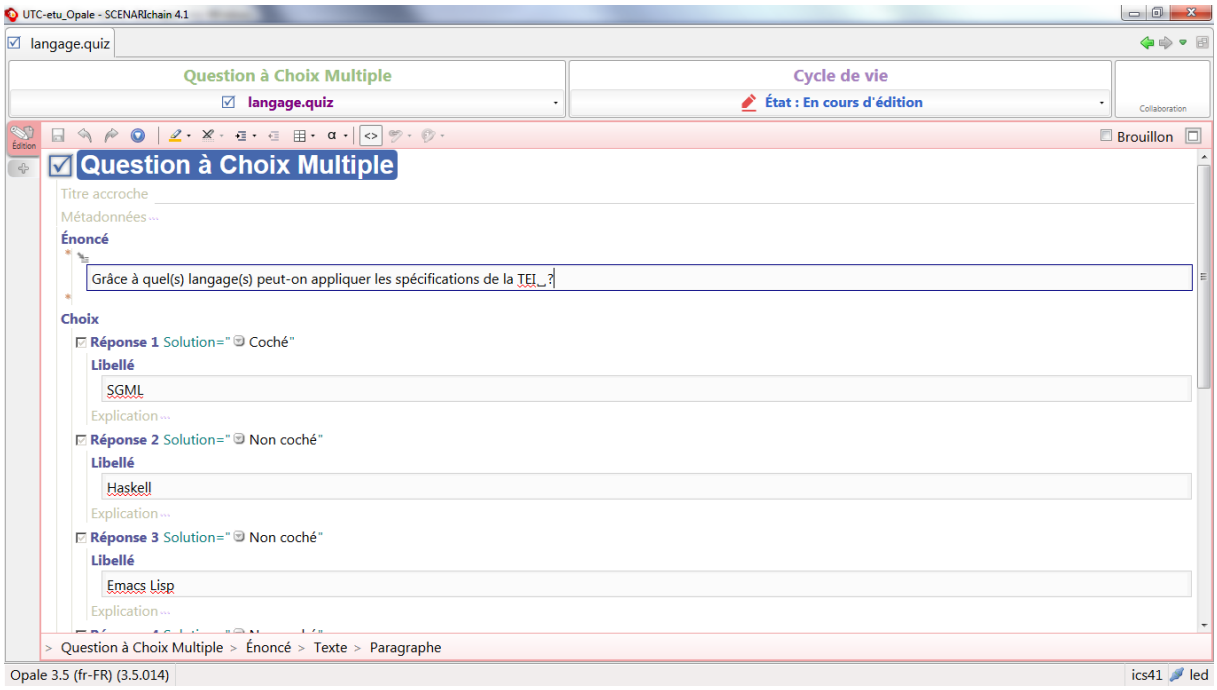


Figure 8 - Édition WYSIWYM d'un QCM.

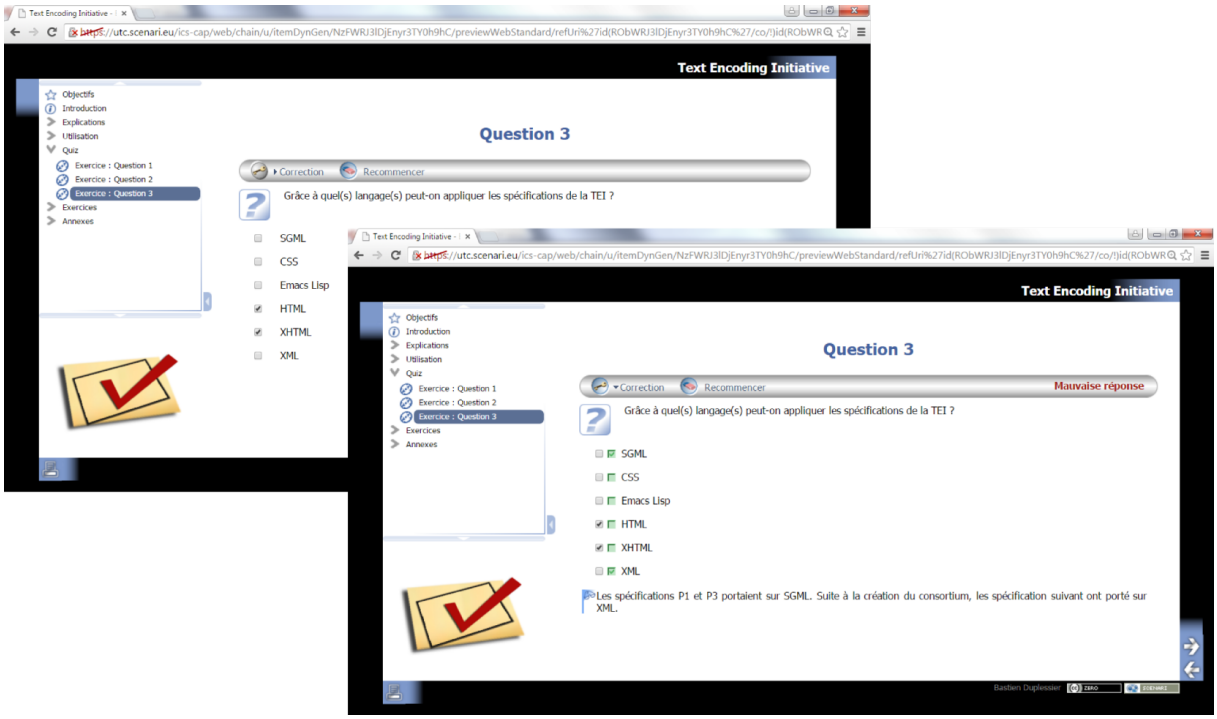


Figure 9 - QCM interactif dans la publication web du module.

Les *exercices rédactionnels* sont quant à eux des exercices "classiques" (TD, examen sur papier...). Pour chaque question, l'auteur peut préciser des solutions et des indices, qui seront masqués par défaut dans la publication web (pour les solutions, un paramètre de publication permet également de ne pas les y inclure).

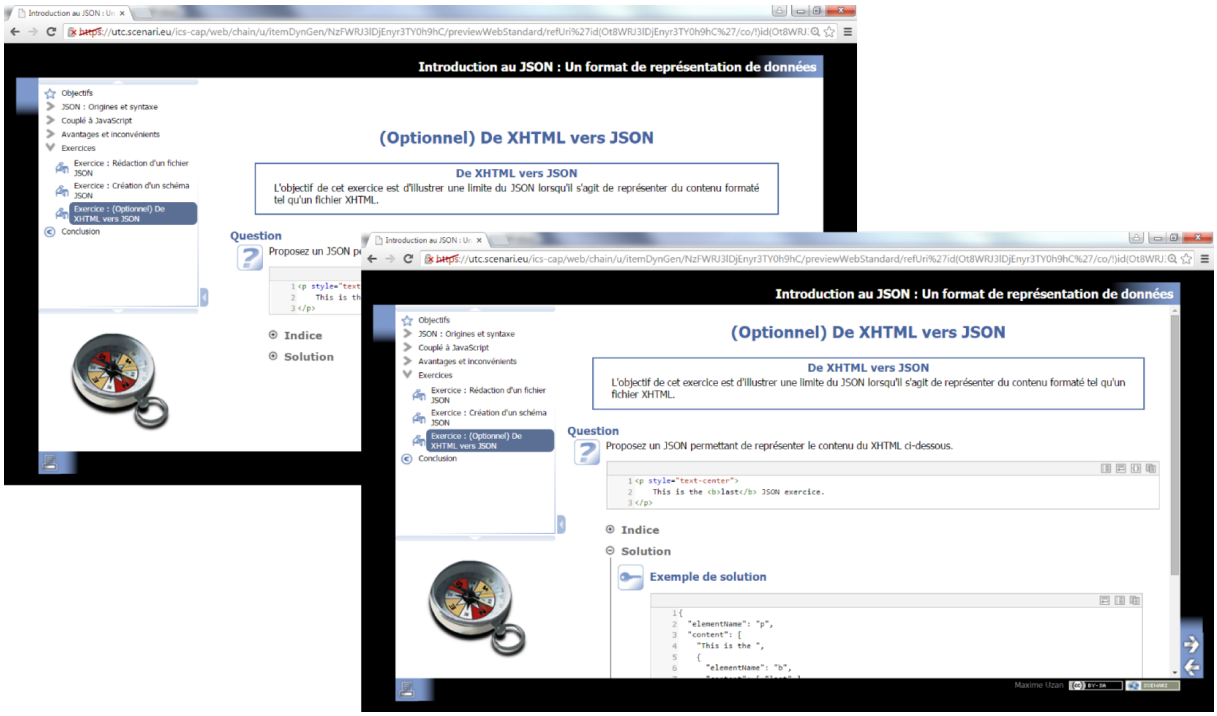


Figure 10 - Blocs dépliables (indices, solutions...) dans la publication du module.

Références

Dans du texte riche, il est possible de faire référence à des fragments tels que des grains (renvoi), des abréviations ou encore des entrées de bibliographie ou de glossaire. Dans la publication web, ces références seront matérialisées par des *incises* (pour les abréviations et les entrées de bibliographie et de glossaire) ou en sur-fenêtre (pour les renvois vers un grain, y compris quand le grain référencé appartient déjà au plan du module).

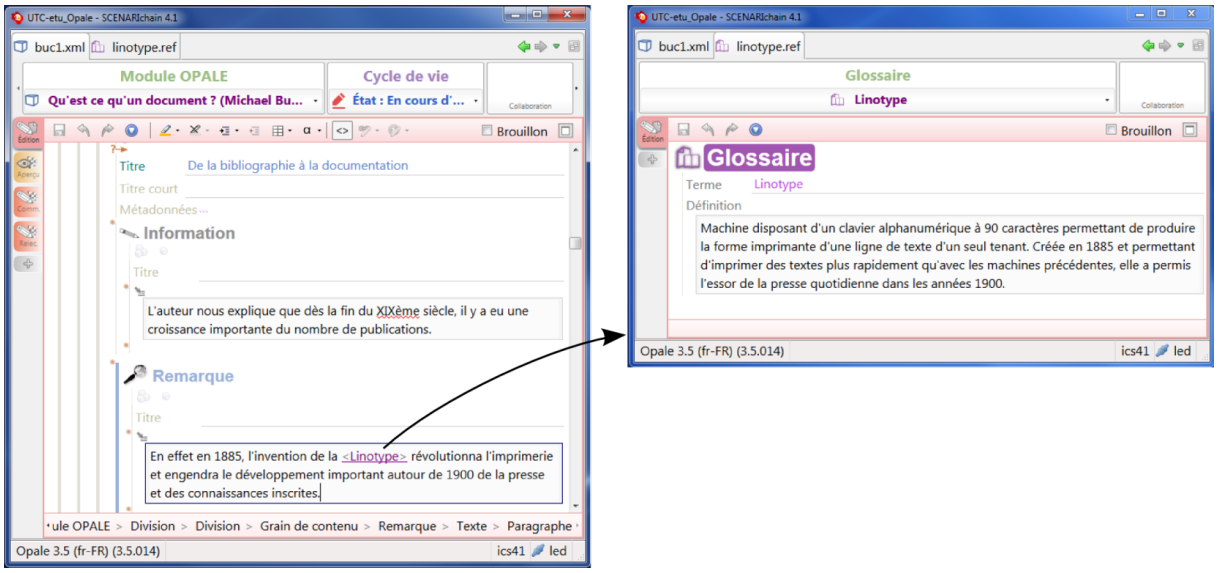


Figure 11 - Référence à une entrée de glossaire dans l'éditeur WYSIWYM.

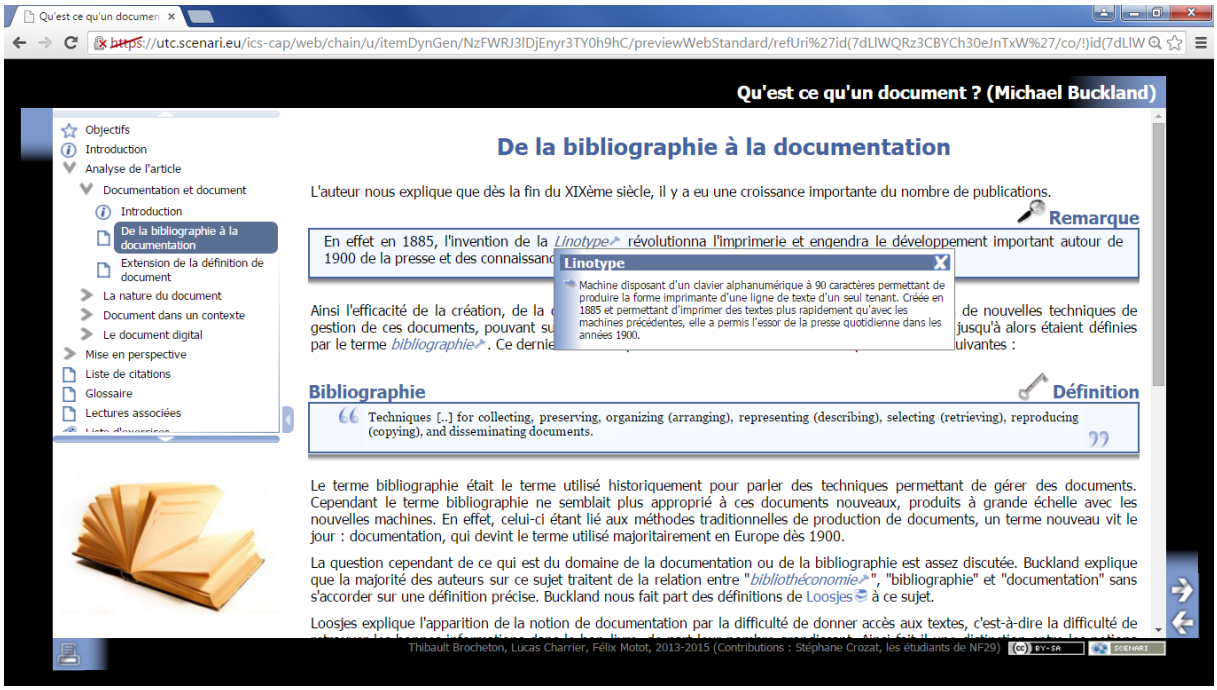


Figure 12 - Publication de l'entrée de glossaire en surimpression (incise).

Figure 13 - Publication d'un grain lié en sur-fenêtre.

Organisation

Le plan du module est constitué à partir des éléments suivants :

- des activités d'apprentissage, pouvant contenir des grains et des exercices ;
- des divisions, pouvant contenir des grains, des exercices, des activités d'apprentissage, ou encore d'autres divisions (récursivité).

Les grains et les exercices sont ainsi les niveaux les plus bas dans le plan ("feuilles").

Figure 14 - Publication web d'un module Opale.

Filtrage et sélection de contenu

Il est possible de filtrer des contenus à tous les niveaux du module (divisions, grains, balises pédagogiques...).

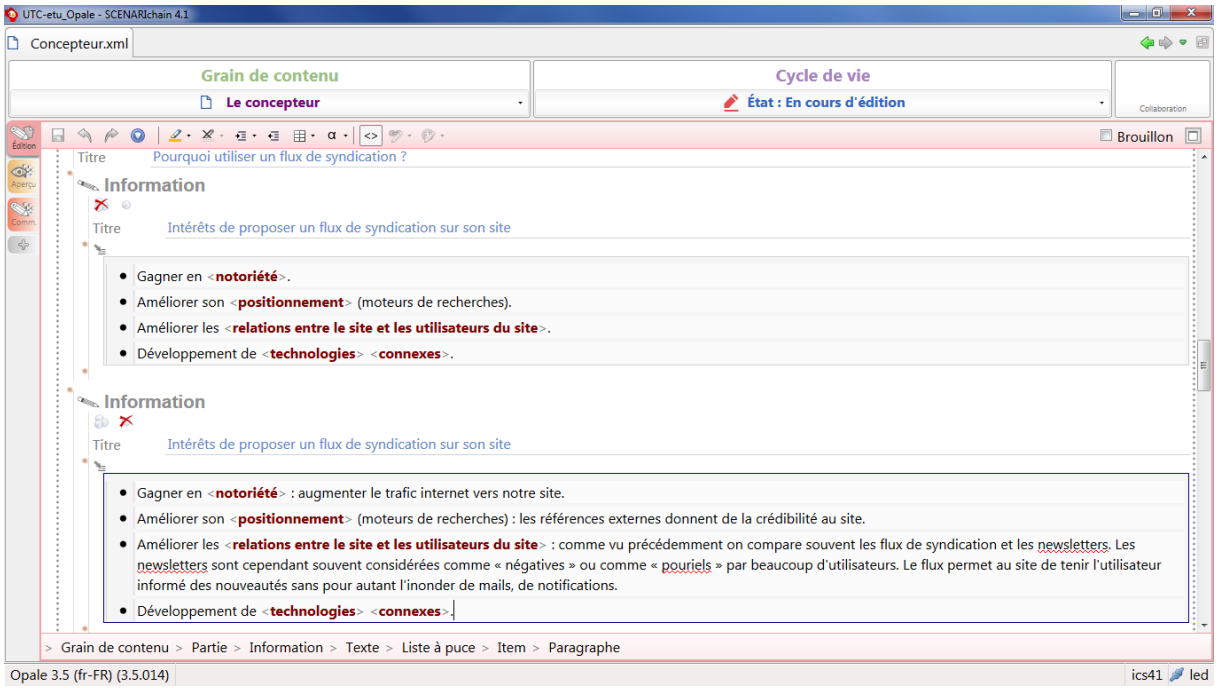


Figure 15 - Utilisation des filtres de version standard/courte dans l'éditeur WYSIWYM.

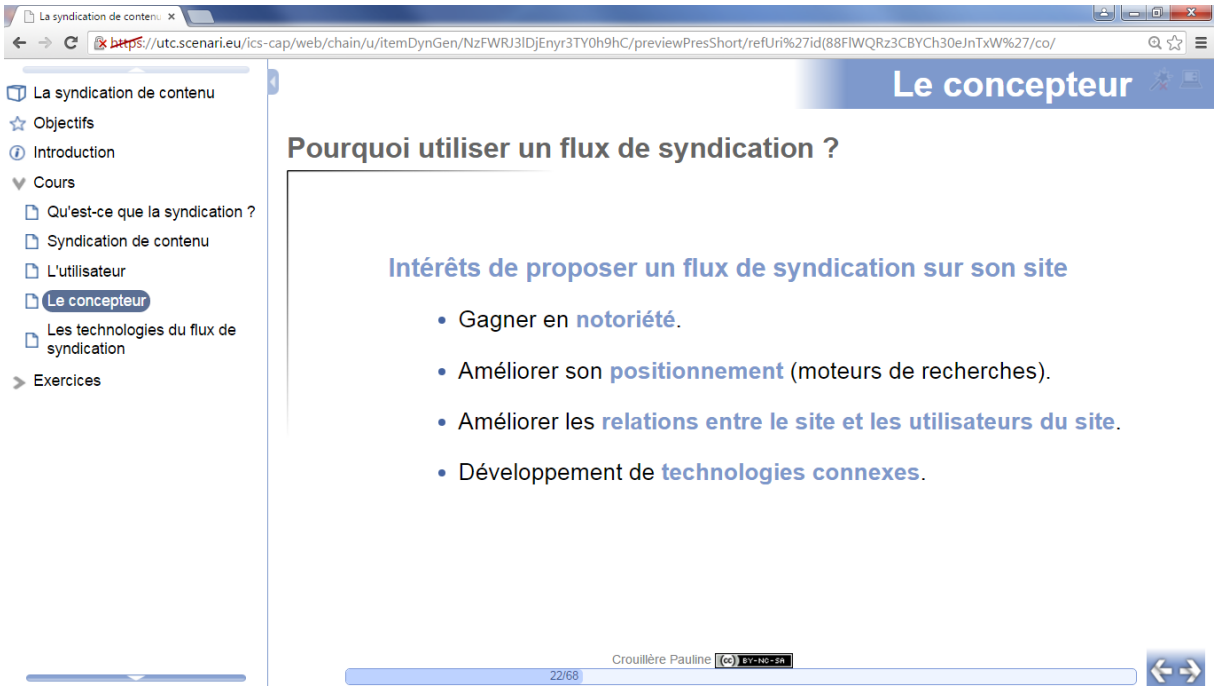


Figure 16 - Sélection des contenus fléchés pour la version courte dans la publication diaporama.

1.1.4 Topaze

Le modèle Topaze est un enrichissement d'Opale permettant de créer des documents multilinéaires au format web. Un document Topaze repose sur un graphe dont les nœuds sont des étapes (de quatre types possibles) et les liens sont des enchaînements entre étapes.

Étapes

Il existe quatre types d'étape :

- les étapes de contenu, composées de balises pédagogiques tel que dans un grain Opale (il est aussi possible de référencer un grain existant dans ce type d'étape) ;
- les étapes de quiz, qui consistent en un ensemble de questions (exercices auto-évalués d'Opale ou spécifiques à Topaze : QCU à points, QCM à addition de points ou à plans de réponse...) ;
- les étapes contenant un module Opale ;
- les étapes d'orientation (carte avec des zones cliquables pour choisir l'étape suivante).

Enchaînements

On distingue par ailleurs plusieurs types d'enchaînement :

- simple : une seule étape suivante sera proposée (lien linéaire) ;
- libre : plusieurs étapes suivantes suivantes seront proposées (liens-bifurcations), chacune pouvant être accompagnée d'un texte de transition ;
- conditionné : l'étape suivante sera proposée en fonction d'une condition, pouvant porter sur la visite préalable (ou non) d'une ou plusieurs étapes et/ou de valeurs calculées tout au long du parcours du lecteur, par exemple à partir des scores obtenus aux étapes de quiz ;

Certains enchaînements peuvent être déterminés dynamiquement en fonctions des réponses du lecteur. À la suite d'une étape de quiz par exemple, l'enchaînement peut être défini en fonction du pourcentage de réussite aux questions (score égal, inférieur ou supérieur à un tel pourcentage) : il peut donc y avoir autant d'enchaînements possibles que de "paliers" de score (25%, 50%, 75%, 100%...).

Une étape de fin comporte un enchaînement spécial ne liant aucune étape suivante (il peut y avoir

plusieurs fins possibles en fonction du parcours suivi).

Exemple : Visite en Questions

La société Exosens (prestataire Scenari), en partenariat avec l'UTC, a réalisé un dispositif de navigation multilinéaire pour le Musée des Beaux-Arts d'Angoulême à l'aide du modèle Topaze. Ce dispositif, appelé "Visite en Questions", vise à être utilisé sur support tactile (tablette ou smartphone) par les visiteurs au cours de leur parcours physique au sein du musée.

Les étapes du dispositif sont relatives aux œuvres du musée. Chaque œuvre est associée à deux étapes : une première présentant une image de l'œuvre et sa localisation (étape de contenu), et une seconde proposant un questionnaire sur l'œuvre (étape de quiz). Il existe plusieurs modes de navigation dans le dispositif, dont celui de la "visite guidée" : après chaque œuvre, le visiteur se voit proposer différentes œuvres pour continuer son parcours à travers le "réseau des œuvres" (on parlera de parcours réticulaires) :

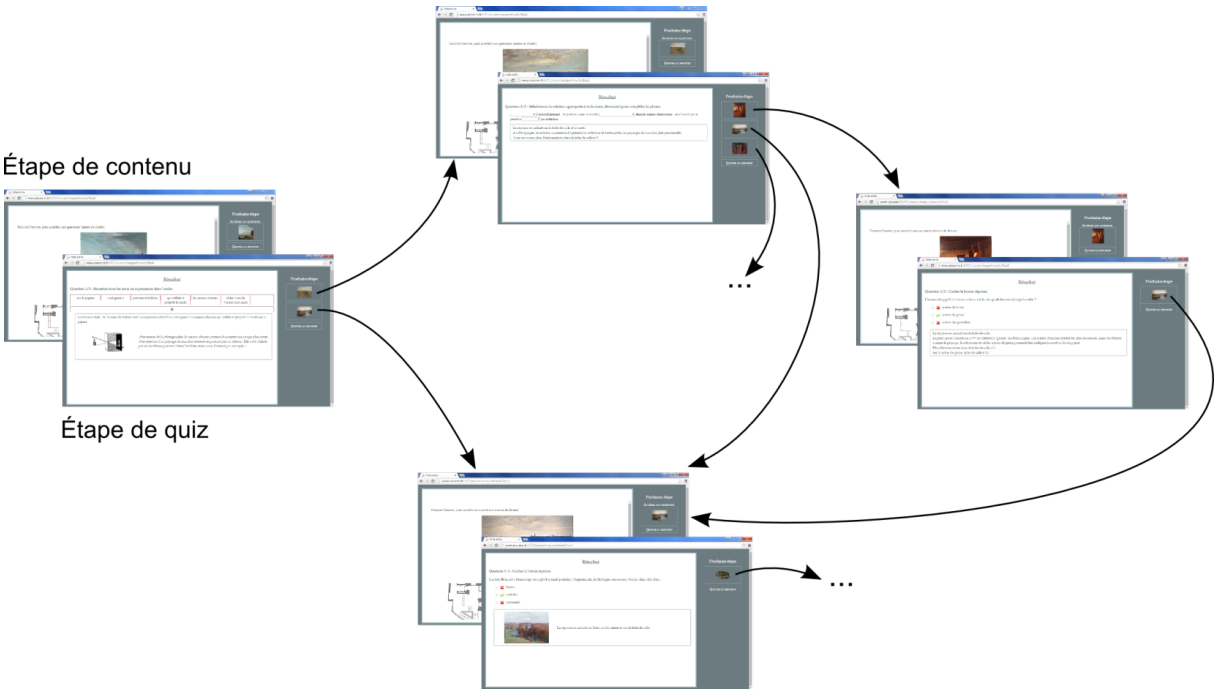


Figure 17 - Schéma des parcours réticulaires de "Visite en Questions".

Détails

- Musée d'Angoulême *(http://musee-angouleme.fr/)*
- Maquette du dispositif *(http://www.exosens.fr/UTC/musee/maquetteweb/final/)*
- http://www.exosens.fr/UTC/musee/recap/co/TopazeMusee.html
- Vidéo de présentation aux Rencontres Scenari 2015 *(https://www.youtube.com/watch?v=kaMe-31B8D0&feature=youtu.be&t=2844)*

Exemple : Faq2Sciences

Faq2Sciences est un site grand public mis en ligne en 2015 suite à un projet piloté par Unisciel (Université des Sciences en ligne). Il s'agit d'un ensemble de tests de positionnement réalisés avec Topaze, à destination des étudiants s'inscrivant dans une licence scientifique et souhaitant évaluer leur niveau de connaissances.

Les tests de type "focus" proposent une première étape de repérage, comportant plusieurs séries de cinq questions relatives à une thématique (par exemple : cinq questions de physique, puis cinq de chimie et cinq de mathématiques). À l'issue de cette étape, l'étudiant peut choisir d'approfondir une

thématique dans laquelle il n'a pas obtenu un score assez élevé : une étape comportant dix questions sur cette thématique lui sera alors proposée. Au niveau de Topaze, cela se traduit par un enchaînement conditionné :

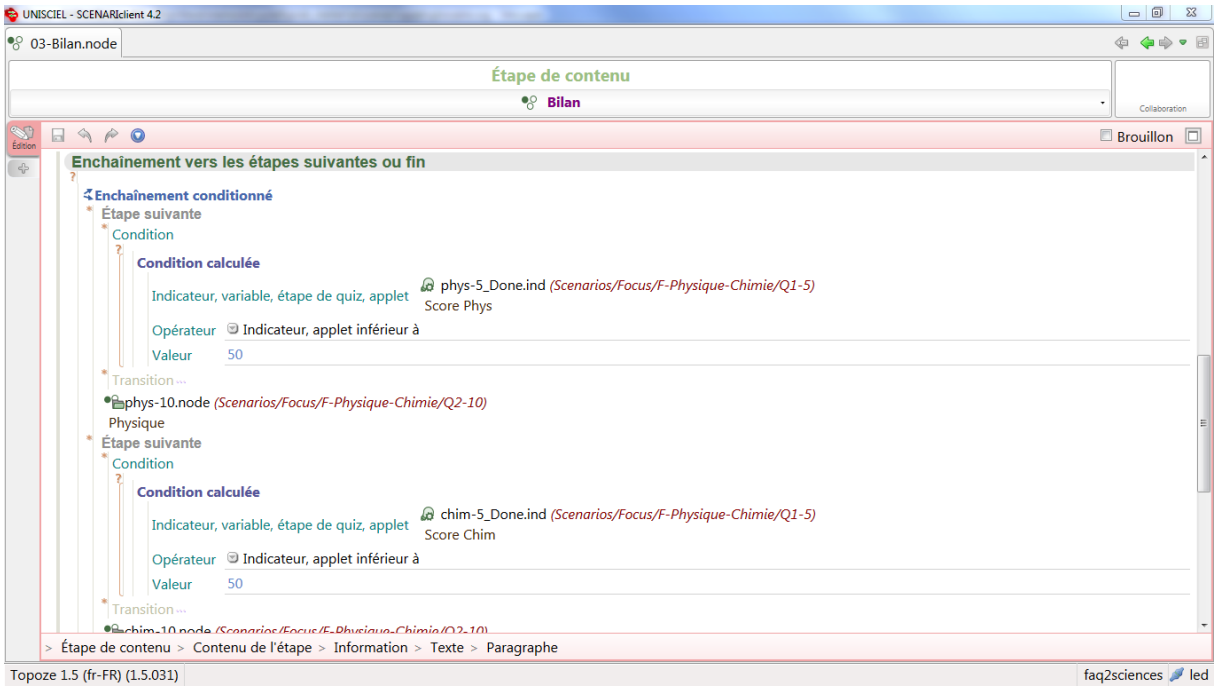


Figure 18 - Enchaînement conditionné.

Détails

- Site Faq2Sciences *(https://www.faq2sciences.fr/)*
- Unisciel *(http://www.unisciel.fr/)*
- Vidéo de présentation aux Rencontres Scenari 2015 *(https://www.youtube.com/watch?v=Aa5XhuF6z2Y)*

1.2 Relecture

La relecture se distingue des autres types de lecture par les objectifs qu'elle poursuit, ce qui d'après O'Hara (1996) induit des rapports particuliers au texte en termes de manipulation et de navigation. Par exemple, les allers-retours dans le texte pour vérifier la cohérence de plusieurs passages (sur le fond ou la forme) sont plus caractéristiques de la relecture que de la lecture de loisir, où le texte tend à être lu dans une seule et même continuité. La relecture amène également à annoter et à corriger le texte, ce que les technologies numériques permettent d'instrumenter.

En mettant en jeu une *lecture critique*, la relecture entraîne un effort cognitif plus important que pour la *lecture de compréhension* d'après Roussey et Piolat : « dans la lecture de révision, on lit non seulement afin de se représenter le sens du texte, mais aussi et surtout dans le but de trouver des problèmes susceptibles d'impacter les buts rhétoriques de l'auteur, ou bien la représentation du contenu textuel que se fera le lecteur » (2008, notre traduction).

1.2.1 En révision professionnelle

La relecture s'inscrit dans le domaine de la révision professionnelle. D'après les principes directeurs établis par l'ACR (Association Canadienne des Réviseurs), « la révision [...] a pour but d'assurer la

qualité de la langue et l'efficacité de la communication » (Arsenault *et al.*, 2014). Les auteurs de ces principes font état de plusieurs activités principales présentées ci-dessous.

Révision de fond

« La révision de fond suppose une lecture attentive et méthodique d'un texte en vue de l'adapter aux destinataires, d'en clarifier le contenu et d'en réorganiser la structure. »

Cette relecture est généralement réalisée par un expert du contenu (un juriste par exemple). Elle cherche à garantir le sens visé par le texte et l'exactitude des informations. Un ensemble de modifications (ajouts, suppressions, remaniement du plan...) est suggéré par le relecteur via des annotations, des commentaires envoyés par mail, etc.. Si le document est éditable et selon la répartition des rôles entre le rédacteur et le relecteur, les modifications peuvent aussi être directement effectuées lors de la révision de fond.

Révision de forme

« La révision de forme vise l'amélioration du style du texte dans son ensemble grâce à des corrections de syntaxe, de vocabulaire, d'orthographe et de ponctuation. »

Cette relecture touche un métier bien précis, celui de correcteur. Dans certains cas, la révision de forme peut ne pas constituer une étape à part entière et être effectuée en même temps que la révision de fond. La poursuite des deux objectifs lors d'une même lecture est souvent difficile : l'immersion dans le texte pour en saisir le sens peut rendre le relecteur moins attentif aux fautes. En particulier, il apparaît que la lecture répétée d'un texte, par son propre auteur typiquement, amène à balayer de plus en plus vite le texte, ce qui est souvent la cause de fautes non corrigées. Outre l'usage de correcteurs automatiques, une technique couramment employée par les correcteurs professionnels pour détecter les fautes est de lire le texte à l'envers afin de se détacher du sens (voir par exemple le blog monbestseller.com *(http://www.monbestseller.com/actualites-litteraire-conseil/2325-comment-relire-son-livre-et-corriger-les-fautes-dorthographe)*).

Préparation de copie

« La préparation de copie consiste à mettre au point un texte déjà révisé en vu de sa mise en pages. Il s'agit notamment d'appliquer de façon uniforme dans tout le document les règles et les conventions en usage, et d'informer le ou la graphiste de toute exigence particulière touchant la production. »

Il s'agit d'un ensemble de vérifications en partie redondantes avec la révision de forme, telles que l'uniformité de la typographie : abréviations, symboles, écriture des nombres en lettres ou en chiffres, écriture des dates, ponctuation, etc.. Cette étape est héritée du monde de l'imprimé, où les tâches relevant de la mise en forme finale (graphisme, façonnage...) sont réalisées par l'éditeur puis l'imprimeur. Elle persiste dans le numérique lorsque chaque document ou presque fait l'objet d'une mise en forme dédiée, tel que dans des secteurs comme la communication (plaquette commerciale réalisée avec InDesign par exemple).

Correction d'épreuves

« La correction d'épreuves comprend toute vérification qui suit l'étape de mise en pages ou d'intégration Web. Qu'il s'agisse de la première épreuve ou d'épreuves subséquentes, il faut examiner notamment la typographie, l'orthographe, la mise en forme du texte et tous les aspects de la présentation visuelle. »

Dans le monde de l'imprimé, cette étape constitue la relecture du document final (révisé puis mis en forme). Les vérifications, encore une fois redondantes avec la révision de forme, visent à s'assurer qu'aucune nouvelle erreur n'a été introduite lors de la mise en forme.

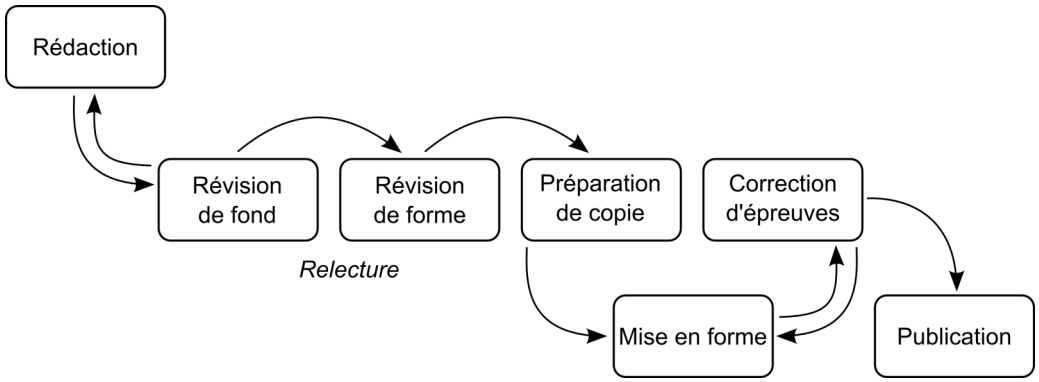


Figure 19 - Processus de relecture en révision professionnelle.

1.2.2 Dans les chaînes éditoriales

Les spécificités des chaînes éditoriales impliquent de réinterroger le processus de relecture tel que nous l'avons vu en révision professionnelle. Tout d'abord, l'automatisation de la mise en forme rend caduques la préparation de copie et la correction d'épreuves. En effet, la mise en forme est pensée en amont de la production documentaire, et de façon générique, lors de la modélisation de la chaîne éditoriale. À ce stade, la validation de la mise en forme se fait à partir de contenus de test ou d'un faux-texte ("*Lorem ipsum...*"), afin d'évaluer sa conformité par rapport à la maquette définie.

> De plus, l'abandon du WYSIWYG fait émerger de nouvelles questions : que doit-on relire entre d'un côté les contenus fragmentés dans SCENARIchain, et de l'autre les contenus mis en forme ? Faut-il relire chacune des mises en forme possibles du contenu ? Par ailleurs, la relecture d'un document ayant un fragment F réutilisé plusieurs fois entraîne-t-elle la relecture de F dans chacun de ses contextes (les fragments liant F), voire des documents partageant ce fragment dans leur ensemble ?

Nous allons étudier le cas concret de l'IFCAM (Université du groupe Crédit Agricole) pour voir comment ces questions se posent en pratique.

L'IFCAM utilise un modèle documentaire similaire à Opale pour maintenir une base de 296 modules publiés aux formats web (e-learning) et papier. Ils sont mobilisés dans le cadre d'une formation diplômante (le CETCA) concernant quotidiennement plus de 3000 collaborateurs du Crédit Agricole, à raison de 15-20 modules par unité d'enseignement (chiffres de 2013).

Une à deux fois par an sont organisées des campagnes de mise à jour de ces modules, afin que les contenus soient conformes au cadre législatif (nouvelles lois de finance, etc.) et réglementaire. Ces mises à jour impliquent un réseau d'une cinquantaine d'experts métier, et sont réalisées en plusieurs phases :

- les experts métier relisent les modules à travers une prévisualisation web, et utilisent le système de commentaires pour proposer des modifications (actualisation de chiffres, reformulations, ajout d'exercices, de ressources...) ;
- une cellule de production (moins d'une dizaine de personnes) intègre les modifications dans SCENARIchain ;
- les modifications sont relues par les experts lors d'une étape de validation.

Supposons maintenant que le choix se soit porté sur une relecture dans SCENARIchain. L'avantage de cette solution est de pouvoir prendre en compte des informations qui ne sont pas communes à toutes les mises en forme : par exemple, des contenus spécifiques au format papier, telles que des alternatives statiques aux contenus multimédias ; ou bien relevant exclusivement de la dimension auctoriale :

typiquement, la réutilisation d'un contenu dans plusieurs documents, qui peut être une information utile au relecteur, n'est pas visible dans la mise en forme de ce contenu. En revanche, l'interface WYSIWYM et la fragmentation empêchent de relire le document dans sa matérialité propre : il ne sera pas possible, par exemple, de tester l'interactivité du site web, ou bien de vérifier la mise en page de la publication papier (absence de veuves et d'orphelins, etc.).

Au-delà de ces questions soulevées au niveau des chaînes éditoriales, il nous semble que c'est le document numérique dans son ensemble qui est réinterrogé par la problématique de relecture.

Chapitre 2

Problématique

2.1 Le document numérique : entre instabilité, interactivité et rééditorialisation 28
2.1.1 Instabilité 28
2.1.2 Interactivité 30
2.1.3 Rééditorialisation 32
2.2 Les chaînes éditoriales : une technologie de rééditorialisation documentaire 34
2.2.1 Polymorphisme et formes documentaires 34
2.2.2 Transclusion et graphe documentaire 35
2.2.3 Dérivation et déclinaison 36
2.3 Vers la conception de formes de relecture 37
2.4 Cadre théorique 38
2.4.1 Du document au document numérique 38
2.4.2 La raison graphique 39
2.4.3 La tendance technique du numérique 40
2.4.4 La raison computationnelle 40
2.4.5 Les tropismes du numérique 41

L'enjeu de ce chapitre est de montrer en quoi le numérique complexifie la relecture tout en donnant les moyens de produire des formes documentaires dédiées à cette activité. Nous caractériserons la complexité via trois propriétés intrinsèques du document numérique : l'instabilité, l'interactivité et la rééditorialisation. Nous verrons ensuite que ces trois propriétés se retrouvent au niveau des chaînes éditoriales, motivant ainsi le fait de les utiliser comme cadre pour nos travaux sur la relecture.

2.1 Le document numérique : entre instabilité, interactivité et rééditorialisation

2.1.1 Instabilité

D'après Stiegler (1995), le numérique marque le passage d'un support statique (« parchemin, manuscrit ou papier imprimé ») à un support dynamique, entraînant ainsi un rapprochement "naturel" de la lecture et de l'écriture. Stiegler prend pour exemple les outils numériques d'annotation mobilisés par le lecteur "savant", qui permettent d'automatiser et de mémoriser de façon infaillible ses opérations de *hiérarchisation* (souligner, commenter dans la marge...) et de *qualification* (indexer par des mots-clés, relier à d'autres documents...). À l'inverse, l'annotation sur un support statique par des lecteurs constitue moins un acte d'écriture car elle est limitée par la « rupture du bon à tirer », qui empêche par exemple de rééditer un ouvrage en y intégrant ces annotations, comme une édition numérique le permettrait. À travers cette *lecture-écriture*, nous voyons donc qu'un contenu numérique admet un nombre illimité de réécritures, aussi bien *autour* du texte (annotation) que *dans* le texte (modifications, ajout de compléments...).

Par ailleurs pour le collectif Pédauque (2006), le numérique amène à désigner par "documents" des productions qui sont en réalité des "proto-documents", par exemple les « différentes versions successives d'un document de travail » (*ibid.*). Dans le cadre des activités de conception coopérative, Zacklad (2005) parle plus précisément de *documents pour l'action*, qu'il caractérise à travers leur inachèvement prolongé.

Le numérique favorise ainsi une écriture fortement itérative en comparaison des pratiques de l'édition traditionnelle. En effet, un ouvrage papier n'est publié que lorsqu'il est définitivement validé par l'éditeur. S'il peut néanmoins faire l'objet de rééditions, celles-ci sont généralement en nombre limité et distantes de plusieurs années. Par ailleurs, quand des fautes ont été détectées après tirage, il est d'usage de consigner les corrections dans un *errata* adjoint à l'ouvrage, plutôt que de le publier à nouveau.

Les technologies Wiki permettent d'illustrer cette écriture itérative et l'instabilité documentaire qu'elle provoque. En effet, les utilisateurs de ces systèmes peuvent apporter des modifications aux documents à travers l'interface web, et gérer ces modifications grâce à l'historisation de chaque version. Le numérique change donc notre façon d'écrire : « Le processus de rédaction est collaboratif [...] et progressif [...] : des notes brutes peuvent être publiées par certains, pour être ensuite améliorées, voire finalisées par d'autres. La tension principale que doivent gérer ces dispositifs est l'opposition manifeste entre documentation de référence d'une part et contenu en perpétuel mouvement d'autre part (modification, validation, ...). » (Crozat *et al.*, 2011).

Des statistiques sur le projet encyclopédique Wikipédia illustrent cette tendance : d'après des études menées de 2001 à 2006 sur la version anglaise (Wilkinson et Huberman, 2007), environ un millier d'articles de moins de deux ans ont dépassé le millier de modifications. Les versions historisées des articles Wikipédia ont d'ailleurs fait l'objet de recherches sur des nouvelles techniques de visualisation de l'information, telles que l'*history flow* (Viégas *et al.*, 2004), qui permet de représenter la dynamique temporelle des modifications d'un article Wikipédia par ses différents auteurs. Dans l'illustration ci-dessous, chaque version est représentée par un rectangle, dont la longueur et la largeur représentent respectivement la taille du contenu et le temps avant qu'une nouvelle version soit créée, et dont les couleurs permettent de distinguer les auteurs ayant modifié telle ou telle partie du contenu.

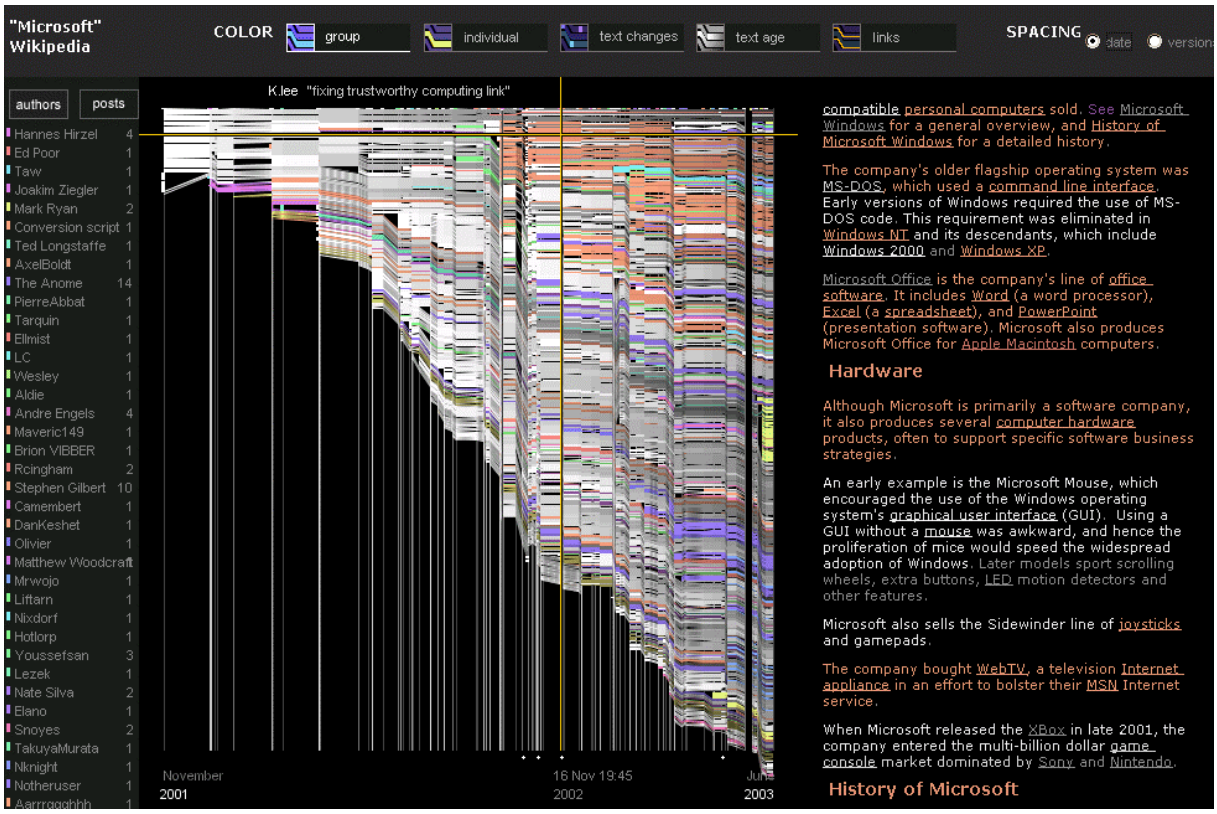


Figure 20 - Visualisation history flow (Viégas et al., 2004).

Prenons l'exemple de l'article Wikipédia (version française) traitant de Philae, l'atterrisseur de l'Agence spatiale européenne qui s'est posé sur la comète "Tchouri" en novembre 2014. Entre le 14 et le 16 juin 2015, l'article a été modifié plus d'une vingtaine de fois suite à une reprise de la communication entre Philae et la sonde spatiale Rosetta :

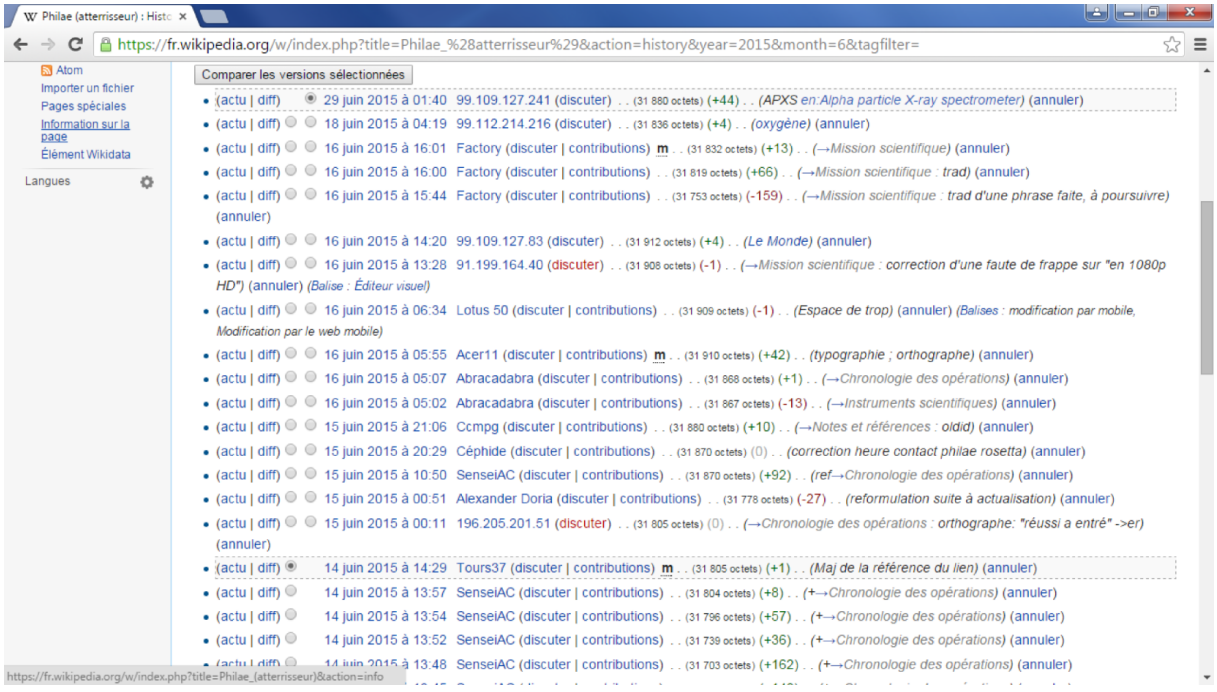


Figure 21 - Historique des modifications de l'article Wikipédia "Philae".

L'outil de comparaison (*différentiel*) est utile pour identifier précisément les changements qui ont

eu lieu entre deux versions (pas forcément consécutives) :

Figure 22 - Différentiel de deux versions de l'article Wikipédia "Philae".

> Le différentiel rend la relecture plus efficace, mais ne permet pas à lui seul de valider entièrement la nouvelle version d'un document. En effet, les modifications apportées peuvent avoir généré des incohérences avec le reste du document, que seule une relecture exhaustive permet de détecter. Cependant, du fait de l'écriture itérative, il devient impossible en pratique de relire entièrement le document à chacune de ses versions.

2.1.2 Interactivité

Le numérique propose une écriture interactive, qui « suppose une forme de programmation informatique, plus ou moins ouverte, des interventions matérielles du lecteur (qui devient ainsi un *interacteur*), ces interventions entraînant des *réponses* de l'ordinateur. » (Bouchardon, 2010, p. 134). L'interactivité est notamment caractérisée par les liens hypertextes, qui donnent au lecteur la possibilité de construire son parcours au sein du document qu'il consulte. D'après Crozat (2015a, 2015b), ces liens peuvent être de plusieurs sortes (augmentation, parenthèse, navigation, hyper-navigation) et permettent de scénariser différents types de parcours (linéaire, arborescent, multilinéaire...). L'interactivité donne également au lecteur la possibilité d'introduire des données qui peuvent ensuite faire l'objet d'un calcul réalisé par un programme (questionnaires interactifs, parcours conditionnés...).

Augmentation

L'augmentation est caractérisée par un contenu venant s'ajouter au contenu de départ, en extension ou en surimpression de ce dernier. Dans le premier cas, on parlera de contenu à profondeur variable (p. 17) ou *stretchtext* (Nelson, 1987) se manifestant par un contenu qui se "déplie". Dans le second, on parlera d'incise (p. 18).

Parenthèse

La parenthèse désigne le remplacement temporaire d'un contenu par un autre, s'affichant par exemple dans une sur-fenêtre masquant le contenu d'origine (p. 19). Il s'agit d'une simple digression, à la suite de laquelle le lecteur ne peut que revenir au contexte initial (en fermant la sur-fenêtre ou à

travers un lien "retour" typiquement).

(Hyper-)navigation

La navigation décrit un déplacement transversal à l'intérieur du document ou bien la sortie complète de ce document pour aller vers un autre : dans ce dernier cas, on parle d'hyper-navigation. Dans le site service-public.fr *(https://www.service-public.fr/)* par exemple, on trouve des liens transversaux allant d'une fiche à une autre (navigation), ou encore des liens pointant vers d'autres sites tels que le portail Legifrance *(http://www.legifrance.gouv.fr/)* pour le référencement d'articles de loi (hyper-navigation).

Parcours

Les parcours peuvent être plus ou moins contraints en fonction de la structure du document. Dans un document linéaire (p. 19), le parcours est suggéré par des liens précédant/suivant, ainsi qu'un plan hiérarchique. Dans un site tel que service-public.fr, le lecteur créé son parcours à travers les différentes fiches classées dans une arborescence (classement par thèmes, sous-thèmes...). Dans un document multilinéaire, le lecteur a le choix sur chaque page entre plusieurs pages suivantes possibles : il existe différents parcours qui s'entrecroisent, et le document repose sur une structure de réseau (p. 22).

Calcul

La dimension computationnelle du document numérique permet de programmer des comportements en fonction des actions effectuées par le lecteur. Dans les questionnaires interactifs (p. 17) par exemple, la correction (bonnes et mauvaises réponses, explications, score...) est individualisée en fonction des réponses. Les enchaînements conditionnés (p. 23) de Topaze permettent quant à eux de proposer des étapes suivantes en fonction de conditions plus complexes (par exemple, les scores obtenus aux étapes de quiz précédentes).

L'interactivité, entre prise en charge et contrôle

Dans le cadre de ses travaux portant sur le récit interactif, Bouchardon (2010) analyse le rapport entre le lecteur et le dispositif : « Avec le numérique, [...] le lecteur délègue à l'outil les fonctions d'accès : l'accès au contenu est entièrement codé, il est dans l'outil. Si l'accès n'a pas été prévu entre deux fragments par le concepteur des liens, ce lien n'existe pas. [...] Dans le numérique, tout cela est *caché* au lecteur. » (*ibid.*, pp. 157-158). Bouchardon parle plus précisément de la « routinisation » d'actions qui, dans le support papier, requièrent une intervention physique du lecteur (tourner la page, sauter des pages pour consulter le glossaire, etc.). Une action routinisée est caractéristique de la *prise en charge* du lecteur par le dispositif. Comme nous l'avons vu avec les questionnaires interactifs et les enchaînements conditionnés, cette prise en charge peut être le fait d'un programme et donc d'un calcul. Mais l'interactivité peut également donner une impression de *contrôle*, par exemple lorsque plusieurs liens possibles sont proposés à un point du récit : « Le lecteur, devant choisir un lien parmi plusieurs, est contraint d'*éliminer* les autres : il aura tendance à légitimer son choix, et par là même sera amené à construire une légitimité du lien choisi et du parcours effectué. » (Bouchardon, 2010, pp. 158). Les liens d'augmentation, de parenthèse, de navigation ou d'hyper-navigation participent aussi de cette impression de contrôle selon nous, le lecteur ayant le choix de les activer ou non (par exemple, on peut choisir de ne pas "déplier" une rubrique d'informations complémentaires).

L'interactivité, à travers les impressions de prise en charge et de contrôle, ne permet pas de mener une relecture exhaustive, comprise comme la capacité à faire la *synthèse* des différentes lectures résultant des possibilités d'interaction. En effet, la prise en charge par un programme empêche d'appréhender toute la combinatoire des résultats possibles de ce programme en fonction des données résultant de l'interaction. Le contrôle entraîne quant à lui le risque d'oublier certains contenus ou parcours au fil de la relecture, lorsque les liens permettant d'y accéder ne sont pas activés.

2.1.3 Rééditorialisation

La rééditorialisation est définie par Crozat (2012a) comme étant « la publication d'une œuvre originale dans son point de vue, sa forme, sa scénarisation, à partir de contenus qui ne le sont pas tous ». Le processus de rééditorialisation peut être décrit en six étapes (*ibid.*) : (1) sélection, (2) déconstruction et (3) transformation (au sens d'une réécriture) de fragments de contenus existants ; (4) production de nouveaux fragments (compléments, glose...) et (5) scénarisation de l'ensemble des contenus ; (6) publication adaptée au nouveau contexte (avec une mise en forme et des métadonnées spécifiques).

Nous illustrons un cas de rééditorialisation à travers des exemples de revue de presse sur les sites des associations de la Quadrature du Net et de l'April. On y trouve par exemple un même article de l'Obs/Rue89 ("Loi numérique, dernier jour : le sursaut des lobbies", du 18/10/2015) rééditorialisé de deux manières différentes :

Figure 23 - Article original sur l'Obs/Rue89 (http://rue89.nouvelobs.com/2015/10/18/loi-numerique-dernier-jour-sursaut-lobbys-261722).

Figure 24 - Rééditorialisation dans la revue de presse de la Quadrature du Net (https://www.laquadrature.net/fr/rue89-loi-numerique-dernier-jour-sursaut-lobbys).

Figure 25 - Rééditorialisation dans la revue de presse de l'April (https://www.april.org/lobs-loi-numerique-dernier-jour-le-sursaut-des-lobbys).

Dans le premier cas, l'article a été résumé en trois extraits, complétés par une capture d'écran du site d'origine. Dans le second cas, il s'agit simplement d'une citation de la première phrase de l'article. Dans les deux cas, on remarque l'ajout de tags propres à chaque revue de presse.

La rééditorialisation de contenus sur le Web peut également être automatisée via par exemple la technologie de syndication (flux RSS et Atom), qui permet à un site (presse en ligne, blogs...) d'exposer son contenu sous la forme d'un fichier au format XML (le flux) auquel d'autres sites, appelés

agrégateurs, peuvent faire référence. Une transformation XSL ainsi qu'une feuille de style CSS peuvent être appliquées au flux afin de rendre son contenu lisible dans l'agrégateur. Plus récemment sont apparues des plateformes dites de "curation de contenu" (Scoop.it, Storify, Pearltrees, etc.), qui sont notamment utilisées comme des outils de veille documentaire. Ces plateformes permettent à leurs utilisateurs de créer des dossiers dans lesquels ils peuvent répertorier les pages web qu'ils jugent pertinentes, les commenter, etc..

La rééditorialisation a pour effet de démultiplier les variations du contenu à relire. En effet, pour un auteur voulant par exemple valider les différents contextes de rééditorialisation de son document, la tâche se heurte à la redondance des contenus invariants, par ailleurs difficiles à distinguer au milieu des autres contenus spécifiques à la rééditorialisation, et parfois même issus d'autres sources.

2.2 Les chaînes éditoriales : une technologie de rééditorialisation documentaire

2.2.1 Polymorphisme et formes documentaires

Le polymorphisme est défini par Crozat comme une « technique d'automatisation de la transformation du codage d'un contenu en un autre codage afin de remplir des objectifs éditoriaux différents » (2012a). Il repose sur l'articulation entre différentes formes documentaires (*ibid.*) :

- *Forme génératrice* (FG). Il s'agit des ressources XML, dont la structuration logique (André *et al.*, 1989) est contrôlée à l'aide d'un schéma XML (DTD, RelaxNG...). Crozat souligne que cette forme n'a en général pas de vocation documentaire : c'est en effet parce qu'elle est abstraite qu'elle se prête particulièrement bien aux transformations (*disponibilité manipulatoire*).
- *Forme(s) d'édition* (FE). La transformation de la FG consiste ici à rendre cette dernière modifiable via un éditeur XML. Le pluriel entre parenthèses laisse entendre qu'un même contenu puisse être édité via des FE spécialisées, par exemple une FE1 pour le contenu d'un module et une FE2 pour les métadonnées, les ressources, etc.. Un éditeur XML WYSIWYM permet de guider l'auteur à l'aide d'une interface de type "formulaire" en s'appuyant sur le schéma XML de la FG. Le paradigme WYSIWYM (Power *et al.*, 1998 ; Van Deemter et Power, 2000) permet ainsi un « compromis entre la posture essentiellement sémiotique du WYSIWYG et celle essentiellement logique de la programmation informatique » (Crozat, 2015b).
- *Formes publiées* (FP). La FG est finalement transformée (à l'aide de feuilles XSL) dans des formats de publication en vue de son interprétation par des lecteurs. Les différentes FP d'une même FG répondent à des logiques multi-supports et multi-usages. **En particulier, une FP au format web est un document interactif.**

Le polymorphisme est en tension avec la relecture. En effet, cette dernière suppose la capacité du document à être objectivé par le relecteur, ce qui selon Bachimont repose sur « le fait de pouvoir accéder à un exemplaire de référence, une version faisant autorité [...] » (2007, p. 172). Or l'objectivation est impactée par le polymorphisme, tant du point de vue du lecteur que de celui de l'auteur :

- Pour le lecteur : « L'individualisation du contenu a pour conséquence d'annuler le contenu comme objet pour n'en faire que le reflet de l'idiosyncrasie du lecteur : l'objectivation est annulée par l'adaptation du contenu. Au lieu de constituer un pôle d'identité et de référence auquel ajuster et confronter sa compréhension et appropriation, le contenu se dissout dans les multiples présentations à chaque fois différentes du contenu. Le lecteur ne peut plus se situer par

rapport au contenu présenté et faire la part entre le contenu et sa présentation, ni rapprocher les différentes présentations possibles à un même noyau de sens. » (Bachimont et Crozat, 2004).

- Pour l'auteur : « La multiplicité des contextes d'usage et de supports de publication conduit l'auteur à un exercice complexe de navigation entre la structure canonique et le dossier des publications. L'expérience a montré que ni la structure canonique, ni aucune des publications prises isolément n'était suffisante pour que l'auteur objective son contenu qui n'existe que dans l'*ensemble* de ses mises en forme. Une question de *confiance*, y compris avec sa dimension irrationnelle, se pose en effet. Le besoin de validation de tous les supports de publication et non uniquement du document canonique est également porté par le fait que seuls ces documents "réels" permettent aux auteurs une projection dans les usages qui donnent corps à leur relecture. » (Crozat et Bachimont, 2004).

2.2.2 Transclusion et graphe documentaire

La transclusion est un concept proposé à l'origine par Nelson (1981) dans le cadre de l'hypertexte. Il désigne par là le fait qu'un document, ou fragment, puisse apparaître comme une partie intégrante d'un autre document dans lequel ce fragment est inclus par référence.

Dans le contexte des chaînes éditoriales, la transclusion consiste à décomposer la FG en fragments dont le contenu est ré-inclus dans la FP par la transformation. En théorie, la transclusion est invisible dans la FP : les deux fragments liés forment un seul et même texte, lisible de manière linéaire par le lecteur. En pratique, on peut aussi parler de transclusion dans un sens plus large, par exemple lorsque le fragment référencé s'affiche dans une page web indépendante au niveau de la FP (cas des grains dans les modules Opale) : cette page n'en reste pas moins intégrée au document en tant que forme publiée.

Il peut exister d'autres modalités d'agrégation entre deux fragments : Scenari propose par exemple non-seulement le lien de transclusion (grains transclus dans les modules), mais aussi le lien de référence (un fragment de définition attaché à une portion de texte dans un grain).

Le fait qu'un fragment puisse être transclus dans plusieurs autres fragments favorise la réutilisation de contenus : typiquement, un grain Opale peut être utilisé dans plusieurs modules à la fois. La production documentaire est ainsi déplacée sur un réseau de fragments, aussi appelé graphe documentaire (Arribe, 2014), dont les arcs sont les liens de transclusion et de référence entre fragments.

Le graphe documentaire a deux conséquences sur la relecture. Premièrement, tout fragment peut être mobilisé dans plusieurs contextes. La mise à jour d'un fragment n'amène donc plus seulement à contrôler la *cohérence éditoriale*, définie par Arribe comme « le bon enchaînement et l'absence de contradictions dans le contenu d'un document » (2014, p. 57), mais également la *cohérence du graphe*, au sens de la cohérence éditoriale de tous les documents engendrés par le graphe (*ibid.*). Cela a pour effet d'entraîner la relecture de plusieurs documents, comme le montre le schéma ci-dessous :

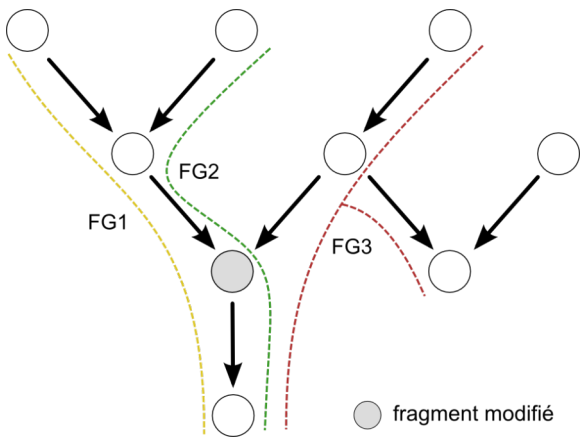


Figure 26 - Trois documents à relire suite à la modification d'un fragment dans le graphe documentaire.

De plus, le graphe documentaire impose la relecture d'un document fragment par fragment. Par exemple, la fonction de différentiel dans l'éditeur Scenari est utilisable sur un fragment et une de ses entrées d'historique :

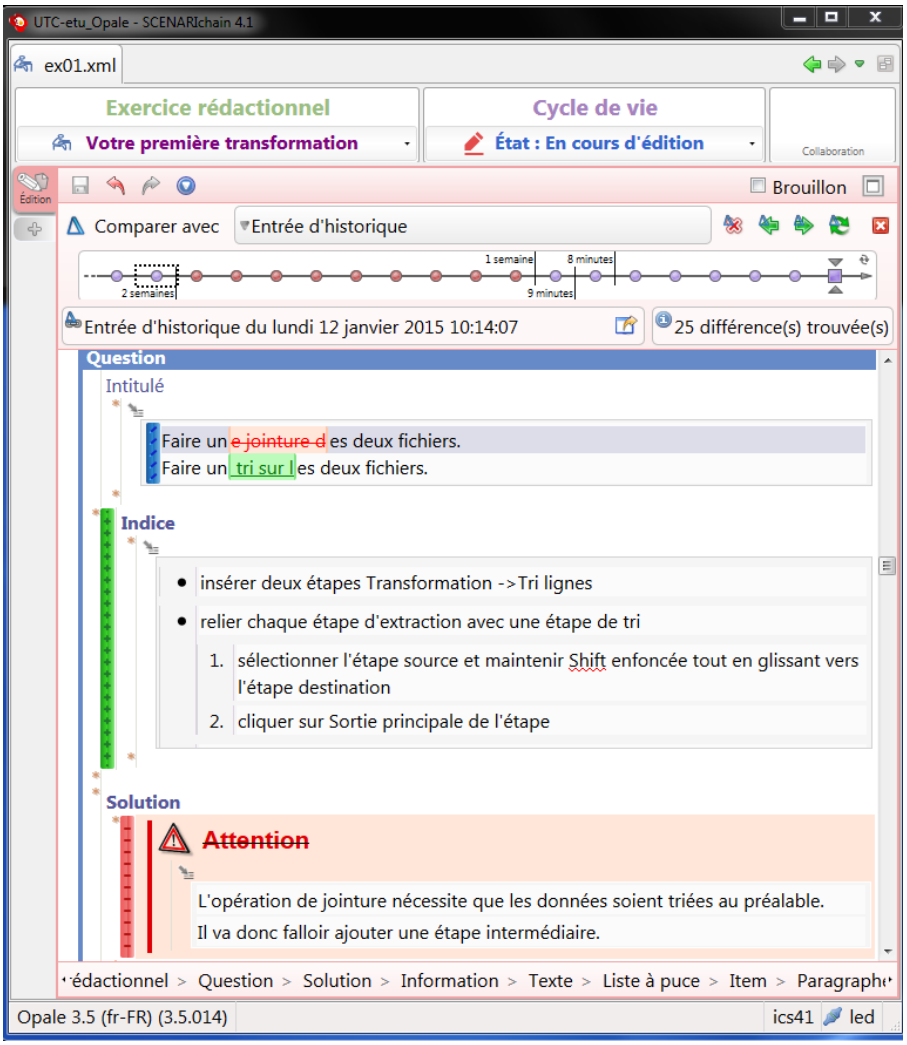


Figure 27 - Différentiel d'un fragment avec une de ses entrées d'historique.

2.2.3 Dérivation et déclinaison

Dérivation

La transclusion ne permet pas de faire une adaptation du fragment locale à un de ses contextes de réutilisation. L'enjeu de la dérivation est de lever cette limite : par exemple, un enseignant devant réaliser une version de son cours pour la formation continue souhaite réutiliser les contenus qu'il a rédigés pour la formation initiale, en les adaptant.

La dérivation d'un fragment F consiste à produire un fragment F' pour pouvoir en modifier son contenu, tout en conservant un lien généalogique de F' vers F. Les modifications de F' n'impactent pas F, mais les modifications de F peuvent avoir un effet sur F' : par exemple, ces modifications peuvent être propagées automatiquement à F' ; au contraire, dans un contexte collaboratif, l'auteur chargé de F' sera averti des modifications et pourra choisir de les réintégrer à F' ou non.

La dérivation est instrumentée dans Scenari à travers les notions d'*atelier premier* et de *calque* (Arribe, 2014). L'atelier premier contient les fragments d'origine, tandis que le calque contient les fragments dérivés et éventuellement d'autres fragments spécifiques. Ainsi, les contextes de rééditorialisation peuvent être vus de manière séparée. On parle de calque *divergeant* pour préciser que les modifications ayant lieu dans le calque n'impactent pas l'atelier premier.

Déclinaison

Enfin, la déclinaison permet à l'auteur de gérer différentes adaptations du contenu d'un fragment non pas via des fragments dérivés, mais au sein du même fragment à l'aide d'un mécanisme de filtres pris en compte lors de la publication. Autrement dit, la déclinaison est la programmation de variations *a priori* du contenu (au sens où elle est prévue par le modèle documentaire), et non *a posteriori* comme l'est la dérivation.

On peut envisager la déclinaison comme un cas particulier de polymorphisme, dans lequel les différentes FP correspondent à autant de variations (et non de mises en forme) de la même FG.

Similairement à la réutilisation par transclusion, toute modification d'un contenu commun à plusieurs variations, obtenues par dérivation ou déclinaison, peut avoir un impact sur la cohérence éditoriale de l'une ou l'autre de ces variations.

2.3 Vers la conception de formes de relecture

Les chaînes éditoriales se positionnent comme des « système[s] de production documentaire cherchant à instrumenter des fonctions d'écriture numérique originales » (Crozat, 2012a), tirant parti de la *tendance technique* du numérique pour automatiser la rééditorialisation.

L'usage de ces fonctions amène à démultiplier les formes documentaires (FE instables, FP interactives...), dont aucune n'est réellement adaptée à la relecture comme nous l'avons vu dans les sections précédentes. Cependant, le polymorphisme qui a permis d'obtenir ces formes à partir d'une même FG peut aussi servir à produire de nouvelles formes dédiées à la relecture. **L'enjeu de notre recherche est alors de définir des stratégies de conception pour ces formes de relecture.**

La rééditorialisation illustre à notre sens un cas de *pharmakon*, concept philosophique de Stiegler caractérisant l'ambivalence de la technique : « *Tout objet technique est pharmacologique* : il est à la fois poison et remède. Le *pharmakon* est à la fois ce qui permet de prendre soin et ce dont il faut prendre soin, au sens où il faut y faire attention : c'est une puissance curative dans la mesure et la démesure où c'est une puissance destructrice » (Stiegler, 2012). La rééditorialisation est effectivement dans un rapport pharmacologique vis-à-vis du document : elle a un rôle curatif vis-à-vis du *clonage* (usage massif du "copier/coller") qui a pour effets néfastes la *redondance* et l'*oubli de la source* (Crozat, 2012a) ; mais ce faisant, elle devient un "poison" pour l'activité de relecture ; elle possède finalement, à travers le polymorphisme, un principe *auto-curatif* permettant de produire une énième forme dédiée cette fois-ci à la relecture.

Nous allons répondre à cette problématique en deux temps. Tout d'abord, l'état de l'art présentera trois fonctions d'aide à la relecture :

- le différentiel, répondant au problème de l'instabilité ;
- l'annotation, qui prend place dans la relecture de fond ;
- la correction automatique, instrumentant la relecture de forme.

Nous chercherons ensuite à compléter cet état de l'art en proposant deux techniques pour la conception de formes de relecture :

- la linéarisation, visant à résoudre le problème posé par l'interactivité ;
- la tabulation, qui s'attachera à traiter le problème posé par la rééditorialisation dans le cas de la déclinaison.

2.4 Cadre théorique

Nous revenons ici sur les hypothèses théoriques qui sous-tendent la notion de document numérique et sont à l'origine de la problématique de nos travaux. En effet, les trois propriétés du numérique que nous avons mobilisées s'appuient sur les *tropismes* proposés par Crozat (2015a) afin de caractériser la *tendance technique* du numérique identifiée par Bachimont (2007). Cette tendance permet non-seulement de mieux comprendre l'évolution du document avec le numérique, mais aussi « d'inventer - pour les exhumer - des formes documentaires qui ouvrent un champ du possible nouveau, pour tenter d'en anticiper les incidences cognitives et sociétales en genèse » (Crozat, 2012a), notamment à travers les fonctions de rééditorialisation proposées par les chaînes éditoriales.

2.4.1 Du document au document numérique

Buckland (1997) observe que la documentation, discipline de gestion des documents, a entraîné la réinterrogation de son objet d'étude. En effet, la vision traditionnelle du document papier était devenue trop étroite pour inclure d'autres médias tels que l'audiovisuel. Les théoriciens de la documentation ont alors proposé de nouvelles définitions, moins attachées à la matérialité des documents qu'aux fonctions qu'ils assurent.

Pour Otlet, dont le *Traité de documentation* publié en 1934 fut à l'origine de la discipline, ce ne sont pas seulement les objets décrits dans les textes ou représentés par des images qui ont le statut de document, mais également les objets eux-mêmes dans la mesure où ils sont porteurs d'informations. Un document peut alors être tri-dimensionnel, tels une sculpture dans un musée ou encore un objet archéologique.

Cette notion du document comme objet est renforcée par Briet, qui en donne la définition suivante : « tout indice concret ou symbolique, conservé ou enregistré, aux fins de représenter, de reconstituer ou de prouver un phénomène ou physique ou intellectuel » (Briet, 1951, p. 7). Briet donne notamment l'exemple d'une antilope qui, si elle est exposée dans un zoo, acquiert le statut de document.

À partir de la réflexion de Briet, Buckland propose les conditions suivantes pour déterminer si un objet est un document : il faut qu'il y ait *matérialité* et *intentionnalité* ; les objets doivent être *traités* (Zacklad (2005) parle plus précisément de *documentarisation*) et *perçus* en tant que document.

Dans ce mémoire, nous partagerons la définition de Bachimont et Crozat : « On définira [...] un document comme une inscription de contenus sur un support pérenne, établie dans un contexte éditorial. Un contenu est une forme d'expression pourvue d'une valeur culturelle associée à un véhicule matériel, il exprime une signification et suscite une interprétation ; une inscription est un contenu fixé sur un support matériel, tel qu'il lui apporte une permanence dans le temps ; un contexte éditorial est l'association d'un contexte de production et d'un contexte de réception. » (Crozat, 2012a). La notion de contexte éditorial est à rapprocher de la distinction faite par Bachimont entre les documents possédant une intentionnalité soit *a priori*, où le support matériel a été constitué pour être un document, soit *a posteriori*, où l'objet matériel précède le document (Bachimont, 2007, p. 180). Si l'intentionnalité *a posteriori* permet de prendre en compte la notion de document comme objet (et les exemples d'Otlet et de Briet), nous nous restreindrons dans ce mémoire à l'intentionnalité *a priori*.

Le concept de document est fortement réinterrogé avec le numérique. Buckland (1998) souligne le fait qu'avec le stockage sous forme binaire, le document perd définitivement sa forme physique traditionnelle, ce qui renforce la vision fonctionnelle du document qu'avaient proposé Otlet et Briet, entre autres. Il illustre cette rupture en donnant l'exemple des tables de logarithmes, un document imprimé conventionnellement utilisé avant l'apparition des calculatrices. Une version numérique de ce document peut tout à fait être le résultat d'un programme dans lequel les valeurs logarithmiques sont calculées dynamiquement plutôt que stockées "en dur".

Pour le collectif Pédauque, cette rupture n'est pas sans conséquence sur les pratiques documentaires : « La manifestation la plus évidente du changement est [...] la perte de la stabilité du document comme objet matériel et sa transformation en un processus construit à la demande, qui ébranle parfois la confiance que l'on mettait en lui » (Pédauque, 2003). Crozat précise que le numérique impose « la séparation entre la forme d'inscription, une ressource binaire sur un support d'enregistrement, et la forme de lecture, une manifestation sémiotique sur un dispositif de lecture, la seconde étant calculée à partir de la première par l'intermédiaire de l'exécution d'un programme. **Ce qu'on lit n'est plus ce qui a été écrit.** » (Crozat, 2012a). Ainsi dans le numérique, il convient de ne plus considérer le document dans son unité traditionnelle, mais d'envisager les différentes formes qu'il prend.

2.4.2 La raison graphique

Pour Bachimont, les propriétés physiques des inscriptions ont un rôle vis-à-vis des connaissances : « La forme d'expression, sa structure matérielle, son apparence sensible et perceptible, conditionnent l'interprétation qui pourra être effectuée et le sens qui pourra être attribué à l'inscription. L'inscription, dans sa matérialité, fournit une matière à réflexion et donne à penser à l'esprit qui l'appréhende. Selon le type de matérialité qui l'incarne, un type d'interprétation se dégagera qui reflétera dans sa conceptualité et ses structures cognitives la matérialité interprétée. » (Bachimont, 2007, p. 69). Les inscriptions de connaissances sont l'objet de la théorie du support [p.126] proposée par Bachimont (2004).

Concernant l'écriture graphique, Goody (1979) a montré qu'elle a donné naissance à de nouvelles structures de pensée du fait des possibilités de spatialisation et de permanence des inscriptions offertes par le support, inconcevables dans le discours oral. Ces structures sont :

- la *liste*, qui permet d'ordonner et de classer des éléments qui, énoncés oralement, seraient confondus dans un seul et même flux ;
- le *tableau*, qui tire parti de la bidimensionnalité du support pour catégoriser chaque unité en fonction de sa position verticale et horizontale ;
- la *formule*, qui permet de donner aux signes écrits une signification formelle mobilisable dans un raisonnement logique par exemple.

En s'appuyant sur ces trois structures, Goody pose l'existence d'une *raison graphique*, rationalité propre à l'écriture qui serait à l'origine des sciences (notamment les mathématiques) et plus largement de la société moderne.

La raison graphique et par suite l'imprimerie ont constitué les pratiques documentaires autour du document papier (dont celles de la documentation) en vigueur depuis plusieurs siècles et encore largement répandues aujourd'hui. Pour Bachimont : « La culture de l'écrit s'est construite sur les propriétés de fixité et de permanence du support, les évolutions ayant la plupart du temps eu lieu dans la perspective de renforcer ces deux propriétés. Un contenu est donc ce qui est fixé sur un support, la matérialité du support apportant la persistance temporelle au contenu ainsi que son intégrité et ses délimitations. » (Bachimont, 2007, p. 222). Dans cette culture de l'écrit, l'interprétation repose sur une instrumentation permettant à la fois l'*objectivation* et l'*appropriation* (*ibid.*, p. 172) :

- « l'objectivation se traduit par le fait de pouvoir accéder à un exemplaire de référence, une version faisant autorité, sur laquelle tout le monde est d'accord (ou presque) pour voir la version authentifiée (à défaut d'authentique) du contenu exprimé. »
- « l'appropriation se traduit par des vues construites à partir de la version de référence [par exemple des vues annotées], pour exprimer et présenter le contenu dans une forme plus accessible et plus familière au lecteur. »

2.4.3 La tendance technique du numérique

Afin de mieux comprendre l'évolution du document avec ce changement de support, Bachimont cherche à identifier la *tendance technique* du numérique.

La tendance technique est un concept proposé par Leroi-Gourhan, anthropologue spécialiste de la Préhistoire. Pour cet auteur, l'évolution technique n'est pas la pure expression d'un "génie de l'invention" plus ou moins présent dans une société particulière. Son hypothèse est que la technique est caractérisée par des *tendances* et des *faits*. La tendance agit sur la matière comme un déterminisme qui, tel que pour un organe biologique, la pousse à s'organiser pour se donner une fonction technique : « [La tendance] pousse le silex tenu à la main à acquérir un manche, le ballot traîné sur deux perces à se munir de roues. » (Leroi-Gourhan, 1943, p. 27). Les faits sont les expressions locales et contingentes d'une tendance, chacune colorée de la singularité du groupe ou de la société mettant au point l'objet technique (environnement, climat, culture...).

Par exemple, Leroi-Gourhan a étudié des formes de charrue retrouvées dans des régions géographiquement proches (Asie du Sud-Est, Japon, Tibet), chacune ayant néanmoins des spécificités locales (par rapport au type de sol cultivé ou encore à la valeur culturelle associée à l'objet). À la vision ethnocentrique selon laquelle une des trois régions aurait diffusé son savoir technique aux deux autres, s'oppose l'hypothèse d'un tendance "charrue" s'exprimant localement à travers des faits, en l'occurrence trois formes voisines de charrue (Stiegler, 1994a).

Bachimont affirme que le support numérique n'est pas orthothétique au sens de Stiegler, qui désigne par là « le fait que des techniques de la mémoire permettent de poser (*thèse*) exactement (*ortho*) ce qu'elles enregistrent » (Bachimont, 2007, p. 258). Le support papier, par exemple, est orthothétique : l'écriture sur ce support permet d'examiner « ce qui s'est *pensé* comme étant ce qui s'est *passé* » (Stiegler, 1994b). Inversement, le support numérique est *autothétique* : les unités binaires ne renvoient à rien d'interprétable, et le calcul qui s'en empare ne fait appel « à aucune autre réalité que lui-même » (Bachimont, 2007, p. 247). Tout contenu numérique est donc « d'emblée falsifiable et possiblement falsifié », car « il ne véhicule pas sur lui sa genèse ni les étapes de sa construction. » (*ibid.*, pp. 34-35).

La manipulation par le calcul est donc l'essence technique du support numérique selon Bachimont. « Ça a été manipulé » : tel est le *noème* du numérique, soit « ce qu'il faut comprendre et penser à propos du numérique » (Bachimont, 2007, p. 33), à l'instar du « Ça a été » de Roland Barthes à propos de la photographie. Crozat (2015a) illustre ce noème en donnant l'exemple de la chaîne manipulatoire d'un mail : le contenu tapé au clavier a d'abord été encodé en séquences d'octets stockées dans la mémoire de l'ordinateur, complétées par d'autres séquences relatives aux métadonnées du mail (date, expéditeur, récepteur...) ; puis ces séquences ont été copiées et transmises, via un protocole de communication, à un serveur qui les a retransmises à l'ordinateur du récepteur ; enfin, elles sont décodées par un programme pour s'afficher à l'écran sous forme de caractères interprétables.

Après en avoir proposé le noème, Bachimont établit par déduction la tendance technique du numérique à travers les concepts de *fragmentation* et de *recombinaison* (Bachimont, 2007, p. 37) :

- la fragmentation est la discrétisation des contenus sous forme d'unités *désémantisées* (codage binaire) ;
- la recombinaison est la manipulation de ces unités discrètes par des règles formelles, pouvant mener à une *resémantisation* des contenus bien qu'étant arbitraires par rapport à eux.

2.4.4 La raison computationnelle

Comme l'a fait Goody avec l'écriture graphique, Bachimont propose de rechercher l'impact du calcul sur nos rapports à la connaissance et les nouvelles structures de pensée qui en émanent.

Si l'écriture graphique permet de spatialiser le temps du discours oral, le calcul traduit le mouvement inverse : « [...] l'informatique permet un déploiement de l'espace en temps : un programme n'est pas autre chose qu'un dispositif réglant un déroulement dans le temps, le calcul ou l'exécution du programme, à partir d'une structure spécifiée dans l'espace, l'algorithme ou programme. » (Bachimont, 2007, p. 73). Bachimont en déduit les structures de pensée d'une *raison computationnelle* qu'il met en regard de celles de la raison graphique (*ibid.*, p. 74) :

- le *programme*, succédant à la liste : la structure spatiale du programme est déployée en temps par l'exécution de ses instructions, ce temps étant celui « nécessaire à l'exploration systématique d'un espace de calcul, comme parcours de tous les cas possibles d'une combinatoire » ;
- le *réseau*, succédant au tableau : le réseau est un tableau "augmenté" dans lequel les cases peuvent se référencer entre elles indépendamment de leur situation spatiale en lignes et en colonnes ;
- la *couche*, succédant à la formule : la formalisation écrite se traduit au niveau informatique par un empilement de couches indépendantes les unes des autres, c'est-à-dire que les calculs effectués au niveau d'une couche peuvent être pensés indépendamment de l'implémentation des couches sous-jacentes.

La raison computationnelle n'est pas sans conséquence sur l'interprétation des inscriptions numériques. Pour Bachimont, celles-ci entraînent une *désorientation* dans le sens où l'intelligibilité des contenus calculés n'est garantie que par la vérification de leurs processus de (re)construction, à savoir l'articulation des programmes, des réseaux et des couches, qui reste difficilement réalisable en pratique. Ainsi : « La désorientation conceptuelle dans laquelle nous entraînent les inscriptions se manifeste par une dispersion du sens : l'interprétation n'aboutit pas car le parcours se perd dans des manifestations matérielles désordonnées » (Bachimont, 2007, p. 78).

Les conditions d'intelligibilité d'un document Web, par exemple, permettent d'illustrer l'idée de désorientation : la lecture du contenu dépend de l'exécution de plusieurs programmes (interprétation du code HTML, exécution des scripts, affichage des styles...) appartenant à des couches logicielles différentes (par niveau d'abstraction décroissant : moteur de rendu du navigateur, langage de programmation, langage machine) et exploitant un réseau d'adresses mémoire non-contiguës (liens vers les autres pages ou vers une ancre de la même page...). La perception simultanée de tous ces éléments calculatoires pour parvenir *in fine* à l'interprétation des inscriptions est en effet une opération complexe.

Le document numérique en tant que reconstruction dynamique met à mal l'interprétation en ce que l'objectivation des contenus est annulée au profit de l'appropriation : « le document n'existe qu'au moment de sa consultation, dans une forme n'existant que dans le contexte singulier d'une consultation par un lecteur individuel donné » (*ibid.*, p. 240). L'objectivation suppose d'accéder à la ressource invariante, ou du moins à sa vue canonique (la ressource en elle-même, binaire, étant inaccessible), par exemple une vue XML reflétant la structure logique du document (Crozat et Bachimont, 2004), et de vérifier sa reconstruction, qui comme nous l'avons vu précédemment se heurte à la désorientation.

Pour finir, Crozat souligne l'ambiguïté portée par la notion de document avec son évolution numérique : « Le document numérique finalement n'existe pas, la locution est oxymorique. Il ne peut exister que des constructions numériques dont le traitement calculatoire permet de simuler un ordre documentaire. » (Crozat, 2012a).

2.4.5 Les tropismes du numérique

Pour Crozat (2015a), un contenu numérique se définit non-seulement dans le fait qu'il a été manipulé, mais également dans le fait qu'il est encore manipulable. Ainsi, le noème du numérique ne s'énonce pas seulement au passé, mais également au futur : « [Ça a été manipulé,] et ça le sera à nouveau [...] ». Dans ce cadre, Crozat propose de caractériser la tendance technique du numérique à travers six *tropismes*, c'est-à-dire les propriétés essentielles sans lesquelles « il n'est pas possible de bien

penser l'objet technique [numérique] » (*ibid.*) :

- *Abstraction* : « Ça a été codé et ce sera recodé ». Les contenus sont conformes à des modèles permettant leur manipulation algorithmique.
- *Adressage* : « Ça a été trouvé et ça sera retrouvé ». Chaque contenu est localisé à une adresse permettant d'y accéder.
- *Connexion* : « Ça a été transmis et ça sera retransmis ». Toute machine peut être connectée à une autre machine du réseau et lui transmettre de l'information.
- *Duplication* : « Ça a été copié et ça sera recopié ». Toute transmission d'information (sur le réseau, de la mémoire vive au disque dur, etc.) est en fait une copie de cette information.
- *Transformation* : « Ça a été changé et ça sera rechangé ». L'information binaire subit une chaîne de transformations algorithmiques jusqu'à sa restitution à l'écran.
- *Universalité* : « Ça a été intégré et ça sera réintégré ». Tout contenu est ramené à un codage binaire, et ce quelle que soit sa forme sémiotique d'origine (texte, image, son...).

Outre le fait que les tropismes permettent de mieux comprendre le numérique, leur intérêt réside dans leur capacité à être déclinés en *fonctions*, au sens de « modalités effectives d'écriture rendues disponibles par les applications » (Crozat *et al.*, 2011). Six de ces fonctions nous intéressent particulièrement, puisque ce sont sur elles que s'appuient les trois propriétés du numérique mobilisées dans la problématique de nos travaux :

- *Itération* (transformation) : « Le numérique permet de faire évoluer progressivement le contenu par étapes successives. »
- *Interactivité* (transformation) : « Le numérique permet de programmer des interactions entre l'utilisateur et la machine. »
- *Polymorphisme* (abstraction) : « Le numérique permet de calculer plusieurs formes de présentation à partir de ressources identiques. »
- *Transclusion* (adressage) : « Le numérique permet d'intégrer des parties de contenus tiers à l'intérieur d'un contenu pour les afficher comme si elles en faisaient partie intégrante. »
- *Dérivation* (duplication) : « Le numérique permet d'élaborer un nouveau contenu à partir de la copie d'un contenu précédemment existant. »
- *Paramétrisation/Déclinaison* (abstraction) : « Le numérique permet de créer des contenus déclinables selon des paramètres fixés a priori. »

Tropismes	**Fonctions**	**Propriétés impactant la relecture**
Transformation	Itération	Instabilité
	Interactivité	Interactivité
Abstraction	Polymorphisme	Rééditorialisation
	Paramétrisation/Déclinaison	
Adressage	Transclusion	
Duplication	Dérivation	

Tableau 1 - Tropismes, fonctions et propriétés du numérique impactant la relecture.

Chapitre 3

État de l'art

3.1 Différentiel 44
3.1.1 Contexte 44
3.1.2 Exemple introductif 45
3.1.3 Deltas 48
3.1.4 Distance d'édition 50
3.1.5 Diff XML 52
3.2 Annotation 54
3.2.1 Lecture active 54
3.2.2 Documentarisation et collaboration 56
3.3 Correction automatique 58
3.3.1 Approches 58
Approche lexicale 59
Approche syntaxique 60
Approche probabiliste 61
Analyse automatique du discours 62
3.3.2 Correcteurs automatiques et relecture de forme 63

Ce chapitre vise à présenter trois fonctions de relecture : le différentiel, l'annotation et la correction automatique. Le différentiel sera détaillé à travers des exemples à partir desquels nous préciserons les différents éléments théoriques et techniques en jeu. L'annotation sera abordée à travers deux champs d'usage : la lecture active et la documentarisation de l'activité. La correction automatique sera finalement illustrée selon les différentes approches qu'elle met en œuvre.

3.1 Différentiel

3.1.1 Contexte

Gestion de versions

Le différentiel est une fonction dont l'usage s'est répandu avec la gestion de versions, dans le cadre du génie logiciel. Un système de gestion de versions, tel que *Subversion* (Collins-Sussman *et al.*, 2004), permet de centraliser le code d'une application (arborescence de fichiers) dans un dépôt (serveur distant en général) afin que plusieurs développeurs puissent synchroniser leurs versions du code. L'action de *checkout* permet à chaque développeur de récupérer une copie locale du dépôt. Cette copie locale peut être mise à jour avec la version courante du dépôt par l'action d'*update*. Les modifications effectuées sur la copie locale peuvent être reversées dans le dépôt par l'action de *commit*. La version courante est alors modifiée, mais le dépôt conserve un historique des anciennes versions des fichiers (on parle de révisions). Il est également possible de créer des branches du dépôt, c'est-à-dire des versions qui évoluent indépendamment du tronc principal (*trunk*).

Par exemple, soient :

- F la version courante d'un fichier ;
- A et B deux développeurs travaillant en même temps sur leur copie locale de F ;
- FA, resp. FB les versions de F modifiée par A, resp. B.

Si FA est commité en premier, le système informera B, lorsque celui-ci voudra commiter FB, que sa version de F n'est plus à jour. Deux cas de figures sont possibles : les modifications de A et de B sur F peuvent être indépendantes, ou bien concerner les mêmes lignes. Dans le premier cas, le système peut fusionner FA et FB. Dans le second, la fusion ne peut être effectuée automatiquement, amenant B à devoir choisir la bonne version sur chaque ligne conflictuelle. Pour cela, B peut s'appuyer sur la fonction de différentiel (proposée, par exemple, par les clients Subversion tels que *TortoiseSVN* ou *Subclipse*) afin de comparer FA et FB. Notons que le différentiel peut aussi être utile à B dans le premier cas, avant de demander au système de fusionner FA et FB (typiquement, pour vérifier que les modifications parallèles sont bien cohérentes entre elles). Dans un cas plus général, une revue de code peut s'appuyer sur le différentiel : par exemple, on cherche à savoir quelle modification a entraîné une régression de l'application.

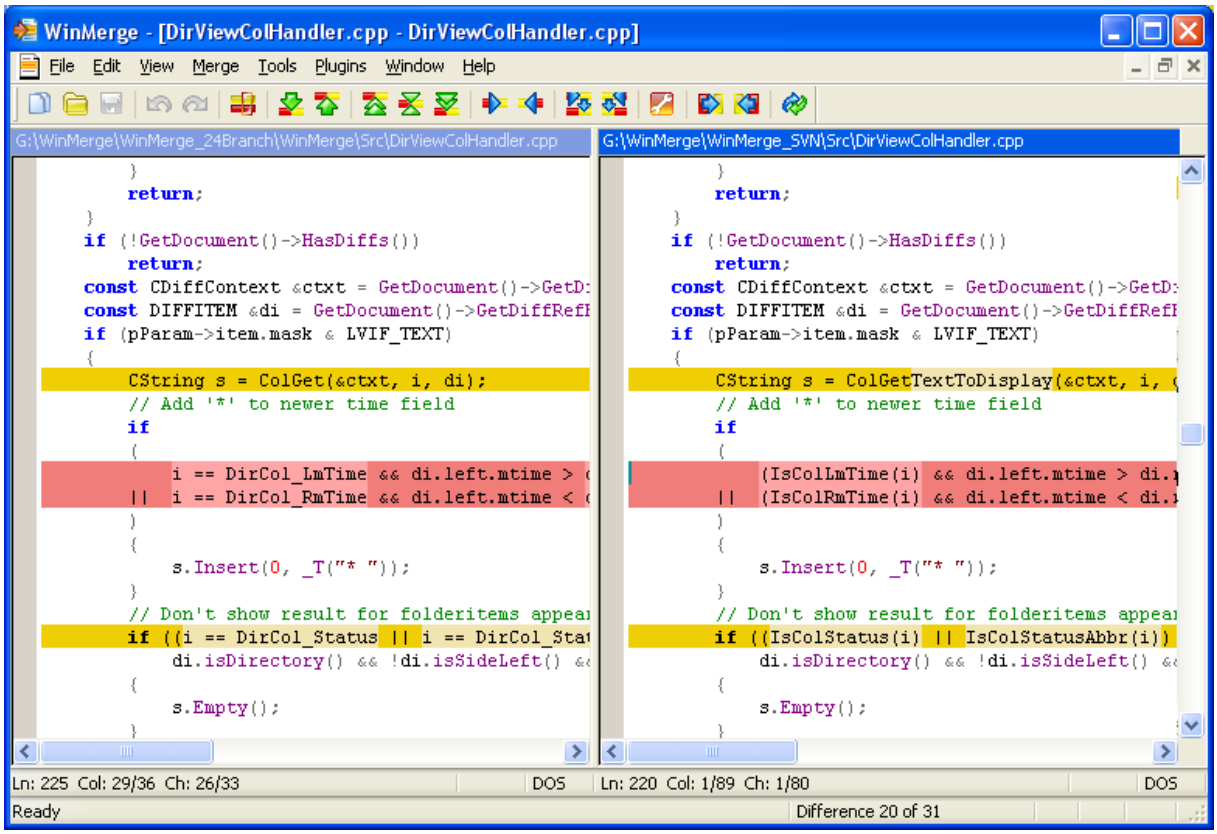


Figure 28 - Capture d'écran du logiciel WinMerge (source : http://winmerge.org/about/screenshots/filecmp_inlinediff.png).

La gestion de versions s'est depuis étendue au monde du document, par exemple dans les éditeurs bureautiques ou encore les Wiki.

Définitions

Soient D1 et D2 deux documents ou versions/variations d'un même document. Un différentiel, abrégé *diff*, est un algorithme comparant D1 et D2 et produisant une liste de différences appelée *delta*. Dans le cas où D2 est comparé *par rapport à* D1 (et non l'inverse), le delta exprime une liste d'opérations (insertions, suppressions...) à appliquer à D1 pour obtenir D2. Un *patch* désigne l'algorithme exécutant le delta dans l'ordre prescrit des opérations. Un *merge* est un algorithme permettant de fusionner D1 et D2 à partir de leur delta, tandis qu'un *tree-way merge* se base également sur l'ancêtre commun de D1 et de D2, noté D0, et nécessite les deltas de D1 avec D0 d'une part et de D2 avec D0 d'autre part.

Nous nous intéressons ici plus particulièrement au diff XML. En effet, dans les chaînes éditoriales, on a d'une part les formes génératrices encodées en XML, et d'autre part les formes publiées web au format HTML.

Nous appelons *forme différentielle* la forme permettant la relecture d'un document dans ses différentes versions/variations, en proposant notamment une visualisation des différences entre ces versions/variations. La forme différentielle entre D1 et D2 peut être construite par application sur D1 du delta résultant du diff de D2 par rapport à D1.

3.1.2 Exemple introductif

Soient les deux fichiers texte suivants (exemple tiré du manuel d'administration fonctionnelle de

SCENARIclient (4.1) *(http://docs.kelis.fr/sc41/adminFonct/site/co/adminAvanc.html)*) :

- *v1.txt* :

```
1 Pour envoyer un message de maintenance aux utilisateurs connectés,
  sélectionnez le message adapté dans la liste déroulante, puis cliquez sur
  "Envoyer".
2 Les utilisateurs le reçoivent alors dans l'interface de l'application
  pendant 30 secondes.
3 Si, avant la fin du décompte, aucun utilisateur ne s'oppose à
  l'opération de maintenance, le message "Aucun utilisateur ne s'oppose à
  l'opération de maintenance" s'affiche.
```

- *v2.txt* :

```
1 Pour envoyer un message de maintenance aux utilisateurs connectés,
  sélectionnez le message adapté dans la liste déroulante, puis cliquez sur
  "Envoyer".
2 Un décompte de 30 secondes s'enclenche alors dans l'onglet
  "Administration avancée".
3 Les utilisateurs reçoivent le message dans l'interface de
  l'application.
4 Si, avant la fin du décompte, aucun utilisateur ne s'oppose à
  l'opération de maintenance, le message "Aucun utilisateur ne s'oppose à
  l'opération de maintenance" s'affiche.
```

La commande Unix *diff* (Hunt et MacIlroy), 1976) calcule les différences entre ces deux fichiers en termes de lignes ajoutées (n'existant pas dans *v1*) et supprimées (n'existant pas dans *v2*). Le résultat s'affiche de la manière suivante dans le terminal (format unifié) :

```
1 diff -u v1.txt v2.txt
2 --- v1.txt  2015-09-15 12:22:02.448751522 +0200
3 +++ v2.txt  2015-09-15 12:22:42.352949397 +0200
4 @@ -1,3 +1,4 @@
5  Pour envoyer un message de maintenance aux utilisateurs connectés,
  sélectionnez le message adapté dans la liste déroulante, puis cliquez sur
  "Envoyer".
6 -Les utilisateurs le reçoivent alors dans l'interface de l'application
  pendant 30 secondes.
7 +Un décompte de 30 secondes s'enclenche alors dans l'onglet
  "Administration avancée".
8 +Les utilisateurs reçoivent le message dans l'interface de l'application.
9  Si, avant la fin du décompte, aucun utilisateur ne s'oppose à l'opération
  de maintenance, le message "Aucun utilisateur ne s'oppose à l'opération de
  maintenance" s'affiche
```

Diff orienté lignes

Les deux figures ci-dessous sont des exemples de formes différentielles pouvant être produites à partir du résultat de *diff*. Les lignes ajoutées (resp. supprimées) sont colorées en bleu (resp. en gris). Dans la première figure, les deux versions sont "fusionnées" (les lignes sont également barrées ou soulignées pour accentuer l'affichage des différences) tandis que dans la seconde, elle sont affichées face à face.

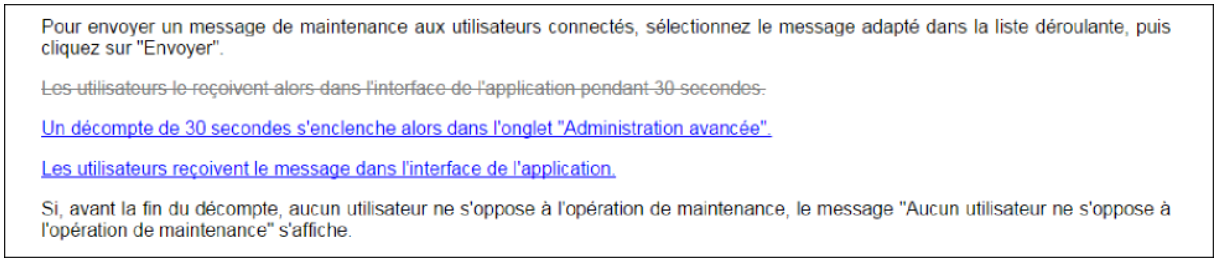


Figure 29 - Affichage "fusionné" des différences.

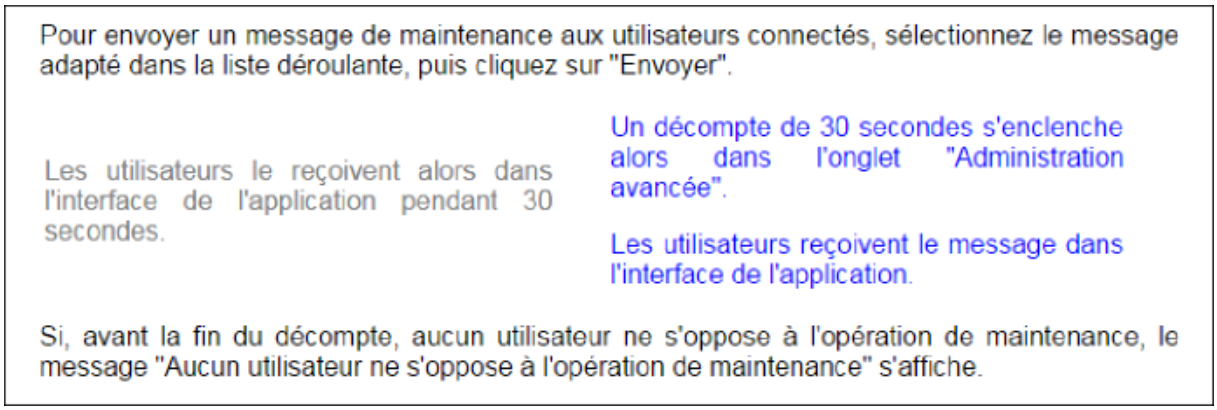


Figure 30 - Affichage en "vis-à-vis" des différences.

Diff orienté caractères

En analysant le contenu, on remarque que la troisième ligne de *v2* est en fait une évolution de la deuxième ligne de *v1*, dans laquelle les chaînes "le", "alors" et "pendant 30 secondes" ont été supprimées, et la chaîne "le message" a été ajoutée. Les deux formes différentielles précédentes créent donc une certaine redondance pour les parties communes de ces deux lignes. Pour améliorer ces formes, la granularité d'expression des différences doit pouvoir être plus fine que la ligne, c'est-à-dire au niveau du caractère, telle que dans la forme ci-dessous :

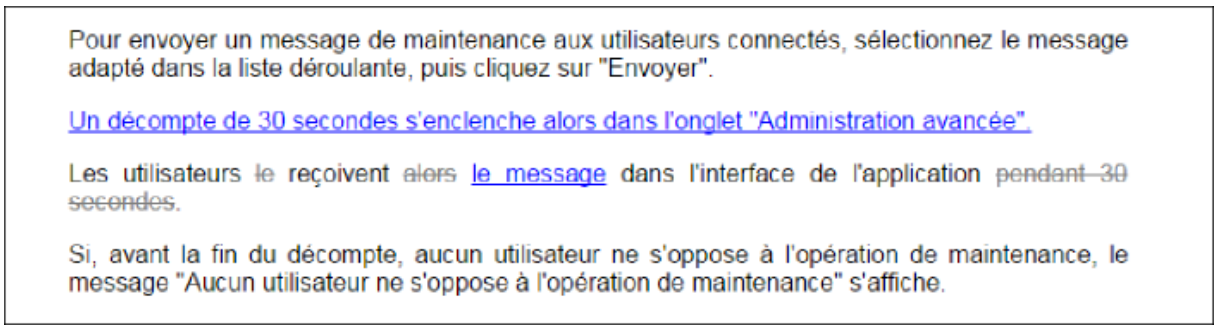


Figure 31 - Affichage des différences au niveau des caractères.

En revanche, l'indication des caractères ajoutés et supprimés au niveau d'une ligne ont un impact sur sa lisibilité. En effet, la ligne acquiert plusieurs niveaux de lecture et ne peut plus se lire tout à fait linéairement. La deuxième forme ci-dessous permet d'éviter cette difficulté, en dupliquant la ligne sur deux colonnes, avec en sur-ajout les caractères propres à chaque version (ce qui minore la redondance). Le fond coloré indique explicitement que cette ligne a été modifiée :

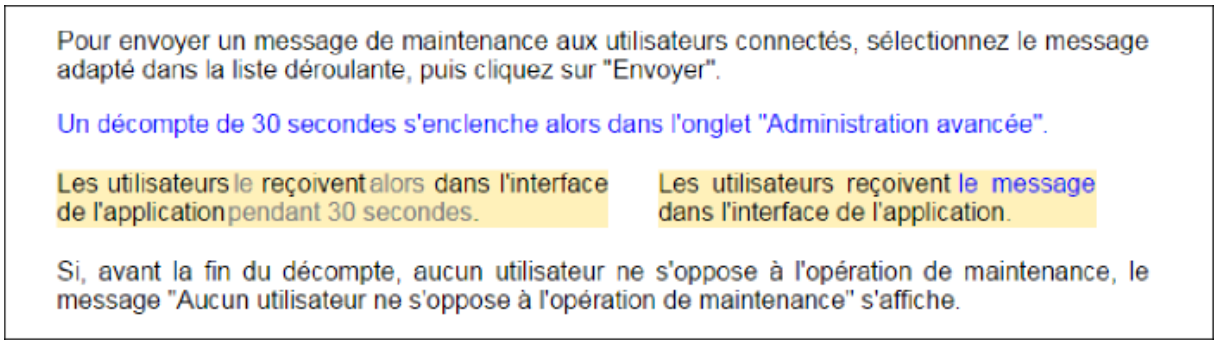


Figure 32 - Affichage en "vis-à-vis" de différences au niveau des caractères.

Autres visualisations

Si les formes différentielles précédentes sont les plus courantes, il existe d'autres façons de visualiser les différences entre versions. Citons par exemple les travaux de Chevalier *et al.* (2010), qui proposent l'utilisation de transitions animées pour naviguer dans l'historique d'un document (une vidéo de démonstration de leur logiciel Diffamation est accessible à l'adresse : https://www.youtube.com/watch?v=17lz5nt5_jg). Cette approche du *replay* est une alternative intéressante aux visualisations statiques de diff. Elle peut en effet sembler plus naturelle et confortable pour l'utilisateur en phase exploratoire, l'animation donnant l'impression de reproduire en temps réel les gestes d'écriture ayant conduit aux modifications. Cependant, l'animation ne nous semble pas adaptée pour un contexte de relecture, qui requiert la synopsis spatiale du texte et des modifications.

3.1.3 Deltas

Le delta est une description *technique* des différences, c'est-à-dire une description pouvant être traitée par la machine en vue de produire un résultat, typiquement un nouveau document (patch, merge) ou bien une forme différentielle. Chaque opération du delta précise l'action à effectuer (insertion, suppression...) ainsi que la localisation de cette action (numéro de ligne, de caractère...).

Soit le delta suivant décrivant les différences entre les fichiers *v1.txt* et *v2.txt* de la section précédente :

```
1 <delta>
2     <removeLine index="2"/>
3     <insertLine afterIndex="1">Un décompte de 30 secondes s'enclenche
  alors dans l'onglet "Administration avancée".</insertLine>
4     <insertLine afterIndex="2">Les utilisateurs reçoivent le message dans
  l'interface de l'application.</insertLine>
5 </delta>
```

Ce delta utilise deux types d'opération, *insertLine* et *removeLine*. L'attribut *afterLine* dans la balise *insertLine* précise l'index de la ligne après laquelle la nouvelle ligne a été ajoutée, dont le contenu est spécifié à l'intérieur de la balise. L'attribut *lineIndex* dans la balise *removeLine* indique l'index de la ligne supprimée. Notons que le contenu de la ligne supprimée n'est pas nécessaire dans le delta, puisqu'il s'agit d'un contenu présent dans la version à laquelle le delta va être appliqué (*v1*).

Pour obtenir *v2* à partir de *v1* à l'aide d'un patch, les opérations du delta doivent être effectuées dans l'ordre prescrit. Par exemple, si les deux premières opérations sont inversées, ce sera la ligne venant juste d'être insérée ("Un décompte de 30 secondes...") qui sera supprimée, et non la ligne 2 de *v1* ("Les utilisateurs le reçoivent alors...") qui aura été décalée à la ligne 3 par l'insertion. D'autres ordonnancements de ces opérations sont possibles en revanche (le delta n'est pas unique), cependant les index des lignes concernées ne seront pas les mêmes, tel que dans le delta suivant :

```
1 <delta>
2     <insertLine afterIndex="2">Un décompte de 30 secondes s'enclenche
  alors dans l'onglet "Administration avancée".</insertLine>
3     <removeLine index="2"/>
4     <insertLine afterIndex="2">Les utilisateurs reçoivent le message dans
  l'interface de l'application.</insertLine>
5 </delta>
```

Ces deux deltas peuvent aussi être utilisés pour construire la première forme différentielle que nous avons proposée. L'application du delta sur *v1* consiste à insérer les lignes de *v2* et à formater les lignes pour l'affichage des différences, par exemple de la manière suivante :

```
1 <p>Pour envoyer un message de maintenance aux utilisateurs connectés,
  sélectionnez le message adapté dans la liste déroulante, puis cliquez sur
  "Envoyer".</p>
2 <p><del>Les utilisateurs le reçoivent alors dans l'interface de
  l'application pendant 30 secondes.</del></p>
3 <p><ins>Un décompte de 30 secondes s'enclenche alors dans l'onglet
  "Administration avancée".</ins></p>
4 <p><ins>Les utilisateurs reçoivent le message dans l'interface de
  l'application.</ins></p>
5 <p>Si, avant la fin du décompte, aucun utilisateur ne s'oppose à
  l'opération de maintenance, le message "Aucun utilisateur ne s'oppose à
  l'opération de maintenance" s'affiche.</p>
```

Notons que les opérations de type *removeLine* ne doivent pas mener à la suppression pure et simple de la ligne, contrairement à la logique d'un patch. Ainsi dans notre exemple, la ligne 2 dans la forme différentielle reste bien la ligne 2 de *v1*. Pour que les opérations suivantes ne soient pas faussées, le programme doit pouvoir réévaluer dynamiquement chaque index en lui ajoutant le nombre de suppressions qui n'auront pas "réellement eu lieu" aux index inférieurs ou égaux à cet index. Par exemple dans le deuxième delta ci-dessus, la dernière insertion devra avoir lieu après la troisième ligne, et non pas la deuxième.

Opérations atomiques et opérations complexes

Barabucci (2013) propose un modèle de delta générique dans lequel il distingue deux types d'opération : atomiques et complexes. Les opérations atomiques sont les opérations qui ne peuvent pas être composées à partir d'autres opérations. Il s'agit typiquement des opérations d'insertion et de suppression, telles que nous les avons utilisées dans les deux deltas ci-dessus. Dans le cadre d'un patch par exemple, ces opérations sont suffisantes. Les opérations complexes sont quant à elles des opérations composées à partir d'opérations atomiques. Dans l'exemple de la seconde forme différentielle, les opérations d'insertion et de suppression ne suffisent pas (du moins, pas directement) pour afficher les différences en face à face : il faut pour cela une indication explicite du remplacement de la seconde ligne de *v1* par les seconde et troisième lignes de *v2*, afin d'opposer ces deux groupes de lignes dans un tableau par exemple. Le remplacement pouvant être obtenu à partir d'un ensemble d'insertions et de suppressions localisées à des index adjacents, il constitue une opération complexe que l'on peut mobiliser dans le delta (ici, pour la seconde forme différentielle) :

```
1 <delta>
2   <replaceLines index="2" length="1">
3     <line>Un décompte de 30 secondes s'enclenche alors dans l'onglet
  "Administration avancée".</line>
4     <line>Les utilisateurs reçoivent le message dans l'interface de
  l'application.</line>
5   </replaceLines>
6 </delta>
```

Concernant les deux dernières formes différentielles, un deuxième niveau d'expression des différences entre en jeu, celui des caractères. Les opérations basiques sont ici *removeChars* et *insertChars* (nous préférons le terme "basique" ici, car les opérations atomiques sont plus logiquement l'insertion et la suppression d'un seul caractère). Les index représentent une position de caractère dans la chaîne (notons que pour chaque opération, l'index prend en compte le nombre de caractères supprimés et ajoutés lors des opérations précédentes). Les balises *removeChars* possèdent également un attribut indiquant la longueur de la sous-chaîne à supprimer. Dans le delta ci-dessous, nous utilisons également l'opération complexe *updateLine*, pouvant être composée à partir des opérations *removeChars* et *insertChars* à condition, par exemple, que le volume de ces différences de caractères par rapport à la longueur moyenne des deux lignes ne soit pas trop important (dans le cas contraire, ces deux lignes sont "trop différentes" et gagnent à être exprimées dans un couple *insertLine*/*removeLine*).

```
<delta>
    <insertLine afterIndex="1">Un décompte de 30 secondes s'enclenche
 alors dans l'onglet "Administration avancée".</insertLine>
    <updateLine index="3">
        <removeChars index="17" length="3"/>
        <removeChars index="27" length="6"/>
        <insertChars afterIndex="27"> le message</insertChars>
        <removeChars index="72" length="20"/>
    </updateLine>
</delta>
```

3.1.4 Distance d'édition

Les algorithmes de diff ont pour fondement théorique le problème de la distance d'édition (*edit distance* en anglais). Il s'agit de mesurer la similarité entre deux chaînes de caractères C1 et C2, ou encore entre deux arbres A1 et A2, en cherchant un ensemble minimum d'opérations (*edit script*) permettant de transformer S1 en S2, respectivement A1 en A2.

Distance de Levenshtein

L'algorithme de Levenshtein (1966) recherche la distance de deux chaînes C1 et C2 en évaluant le coût minimal d'édition en termes d'ajout, de suppression et de substitution de caractères. Par exemple, la chaîne "panorama" peut être transformée en "paronomase" à l'aide des cinq opérations suivantes :

1. par**o**rama (substitution de "n" en "r")
2. paro**n**ama (substitution de "r" en "n")
3. parono**m**a (substitution de "a" en "o")
4. paronoma**s** (ajout d'un "s")
5. paronomas**e** (ajout d'un "e")

On peut retrouver ce résultat avec l'algorithme de Levenshtein et vérifier qu'il s'agit bien d'une transformation de coût minimal. Il s'agit de remplir la matrice suivante :

		P	A	R	O	N	O	M	A	S	E
	0	1	2	3	4	5	6	7	8	9	10
P	1	**0**	1	2	3	4	5	6	7	8	9
A	2	1	**0**	1	2	3	4	5	6	7	8
N	3	2	1	**1**	2	2	3	4	5	6	7
O	4	3	2	2	**1**	2	2	3	4	5	6
R	5	4	3	2	2	**2**	3	3	4	5	6
A	6	5	4	3	3	3	**3**	4	3	4	5
M	7	6	5	4	4	4	4	**3**	4	4	5
A	8	7	6	5	5	5	5	4	**3**	**4**	**5**

Tableau 2 - Matrice de calcul de la distance de Levenshtein.

Pour chaque case M[i, j], i et j>0, la valeur affectée est le minimum entre :

- M[i-1, j] + 1 (case directement à gauche) ;

- M[i, j-1] + 1 (case directement en haut) ;
- M[i-1,j-1] + 0 si les deux caractères sont égaux, M[i-1,j-1] + 1 sinon.

La dernière case de la matrice donne la distance d'édition entre C1 et C2, qui est bien celle de la transformation que nous avions proposée. En allant jusqu'à la dernière case tout en choisissant à chaque pas la case suivante la moins coûteuse, on trouve un ensemble minimal d'opérations :

- un saut de colonne (de gauche à droite), par exemple de M[8, 8] = 3 à M[8, 9] = 4, correspond à un ajout, ici l'ajout d'un "s" ;
- un saut de ligne (de haut en bas), inversement, correspond à une suppression ;
- un saut de ligne et de colonne (case juste en bas à droite), par exemple de M[2, 2] = 0 à M[3, 3] = 1, correspond à une substitution, ici la substitution de "n" en "r".

Le chemin mis en exergue dans la matrice correspond aux opérations proposées intuitivement plus haut. Cependant, d'autres chemins de même distance sont possibles.

Plus longue sous-séquence commune

Le problème de la plus longue sous-séquence commune (*LCS* pour *Longest Common Subsequence*) correspond à la recherche de l'edit script le plus court entre deux chaînes en termes d'ajout et de suppression (Myers, 1986). Contrairement à la distance de Levenshtein, les substitutions ne sont pas considérées ici. Par exemple, la LCS entre les chaînes "panorama" et "paronomase" est "panoma", d'après le résultat de l'algorithme de Myers (*ibid.*) implémenté dans la librairie *google-diff-match-patch* (Fraser) :

- **pano**rama
- **paronoma**se

D'où l'edit script minimal suivant pour transformer la première chaîne en la seconde :

- pa**ro**norama (insertion de "ro")
- paronoma (suppression de "ra")
- paronoma**se** (insertion de "se").

Les diffs permettant de comparer deux fichiers texte (*diff textuel*), c'est-à-dire deux séquences de caractères encodés en ASCII ou en Unicode par exemple, s'appuient généralement sur la recherche de la LCS entre ces deux séquences pour calculer le delta. Dans l'affichage ci-dessous, qui résulte de l'application de *google-diff* (s'appuyant sur l'algorithme de Myers (1986)) à notre exemple, la LCS correspond aux caractères sur fond blanc tandis que les différences sont sur fond coloré. Notons que les fins de ligne sont également indiquées comme caractères, appartenant à la LCS ou non.

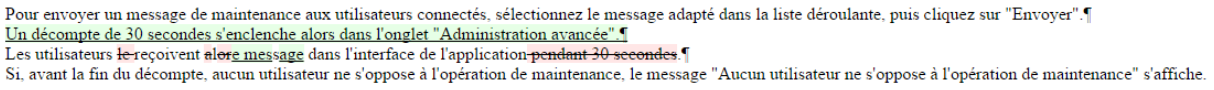


Figure 33 - Résultat de l'algorithme de Myers (google-diff).

Ce résultat est assez proche de la troisième forme différentielle (diff orienté caractères) que nous avions proposée plus haut. Toutefois, concernant la troisième ligne, on remarque un affichage peu lisible à cause de lettres communes entre "alors" et "le message". En effet, l'intérêt d'un edit script *minimal* entre deux fichiers F1 et F2 est avant tout d'avoir de meilleures performances lors de la transformation de F1 en F2 par le patch, plus que de permettre la bonne lisibilité de la forme différentielle. Pour améliorer la *sémantique* des différences, Fraser (2009) propose un post-traitement de l'edit script permettant d'étendre les différences aux "petites parties communes" qu'elles encadrent (principe du *semantic cleanup*). Par exemple, l'application de ce post-traitement avec *google-diff* donne la ligne suivante pour notre exemple :

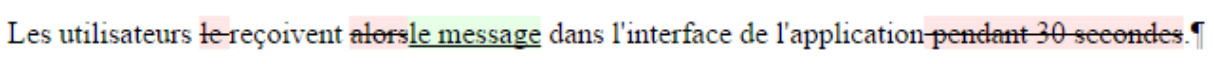


Figure 34 - Résultat du "semantic cleanup" (google-diff).

Le diff orienté lignes, tel que nous l'avons vu plus haut avec la commande Unix *diff* (Hunt&McIlroy, 1976), est un cas particulier de diff textuel dans lequel la LCS est recherchée au niveau des lignes et non au niveau des caractères. En ASCII par exemple, les lignes sont les séquences de caractères délimitées par la combinaison *CR+LF* (caractères spéciaux pour indiquer un retour chariot puis un saut de ligne, unifiés en un seul caractère *CRLF* sous Windows). Dans un diff orienté lignes, chaque ligne est préalablement codée sur un seul nombre par une fonction de hachage. Celle-ci assure que deux lignes identiques seront codées de la même façon, mais deux lignes non-identiques peuvent théoriquement être codées par le même nombre : en effet, la fonction de hachage n'est pas injective. Cependant, plus le nombre d'octets sur lequel est codée la valeur de hachage est grand, moins ces cas de "collision" sont probables. Par ailleurs, des mécanismes existent pour repérer de telles situations (*ibid.*). La LCS est alors recherchée sur les deux séquences de lignes codées, et les différences qui en résultent représentent des ajouts et suppressions de lignes. Sur notre exemple, la fonction de hachage pourrait transformer les deux versions ainsi :

1. 123
2. 1453

D'où la LCS "13" et l'edit script composé de la suppression de "2" et l'insertion de "4" et "5" à la place.

Distance entre deux arbres

Un arbre est une structure de données composée de nœuds ayant les propriétés suivantes :

- chaque nœud est décrit par un label (une chaîne de caractère) ;
- chaque nœud peut contenir plusieurs nœuds enfants ;
- chaque nœud a au plus un parent ;
- un nœud sans enfant est une feuille, ou nœud terminal ;
- le nœud racine est l'ancêtre de tous les nœuds et est le seul à ne pas avoir de parent.

Outre la relation parent/enfant, les nœuds peuvent être frères (même parent), ancêtre/descendant, cousins, oncle/neveu, etc.. De plus, un arbre dans lequel les nœuds enfants se lisent dans un certain ordre (de gauche à droite) est dit *ordonné*.

Les opérations d'édition sur un arbre ne sont pas de même nature que les opérations résultant de la recherche de la LCS de deux chaînes. En effet, elles portent sur les nœuds de l'arbre et prennent en compte les relations hiérarchiques (parent/enfant) entre ces nœuds. La notion de distance entre deux arbres ordonnés a été introduite par Tai (1979), qui redéfinit les opérations d'insertion, de suppression et de substitution dans ce cadre :

- une insertion d'un nœud N1 au niveau d'un nœud N2 correspond au fait que le parent de N2 devienne celui de N1 et que N1 devienne le parent de N2 (et l'ancêtre de tous les nœuds du sous-arbre de N2) ;
- une suppression d'un nœud N correspond au rattachement de ses enfants au parent de N (en respectant l'ordre par rapport aux nœuds frères de N) ;
- une substitution correspond au changement du label d'un nœud.

La LCS peut toujours s'appliquer à une *linéarisation* de l'arbre, mais avec une expression des différences plus pauvre qu'avec les opérations d'édition ci-dessus.

3.1.5 Diff XML

Un document XML est un fichier texte (encodé en Unicode) qui respecte la syntaxe arborescente définie par le standard XML. Cette syntaxe permet la structuration logique du contenu : « un livre est organisé en chapitres, chaque chapitre en sections, sous-sections, paragraphes, etc. » (André *et al.*, 1989). La représentation en mémoire d'un document XML est donc un arbre, dont les nœuds sont de

différents types, parmi lesquels : nœuds document, élément, attribut, texte (ici, nous ne considérerons pas les autres types de nœud tels que les commentaires ou les *processing instructions*). Le standard XML ajoute des contraintes syntaxiques supplémentaires en fonction de ces types :

- un document XML possède un unique nœud document qui est la racine de l'arbre ;
- les éléments sont identifiés par un nom et peuvent contenir des attributs, des nœuds textes et d'autres éléments ;
- un élément terminal est dit élément vide ;
- les attributs sont identifiés par un nom unique au sein de l'élément et ne peuvent contenir qu'un seul nœud texte ;
- les nœuds textes sont des feuilles et ont une valeur (chaîne de caractères) ;
- on peut considérer que le label d'un nœud élément ou attribut correspond à son nom, et que le label d'un nœud texte correspond à sa valeur (ou bien un *hash* de sa valeur, tel que décrit dans (Rönnau *et al.*, 2009) et comme nous l'avons vu plus haut avec le diff orienté lignes).

Un diff XML est donc un cas particulier de calcul de la distance d'édition entre deux arbres. L'ordre des nœuds enfants peut être pris en compte de différentes façons selon le domaine d'application ou d'usage du document XML. Typiquement :

- Dans les langages XML orientés données (sérialisation d'une table de base de données par exemple), l'ordre des nœuds enfants n'est pas signifiant. Par exemple, les entrées de la table *Catalogue* ci-dessous peuvent être sérialisées dans un ordre arbitraire (par un tri alphanumérique sur le numéro, la désignation, le stock...), tous les ordres possibles étant potentiellement signifiants.

```
1 <catalogue>
2     <ouvrage num="1" designation="La raison graphique" auteur="Goody,
  J." stock="5"/>
3     <ouvrage num="2" designation="La technique et le temps, tome 1"
  auteur="Stiegler, B." stock="3"/>
4     <ouvrage num="3" designation="Du mode d'existence des objets
  techniques" auteur="Simondon, G." stock="1"/>
5     ...
6 </catalogue>
```

- Dans les langages XML orientés documents (formats portés avant tout sur la structuration logique, comme DocBook, TEI, DITA... ou également dédiés à la présentation, comme XHTML), l'ordre des nœuds enfants (hors attributs) dans le fichier sérialisé est signifiant. Par exemple dans le fragment XHTML ci-dessous, la balise *h1* est affichée par le navigateur avant la balise *p*, sans quoi le sens n'est pas le même. En revanche, l'inversion des deux attributs *id* et *class* n'aura aucune conséquence visuelle.

```
1 <h1 id="456" class="section_title">Introduction à Oracle</h1>
2 <p>Oracle est le premier SGBDR commercialisé en 1979.</p>
3 ...
```

  C'est dans ce second cas que nous nous situons.

Le delta produit par un diff XML est donc une liste d'opérations concernant les nœuds de l'arbre XML. Il est difficile d'établir un consensus sur un modèle de delta du point de vue des types d'opérations à considérer, qui varient souvent selon l'algorithme, classiquement parmi l'insertion, la suppression, la modification, le déplacement ou encore la copie d'un nœud. En effet, comme le souligne Vion-Dury (2011), la plupart des algorithmes sont optimisés pour une certaine application et/ou une certaine taille de document. Seules les opérations d'insertion et de suppression de nœuds (*avec* leur descendance, éventuellement aucune si le nœud est une feuille) sont communément supportées par les diffs, et ce de manière univoque.

Dans le modèle formel proposé par Vion-Dury (*ibid.*), un delta est composé uniquement de ces deux types d'opérations, *ins* et *del*, qualifiées d'*atomiques*. Chaque nœud de l'arbre XML doit être accessible

via un unique *chemin*. Dans l'exemple suivant (inspiré de (*ibid.*)), les chemins des nœuds éléments *a*, *b*, *c* et *d* sont respectivement 1, 1/1, 1/1/1 et 1/2.

```
<a>
    <b>
        <c/>
    </b>
    <d/>
</a>
```

La suppression de *d* s'écrit *del(1/2)*, tandis que l'ajout de *e* en tant qu'élément frère de b s'écrit *ins(1/1/2, e)*. Comme nous l'avons vu plus haut, les opérations du delta ne peuvent être exécutées dans un ordre quelconque que si elles n'interfèrent pas les unes avec les autres. Dans le cas du diff XML, Vion-Dury formalise cette exigence avec le concept d'*orthogonalité* des chemins. Voici un pseudo-algorithme que nous proposons à partir des propriétés formelles de l'orthogonalité caractérisées dans (*ibid.*) :

```
Si les deux chemins sont égaux ou désignent des nœuds frères (de même nœud parent), ils ne sont pas orthogonaux
 Sinon :
   Si les deux chemins sont de même profondeur, ils sont orthogonaux
   Sinon, on remonte le plus long chemin jusqu'à arriver au niveau du plus court :
     Si le nœud "remonté" précède le nœud désigné par le chemin le plus court, alors les chemins sont orthogonaux
     Sinon, les chemins ne sont pas orthogonaux
```

Dans l'exemple précédant, les chemins 1/1/2 et 1/2 sont orthogonaux car 1/1/2 est plus profond que 1/2 d'une part, et 1/1 (sous-partie de 1/1/2 de même niveau que 1/2) précède bien 1/2 dans l'ordre logique de l'arbre d'autre part. Par ailleurs, 1/2/1 et 1/1 ne sont pas orthogonaux car 1/2 (sous-partie de 1/2/1 de même niveau que 1/1) ne précède pas 1/1 ; 1/1 et 1/2 ne le sont pas non plus car ce sont des nœuds frères. Les opérations *ins(1/1/2, e)* et *del(1/2)* portant sur des chemins orthogonaux, le résultat final ne dépendra pas de leur ordre d'exécution.

Par suite, Vion-Dury redéfinit le delta comme un groupe de transformations, qui sont soit des opérations atomiques, soit des deltas, pouvant être exécutés :

- séquentiellement, c'est-à-dire selon un ordre prescrit ;
- parallèlement, c'est-à-dire que la séquence choisie n'influence pas le résultat final, à condition que toutes les transformations du delta soient orthogonales deux à deux.

Un troisième type de composition, nommé *snapshot*, concerne le cas où deux transformations ont des chemins orthogonaux d'une part, et où le chemin de la première précède le chemin de la seconde dans l'ordre total des nœuds (ordre dans le document XML sérialisé) d'autre part. Par exemple, considérons deux opérations *del* sur les chemins 1/1 et 1/2/1, qui ne sont pas orthogonaux comme nous l'avons vu plus haut. La séquence *del(1/2/1);del(1/1)* peut être considérée comme correcte (1/2/1 ne précède pas 1/1 dans l'ordre des nœuds), la séquence inverse *del(1/1);del(1/2/1)* doit être réécrite à l'aide de la composition snapshot comme *del(1/1);del(1/1/1)*.

3.2 Annotation

3.2.1 Lecture active

Définitions

La lecture active est une lecture qui s'accompagne d'une réflexion critique et d'un apprentissage (Schilit *et al.*, 1998). Ce type de lecture est proche de la lecture savante, qui d'après Gebers est une

« lecture intensive d'un ensemble de documents dont l'objectif est de produire un nouveau contenu qui réifie l'interprétation d'un corpus par un lecteur » (2008) ; ou encore de la lecture critique, qui d'après Bottini consiste à « élaborer de nouvelles configurations documentaires faisant sens à partir notamment des fragments résultant de la discrétisation opérée sur les ressources du corpus » (2010). Ces deux dernières définitions insistent sur l'engagement du lecteur au cours de son activité. En effet pour Stiegler, il s'agit d'« inscrire sa lecture à même le texte lu », à tel point que « lire et écrire deviennent proprement inséparables » (1995). La lecture active se prolonge également avec la production de nouveaux contenus ou documents. Elle réalise alors un but de lecture, au sens de O'Hara (1996) : un texte ou un ensemble de textes est lu en vue de produire un résumé, une revue critique, etc..

La lecture active entraîne la production d'annotations, définies par Prié comme « les inscriptions premières de la lecture, qui correspondent aux notes (au sens large) que le lecteur a souhaité inscrire au cours de celle-ci. » (2011, p. 104). Pour Virbel, elles sont un moyen pour le lecteur de gérer son interprétation à long terme : « [...] l'annotation apparaît comme une technique empirique et individualisée de mémorisation et de la capitalisation de résultats de lecture, dès lors remployables dans la suite même de la lecture en cours, mais aussi pour la gestion de relectures ou de consultations ultérieures, et pour la facilitation de l'accès transversal ou diagonal du texte [...] » (1994). Les annotations constituent ainsi un *espace rétentionnel virtuel*, au sens de Stiegler (1995), virtuel en ce que cet espace est externe à la mémoire biologique. Pour ce même auteur, le numérique ouvre la voie vers l'automatisation de l'exploitation des annotations, qui sont effectivement des inscriptions manipulables comme les autres au sein du système numérique dit *global* (Prié, 2011).

L'instrumentation numérique de la lecture active fait l'objet de plusieurs recherches, concernant les documents textuels (Gebers, 2008), audiovisuels (Aubert et Prié, 2005 ; Richard *et al.*, 2007 ; Prié, 2011) ou encore multimédia (Bottini, 2010).

Par exemple dans le cas de l'instrumentation de la lecture savante, Gebers (2008) a proposé un prototype d'environnement où le lecteur constitue un dossier documentaire et créé des annotations textuelles multi-ancres et multi-cibles entre les contenus du dossier. À travers les différents parcours proposés par l'environnement (linéarisation du dossier, vue réseau, accès transversal aux annotations), le lecteur peut (ré)organiser sa lecture du dossier.

Dans le cadre du projet Advene (Aubert et Prié, 2005), c'est la lecture active de documents audiovisuels qui a été étudiée. Les documents audiovisuels ont la particularité de prescrire un rythme de lecture par leur temporalité intrinsèque. L'annotation dans de tels documents prend la forme d'une « information attachée à un fragment spatio-temporel d'une vidéo » (Prié, 2011, p. 121). À partir des annotations, l'environnement permet de produire une hypervidéo, soit un ensemble de vues (table des matières, mosaïque d'images, mise en relation de fragments...) exploitant les annotations et les reliant aux fragments concernés. Le lecteur peut alors naviguer entre ces fragments mais aussi les jouer, marquant ainsi un retour au rythme du flux temporel.

Enfin, Bottini (2010) propose un environnement de lecture critique multimédia reposant sur un modèle conceptuel prenant en compte l'hétérogénéité des contenus manipulés (texte, image, son...). Ce modèle repose sur trois niveaux : matériel, annotatif et organisationnel. Le niveau matériel correspond aux différents contenus pouvant être importés dans l'environnement (entités matérielles) et à partir desquels peuvent être opérées des sélections. La sélection est définie en fonction du type de contenu : par exemple, une sélection dans un contenu sonore est décrite par des *timecodes* de début et de fin ; une sélection dans une image est décrite par une zone géométrique. Le lecteur peut associer ces sélections à des entités sémantiques au niveau annotatif, qui permettent d'ajouter des gloses textuelles ou bien de créer des liens vers d'autres entités sémantiques. Enfin au niveau organisationnel, le lecteur peut découper une entité matérielle dans une arborescence d'entités structurelles. Par exemple, un flux audio peut être subdivisé sur autant de niveaux que le lecteur le souhaite ; une partition musicale peut être décomposée en pages puis en systèmes (ensemble de portées). À l'issue de ces phases de sémantisation et d'organisation, le lecteur peut par exemple produire le dossier synchronisé d'une œuvre, à partir de plusieurs de ses interprétations (entités sonores) et partitions (entités graphiques).

Ces instrumentations de la lecture active ont pour origine les travaux des pionniers de l'informatique

documentaire tels que Nelson et, quelques dizaines d'années avant cela, Vannevar Bush. Dans son célèbre article *As we may think* (1945), Bush avait décrit une machine, le Memex, stockant sur des microfilms une grande quantité de documents que l'utilisateur peut lire, commenter, lier et comparer entre eux. D'après Crozat (2015a), l'idée de Bush était de pouvoir *mécaniser* les actes de pensée répétitifs (et donc répétables par la machine) qui ont lieu au cours de la lecture et de l'écriture (rechercher un texte, le manipuler, suivre une référence vers un autre texte...). La pensée de Bush fut à la base du concept d'hypertexte, théorisé pour la première fois par Nelson en 1965. Dans *Literary Machines* (1981), Nelson défend une vision non-linéaire du texte, censée correspondre au mode réticulaire de la pensée humaine (par association d'idées). Au contraire, la linéarité traditionnelle du texte aurait tendance à imposer une seule interprétation, là où l'hypertexte donne la possibilité au lecteur de construire ses propres parcours, ou bien à l'auteur d'en prévoir plusieurs.

Critique de la lecture numérique

Nous voyons que le numérique permet d'instrumenter la lecture active à travers des environnements de lecture proposant des fonctions avancées d'annotation et d'exploitation des annotations.

Pour Giffard, il en va autrement des pratiques usuelles de lecture numérique, c'est-à-dire de la lecture sur le web : « la lecture numérique existe [...] en tant que pratique culturelle, mais elle ne remplit pas les conditions nécessaires d'une lecture générique parce qu'elle n'arrive pas à intégrer la lecture approfondie, attentive, associée à la réflexion » (2013). Giffard rappelle en effet que la réflexion occupe une place fondamentale, bien que distincte de la lecture elle-même, dans l'appropriation d'un texte. Elle est à la base de la tradition de lecture qui s'est développée dans les monastères au Moyen-âge, où la lecture silencieuse et intensive (la lecture et relecture d'un ensemble de textes) s'entrecoupait de moments de méditation, afin de favoriser une intériorisation du texte.

Le web transforme la lecture approfondie, ou lecture d'étude, en une simple lecture d'information non-suivie par la réflexion. Giffard s'appuie sur l'analyse de Nicholas Carr et la notion de surcharge cognitive mise en évidence par les études sur la lecture sur le web. En plus des problèmes de lisibilité des textes à l'écran, on retrouve la désorientation induite par les liens hypertextes : « La prise en compte des hyperliens à l'intérieur des textes et des sites est un bon exemple de surcharge cognitive dans le temps. Tout en lisant, le cerveau doit considérer l'intérêt éventuel des hyperliens et prendre la décision de les activer (ou pas). » L'attention devant être portée au texte et la prolongation de la lecture par la réflexion sont donc en quelque sorte détournées par l'outil. En cela, le web n'est pas une technologie de lecture selon Giffard, mais une *technologie par défaut*.

Inventée au tournant des années 1990 par Tim Berners-Lee, cette technologie s'inspire des travaux de Bush et de Nelson. S'il s'agit bien d'un système hypertexte (mais dont les liens sont seulement unidirectionnels), Giffard rappelle que le web ne propose pas, du moins en dehors de plate-formes spécialisées, de mécanismes standards permettant au lecteur de créer ses propre liens entre documents ou fragments de documents ou encore d'annoter les contenus. Ainsi pour Giffard : « la possibilité pour le lecteur de produire ses propres parcours de lecture dans le texte numérique, centrale dans l'orientation hypertextuelle n'a pas été actualisée dans le dispositif du web ».

3.2.2 Documentarisation et collaboration

Définitions

La notion de *documentarisation* est tout d'abord employée chez le collectif Pédauque (2006), qui désigne par là « l'omniprésence documentaire dans l'organisation sociale moderne » (Crozat, 2012a).

Chez Zacklad (2005), la documentarisation revêt un sens plus précis : « [La documentarisation] consiste à doter [les] supports d'attributs spécifiques permettant de faciliter (i) leur gestion parmi d'autres supports, (ii) leur manipulation physique, condition d'une navigation sémantique à l'intérieur du contenu sémiotique et enfin, (iii) l'orientation des récepteurs [...] » (*ibid.*). Les documents sont ici étudiés sous l'angle des *transactions communicationnelles*, c'est-à-dire comme « supports à la coordination d'un collectif distribué engagé dans une activité commune finalisée » (*ibid.*). Dans ce contexte, une nouvelle classe documentaire émerge : celle des documents pour l'action - DopA (*ibid.*).

Le DopA se caractérise notamment par son soutien aux activités de conception coopérative et son inachèvement prolongé. Ce concept permet de caractériser des objets documentaires nouveaux, comme les forums ou les groupes de discussion.

Dans le cadre du DopA, Zacklad définit l'annotation de la manière suivante : « toute forme d'ajout visant à enrichir une inscription ou un enregistrement pour attirer l'attention du récepteur sur un passage ou pour compléter le contenu sémiotique par la mise en relation avec d'autres contenus sémiotiques pré-existants ou par une contribution originale » (2007). Il propose une typologie d'annotation dans laquelle il distingue (*ibid.*) :

- l'*annotation-attentionnelle*, qui correspond à une simple indication de lecture (passage surligné...) ;
- l'*annotation-associative*, qui met en jeu le renvoi vers un autre fragment interne ou externe au document (référence bibliographique, lien hypertexte...) ;
- l'*annotation-contributive*, qui se distingue par un une production sémiotique (texte, image...) venant se surajouter au fragment annoté.

Dans cette typologie, les annotations sont réalisées à l'intention d'un bénéficiaire potentiellement (mais la plupart du temps) différent du lecteur-annotateur. Zacklad distingue la prise de notes comme un cas particulier dans lequel le lecteur est lui-même le bénéficiaire de ses annotations, tel que dans la lecture active le plus souvent.

Annotations-contributives

Dans le cadre de l'activité de relecture, nous nous intéressons principalement aux annotations-contributives réalisées par le relecteur et adressées au rédacteur (pouvant être le relecteur de son propre texte). Celles-ci jouent plus spécifiquement un rôle de *critique* du contenu ou de *planification* d'une action à entreprendre sur le contenu (Zacklad *et al.*, 2003). De plus, leur fonction n'est pas la même selon qu'elles concernent un document inachevé ou achevé (Zacklad, 2007) :

- Dans le premier cas, on se situe plutôt dans un contexte de correction (le document est voué à être modifié après la phase d'annotation). Le relecteur évalue le degré d'achèvement du document et propose des pistes de réécriture via les annotations. Cette activité est par exemple instrumentée dans les logiciels bureautiques tels que MS Word ou OpenOffice, avec le *mode révision* : « [...] les éditeurs de texte électroniques permettent d'introduire des annotations-contributives à la fois sous la forme de commentaires et sous la forme de corrections ou de compléments au texte initial qui apparaissent dans des couleurs différentes et qui peuvent être progressivement acceptés ou rejetés par les co-transactants lors de la réception du document ».
- Dans le second cas, « l'annotation s'inscrit [...] dans un processus de "reprise", dans lequel le document [...] fait l'objet d'un processus de "ré-appropriation" et de "re-documentarisation" ». Le relecteur cherche à actualiser le contenu du document (« [formulation] des objections associées à la prise en compte de nouvelles informations [...] ») et ses annotations mènent éventuellement à un « projet de "ré-édition" ou de réutilisation du document dans un contexte différent ».

Compte tenu des propriétés intrinsèques du document numérique (réécriture permanente, instabilité...), il ne nous parait pas nécessaire de maintenir la distinction entre document achevé et inachevé.

Fragments pragmatiques

Dans le cadre des chaînes éditoriales, Arribe définit la *documentarisation de l'activité* comme « la production d'une documentation liée à l'activité des rédacteurs sur un graphe de fragments » (2014). La documentarisation entraîne la production de nouveaux fragments appelés *fragments pour l'action*, par opposition aux autres fragments du graphe à vocation documentaire. La tâche et la liste de tâches sont des exemples de fragments pour l'action. Une tâche décrit typiquement une action à entreprendre sur un ou plusieurs fragments documentaires (mise à jour, relecture...).

Arribe souligne cependant que la dichotomie entre fragments pour l'action et fragments documentaires s'avère être une limite dans certains usages. Il existe en effet des situations hybrides dans lesquelles des informations du support à l'action ont vocation à être réutilisées dans un contexte de production documentaire et inversement, comme illustré par les dossiers d'homologation rédigés avec la chaîne éditoriale utilisée au sein de la société Quick :

- ils comportent des métadonnées renseignant l'auteur et la date de révision ;
- leur validation peut servir d'*input* à la création de tâches de mise à jour des fragments de la documentation de référence.

Pour dépasser cette limite, Arribe propose le concept de *fragment pragmatique*. Un fragment pragmatique peut incorporer des structures relevant aussi bien de la production documentaire que du support à l'action, et se positionne entre ces deux dimensions selon leur importance relative. Trois exemples de fragments pragmatiques dans les chaînes éditoriales sont proposés, avec leur instrumentation dans Scenari : le moteur de tâches, le système de commentaires et l'enrichissement de fragments documentaires.

Le moteur de tâches consiste en un panneau de gestion des tâches de l'utilisateur. Une tâche est répertoriée selon son état ("à venir", "en cours" ou "close") et peut avoir des fragments documentaires associés ainsi que des dates de planification (à laquelle elle passe de l'état "à venir" à l'état "en cours") et d'échéance. La tâche est également décrite par un cycle de vie, soit un ensemble de statuts et de transitions entre ces statuts. Le cycle de vie est modélisé dans SCENARIbuilder (primitives de collaboration). Par exemple, une tâche de type "revue" peut avoir les statuts "à valider" (un relecteur demande la validation d'un fragment par un relecteur), "à modifier" (le relecteur demande des corrections au rédacteur, qui repassera ensuite la tâche au statut "à valider"...) et "terminé" (le relecteur valide le fragment et clôt la tâche).

Le système de commentaires permet d'intégrer la dimension pragmatique aux fragments documentaires classiques via l'annotation. Les commentaires peuvent être ajoutés à tout niveau du contenu d'un fragment, sous la forme de fils de discussion auxquels il est possible de répondre par d'autres commentaires, et pouvant aussi être clos ou supprimés. L'annotation peut être réalisée dans l'éditeur Scenari (p. 15) ou bien dans une prévisualisation dédiée (p. 15).

L'enrichissement des fragments documentaires consiste à utiliser des informations relatives au support à l'action non pas dans une partie dédiée de l'application, le moteur de tâches, mais directement au sein des fragments en vue de leur exploitation dans la production documentaire (par exemple, les métadonnées des dossiers d'homologation chez Quick). Cet enrichissement peut se faire en associant un cycle de vie (modélisé dans SCENARIbuilder) à une ou plusieurs classes de fragments. Par exemple, le cycle de vie est constitué des états "brouillon", "à valider" et "validé". Les droits d'écriture peuvent varier en fonction de l'état du fragment et du rôle de l'utilisateur : par exemple, un fragment dans l'état "à valider" est en lecture seule pour les rédacteurs tant que son état n'est pas revenu à "brouillon". De plus, la responsabilité d'un fragment peut être confiée à un utilisateur. Les informations de l'état et du responsable du fragment sont non-seulement visibles dans la forme d'édition (dans le bandeau de gestion), mais également exploitées par le moteur de recherche des fragments (l'utilisateur peut par exemple chercher tous les fragments validés).

3.3 Correction automatique

3.3.1 Approches

Le principe d'un correcteur automatique est d'analyser un texte pour y rechercher d'éventuelles erreurs (orthographiques, grammaticales...) et proposer pour chaque erreur une ou plusieurs corrections à l'utilisateur. Nous présenterons ce domaine en nous appuyant sur les travaux de Kukich (1992), qui identifie trois sous-problèmes de recherche : la détection d'un non-mot (*non-word error* en anglais), qui repose sur un dictionnaire de mots autorisés ; la correction d'un mot isolé, qui consiste à chercher dans

le dictionnaire les mots ayant une faible distance d'édition par rapport au mot à corriger ; la détection et correction d'erreurs dépendant du contexte, qui permettent de gérer les fautes non-détectées (*real-word errors*). On distingue ainsi trois approches suivies par les correcteurs automatiques : les approches lexicales, syntaxiques et probabilistes.

Nous terminerons ce panorama en évoquant des recherches plus récentes portant sur l'analyse automatique du discours permettant de détecter des formulations inappropriées.

Approche lexicale

Dictionnaire

Pour chaque mot du document, le dictionnaire est parcouru pour vérifier si ce mot y figure. Dans le cas où il n'y figure pas, le mot est détecté comme un non-mot et sera sujet à une correction. Le parcours du dictionnaire pose souvent un problème de performance à cause de la taille de ce dernier (plusieurs dizaines de milliers de mots). Différentes techniques permettent de gagner en efficacité.

Par exemple, une table de hachage consiste à indexer les mots du dictionnaire par un code permettant d'y accéder directement, évitant ainsi le parcours complet du dictionnaire. Lorsqu'un mot est testé, son code est calculé afin de pouvoir le comparer au mot indexé par ce code, s'il existe. L'inconvénient de cette méthode est que la fonction de hachage utilisée pour calculer les codes peut entraîner des collisions, c'est-à-dire que deux mots différents peuvent avoir le même code (la fonction de hachage n'est pas injective).

Il est également possible de diviser (partitionner) le dictionnaire sur différents niveaux en fonction de la fréquence d'apparition des mots dans le document, ou bien dans la langue en général : si un mot n'est pas trouvé au premier niveau, on le cherche au second, et ainsi de suite.

Enfin, les différentes déclinaisons morphologiques d'un mot (pluriel, formes conjuguées...) peuvent être "factorisées" sous la forme d'un radical (exemple : "décis" pour "décisif", "décision"...) pour réduire la taille du dictionnaire. Les déclinaisons doivent ensuite pouvoir être reconstruites à partir du radical.

Distance d'édition

La recherche de corrections potentielles (candidats) d'un non-mot repose sur le calcul de sa distance minimale d'édition avec les mots du dictionnaire. Nous avons abordé ce sujet dans la section concernant le différentiel (p. 44), notamment à travers l'algorithme de Levenshtein, qui permet de calculer la distance minimale de deux chaînes en termes d'insertion, de suppression et de substitution de caractères. Dans une étude sur les types de fautes d'orthographe, Damerau (1964) observe que ces trois opérations, auxquelles il ajoute également la transposition (inversion de deux caractères adjacents dans un mot, par exemple : "attendre" et "attender"), sont à l'origine de 80% des fautes humaines. La distance de Damerau-Levenshtein, qui peut être calculée par un algorithme similaire à celui de Levenshtein, incluant en plus le calcul des transpositions, est ainsi une métrique de base pour la plupart des techniques de correction (Kukich, 1992). Les candidats proposés à l'utilisateur sont les mots dont la distance avec le non-mot est inférieure à un certain seuil, classés dans l'ordre croissant des distances. Dans certaines applications, il existe aussi un mode d'auto-correction, où c'est généralement le candidat ayant la plus faible distance qui est sélectionné.

Faux positifs et faux négatifs

Un faux positif est une erreur humaine non-détectée par le correcteur (*real-word errors*). Inversement, un faux négatif est un mot détecté par le correcteur comme fautif alors qu'il ne l'est pas du point de vue de l'humain. Trop étroit, un dictionnaire entraîne des faux négatifs alors que trop large, il entraîne des faux positifs (Kukich, 1992). Les correcteurs modernes utilisant des dictionnaires de grande taille (à titre d'exemple, le dictionnaire de ProLexis contient plus de 700 000 flexions de mots), le problème porte principalement sur les faux positifs. Ces erreurs constituent 25 à 50% des erreurs totales (*ibid.*). En voici quelques exemples typiques :

- Erreurs grammaticales : mauvais accord d'un verbe, utilisation d'un infinitif au lieu du participe

passé pour un verbe du premier groupe... en effet, toutes les formes (conjuguées, infinitive...) sont présentes dans le dictionnaire, telles quelles ou bien sous la forme d'un radical pouvant être décliné.

- Erreurs de sens :
  - Homonymes ("chat" à la place de "chah" dans "le chah d'Iran")
  - Typographie/coquilles ("poison" et "poisson", "trier" et "tirer"...)
  - Mauvaise délimitation entre deux mots ("puis que" et "puisque")
- Erreurs de syntaxe :
  - Répétition d'un mot par inadvertance (doublon), comme dans "erreurs de de syntaxe".

    Inversement, un mot peut être oublié (bourdon), comme dans "erreurs syntaxe", bien qu'il ne s'agisse pas là d'une erreur due à un faux négatif.
  - Les erreurs de sens peuvent dans certains cas être considérées comme des erreurs de syntaxe, typiquement lorsque qu'il ne s'agit pas de la même fonction syntaxique ("avions" en tant que verbe conjugué ou nom au pluriel) dans un cas et dans l'autre.

Hirst et Budanitsky (2005) soulignent que l'auto-correction peut elle-même introduire des faux positifs : typiquement, un mot mal orthographié est remplacé par sa correction la plus probable du point de vue de la distance d'édition (exemple en anglais : "eyt" est remplacé par "yet"), mais ne correspondant pas à ce que voulait dire l'auteur ("eye").

Approche syntaxique

Certaines erreurs dépendant du contexte peuvent être détectées grâce à une analyse syntaxique de la phrase. Les algorithmes permettant une telle analyse relèvent du domaine du traitement automatique du langage naturel (TALN). D'après Kukich (1992), un système de TALN est composé de deux entités :

- Un lexique, où chaque mot est décrit par les différentes informations morphosyntaxiques le concernant (partie du discours, genre, nombre, etc.). Un mot peut appartenir à différentes catégories morphosyntaxiques, par exemple "porte" est à la fois un nom féminin singulier et un verbe conjugué à la première et troisième personne du singulier.
- Une grammaire, qui définit les règles d'agencement des catégories morphosyntaxiques en syntagmes et des syntagmes en d'autres syntagmes plus complexes. Par exemple, un syntagme nominal peut être composé d'un déterminant suivi d'un nom du même genre et du même nombre ("la porte") ; un syntagme verbal consiste en un verbe éventuellement suivi d'un syntagme nominal jouant le rôle de complément d'objet direct ("ferme la porte") ; etc..

Pour vérifier la validité syntaxique d'une phrase, l'analyseur tente de la décomposer (*parsing*) en syntagmes pouvant être agencés selon les règles de la grammaire. Par exemple, la phrase "le pilote ferme la porte" est valide car elle peut se décomposer de la manière suivante :

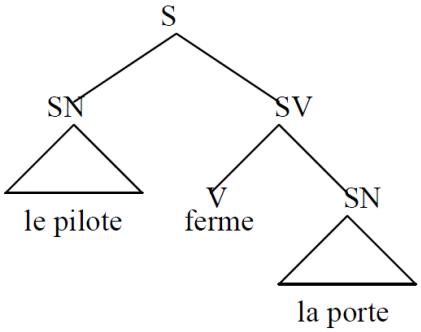


Figure 35 - Première décomposition (source : http://lecomte.al.free.fr/ressources/PARIS8_LSL/Cours1.pdf).

Notons par ailleurs que cette phrase est un exemple d'ambiguïté. En effet, "ferme" peut être vu

comme un adjectif, "la" comme un pronom et "porte" comme un verbe :

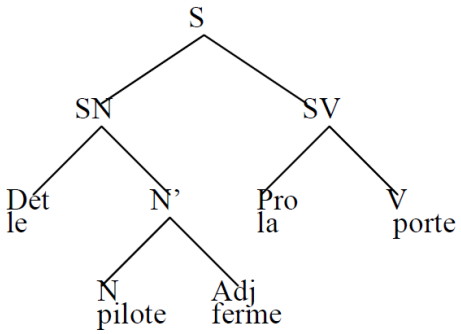


Figure 36 - Seconde décomposition (source : http://lecomte.al.free.fr/ressources/PARIS8_LSL/Cours1.pdf)

.

L'ambiguïté n'est pas un problème *in fine* pour la vérification syntaxique des phrases : en effet, le fait qu'une phrase admette plusieurs décompositions ne change rien à sa validité. En revanche, cela entraîne la nécessité pour l'analyseur syntaxique de construire plusieurs décompositions en parallèle, et ce jusqu'à ce que l'ambiguïté de la phrase soit (éventuellement) résolue (Kukich, 1992). En théorie, une phrase qui ne peut pas être décomposée avec succès par l'analyseur syntaxique comporte une ou plusieurs erreurs qui pourront être détectées voire même corrigées dans certains cas : mauvais accord d'un verbe, répétition d'un mot, confusion entre deux homonymes correspondant à des parties du discours différentes... En pratique cependant, il est difficile de désigner précisément quelle règle syntaxique a causé l'échec de la décomposition, en ajoutant à cela le fait que la grammaire est souvent incomplète (*ibid.*).

Approche probabiliste

N-grammes

L'approche probabiliste s'appuie sur les fréquences d'apparition de *N-grammes* dans un corpus. Soit par exemple le corpus très simple "de la rigueur de la science". Le tableau suivant donne la fréquence des différents unigrammes, autrement dit des mots rencontrés dans ce corpus :

Unigramme	Fréquence
de	2
la	2
rigueur	1
science	1

Tableau 3 - Fréquence des unigrammes.

Un bigramme est une suite de deux mots. La matrice suivante donne la fréquence des différents bigrammes possibles dans le corpus :

	de	la	rigueur	science
de	0	2	0	0
la	0	0	1	1
rigueur	1	0	0	0
science	0	0	0	0

Tableau 4 - Fréquence des bigrammes.

À partir de ces fréquences, on peut établir une probabilité d'apparence d'un mot en fonction du mot qui le précède. Par exemple :

- la probabilité que "de" soit suivi de "la" est de 1 ;
- la probabilité que "la" soit suivi de "rigueur" est de 0,5 ;
- la probabilité que "la" soit suivi de "science" est de 0,5 ; etc.

Pour plus de précision, il est également possible de calculer la fréquence des trigrammes (suite de trois mots) et d'en déduire la probabilité d'apparence d'un mot en fonction des deux mots qui le précèdent. Par exemple, les probabilités que "de la" soit suivi de "rigueur" ou "science" sont toutes les deux de 0,5.

Calculée sur un corpus comportant plusieurs dizaines de milliers de mots distincts, une matrice de fréquence de bigrammes ou de trigrammes peut être utilisée comme modèle statistique de la langue par un correcteur automatique pour détecter et éventuellement corriger les faux positifs.

Mays *et al.* (1991) ont expérimenté l'approche probabiliste à partir d'une matrice de trigrammes basée sur un lexique de 20 000 mots. Leur test est basé sur 100 phrases correctes utilisant les 20 000 mots du lexique. Pour chaque phrase correcte, un ensemble de nouvelles phrases est généré en dérivant cette phrase autant de fois qu'il qu'il est possible de remplacer un de ses mots par un faux positif (chaque phrase dérivée ne continent qu'un seul faux positif). Pour chaque ensemble ainsi constitué, la probabilité d'une phrase est calculée à partir des probabilités des différentes trigrammes qu'elle comporte (pour plus de détails sur le calcul de cette probabilité, voir (Kukich, 1992) ou (Wilcox-O'Hearn *et al.*, 2008)). Pour qu'une phrase soit détectée comme fautive, il faut que sa probabilité ne soit pas la plus forte dans son ensemble ; pour qu'elle puisse être corrigée, il faut que la phrase correcte ait la probabilité la plus forte dans son ensemble. Sur cette expérience, Mays *et al.* obtiennent des scores de 76% pour la détection et 47% pour la correction.

Analyse automatique du discours

Des recherches récentes dans le cadre du projet *Lelie* (Barcellini *et al.*, 2012 ; Kang et Saint-Dizier, 2013 ; 2015) se sont intéressées à l'analyse du discours dans des domaines ciblés tels que la rédaction de procédures (documents techniques) ou d'exigences (cahiers des charges, spécifications, etc.). En effet, « [ces documents] forment un genre conceptuel particulier qui suit des contraintes linguistiques fortes en termes de choix lexical (...), de syntaxe (...), typographique, de style et de contraintes métier variées » (Kang et Saint-Dizier, 2015). Typiquement, les termes flous ou ambigus doivent être évités, de même que les doubles négations (par exemple : "cette manipulation *n'est pas sans* danger") ou encore l'utilisation du passif, les phrases trop longues, etc.. Les auteurs soulignent que le respect de ces principes est souvent contraignant pour les rédacteurs techniques, et occasionne une relecture fastidieuse (*ibid.*). Pour faciliter cette relecture, ils proposent un système d'alertes basé sur une analyse automatique du discours. Cette analyse s'appuie notamment sur la théorie des structures rhétoriques, et est instrumentée via deux briques technologiques :

- *Dislog* (Saint-Dizier, 2012), un langage combinant la programmation logique et les expressions régulières, et permettant de formaliser un ensemble de règles via des *templates*.

- Le méta-interpréteur *<TextCoop>* (*ibid.*), parcourant le texte et confrontant les phrases aux différents templates.

Un template décrit une structure discursive valide ou non-valide. Dans le second cas, le template comporte une règle de réécriture insérant l'alerte (balise XML) qui sera signalée au rédacteur. Un exemple simplifié de template est donné dans (Kang et Saint-Dizier, 2015) :

```
1 [pro(it), modal, aux(be), gap, verb(past participle)]  [pro(it), modal,
  <passif>, aux(be), gap, verb(past participle), </passif>], {diff(verb,[liste
  de verbes non concernés par l'alerte])}.
```

Ce template permet de détecter l'utilisation du passif dans les phrases du type "it shall be demonstrated..." (partie gauche) et de réécrire la phrase détectée avec l'alerte (partie droite). La partie entre accolades est un filtre permettant d'exempter les phrases avec certains verbes : par exemple, "it must be done" ne générera pas d'alerte si le verbe "do" appartient au filtre.

Ces recherches ont été appliquées au niveau industriel avec le logiciel *Lelie for requirements*. Ce logiciel peut s'intégrer à MS Word mais également à une chaîne éditoriale Scenari.

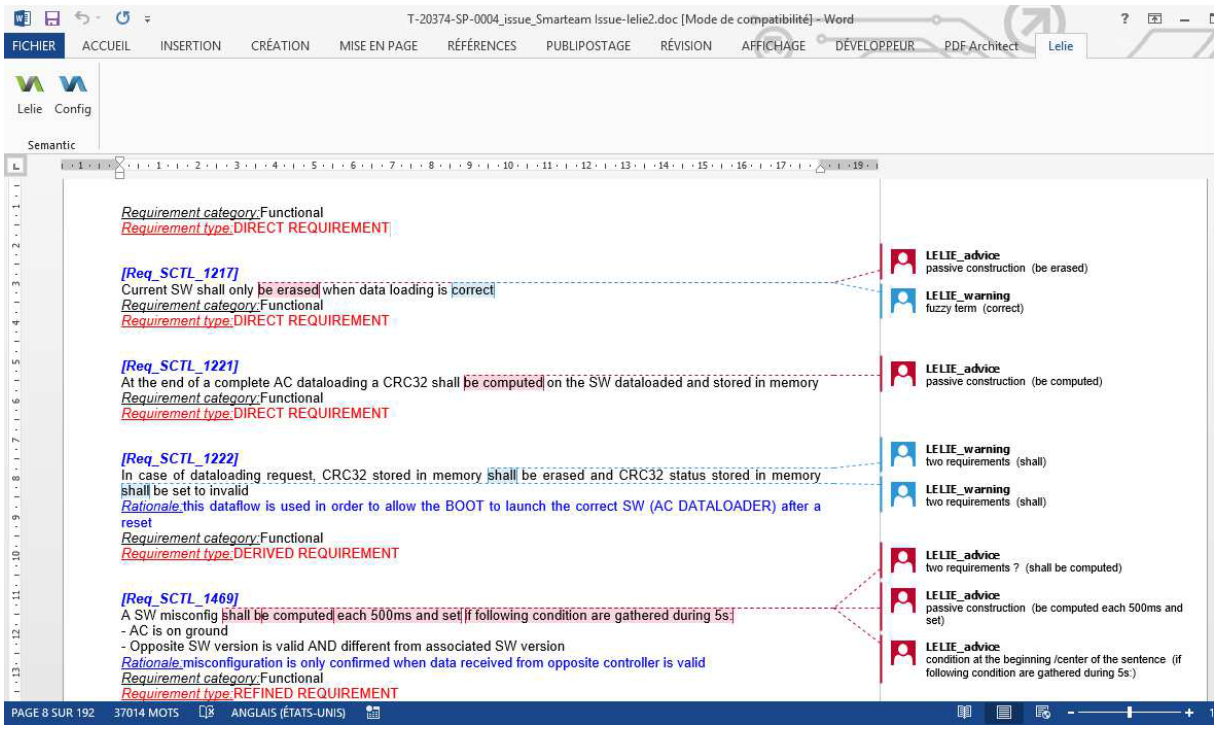


Figure 37 - Résultats de Lelie sur un document Word (Kang et Saint-Dizier, 2015).

3.3.2 Correcteurs automatiques et relecture de forme

La relecture de forme, telle que nous l'avons vue en révision professionnelle (p. 23), peut paraître en contradiction avec l'usage aujourd'hui largement répandu des correcteurs automatiques, dont les capacités de détection et de correction d'erreurs sont croissantes. À titre d'exemple, voici le genre d'erreurs que ProLexis est capable de traiter :

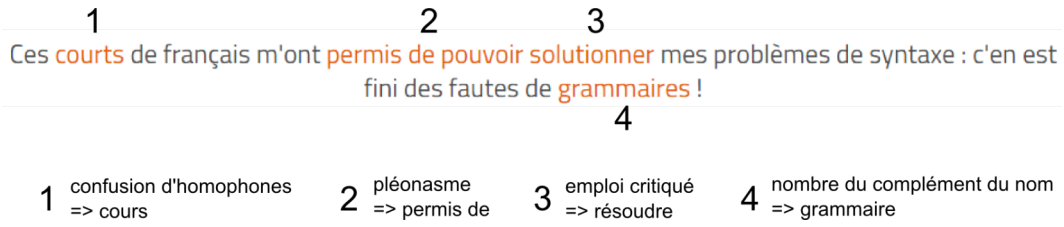


Figure 38 - source : http://www.prolexis.com/correction-professionnelle.

Avant même l'arrivée de ces logiciels de dernière génération, la profession des correcteurs s'inquiétait de voir ce métier disparaître (voir par exemple (Brissaud, 1998)). Loin d'imaginer que le progrès technologique permette à terme le remplacement définitif du correcteur humain, nous pensons que chaque étape de ce progrès peut être vue comme une occasion de réaffirmer la valeur ajoutée de sa relecture compte tenu de ce que le correcteur automatique ne peut toujours pas traiter. Ainsi, la relecture humaine est toujours nécessaire pour certains faux positifs (mots ambigus typiquement) qui ne sont pas détectés avec les correcteurs standards, signe que les recherches dans ce domaine ne sont pas encore arrivées à une maturité suffisante pour faire l'objet d'une industrialisation (excepté dans certains logiciels de pointe tel que ProLexis). De plus, il nous paraît essentiel qu'un correcteur humain puisse conserver une posture critique vis-à-vis des corrections proposées par un logiciel, y compris les corrections très détaillées de l'illustration ci-dessus : par exemple, si le mot "solutionner" est indiqué comme "emploi critiqué" (d'après le Wiktionnaire, ce verbe est rejeté par de nombreux professeurs de français), il appartient au relecteur de juger si cet emploi est légitime dans le contexte éditorial visé. Enfin, la compétence d'un relecteur en matière d'ajustements stylistiques (homogénéisation, allègement...) nous semble encore supérieure à ce qui peut être traité grâce à l'analyse automatique du discours, dont les applications industrielles sont pour le moment confinées à des domaines bien précis (rédaction de cahiers des charges, de manuels techniques...).

Afin de mieux outiller le relecteur dans la recherche d'erreurs non-détectées par un correcteur automatique, une solution pourrait être de proposer un mode d'affichage aléatoire des phrases du document. Cette solution s'inspire de la technique dite de relecture "à l'envers" évoquée plus haut, qui permet de se détacher du fond pour ne se concentrer que sur la forme. De la même manière, cet affichage aléatoire permettrait de traiter les phrases hors de leur contexte et sans logique d'enchaînement entre elles :

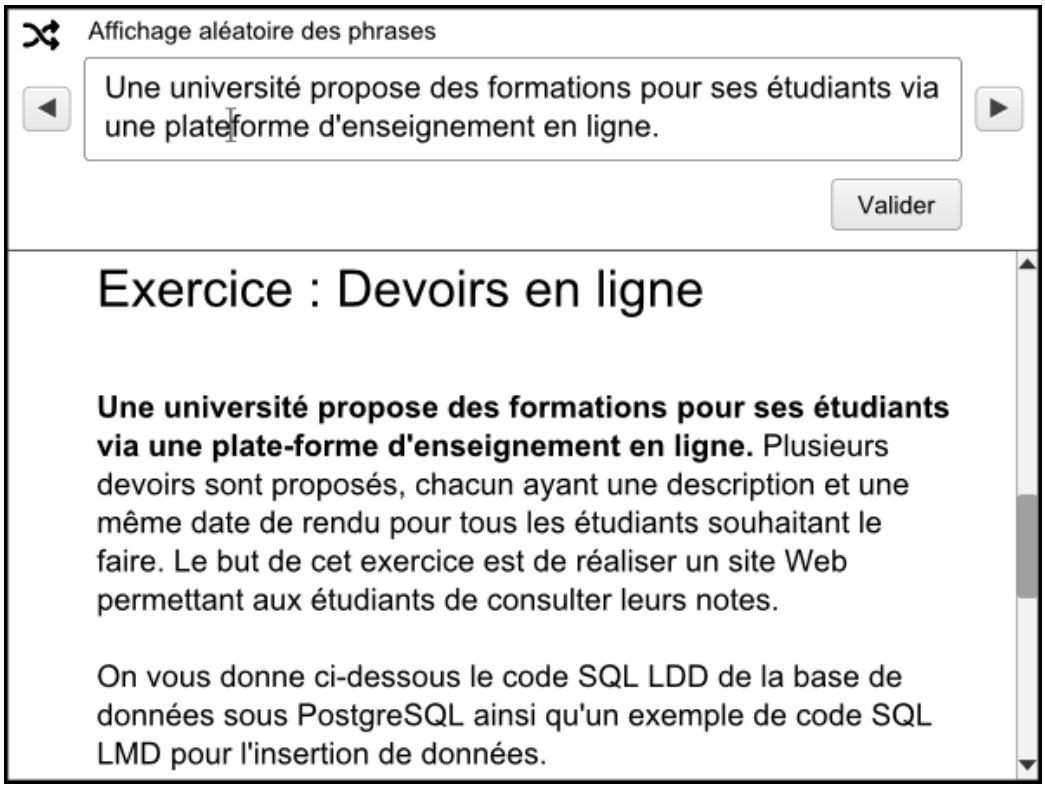


Figure 39 - Maquette d'un mode d'affichage aléatoire des phrases pour la relecture de forme.

Dans la maquette ci-dessus, le relecteur peut passer les phrases une à une à l'aide de boutons précédant/suivant, proposer une correction de la phrase en éditant le champ (par exemple, changer "plate-forme" en "plateforme"), et si besoin visualiser la phrase dans son contexte (phrase mise en surbrillance dans le texte). Notons que la correction pourrait aussi être proposée dans une annotation (et non directement intégrée au contenu), typiquement dans le cas où elle doit être validée par un autre

auteur.

Pour terminer ce panorama sur la question de la relecture de forme, citons brièvement le cas des documents numérisés par des techniques d'OCR (*optical character recognition*). Ces documents peuvent contenir plus ou moins de fautes en fonction de la qualité de la reconnaissance des caractères par la machine et/ou de la qualité des documents préservés. Kukich (1992) souligne que les erreurs générées par la numérisation ne suivent pas les mêmes schémas que les erreurs humaines (par exemple, des lettres telles que "o" et "d" sont confondues à cause de leur forme similaire dans certaines écritures graphiques). Si certaines erreurs peuvent être traitées facilement à l'aide d'une approche probabiliste (typiquement, un trigramme de lettres dont la probabilité est nulle), d'autres sont plus inattendues et varient significativement en fonction du document, ce qui constitue une limite à une correction automatique complète.

Dans le cadre du projet *Ozalid*, la plateforme *Correct* (Josse, 2014) a été développée pour expérimenter la correction collaborative des livres numérisés de la bibliothèque Gallica (BNF). L'interface de correction propose notamment un mode "ligne à ligne" pour une comparaison fine du texte obtenu par OCR avec l'image originale :

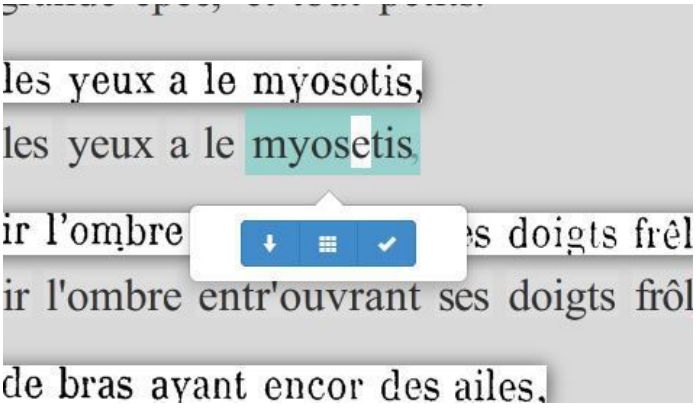


Figure 40 - Correct - mode ligne à ligne (source : http://blog.bnf.fr/uploads/gallica/2014/11/IllustrationCorrect.jpg).

La vérification des corrections effectuées sur cette plateforme constitue une nouvelle problématique de relecture : typiquement, un utilisateur peut avoir corrigé un document différemment qu'un autre utilisateur, ou bien avoir corrigé une erreur qui n'en était pas réellement une (lorsque le livre a été écrit dans une vieille orthographe par exemple...), etc.. Un protocole de qualification automatique de ces corrections collaboratives, basé sur des indicateurs sémiotiques et sémantiques, a été proposé dans (Lagarrigue *et al.*, 2014).

Chapitre 4

Propositions

4.1 Linéarisation 67
4.1.1 Documents multi-pages 68
4.1.2 Parcours réticulaires 69
Notion de graphe 69
Parcours primaire et secondaires 70
Application 71
Optimisation du parcours primaire 73
4.1.3 Contenus interactifs 74
Réintégration 74
Zone de contenus référencés 75
Affichage en sur-fenêtre 77
Synthèse 79
4.1.4 Contenus calculés 79
Linéarisation des questionnaires interactifs 79
Diagrammes d'activité 80
4.2 Tabulation 82
4.2.1 Définition d'une déclinaison 83
4.2.2 Représentation visuelle 84
4.2.3 Divergence et déclinaisons sous-jacentes 86
4.2.4 Contenu partiellement divergeant 88
4.2.5 Définitions généralisées 91
4.2.6 Tabulation sur un modèle à trois déclinaisons 92

Dans ce chapitre, nous proposons deux stratégies de conception de formes de relecture : la linéarisation et la tabulation. La première se fonde sur l'idée d'une relecture exhaustive passant par la

restauration d'une linéarité matérielle des contenus, que nous aborderons via quatre sous-problèmes posés par l'interactivité. La seconde a pour enjeu la relecture parallèle de plusieurs contextes de rééditorialisation d'un document, que nous étudierons dans le cas de la déclinaison.

4.1 Linéarisation

Le concept de linéarité est lié à l'oralité, au flux temporel de la parole. Pour Bachimont (2007), l'écriture et notamment les structures de liste et de tableau décrites par Goody délinéarisent le discours : ces structures offrent une *synospsis* spatiale aux éléments dispersés dans le discours. Mais la linéarité, telle que la définit Vandendorpe, se comprend également dans l'espace, notamment l'espace géométrique : « La linéarité se dit d'une série d'éléments qui se suivent dans un ordre intangible ou préétabli. Parfaitement exemplifiée par la succession des heures et des jours, elle relève essentiellement de l'ordre et du temps, mais s'applique aussi à un espace réduit aux points d'une droite. » (Vandendorpe, 1999, p. 41). Dans ses travaux sur la spatialisation de l'information, Pfaender fait l'hypothèse suivante : « [...] la ligne structure l'espace perceptif en attirant à elle les actions du lecteur/explorateur de cet espace. Elle joue le rôle d'un aimant si bien que lorsque l'activité perceptive passe à proximité d'une ligne, on est invariablement attiré vers elle et il n'est possible d'en ressortir qu'au prix d'un effort non négligeable. » (Pfaender, 2009, p. 87).

Dans le cadre de ses travaux sur le récit interactif, Bouchardon (2010) réinterroge le concept de linéarité qu'il caractérise à la fois comme une certaine structure du récit (la structure linéaire), comme un type d'organisation matérielle (la contiguïté spatiale des contenus) et enfin comme l'idée de continuité de lecture. Ces trois caractéristiques sont illustrées à travers l'étude de l'œuvre *Un conte à votre façon* de Raymond Queneau, "cas frontière" entre linéarité et non-linéarité, que nous rappelons en annexe [p.128].

Structure du récit

La structure linéaire repose sur le *chaînage contraint*, prévu par l'auteur, entre les unités narratives : le récit ne fait sens que si ces unités sont lues dans l'ordre prescrit. Bouchardon rappelle à ce titre la notion d'irréversibilité chez Roland Barthes : « Ce qui importe, c'est que la logique des séquences d'action assure à la suite des événements racontés un "ordre irréversible (logico-temporel) : c'est l'irréversibilité qui fait la lisibilité du récit classique". » (Bouchardon, 2010, p. 74).

Les structures arborescentes proposent quant à elle des *chaînages à choix*, laissant ainsi au lecteur la possibilité de choisir une unité narrative suivante parmi plusieurs possibles. Comme le rappelle Bouchardon en s'appuyant sur les travaux de Clément, une arborescence peut être divergente (le récit a autant de fins possibles que de feuilles dans l'arbre) ou en boucles (les différents chemins possibles se rejoignent à un certain point du récit). Dans la suite, nous parlerons plus généralement de structure réticulaire (en effet, les deux types d'arborescence sont des cas de réseau). Pour Bouchardon, ce type de structure « [déplace l'intérêt du récit] de la lecture d'un devenir représenté en l'actualisation de devenirs contradictoires. » (Bouchardon, 2010, p. 71).

Organisation matérielle

La linéarité matérielle est incarnée par le support papier, dont la contiguïté spatiale suggère un sens pour la lecture des contenus (et ce indépendamment des intentions de l'auteur). Dans l'histoire de l'écriture, le *volumen* est le support linéaire par excellence, alors que le codex introduit un premier degré de délinéarisation avec la pagination.

Le support numérique est caractérisé quant à lui par une fragmentation matérielle, soit la non-contiguïté spatiale des contenus (on peut également rappeler la figure du *réseau* dans la raison computationnelle chez Bachimont). On notera cependant l'exception du document web "mono-page", typiquement un long article sur un blog, qui est paradoxalement plus linéaire que le codex, le défilement de la barre de *scroll* étant analogue au déroulement du *volumen*.

(Dis)continuité de lecture

La lecture linéaire est la convention selon laquelle on lit de manière continue « en parcourant la page ligne après ligne, de gauche à droite et de haut en bas, puis en passant à la page suivante » (Bouchardon, 2010, p. 82). Il y a en revanche discontinuité lorsque le lecteur doit "sauter du texte" :

- soit pour consulter des éléments paratextuels (« tout ce qui entoure le texte sans être le texte proprement dit » (Bouchardon, 2010, p. 95)) tels qu'une note de bas de page ou une annexe, avant de reprendre le fil de la lecture ;
- soit pour suivre un renvoi vers une autre partie du texte, tel que dans un récit de structure réticulaire comme *Un conte à votre façon*.

Nous proposons la linéarisation au sens d'une transformation restaurant une certaine linéarité *matérielle* des contenus. En nous basant sur les exemples des modèles Opale et Topaze, nous allons étudier la linéarisation à travers quatre sous-problèmes :

- les documents linéaires du point de vue de leur structure mais pas de leur organisation matérielle, que nous appellerons documents "multi-pages" (modules Opale) ;
- les parcours réticulaires au sein d'un document multilinéaire (documents Topaze) ;
- les contenus interactifs ;
- les contenus calculés.

4.1.1 Documents multi-pages

Dans la publication web d'un module Opale (p. 19), chaque élément hiérarchique jusqu'au niveau du grain a sa propre page HTML. Malgré cette fragmentation matérielle du contenu, il est possible de lire le document de façon linéaire à l'aide des liens *précédant/suivant*. En outre, les liens du plan permettent d'accéder à chaque page du module, quelle que soit la page où l'on se trouve.

Finitude synoptique

La finitude synoptique, formulée par Bachimont (2001) dans le cadre d'une étude sur l'informatisation du dossier patient en médecine, est une propriété des documents papier (nombre de pages, épaisseur d'un livre...) qui est absente des documents hypertextes. En soi, cette absence ne constitue pas un risque de désorientation pour le lecteur : « la lecture, devenue inconfortable, n'est pas impossible puisque tout écran consulté peut être rapporté à une position dans le parcours canonique [...] le lecteur dispose toujours de la boussole de la succession linéaire pour interpréter et donner une signification à ce qu'il aperçoit à l'écran. » (*ibid.*).

En revanche dans le cas du dossier patient, composé d'un ensemble de comptes-rendus d'hospitalisation, radios... dont il n'existe pas de parcours canonique *a priori*, il a été observé que la lecture était organisée grâce à la synopsis (*voir ensemble*) des différents éléments du dossier, étalés sur une table, et de leurs propriétés physiques (par exemple l'épaisseur relative des différents comptes-rendus). L'enjeu est alors de remplacer cette synopsis *spatiale* par une synopsis *calculée* (*ibid.*).

La finitude synoptique nous semble également fondamentale du point de vue de la relecture : sans elle, l'ensemble de contenu ne peut être appréhendé qu'en parcourant tous les liens. L'enjeu de la linéarisation est donc de calculer une synopsis des contenus en les projetant sur une même page web, tel qu'illustré dans la figure ci-dessous. On se retrouve alors avec une continuité matérielle identique au volumen (défilement à l'aide de la barre de *scroll*), augmentée d'aides à la lecture rendues possibles grâce au caractère dynamique du support, par exemple :

- la taille et la position de la barre de scroll indiquent le reste du contenu à relire ;
- le plan du document, avec des liens vers les ancres HTML des différents niveaux du contenu (liens *intra-page*), est disponible en marge permanente (i.e. non impactée par le défilement

vertical du contenu de droite) ;

- lorsque le relecteur active un lien dans le plan, une transition animée vers le contenu (*smooth scrolling*) permet de conserver une impression de continuité ;
- etc.

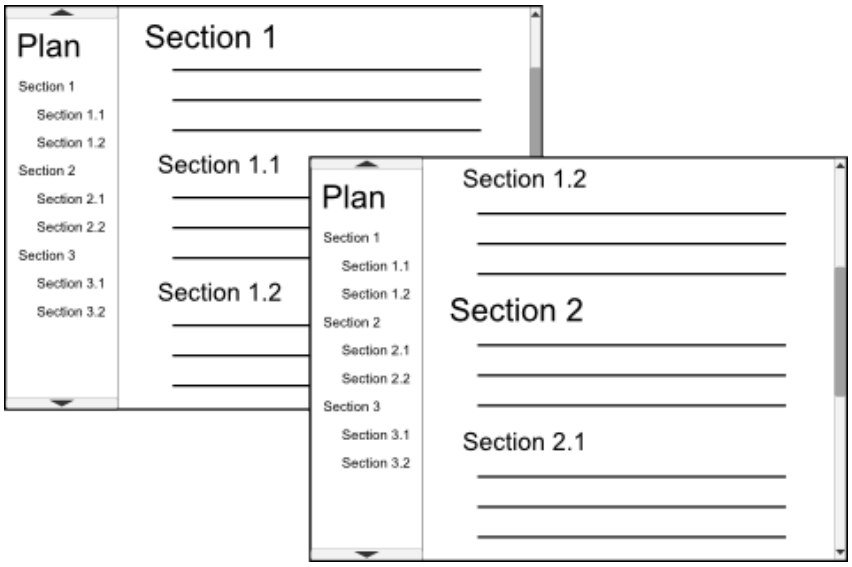


Figure 41 - Maquette d'une forme de relecture linéarisée obtenue par une transformation mono-page des fragments documentaires.

4.1.2 Parcours réticulaires

Notion de graphe

Nous allons maintenant nous intéresser à la linéarisation des parcours réticulaires dans les documents multilinéaires, tels que nous les avons vus à travers l'exemple de Topaze (p. 22).

D'un point de vue informatique, un ensemble de parcours réticulaires est représentable via la structure de graphe. Ce graphe est orienté (on distingue donc les arcs sortants des arcs entrants) et admet un unique point d'entrée, appelé nœud *racine*, permettant de parcourir le graphe par un des chemins possibles. Un graphe peut contenir un circuit, c'est-à-dire une suite d'arcs bouclant sur un même nœud. Soit par exemple le graphe d'*Un conte à votre façon* [p.128] de Raymond Queneau, dans lequel chaque nœud est numéroté en fonction du fragment qu'il désigne :

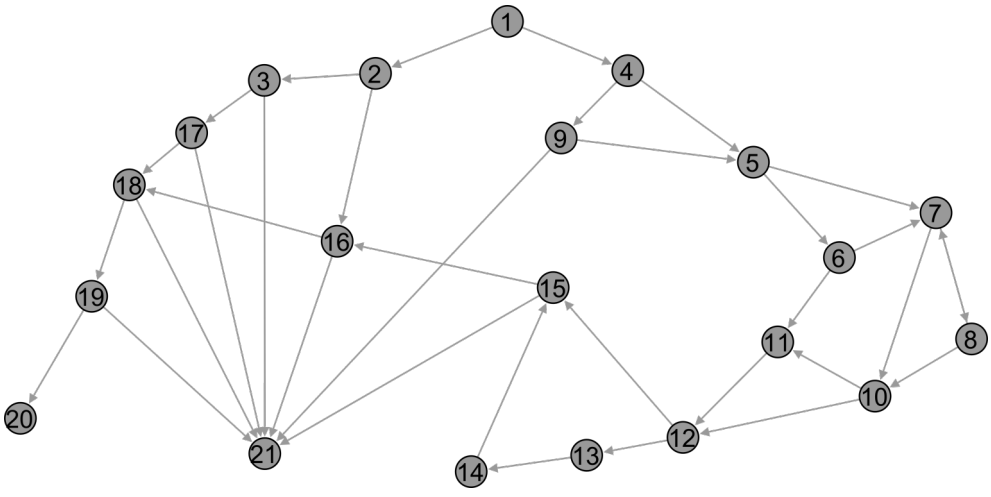


Figure 42 - Graphe représentant Un conte à votre façon de Raymond Queneau.

Ce graphe a pour racine le fragment 1 et comporte un circuit entre les fragments 7 et 8. On retrouve également les deux fins alternatives du conte, à savoir les fragments 20 et 21, qui sont accessibles depuis la racine et n'ont aucun arc sortant.

Parcours primaire et secondaires

Une première solution de linéarisation des parcours réticulaires d'un graphe consiste à générer un à un les différents parcours possibles de ce graphe. Cette solution est en revanche fortement limitée dans le cadre de la relecture, ces parcours étant en nombre élevé voire infini si le graphe comporte un circuit.

Une seconde solution s'inspire directement du lecteur, qui construit un parcours par choix successifs et élimination progressive des autres parcours possibles. Le parcours construit peut alors servir de référence pour les parcours qu'il reste à explorer. En référence aux notions de récit primaire (lu, vécu par le lecteur) et de récit secondaire (possible narratif prévu par l'auteur, mais éliminé par le lecteur), proposées par Clément dans ses travaux sur les structures arborescentes, nous parlerons d'une part de *parcours primaire* pour désigner le parcours construit par le relecteur, et d'autre part de *parcours secondaires* exprimés relativement au parcours primaire. Notons que l'adjectif "secondaire" n'est pas à comprendre ici comme un second niveau de discours inscrit dans le paratexte, mais bien comme un récit/parcours indépendant, qui aurait tout à fait pu être lu à la place du récit/parcours primaire.

Voici un exemple de parcours primaire possible, représenté en vert, dans le graphe d'*Un conte à votre façon*, ainsi que les parcours secondaires relatifs, représentés en gris :

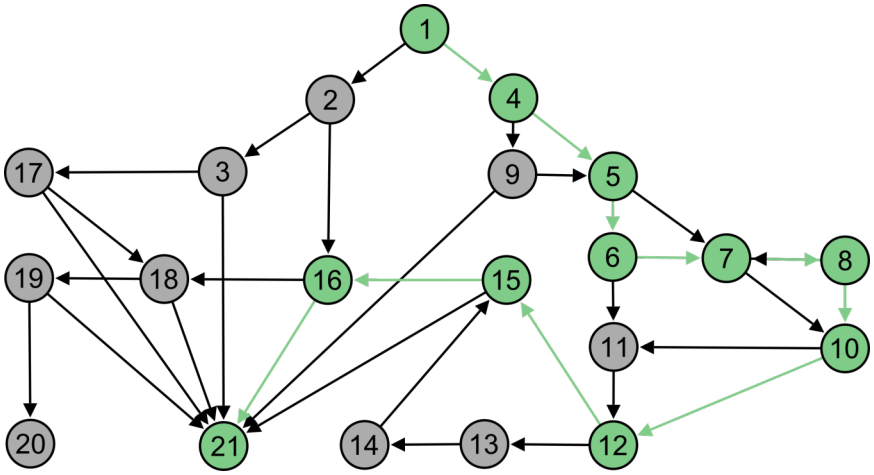


Figure 43 - Exemples de parcours primaire (en vert) et secondaires (en gris).

Un parcours primaire est signifiant pour une relecture linéaire en ce qu'il est un parcours possible du graphe (du moins, si ce parcours n'est pas signifiant, il permet de refléter un éventuel défaut de conception du document...).

Sauts et divergences

Les liens éliminés dans la construction du parcours primaire sont à distinguer selon plusieurs types. Lorsque le nœud au bout du lien éliminé appartient au parcours, on dira que le lien est un *saut* si le nœud est rencontré plus loin dans le parcours, ou un *retour* s'il a déjà été rencontré (cas du circuit). Par exemple, le sous-parcours ci-dessous présente un saut de 7 à 10 et un retour de 8 à 7 :

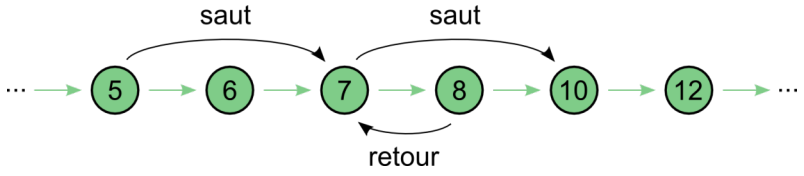


Figure 44 - Sauts et retour.

Si au contraire, le nœud au bout du lien éliminé n'appartient pas au parcours primaire, on parlera de *divergence*. Une divergence est de surcroît *définitive* si aucun nœud du parcours primaire n'est présent dans le sous-graphe lié. Autrement dit, un parcours secondaire est définitivement divergeant s'il ne converge pas *in fine* vers un nœud du parcours primaire. Par exemple, le parcours secondaire 1-2-3-17-18-19-21 est divergeant, tandis que 1-2-3-17-18-19-20 est définitivement divergeant :

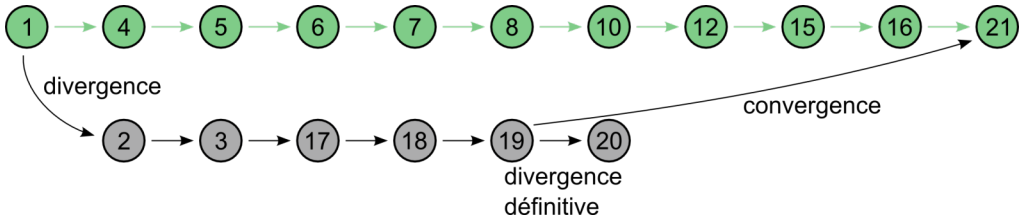


Figure 45 - Divergences simple et définitive.

Notons que tous ces types de lien peuvent aussi exister au niveau d'un parcours secondaire.

Application

En nous appuyant sur les notions de parcours primaire et secondaire d'un graphe, nous proposons à travers la maquette ci-dessous une forme de relecture dédiée aux documents multilinéaires. L'enjeu pour le relecteur est dans un premier temps de construire un parcours primaire, avant d'explorer les parcours secondaires à partir des liens écartés.

Construction du parcours primaire

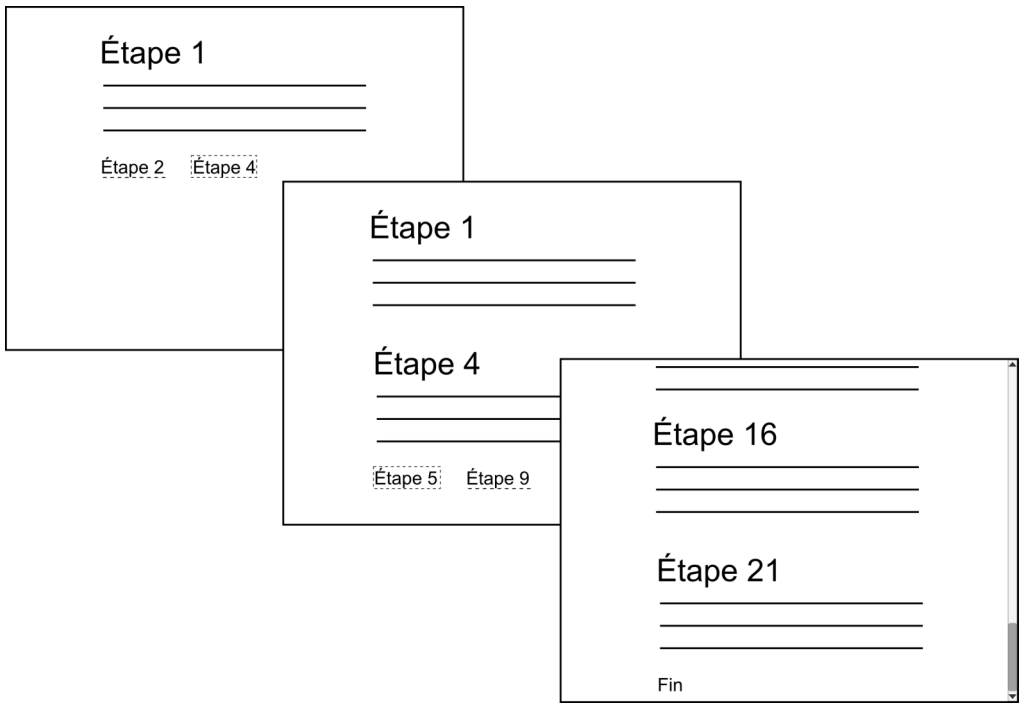


Figure 46 - Construction du parcours primaire par le relecteur.

Après chaque étape, le relecteur se voit proposer un ensemble de liens vers les prochaines étapes possibles, tel que dans la forme classique du document multilinéaire. En revanche, l'étape choisie ne s'affiche pas dans une nouvelle page, mais à la suite de l'étape précédente. De cette façon, le parcours primaire est progressivement linéarisé jusqu'à ce que l'étape finale soit atteinte. Lorsque le contenu déborde par rapport à la hauteur de la fenêtre, la barre de scroll est automatiquement positionnée au niveau de la dernière étape choisie.

Exploration des parcours secondaires

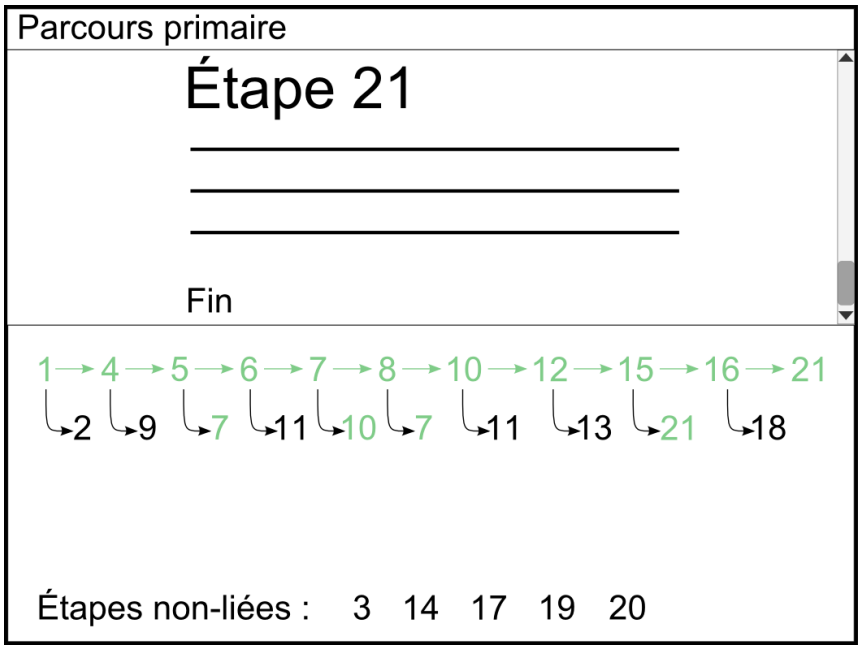


Figure 47 - Visualisation du parcours primaire et des liens écartés.

La zone en bas de fenêtre permet de visualiser le parcours primaire et les liens écartés lors de sa construction. Dans ce second niveau, les étapes appartenant au parcours primaire sont colorées en vert, de façon à repérer les sauts et les retours. Un clic sur une telle étape permet de positionner la barre de scroll à la bonne hauteur dans le parcours primaire. Toutes les autres étapes (en noir) sont cliquables pour permettre l'exploration des parcours secondaires. Par ailleurs, les étapes qui ne sont pas encore atteignables au stade de dévoilement du graphe sont affichées en bas de la zone.

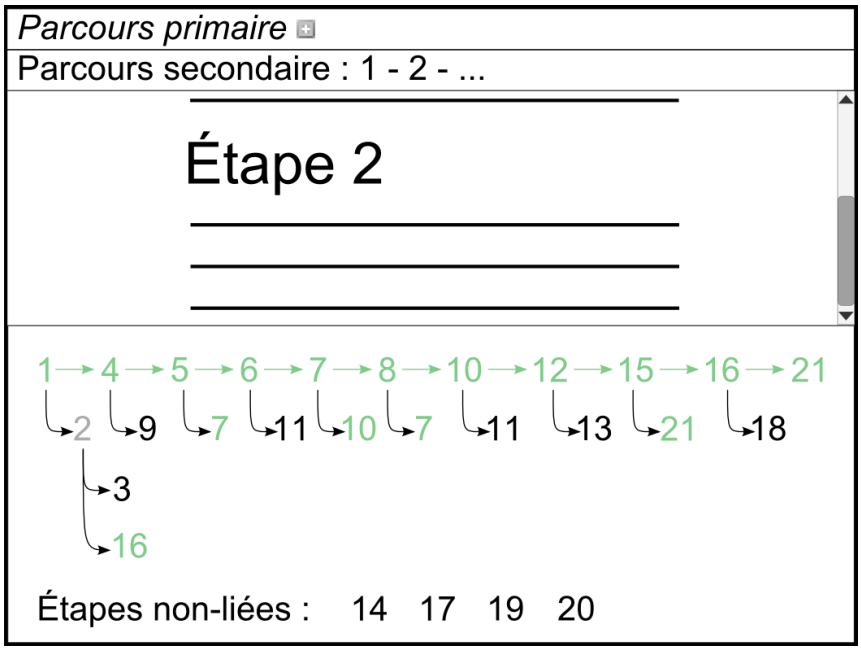


Figure 48 - Exploration d'un parcours secondaire.

Le clic sur une étape non-visitée (l'étape 2 dans la figure ci-dessus) a pour effet :

- d'afficher dans un nouvel onglet (à l'aide d'un menu de type "accordéon") le parcours secondaire avec cette étape, ou de compléter ce parcours s'il a déjà été initié ;
- d'afficher les liens vers les étapes suivantes (en mettant à jour la liste des étapes non-liées) ;

- de colorer cette étape (et ses autres occurrences) en gris, de façon à mémoriser qu'elle a été lue dans un parcours secondaire.

La coloration des étapes déjà parcourues permet de repérer les convergences vers le parcours primaire (par exemple, l'étape 16) ou vers un parcours secondaire (pas d'exemple dans la figure ci-dessus). Notons que nous avons préféré un affichage où les étapes sont dupliquées (par exemple, l'étape 11 est présente deux fois puisqu'elle est liée par les étapes 6 et 10) à un graphe, qui nous aurait semblé peu lisible à cause des liens s'entrecroisant. Une solution consiste à indiquer, au survol d'une étape, ses autres occurrences dans la visualisation, en les soulignant par exemple.

Optimisation du parcours primaire

La forme de relecture précédente nous amène à remarquer le fait qu'un parcours primaire permettra une relecture d'autant plus efficace qu'il contiendra un maximum d'étapes. En effet, l'analyse des parcours secondaires constitue la partie la plus coûteuse de la relecture, puisqu'elle n'évacue pas le problème de la non-linéarité. D'une certaine manière, le fait de laisser le relecteur construire lui-même le parcours primaire comporte le risque que ce dernier soit trop court (par exemple : 1-2-16-21), et que l'essentiel des étapes soit encore à découvrir via les parcours secondaires.

Cette limite nous amène à réfléchir à une manière d'optimiser le parcours primaire. Pour cela, nous faisons appel à la théorie des graphes, et au problème de la recherche d'un plus long chemin dans un graphe orienté, détaillé en annexe [p.130].

Dans notre exemple, l'algorithme de recherche d'un plus long chemin partant de l'étape 1 donne deux parcours candidats, qui ne diffèrent que par leur dernière étape :

1. 1-4-9-5-6-7-8-10-11-12-13-14-15-16-18-19-*20*
2. 1-4-9-5-6-7-8-10-11-12-13-14-15-16-18-19-*21*

Les deux parcours comportent en outre les sauts 4→5, 5→7, 7→10, 10→12 et 12→15, ainsi que le retour 8→7. Le second comporte par ailleurs quatre sauts de plus : 9→21, 15→21, 16→21 et 18→21. Nous faisons l'hypothèse qu'entre plusieurs parcours de longueur maximale, il est préférable de choisir celui comportant le plus de sauts et de retours. En effet, ces derniers permettent non-seulement de libérer le relecteur de l'actualisation du lien sans pour autant lui masquer l'information de la bifurcation possible, mais surtout de lui indiquer que cette bifurcation converge *in fine* vers le parcours primaire. Dans l'algorithme de recherche du plus long chemin, il est donc nécessaire de comptabiliser, sur chaque nœud de chaque parcours candidat, le nombre de liens n'appartenant pas au parcours mais liant ce nœud à un autre nœud du même parcours.

Voici finalement le parcours primaire optimal pour la relecture d'*Un conte à votre façon* :

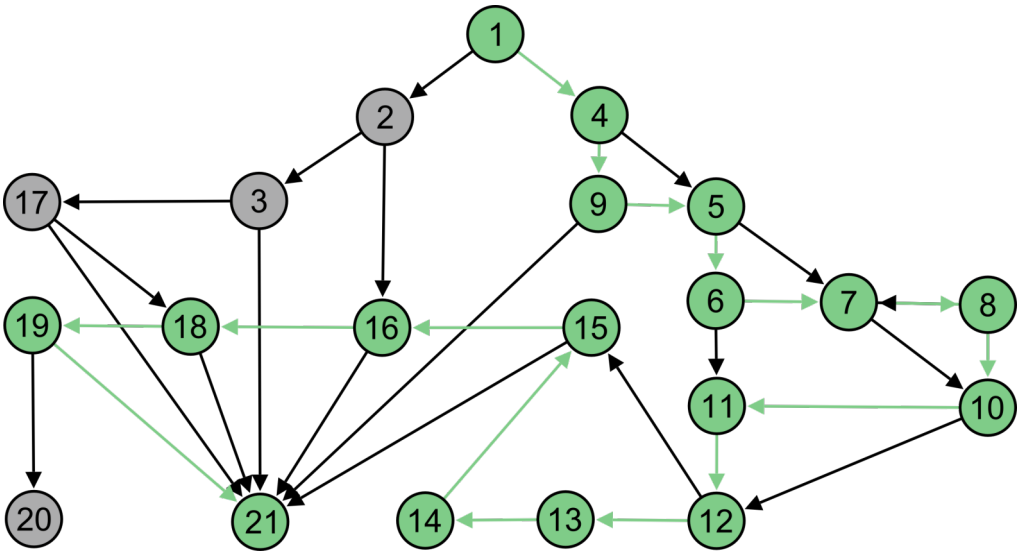


Figure 49 - Parcours primaire optimal dans le graphe d'Un conte à votre façon.

Nous envisageons deux façons d'exploiter le parcours primaire optimal au niveau de la forme de relecture :

1. il peut être proposé directement au relecteur, lui permettant ainsi de passer directement à la phase d'analyse des parcours secondaires ;
2. il peut apparaître de manière implicite lors de la construction du parcours primaire par le relecteur, sous la forme d'étapes suivantes "conseillées" après chaque étape.

La seconde solution nous semble préférable car elle laisse plus de liberté au relecteur, qui peut alors combiner efficacité et découverte. Notons cependant que cette solution nécessite de pouvoir recalculer un parcours primaire optimal à chaque fois que le relecteur choisit une autre étape que celle qui est conseillée.

4.1.3 Contenus interactifs

Réintégration

Dans l'exemple des contenus à profondeur variable, on remarque qu'une solution intuitive de linéarisation des blocs dépliables (p. 17) consiste à inverser leur comportement interactif, c'est-à-dire d'afficher leur contenu par défaut et de le masquer sur demande. Nous allons étudier la possibilité d'appliquer cette solution, que nous appelons *réintégration*, pour les exemples d'incises et de renvois que nous avons donnés. À la différence des blocs dépliables, ces contenus interactifs ont la particularité d'avoir un ancrage *inline*, c'est-à-dire que le lien se situe à l'intérieur même du texte. Leur réintégration doit donc faire face à des contraintes syntaxiques, que ce soit au niveau de la DTD HTML ou bien de la typographie.

Contraintes de la syntaxe HTML

Le résultat de la réintégration doit être valide par rapport à la DTD HTML. Dans la majorité des cas, le lien vers le contenu interactif est ancré à l'intérieur d'une balise *p* (paragraphe), qui d'après le standard HTML5 n'autorise que des balises de type *phrasing content* (*abbr*, *em*, *strong*, *span*...). Or, ces dernières n'incluent pas les balises *p* et *div*, qui sont de type *flow content* (type incluant les balises de type *phrasing content*). Pour réintégrer un contenu interactif, il est donc nécessaire de le transformer, lorsque c'est possible, en n'utilisant que des balises de type *phrasing content*.

En considérant le modèle documentaire des fragments de type *abréviation*, *glossaire* et *bibliographie* dans Opale, on remarque que le champ *Signification* (resp. *Entrée bibliographique*) d'une abréviation (resp. d'une référence bibliographique) est limité à un seul paragraphe, là où le champ *Définition* d'une entrée de glossaire est pluri-paragraphe. Ainsi, les incises d'abréviation et de bibliographie peuvent être réintégrées sans problème du moment que la balise englobant leur contenu est de type *phrasing content* (tel que la balise *span* par exemple). En revanche, les incises de glossaire ne peuvent être réintégrées sans dommage sur la sémantique du contenu : en effet, la transformation de plusieurs paragraphes en autant de balises *span* a pour effet de court-circuiter le retour à la ligne.

Contraintes typographiques

La réintégration ne doit pas impacter la cohérence sémantique de la phrase dans laquelle le lien est ancré, ce dernier balisant la plupart du temps un mot ou un ensemble de mots au milieu de cette phrase. Un contenu s'étendant potentiellement sur plusieurs paragraphes ne peut donc pas être réintégré, puisque ceci aurait pour effet de déstructurer la phrase et le paragraphe en cours. Cette conclusion corrobore la contrainte syntaxique de HTML que nous avons évoquée plus haut.

Intuitivement, le contenu réintégré doit être entouré de signes de ponctuation délimitant une incise (au sens rhétorique/grammatical, d'après le Wiktionnaire : « Proposition indépendante insérée dans une phrase, entre virgules ou tirets ou parenthèses, et qui forme un sens partiel [...] »). Les parenthèses semblent être une bonne solution car elles peuvent contenir une (ou plus) phrase indépendante vis-à-vis de la phrase principale. Ainsi, la réintégration de la définition mono-paragraphe suivante est correcte

d'un point de vue typographique :

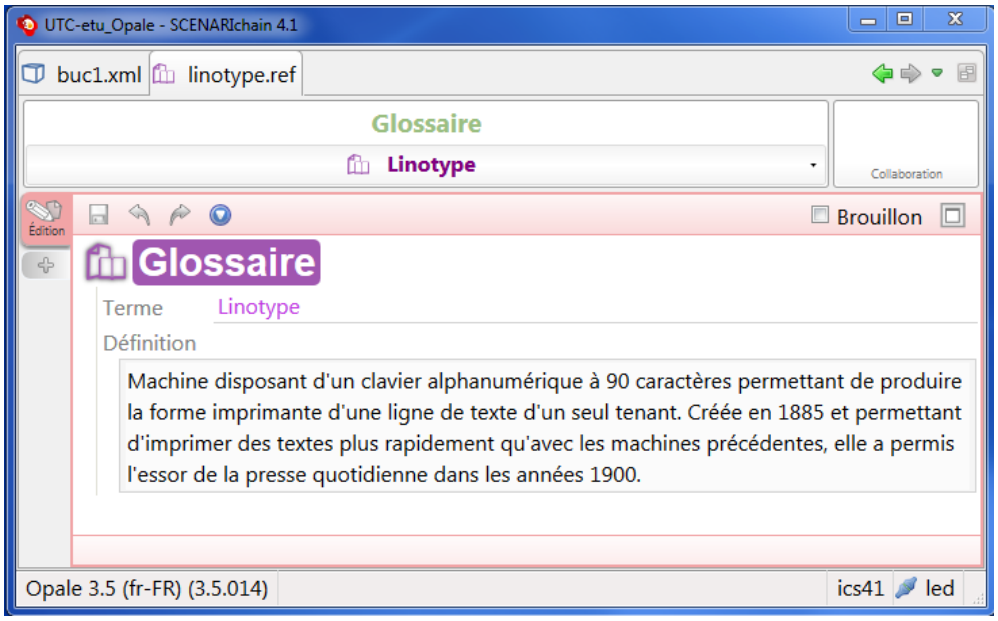


Figure 50 - Définition ne comportant qu'un seul paragraphe.

Dans le cas où l'ancrage du lien est lui-même situé entre parenthèses, cette solution reste viable car une proposition entre parenthèses peut en contenir une autre, même si cela est peu fréquent.

Zone de contenus référencés

Si nous avons vu que la réintégration était viable pour la relecture des incises mono-paragraphe telles que les incises d'abréviation et de bibliographie dans Opale, un autre paramètre peut rendre cette solution fortement redondante, à savoir le fait que ces contenus (ainsi que les incises pluri-paragraphe telles que les entrées de glossaire) sont des fragments à part entière et sont donc réutilisables à l'intérieur du même document. Cette réutilisation intra-documentaire, qui dépend bien sûr de l'intention de l'auteur, peut dans certains cas être *systématique* : par exemple dans les fiches juridiques du modèle Juriguide utilisé à l'UCANSS (Union des Caisses Nationales de Sécurité Sociale), un article de loi est référencé plusieurs fois dans la même fiche, à chaque citation de la loi en question.

Appareil de référence et liens inverses

Une alternative à la réintégration est l'utilisation des appareils de référence classiquement utilisés dans les documents papier, placés soit au début (liste des abréviations utilisées), soit à la fin (glossaire, bibliographie, les annexes, etc.). L'avantage est que le relecteur ne relit qu'une seule fois chaque contenu référencé. Ces structures sont parfois complétées par des renvois inverses : par exemple, dans un index en fin d'ouvrage sont indiquées, pour chaque mot, les pages d'occurrence de ce mot. Cette logique de renvois bi-directionnels peut être transposée dans une forme de relecture à l'aide de liens et de liens inverses :

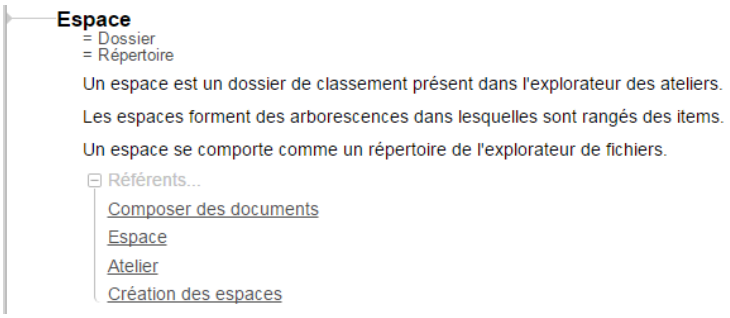


Figure 51 - Entrée de glossaire avec liens inverses dans une documentation réalisée avec le modèle Dokiel (http://docs.kelis.fr/models/dokiel/40/site/co/guideWeb_02_1.html).

L'inconvénient d'un appareil de référence placé dans le flux (au début ou à la fin) de la forme de relecture est de réintroduire une navigation intra-page et donc une impression de discontinuité dans la lecture : par exemple, si le relecteur suit un lien vers une entrée de glossaire, il devra choisir le bon lien inverse parmi plusieurs, si ce contenu est référencé plusieurs fois, pour reprendre le fil de sa lecture (sauf à disposer d'un lien "retour" calculé dynamiquement en fonction de l'ancre qui a appelé le lien).

Afin de restaurer une synopsis spatiale entre le texte et les contenus référencés, nous faisons le choix d'afficher ces derniers dans une zone située en marge permanente de la forme de relecture, dans leur ordre d'apparition dans le document. Lors de l'activation d'un lien de référence, le contenu est mis en évidence dans cette zone (éventuellement à l'aide d'un *scroll* vertical automatique si la zone comporte beaucoup de contenu). Si ce contenu comporte plusieurs liens inverses, une case à cocher permet de le marquer explicitement comme "relu", ce qui aura pour effet de marquer de la même façon toutes les références à ce contenu (notamment celles situées plus loin dans le contenu) et ainsi d'éviter l'effet de redondance.

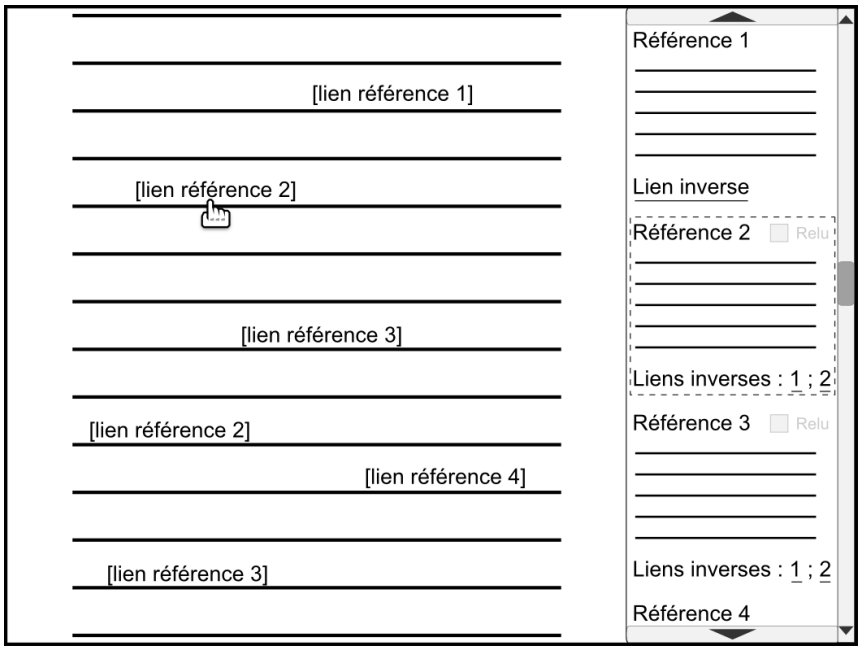


Figure 52 - Mise en évidence d'un contenu référencé.

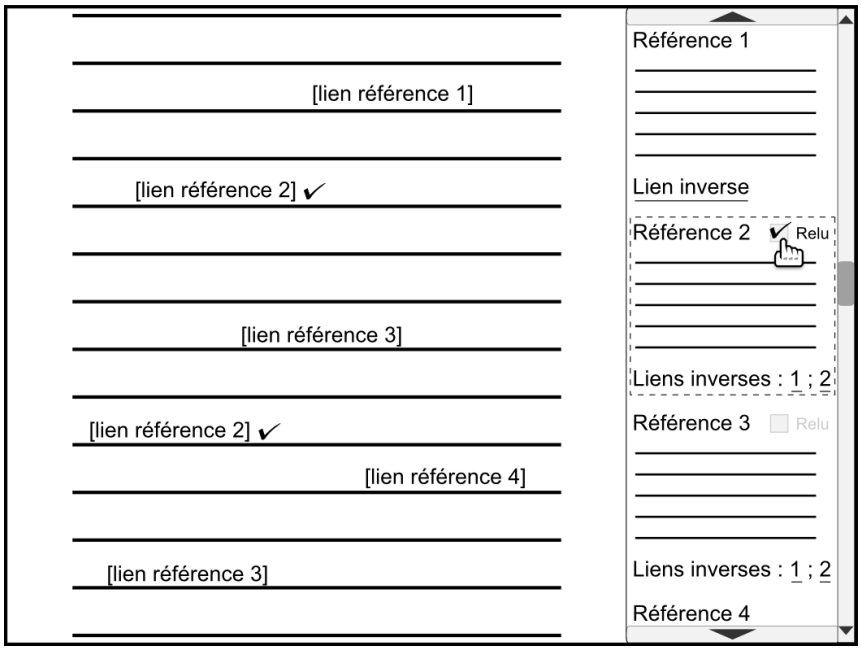


Figure 53 - Marquage explicite de la relecture d'un contenu référencé.

De cette façon, nous laissons une indétermination sur la manière dont ces contenus seront relus, c'est-à-dire au fur et à mesure et/ou à la fin, lors d'une relecture globale des contenus référencés, en utilisant les liens inverses pour "retrouver" les ancrages de ces contenus dans le texte. Ainsi, et cela nous semble important, nous nous détachons de telle ou telle hypothèse quant à la stratégie *réelle* du relecteur.

Affichage en sur-fenêtre

La solution de l'affichage des contenus interactifs dans la zone en marge permanente est limitée pour les contenus "fortement structurés". Nous entendons par là les contenus dont la structure ne se limite pas à du texte (ensemble de paragraphes, de listes, etc.) sur un seul niveau : par exemple, un grain Opale est composé d'un titre, de balises titrées (balises pédagogiques) pouvant comporter du texte, des ressources (images ou autres), des sous-parties, etc.. La figure interactive utilisée pour ce genre de contenu est généralement la parenthèse, tel que pour les grains liés dans Opale (p. 19), ou encore la navigation.

Nous proposons pour ce type de contenu un affichage en sur-fenêtre. Si le document comporte plusieurs références au même contenu et/ou si ce contenu appartient par ailleurs au plan (ce qui est possible pour un grain Opale), il est proposé au relecteur de marquer le contenu s'affichant dans la sur-fenêtre comme "relu". Dans le premier cas, les autres références seront marquées (comme dans la solution précédente). Dans le deuxième cas, le contenu "dans le flux" (c'est-à-dire le contenu tel qu'il est lu dans la linéarité du plan) sera marqué.

Les maquettes suivantes sont inspirées du corpus NF17 dans lequel le grain "Agrégation" est référencé par le grain "Composition", qui par ailleurs le précède dans le plan.

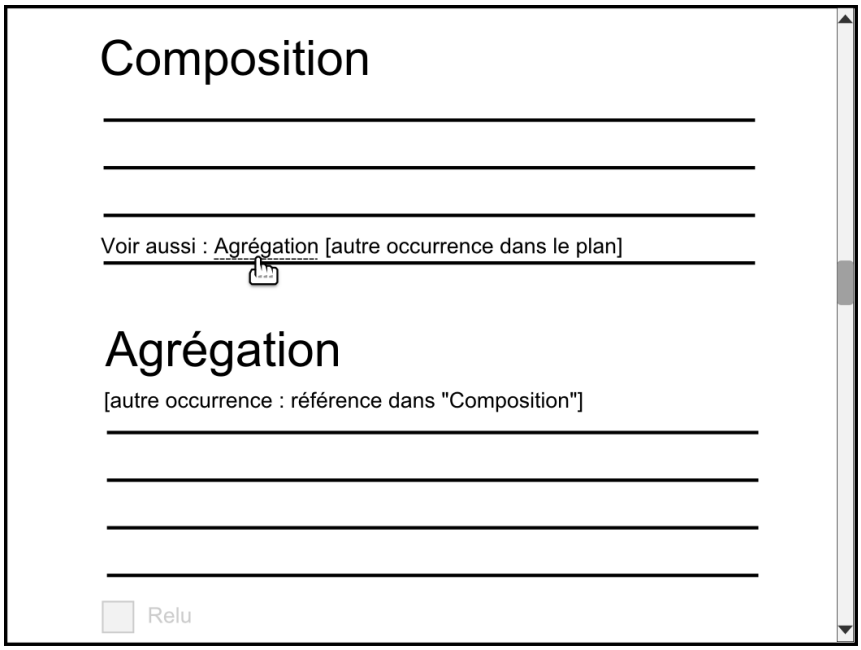


Figure 54 - Indications des différentes occurrences d'un grain dans le document.

Avant d'activer le lien, le relecteur peut savoir s'il existe une autre occurrence du contenu référencé dans le document. De façon symétrique, cette référence est indiquée au niveau du grain "Agrégation", cette fois-ci dans le flux. Ainsi, le relecteur peut choisir de relire ce grain dans la sur-fenêtre tel qu'illustré dans les figures ci-dessous, ou bien de ne pas activer le lien et d'attendre de relire le grain dans le flux (une case à cocher, apparaissant en bas de la figure ci-dessus, permet également de marquer le grain comme "relu").

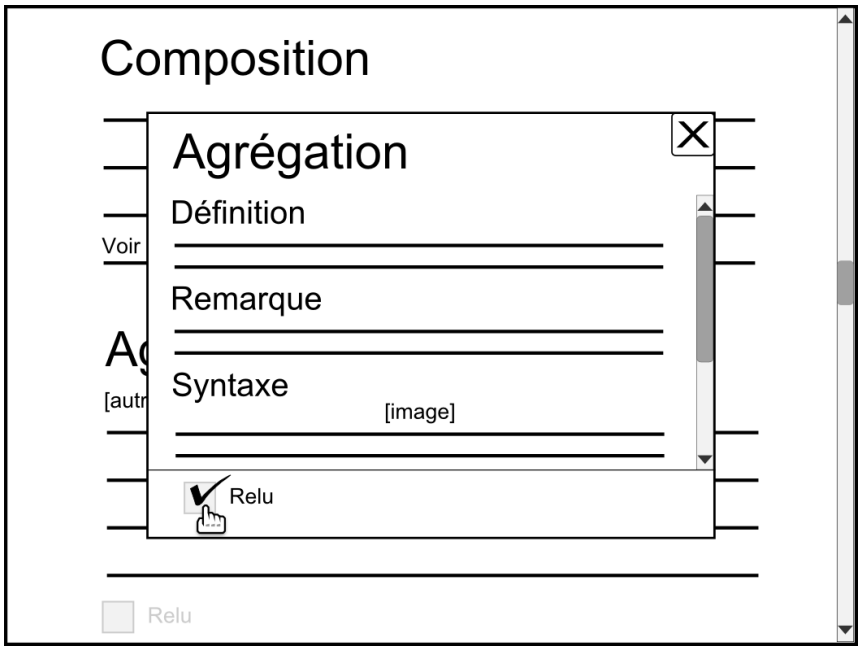


Figure 55 - Affichage du grain référencé en sur-fenêtre et confirmation de la relecture du grain.

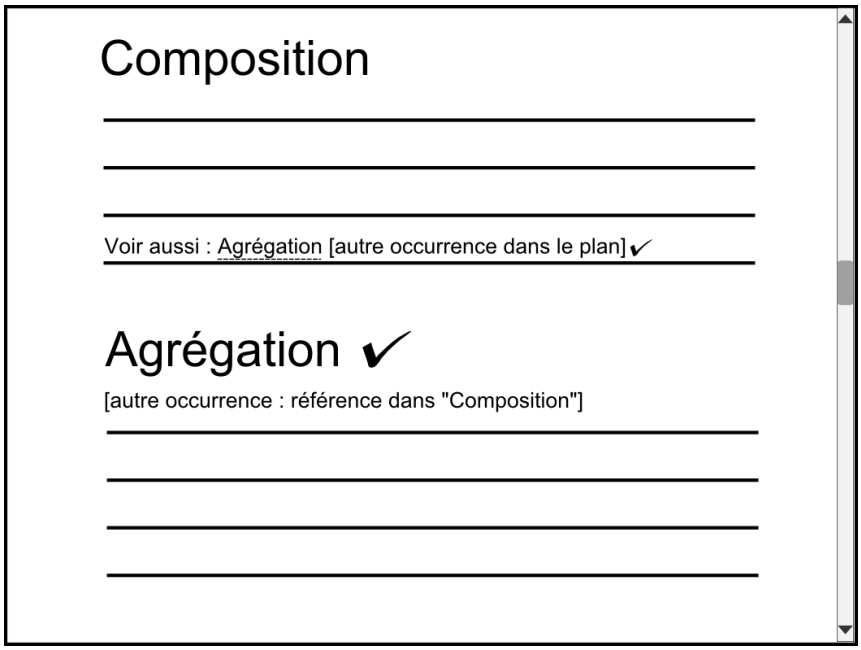


Figure 56 - Propagation de la trace de relecture.

Synthèse

Figure interactive		Ancrage *inline*	Réutilisabilité intra-documentaire	Structuration du contenu	Solution
Augmentation	Contenus à profondeur variable	non	non	Pluri-paragraphe	Réintégration
	Incises	oui	oui	Mono ou pluri-paragraphe	Zone de contenus référencés
Parenthèse		oui	oui	Forte	Affichage en sur-fenêtre

Tableau 5 - Synthèse des propositions pour la linéarisation des contenus interactifs.

4.1.4 Contenus calculés

Linéarisation des questionnaires interactifs

La linéarisation des questionnaires interactifs (p. 17) consiste à présenter leurs contenus sous une forme statique synthétisant les différents résultats possibles quelles que soient les réponses données par le lecteur. Pour les questions de type QCU et QCM notamment, il s'agit de remplacer l'IHM de réponse à la question (boutons radio, cases à cocher...) par un affichage direct des bonne(s) et mauvaises réponses, ainsi que des explications. En outre, l'ensemble du questionnaire (c'est-à-dire toutes ses questions) doit pouvoir être affiché dans une même page, ceci afin d'en permettre une relecture exhaustive.

Il s'agit donc de transformer les contenus de façon à passer de la première à la seconde figure ci-dessous :

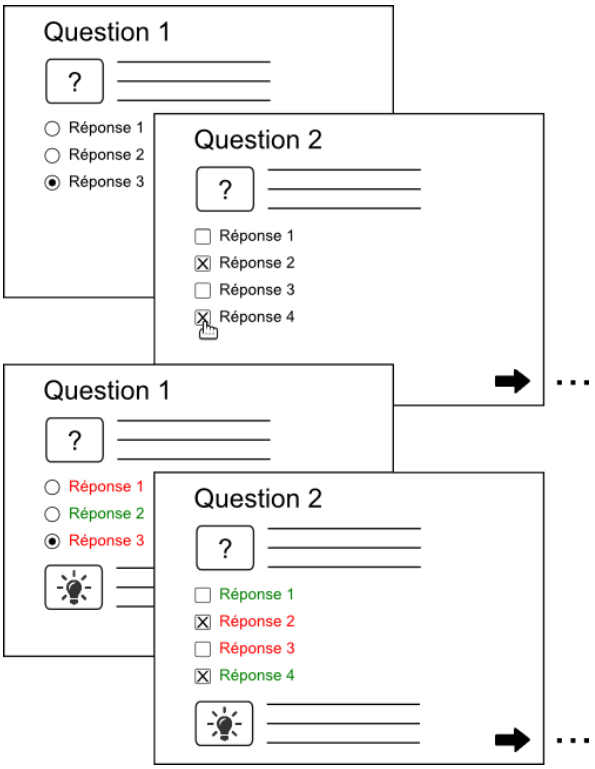


Figure 57 - Questionnaire interactif dans sa forme "traditionnelle".

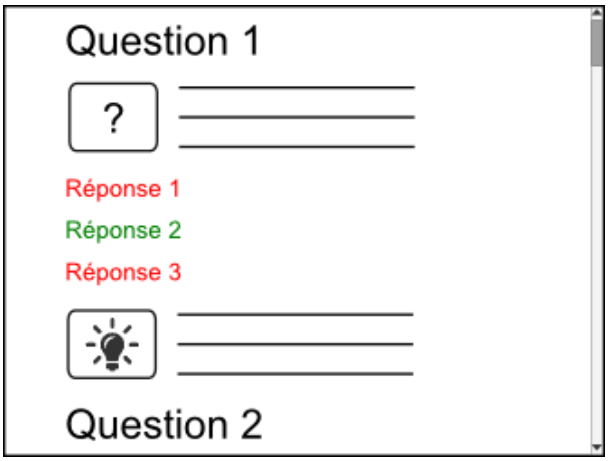


Figure 58 - Questionnaire interactif linéarisé.

Diagrammes d'activité

Dans le cadre des documents multilinéaires, la structure classique de graphe orienté ne permet pas de représenter les parcours conditionnés tels que ceux pouvant être créés avec Topaze (enchaînement conditionné, enchaînement par QCU/QCM, feedback d'une étape de quiz...). Il faut pour cela recourir à des formalismes graphiques disposant d'une typologie des nœuds et des liens adaptée. À ce titre, nous nous intéressons aux diagrammes d'activité du standard *UML (http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-activites)*, et en particulier, les nœuds de type *action* et *décision* (représentés respectivement par des rectangles à bords arrondis et par des losanges) qui permettent de préciser le comportement d'un programme en fonction des actions de l'utilisateur.

Correspondance Topaze-Diagramme d'activité

Nous allons utiliser le formalisme du diagramme d'activité afin de proposer une forme de relecture,

au niveau macro (sans contenu), de documents Topaze. Dans le tableau ci-dessous, nous mettons en correspondance les éléments de Topaze avec les composants graphiques du diagramme d'activité :

Topaze	**Diagramme d'activité**
Accueil de l'étude de cas	Nœud initial : ●
Fin de l'étude de cas	Nœud final : ◉
Étape de contenu Étape de quiz ...	Nœud d'action : (Étape)
Enchaînement simple Enchaînement libre	Transition(s) : →
Enchaînement conditionné Enchaînement par QCU/QCM Feedback d'une étape de quiz	Nœud de décision : ◇
Condition calculée Réponse juste/fausse Intervalle de score	Condition de garde : —[condition]→

Tableau 6 - Mise en correspondance Topaze/Diagramme d'activité.

Exemple

Le diagramme ci-dessous permet de relire la structure conditionnelle d'un test de positionnement Faq2Sciences (p. 23) :

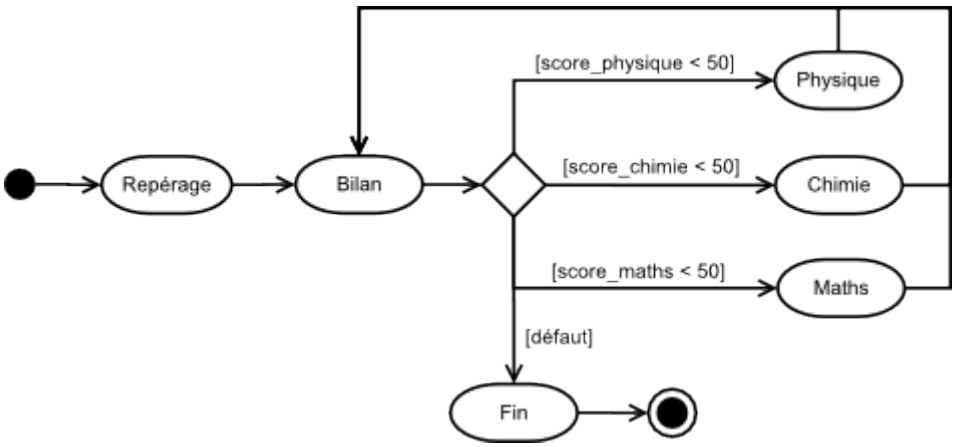


Figure 59 - Diagramme d'activité UML représentant les enchaînements conditionnés d'un test de positionnement.

Notons que si ce diagramme est ici relativement simple (peu de nœuds, peu de liens...), il peut être beaucoup plus complexe dans d'autres cas et devenir illisible (entrecroisement de liens, chevauchement des liens sur les nœuds, etc.). Une version interactive de ce diagramme permettrait de pallier ce problème, en proposant une découverte progressive des nœuds.

4.2 Tabulation

La tabulation d'un contenu, au sens d'une *mise en tableau*, consiste à afficher au sein d'une même forme de relecture les différents contextes de rééditorialisation de ce contenu dans autant de colonnes. Les cases d'une même ligne peuvent entretenir une relation d'équivalence (même contenu dans différentes colonnes) ou bien d'opposition (différences entre colonnes). Prenons par exemple le grain Opale suivant, dont certains contenus sont déclinés pour les versions courte et standard du module :

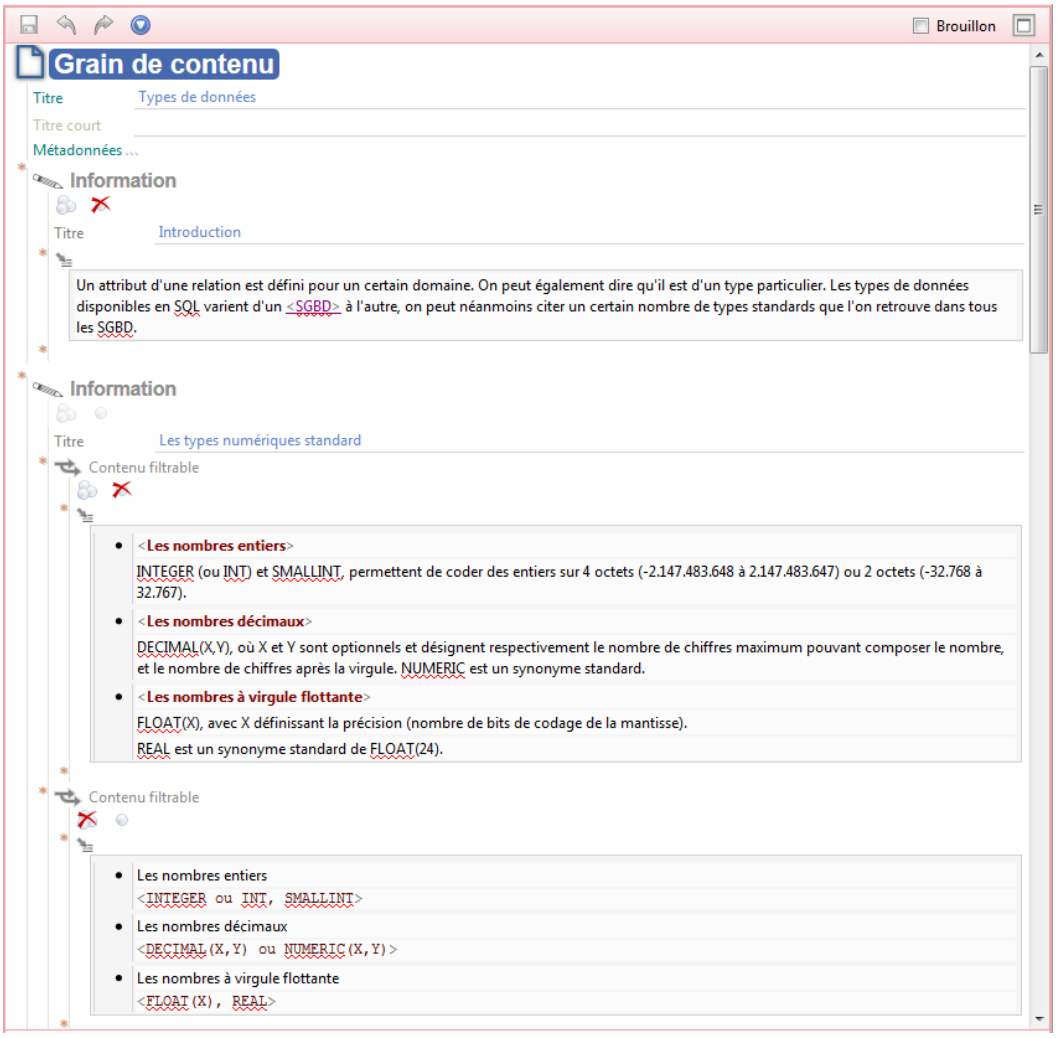
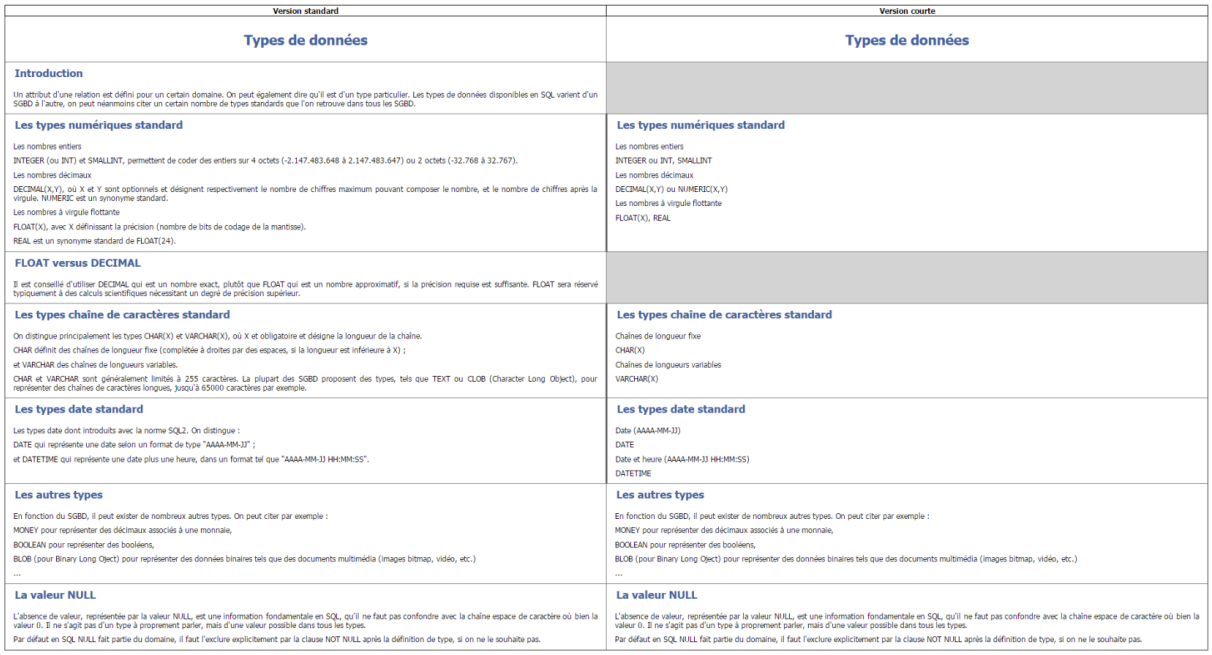


Figure 60 - Grain de contenu décliné.

La maquette ci-dessous illustre la tabulation de ce grain, où chaque balise pédagogique correspond à une ligne du tableau. L'équivalence est marquée par une bordure centrale fine, tandis que l'opposition est marquée par une bordure plus épaisse.



Version standard	Version courte
Types de données	**Types de données**
Introduction Un attribut d'une relation est défini pour un certain domaine. On peut également dire qu'il est d'un type particulier. Les types de données disponibles en SQL varient d'un SGBD à l'autre, on peut néanmoins citer un certain nombre de types standards que l'on retrouve dans tous les SGBD.	
Les types numériques standard Les nombres entiers INTEGER (ou INT) et SMALLINT, permettent de coder des entiers sur 4 octets (-2.147.483.648 à 2.147.483.647) ou 2 octets (-32.768 à 32.767). Les nombres décimaux DECIMAL(X,Y), où X et Y sont optionnels et désignent respectivement le nombre de chiffres maximum pouvant composer le nombre, et le nombre de chiffres après la virgule. NUMERIC est un synonyme standard. Les nombres à virgule flottante FLOAT(X), avec X définissant la précision (nombre de bits de codage de la mantisse). REAL est un synonyme standard de FLOAT(24).	**Les types numériques standard** Les nombres entiers INTEGER ou INT, SMALLINT Les nombres décimaux DECIMAL(X,Y) ou NUMERIC(X,Y) Les nombres à virgule flottante FLOAT(X), REAL
FLOAT versus DECIMAL Il est conseillé d'utiliser DECIMAL qui est un nombre exact, plutôt que FLOAT qui est un nombre approximatif, si la précision requise est suffisante. FLOAT sera réservé typiquement à des calculs scientifiques nécessitant un degré de précision supérieur.	
Les types chaîne de caractères standard On distingue principalement les types CHAR(X) et VARCHAR(X), où X et obligatoire et désigne la longueur de la chaîne. CHAR définit des chaînes de longueur fixe (complétée à droites par des espaces, si la longueur est inférieure à X) ; et VARCHAR des chaînes de longueurs variables. CHAR et VARCHAR sont généralement limités à 255 caractères. La plupart des SGBD proposent des types, tels que TEXT ou CLOB (Character Long Object), pour représenter des chaînes de caractères longues, jusqu'à 65000 caractères par exemple.	**Les types chaîne de caractères standard** Chaînes de longueur fixe CHAR(X) Chaînes de longueurs variables VARCHAR(X)
Les types date standard Les types date dont introduits avec la norme SQL2. On distingue : DATE qui représente une date selon un format de type "AAAA-MM-JJ" ; et DATETIME qui représente une date plus une heure, dans un format tel que "AAAA-MM-JJ HH:MM:SS".	**Les types date standard** Date (AAAA-MM-JJ) DATE Date et heure (AAAA-MM-JJ HH:MM:SS) DATETIME
Les autres types En fonction du SGBD, il peut exister de nombreux autres types. On peut citer par exemple : MONEY pour représenter des décimaux associés à une monnaie, BOOLEAN pour représenter des booléens, BLOB (pour Binary Long Oject) pour représenter des données binaires tels que des documents multimédia (images bitmap, vidéo, etc.) ...	**Les autres types** En fonction du SGBD, il peut exister de nombreux autres types. On peut citer par exemple : MONEY pour représenter des décimaux associés à une monnaie, BOOLEAN pour représenter des booléens, BLOB (pour Binary Long Oject) pour représenter des données binaires tels que des documents multimédia (images bitmap, vidéo, etc.) ...
La valeur NULL L'absence de valeur, représentée par la valeur NULL, est une information fondamentale en SQL, qu'il ne faut pas confondre avec la chaîne espace de caractère où bien la valeur 0. Il ne s'agit pas d'un type à proprement parler, mais d'une valeur possible dans tous les types. Par défaut en SQL NULL fait partie du domaine, il faut l'exclure explicitement par la clause NOT NULL après la définition de type, si on ne le souhaite pas.	**La valeur NULL** L'absence de valeur, représentée par la valeur NULL, est une information fondamentale en SQL, qu'il ne faut pas confondre avec la chaîne espace de caractère où bien la valeur 0. Il ne s'agit pas d'un type à proprement parler, mais d'une valeur possible dans tous les types. Par défaut en SQL NULL fait partie du domaine, il faut l'exclure explicitement par la clause NOT NULL après la définition de type, si on ne le souhaite pas.

Figure 61 - Maquette de tabulation d'un grain de contenu Opale.

Dans certains cas, deux contenus en opposition peuvent être très ressemblants (par exemple dans le cas Quick, seul un chiffre peut changer entre deux déclinaisons d'une procédure, selon le pays dans lequel elle s'applique), ce qui rend difficile l'identification de leurs différences. Nous proposons d'enrichir la tabulation d'une option permettant d'afficher le différentiel entre les deux contenus.

Nous faisons l'hypothèse que la tabulation rend la relecture du contenu plus efficace :

- les contenus communs aux deux déclinaisons peuvent être relus ensemble grâce à leur synchronisation verticale ;
- les contenus singuliers de l'une ou l'autre des déclinaisons sont rapidement identifiables.

Dans cette section, nous allons analyser quelques propriétés formelles des déclinaisons afin de proposer un algorithme de tabulation. Nous allons étudier ces propriétés dans le cas d'un modèle à deux déclinaisons (Opale), puis trois (Juriguide).

4.2.1 Définition d'une déclinaison

Soit un modèle documentaire permettant d'associer à certains contenus d'un fragment la métadonnée suivante :

```
1 Filter {
2   excludeFromD1 : bool;
3   excludeFromD2 : bool;
4 }
```

La déclinaison D1 (resp. D2) est la transformation du fragment filtrant (ignorant, excluant) les contenus pour lesquels *excludeFromD1 = true* (resp. *excludeFromD2 = true*). Autrement dit, D1 est la projection du fragment sur F1, F1 désignant l'ensemble des valeurs de la métadonnée telles que :

- *excludeFromD1 = false*,
- *excludeFromD2 = false* ou *excludeFromD2 = true*.

On dira d'un contenu dont la valeur de la métadonnée appartient à F1 (resp. F2) qu'il est fléché pour D1 (resp. D2). Les contenus fléchés simultanément pour F1 et F2 seront dits communs, tandis que les

contenus fléchés exclusivement pour F1 ou F2 seront dits singuliers.

Exemple

Dans Opale, un module a deux déclinaisons : les versions standard et courte. La métadonnée permettant de filtrer les contenus se matérialise dans la forme d'édition avec l'iconographie suivante :

- : *excludeFromStandard = false*
- : *excludeFromStandard = true*
- : *excludeFromShort = false*
- : *excludeFromShort = true*

On retrouve cette métadonnée :

- au niveau du grain, pour filtrer une balise pédagogique entière ou bien une partie du contenu de celle-ci via la balise *Contenu filtrable* (cf. exemple précédant (p. 82)) ;
- au niveau d'un module, d'une division, d'une activité d'apprentissage, etc. où il est possible de filtrer un grain, une division, etc., tel que dans l'exemple ci-dessous.

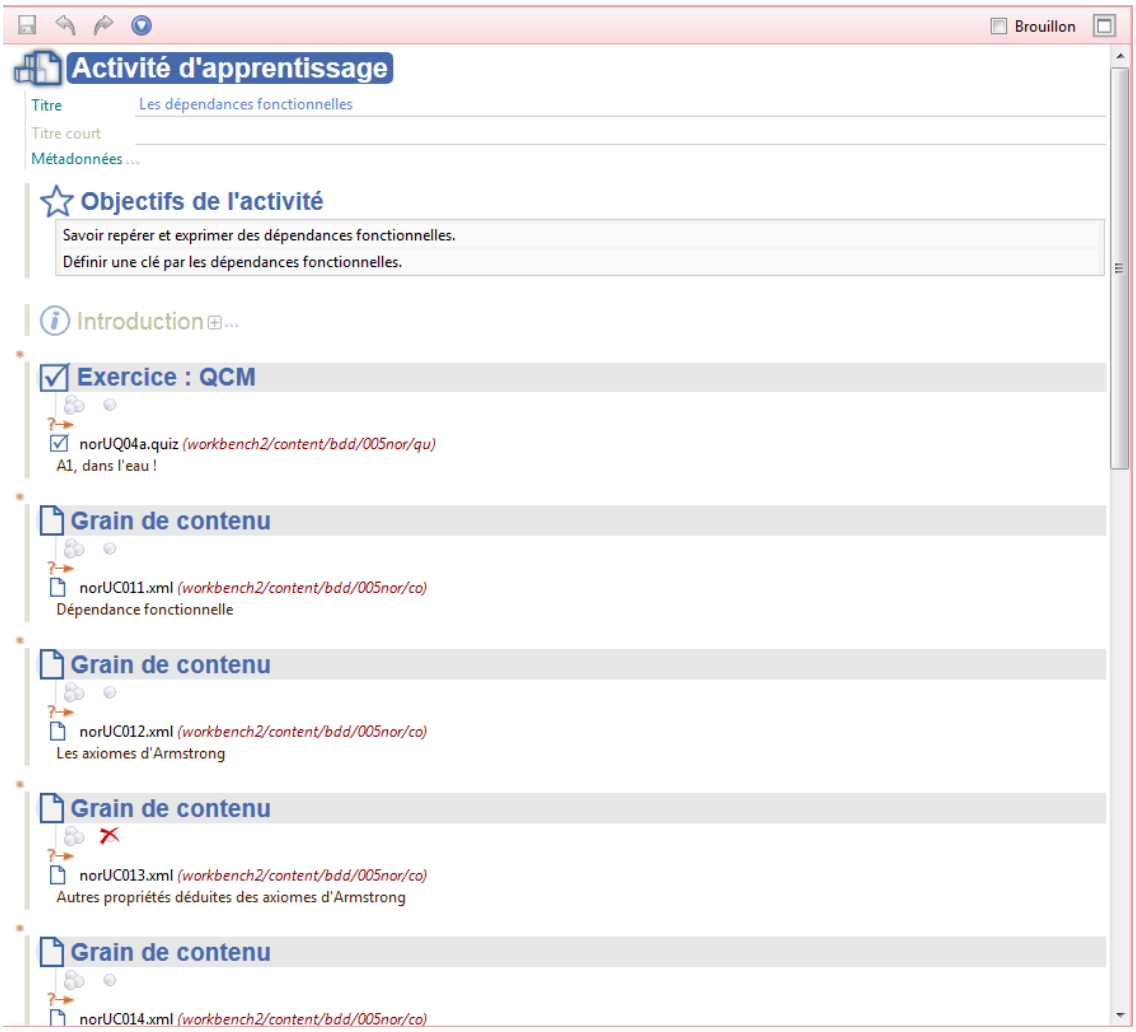


Figure 62 - Activité d'apprentissage déclinée.

4.2.2 Représentation visuelle

Dans la suite, nous utiliserons un *treemap* pour visualiser l'ensemble des déclinaisons d'un fragment ainsi que de ses fragments inférieurs (les activités et les grains d'un module par exemple). Cette représentation permet également d'abstraire, à partir de motifs visuels, certains schémas de déclinaison

pouvant refléter des usages propres au corpus étudié (habitudes ou intentions de l'auteur) et/ou au modèle documentaire correspondant. Ces schémas de déclinaison nous aideront par la suite pour définir une stratégie de tabulation.

Un treemap est une visualisation permettant de représenter une hiérarchie d'informations, soit un arbre, dans un zone rectangulaire de taille fixe. On considère qu'à chaque feuille (nœud terminal de l'arbre) est associée une valeur de taille, et à chaque branche (nœud ayant des nœuds fils) un label. Dans le treemap, chaque feuille est représentée par un rectangle dont l'aire est proportionnelle à la taille de cette feuille. Au niveau le plus bas, les feuilles d'une même branche sont regroupées spatialement de façon à ce qu'elles forment un rectangle, lui même regroupé avec les autres rectangles (feuilles ou branches) du même niveau, et ainsi de suite jusqu'à ce que l'arbre complet soit représenté. Le label d'une branche apparaît au-dessus du rectangle formé par regroupement de ses feuilles et sous-branches. La position relative de deux rectangles d'un même niveau, c'est-à-dire correspondant à deux nœuds frères, ne respecte pas nécessairement l'ordre de ces nœuds dans l'arbre (un nœud peut être spatialisé en dessous ou à droite d'un autre, même si le premier précède le second dans l'ordre des nœuds fils du nœud parent commun).

La première visualisation de ce type a été proposée par Ben Shneiderman (1992), qui l'a utilisée pour représenter son système de fichiers afin de voir la taille relative de ses dossiers, sous-dossiers et fichiers. Dans cette arborescence, les branches sont les dossiers, les feuilles sont les fichiers et leur taille est l'espace mémoire occupé par le fichier sur le disque.

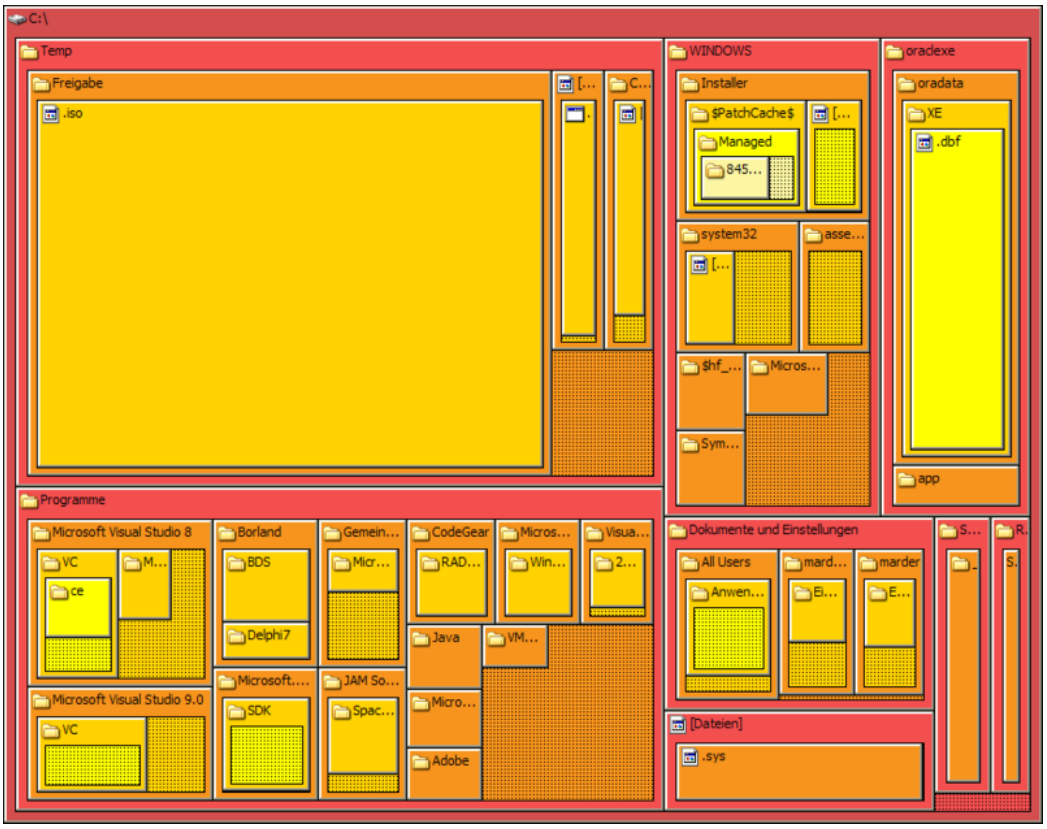


Figure 63 - Treemap représentant l'utilisation de l'espace sur un disque dur (source : https://en.wikipedia.org/wiki/File:Tree_Map.png).

Dans l'utilisation que nous faisons du treemap, nous considérons comme feuille chaque contenu en dessous duquel le contenu ne peut plus être filtré (par restriction du modèle documentaire), et comme branche, inversement, chaque contenu en dessous duquel le contenu peut (encore) être filtré. La taille d'une feuille est calculée en sommant le nombre de caractères du contenu, tandis que la couleur du rectangle correspondant à cette feuille est déterminée par le fléchage du contenu. Seront ainsi de même couleur les feuilles de D1 d'une part, les feuilles de D2 d'autre part et enfin les feuilles communes à D1 et D2.

Exemple : treemap d'un module

Le treemap ci-dessous représente le module "Le niveau physique : Le langage SQL". Chaque rectangle correspond à un niveau hiérarchique du contenu (module, division, activité, grain, balise pédagogique...), dont le titre est donné dans le label (à défaut de titre, on utilise le type de contenu). On retrouve en orange les contenus fléchés pour la version standard, en rouge les contenus fléchés pour la version courte, et en bleu les contenus communs aux deux versions.

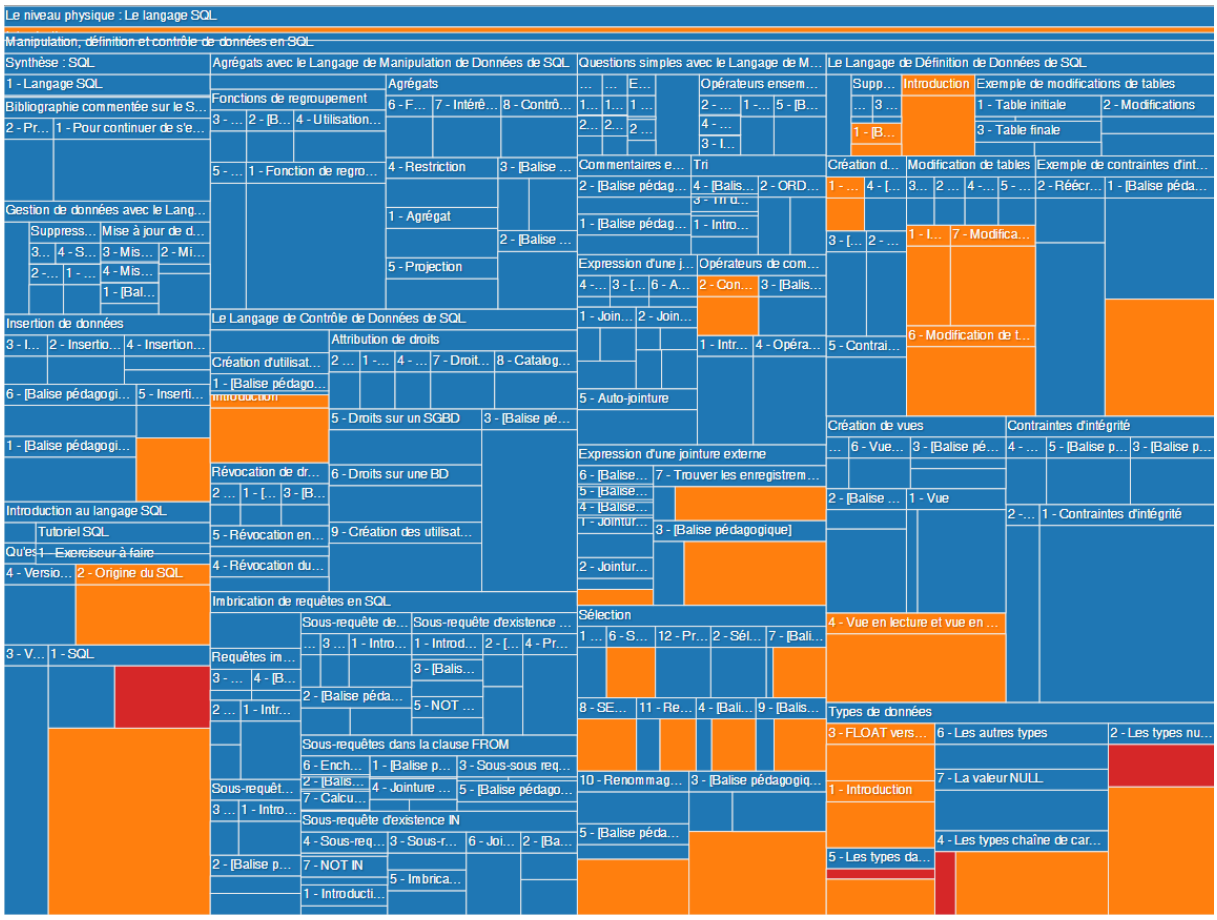


Figure 64 - Treemap d'un module Opale.

Zoom du treemap

Dans la suite, nous présenterons des treemaps plus détaillés en "zoomant" le treemap du module sur des contenus de niveau inférieur (grains, activités d'apprentissage...). Dans ces zooms, nous numéroterons chaque rectangle afin d'identifier son ordre dans le nœud parent (en effet, comme nous l'avons dit plus haut, la position spatiale d'un rectangle est indépendante de l'ordre d'apparition des contenus).

4.2.3 Divergence et déclinaisons sous-jacentes

Définitions

Soit un fragment et C un contenu quelconque de ce fragment. On note $\alpha F1(C)$ (resp. $\alpha F2(C)$) l'ensemble constitué des contenus enfants de C fléchés pour D1 (resp. D2), et $\beta F1(C)$ (resp. $\beta F2(C)$) l'ensemble constitué des contenus descendants de C fléchés pour D1 (resp. D2). Si C est fléché pour D1 et D2, on dira que :

1. C est divergeant si $\alpha F1(C) \cap \alpha F2(C) = \varnothing$ (aucun contenu enfant n'est fléché pour D1 et D2 à la fois) ;

2. C n'a pas de déclinaison sous-jacente si βF1(C) = βF2(C).

On note πF1(C), πF2(C) et πF1,F2(C) les projections de C respectivement sur F1, F2, F1 et F2.

Exemple

Le treemap suivant représente le grain "Types de données" vu précédemment (p. 82) :

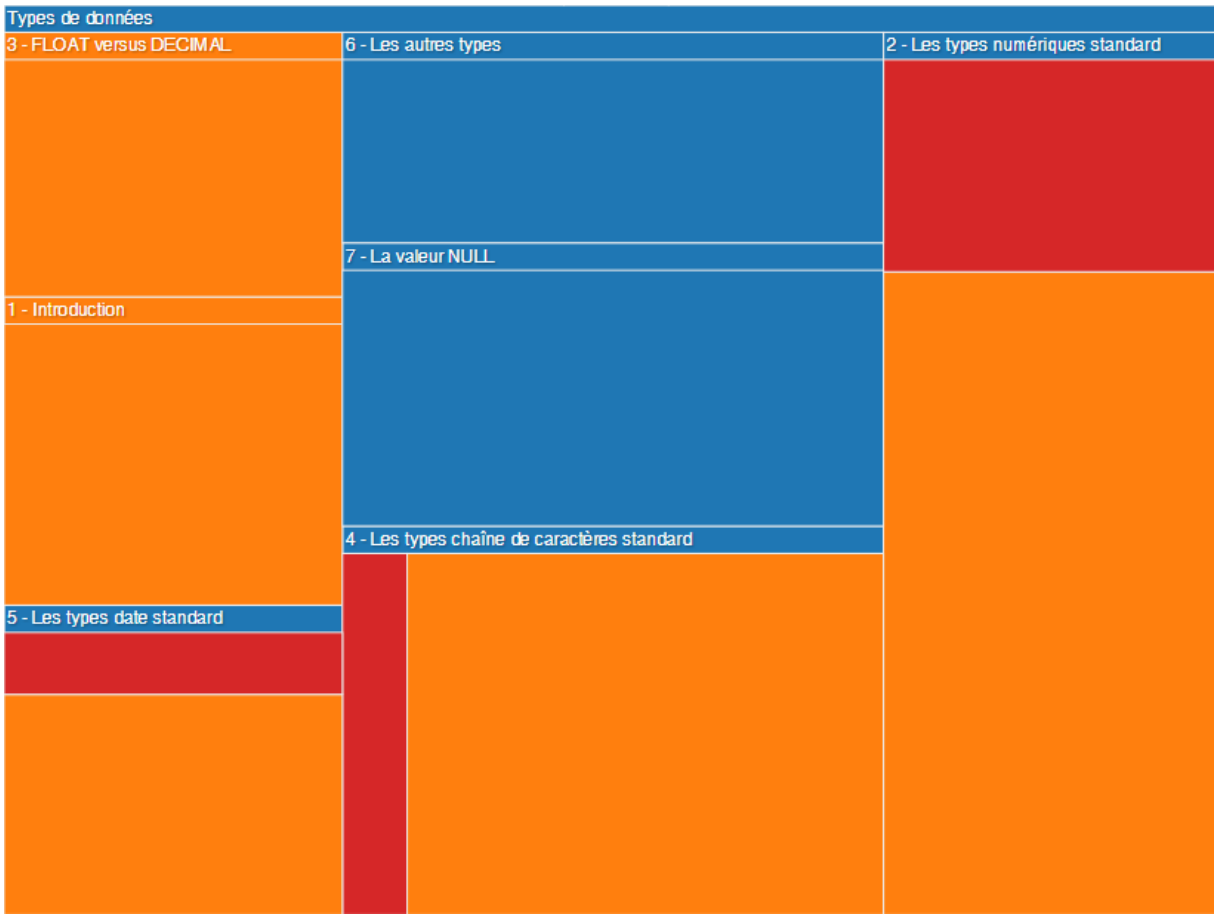


Figure 65 - Treemap d'un grain de contenu (1).

On établit les ensembles suivants :

- αs0 (resp. αc0), soit l'ensemble des enfants du grain fléchés pour la version standard (resp. version courte) :
  - αs0 = {1, 2, 3, 4, 5, 6, 7}
  - αc0 = {2, 4, 5, 6, 7}
- βs0 (resp. βc0), soit l'ensemble des descendants du grain fléchés pour la version standard (resp. version courte) :
  - βs0 = {1, 1.1, 2, 2.2, 3, 3.1, 4, 4.2, 5, 5.2, 6, 6.1, 7, 7.1}
  - βc0 = {2, 2.1, 4, 4.1, 5, 5.1, 6, 6.1, 7, 7.1}
- αsi, αci, βsi, βci (i > 0), c'est-à-dire l'équivalent des ensembles précédant au niveau des balises pédagogiques, prises dans l'ordre des i :
  - αs1 = {1.1}
  - αs3 = {3.1}
  - αc1 = αc3 = ∅
  - αs2 = {2.2}
  - αc2 = {2.1}

- αs4 = {4.2}
- αc4 = {4.1}
- αs5 = {5.2}
- αc5 = {5.1}
- αs6 = {6.1}
- αc6 = {6.1}
- αs7 = {7.1}
- αc7 = {7.1}
- pour tous les βsi (resp. βci), i > 1 αsi = βsi (resp. αci = βci)

Nous déduisons de ces ensembles les propriétés suivantes :

- Le grain "Types de données" :
  - a des déclinaisons sous-jacentes ;
  - n'est pas divergeant (présence de balises pédagogiques communes) ;
- Les balises pédagogiques n°2, 4 et 5 :
  - ont des déclinaisons sous-jacentes ;
  - sont divergentes (il n'y a aucun contenu commun) ;
- Les balises pédagogiques n°6 et 7 :
  - n'ont pas de déclinaison sous-jacente ;
  - par conséquent, ne sont pas divergentes (elles sont chacune strictement identiques dans les deux versions).

Notons que l'on peut aussi repérer visuellement ces propriétés de la manière suivante :

- les contenus sans déclinaison sous-jacente sont les rectangles de label bleu imbriquant uniquement des rectangles bleus ;
- les contenus divergents sont les rectangles de label bleu n'imbriquant que des rectangles oranges ou rouges.

Algorithme de tabulation

Soit un contenu C0 commun à D1 et D2 dont on supposera qu'il a des déclinaisons sous-jacentes et qu'il n'est pas divergeant. Pour chaque contenu enfant de C0, noté C, nous proposons de construire la tabulation de la manière suivante :

- si C est commun à D1 et D2 et sans déclinaison sous-jacente (exemples : BP n°6 et 7) : une ligne d'équivalence comportant πF1,F2(C) dans chaque case (à partir de C, D1 et D2 sont identiques "jusqu'au bout") ;
- si C est commun à D1 et D2 et divergeant (exemples : BP n°2, 4 et 5) : une ligne d'alternative opposant πF1(C) dans la case de gauche à πF2(C) dans la case de droite ;
- si C est fléché uniquement pour D1 (resp. D2) : une ligne opposant πF1(C) dans la case de gauche (resp. πF2(C) dans la case de droite) à une case vide dans la case de droite (resp. de gauche).

4.2.4 Contenu partiellement divergeant

Définition

Soit un contenu C commun à D1 et D2, non-divergeant. On dira que C est partiellement divergeant si αF1(C) ≠ ∅ et αF2(C) ≠ ∅.

Exemple

Le grain suivant est un exemple de contenu partiellement divergeant :

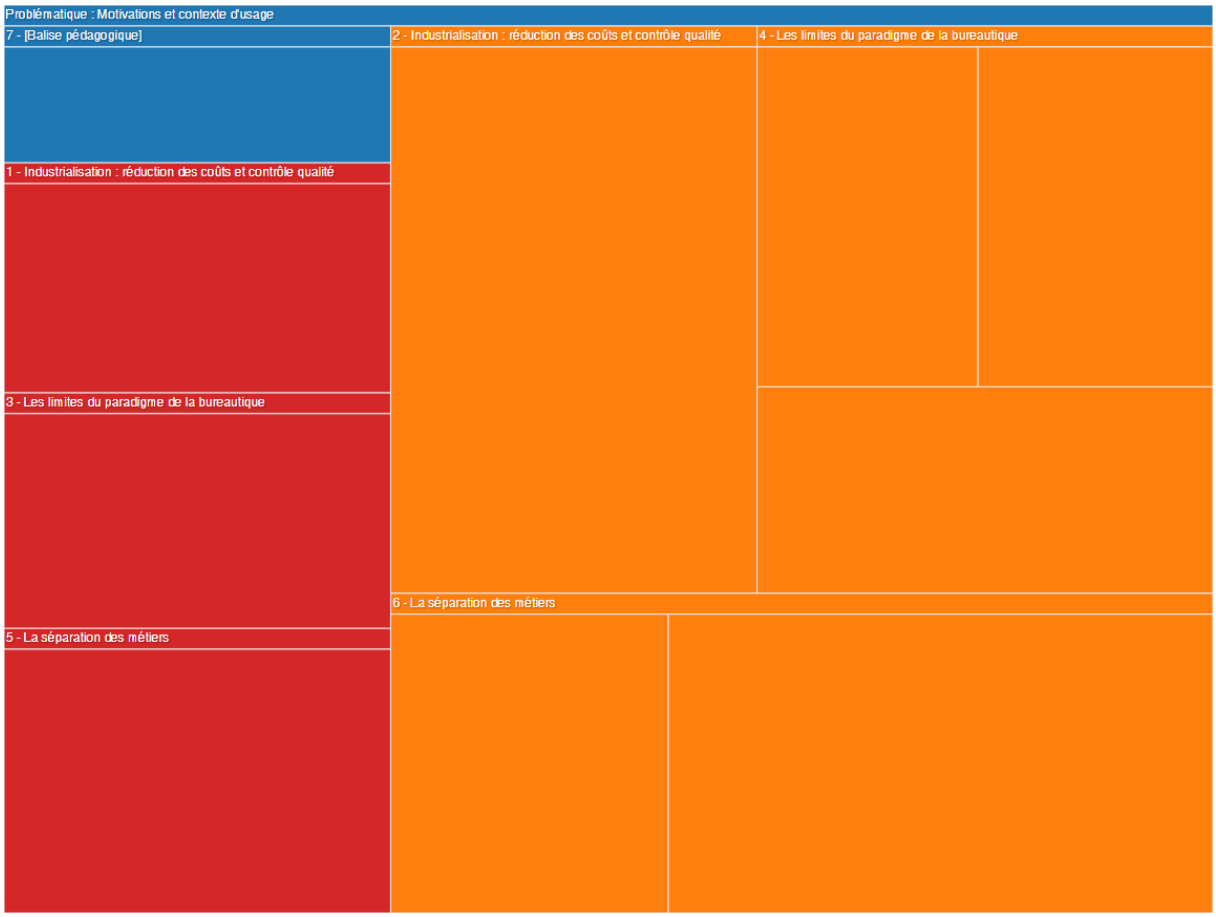


Figure 66 - Treemap d'un grain de contenu (2).

Il est possible de repérer visuellement ce type de contenu : il s'agit de rectangles de label bleu imbriquant un mélange de rectangles bleus, oranges et rouges.

Si l'on applique à ce grain l'algorithme de tabulation proposé plus haut, on obtient le résultat ci-dessous. Notons que cela produit un effet de "quinconce" sur les BP, ce qui rend la relecture plus difficile. Étant donné que le contenu diverge hormis au niveau de la dernière BP, il serait plus judicieux d'opposer les BP n°1, 3 et 5 d'une part aux BP n°2, 4 et 6 d'autre part, ceci afin de "condenser" la tabulation.

Figure 67 - Effet de "quinconce" dans la tabulation d'un grain partiellement divergeant.

L'algorithme doit donc être complété en prenant en compte le cas des contenus partiellement divergeant. Une première solution consiste à projeter entièrement C (une ligne d'alternative opposant πF1(C) dans la case de gauche à πF2(C) dans la case de droite) :

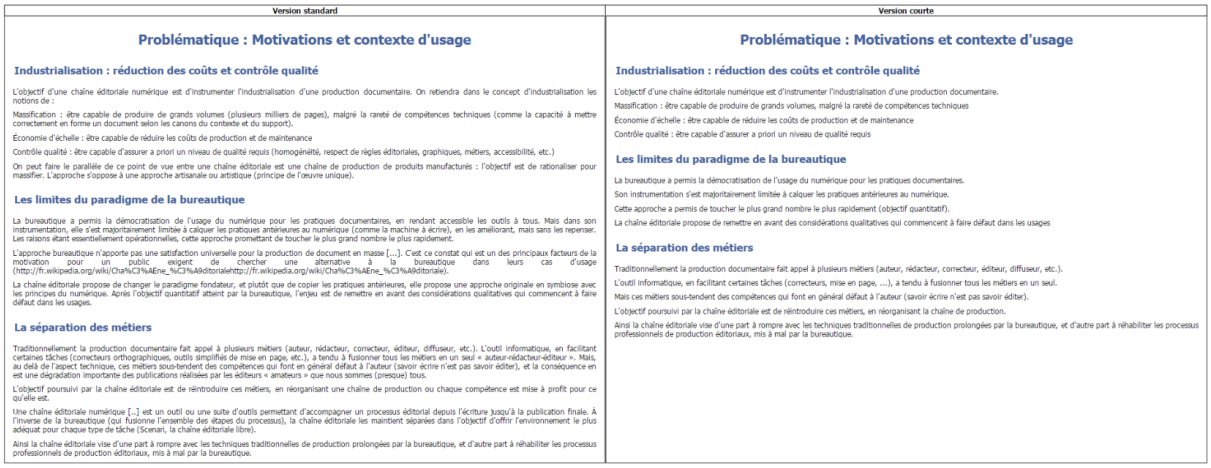


Figure 68 - Projection complète des deux déclinaisons d'un grain partiellement divergeant.

Cette solution paraît bancale dans la mesure où elle mène potentiellement à des déphasages entre les contenus communs à D1 et D2 dans C.

Segment divergeant

Soit αF1(C, i, j) (resp. αF2(C, i, j)) l'ensemble constitué des contenus enfants de C fléchés pour D1 (resp. D2) *entre les indices i et j*. On définira également les projections *locales* de C, πF1(C, i, j) (resp. πF2(C, i, j)). Enfin, on appellera *segment divergeant* un couple (i, j) tel que :

- $\alpha F1(C, i, j) \cap \alpha F2(C, i, j) = \varnothing$
- le contenu à l'index i-1 (resp. j+1), s'il existe, est commun à D1 et D2.

Une seconde solution consiste à poser, pour tout segment divergeant (i, j) de C, une ligne d'alternative opposant πF1(C, i, j) dans la case de gauche à πF2(C, i, j) dans la case de droite :

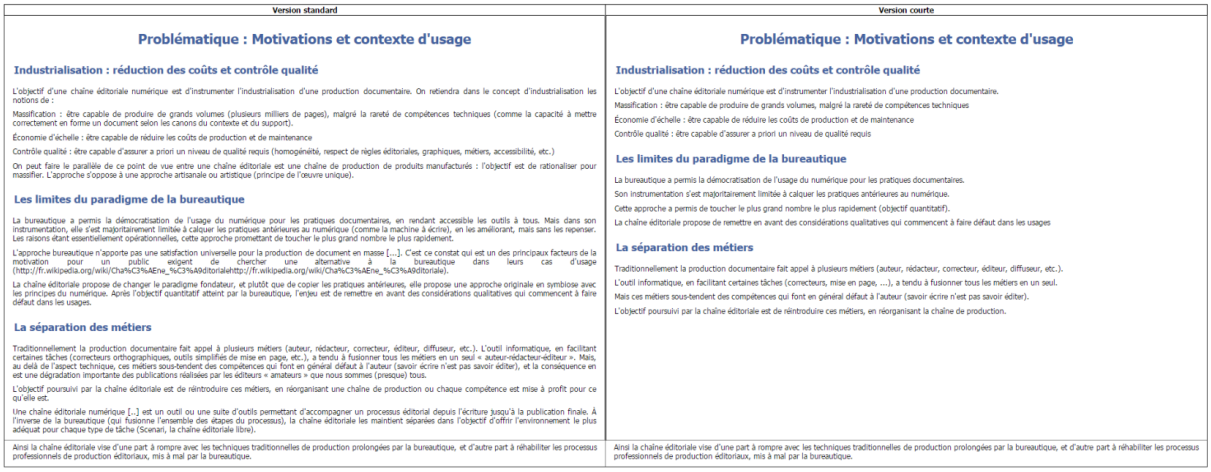


Figure 69 - Tabulation d'un grain prenant en compte la présence d'un segment divergeant.

4.2.5 Définitions généralisées

Soit un modèle documentaire permettant d'associer à certains contenus d'un fragment la métadonnée suivante :

```
Filter {
  excludeFromD1 : bool;
  excludeFromD2 : bool;
  ...
  excludeFromDN : bool;
}
```

La déclinaison Di est la transformation du fragment filtrant (ignorant, excluant) les contenus pour lesquels *excludeFromDi = true*. Autrement dit, Di est la projection de FG sur Fi, Fi désignant l'ensemble des valeurs de la métadonnée telles que :

- *excludeFromDi = false*,
- *excludeFromDj = false* ou *excludeFromDj = true* pour tout j différent de i.

On dira d'un contenu qu'il est fléché *exclusivement* pour Di si la valeur de sa métadonnée appartient à Fi et n'appartient à aucun autre Fj, j différent de i. Si la deuxième condition n'est pas respectée, ce contenu est fléché *entre autres* pour Di, et il faut le considérer comme commun à l'ensemble des déclinaisons dont il n'est pas exclu. En effet, les Fi peuvent s'intersecter : par exemple F1 ∩ F2 implique que *excludeFromD1 = false* et *excludeFromD2 = false*. Le nombre de combinaisons possibles des Fi est donné par la formule $\sum_{k=1}^{N} C_N^k$, où C_N^k désigne le nombre de combinaisons de k filtres parmi N. Ce nombre vaut :

- pour N = 2 : 3 combinaisons (dans Opale : standard, court et standard-court = commun)
- pour N = 3 : 7 combinaisons (dans Juriguide : EC, AD, PC, EC-AD, AD-PC, EC-PC et EC-AD-PC = commun)
- pour N = 4 : 15 combinaisons (pas d'exemple connu)
- etc.

Par conséquent, les ensembles α, β ainsi que la projection π sont généralisés à toutes les combinaisons de Fi et on les notera αFi1[, Fi2, ...](C), βFi1[, Fi2, ...](C) et πFi1[, Fi2, ...](C).

4.2.6 Tabulation sur un modèle à trois déclinaisons

Le modèle Juriguide permet de publier des fiches juridiques dont les contenus peuvent être filtrés en fonction de la convention collective concernée : EC, AD et PC. Les contenus peuvent être filtrés au niveau des sections ou des blocs :

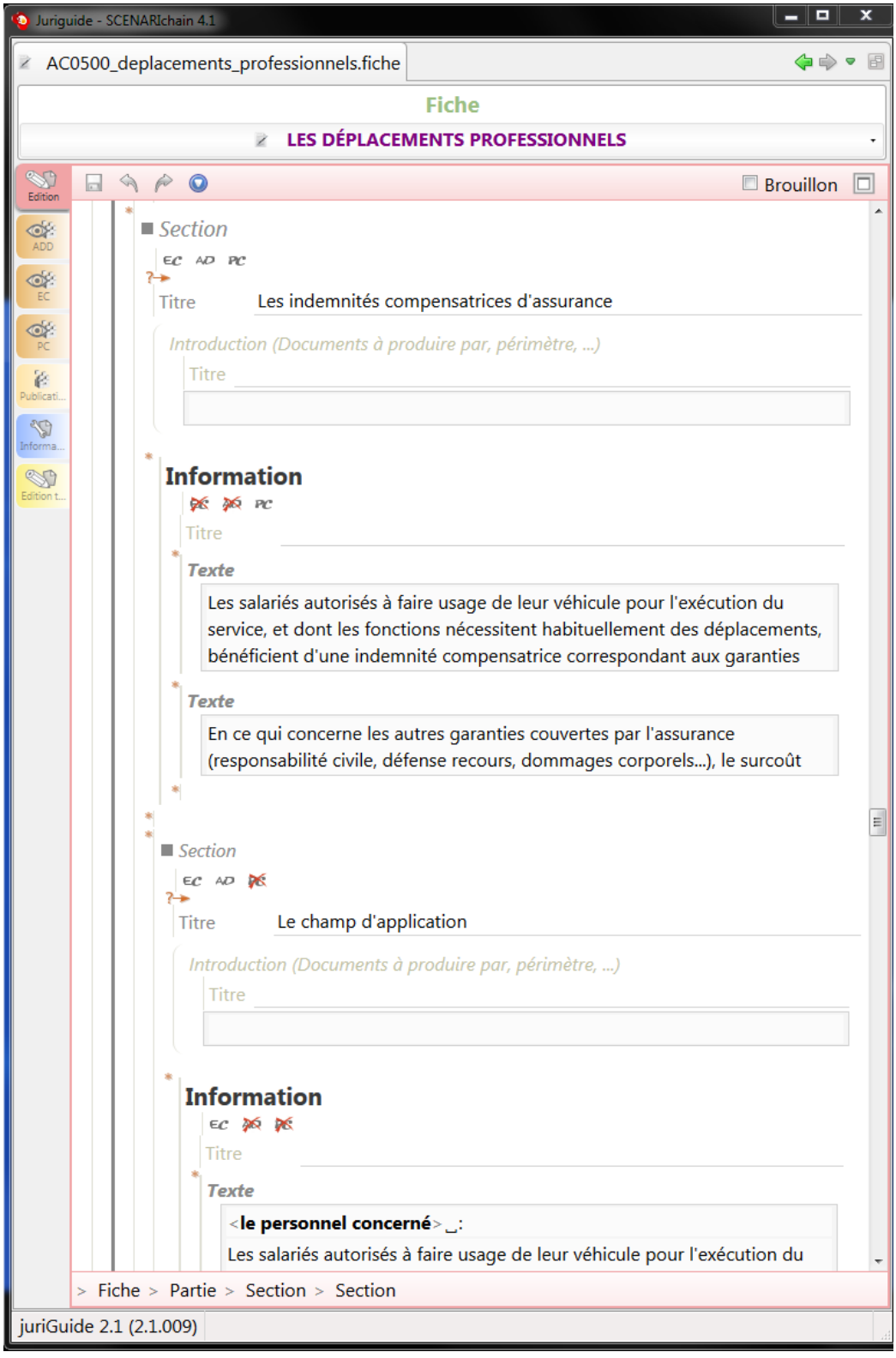


Figure 70 - Fiche juridique déclinée (modèle Juriguide).

Exemple : treemap d'une fiche

Le treemap ci-dessous représente la fiche "Les déplacements professionnels". Les labels des rectangles représentent les titres des sections et sous-sections, tandis que les rectangles terminaux (feuilles) correspondent aux blocs d'intentionnalité. Les correspondances entre les couleurs et les déclinaisons sont les suivantes :

- bleu : commun à EC, AD et PC ;
- violet : commun à EC et AD, exclu de PC ;
- marron : commun à AD et PC, exclu de EC ;
- rose : commun à EC et PC, exclu de AD :
- orange : EC seul ;
- rouge : AD seul ;
- vert : PC seul.

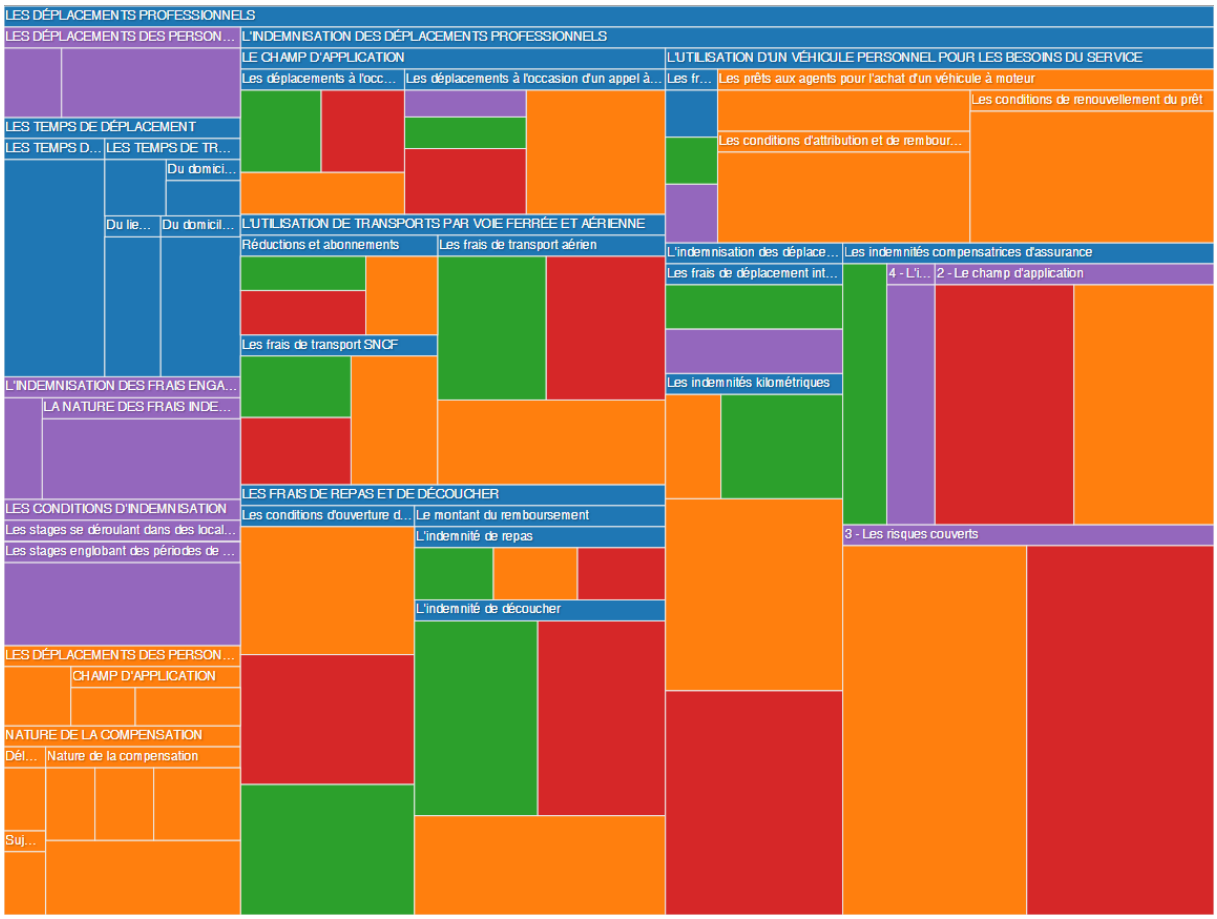


Figure 71 - Treemap d'une fiche.

Dans la suite, on appellera C un contenu fléché pour au moins deux déclinaisons. Visuellement, un tel contenu a un label bleu, violet, marron ou rose.

Redéfinition de la divergence

C est divergent si l'une des conditions suivantes est respectée :

- $\alpha EC(C) \cap \alpha AD(C) \cap \alpha PC(C) = \varnothing$ (rectangles oranges, rouges et verts uniquement)
- $\alpha EC,AD(C) \cap \alpha PC(C) = \varnothing$ (rectangles violets et verts uniquement)
- $\alpha AD,PC(C) \cap \alpha EC(C) = \varnothing$ (rectangles marrons et oranges uniquement)
- $\alpha EC,PC(C) \cap \alpha AD(C) = \varnothing$ (rectangles roses et rouges uniquement)

Redéfinition de l'absence de déclinaison sous-jacente

C n'a pas de déclinaison sous-jacente si l'une des conditions suivantes est respectée :

- $\beta EC(C) = \beta AD(C) = \beta PC(C)$ (sous-arbre bleu)
- $\beta EC,AD(C) = \varnothing$ (sous-arbre vert) ou $\beta AD,PC(C) = \varnothing$ (sous-arbre orange) ou $\beta EC,PC(C) = \varnothing$ (sous-arbre rouge)
- $\beta EC(C) = \beta AD(C)$ et $\beta PC(C) = \varnothing$ (sous-arbre violet)
- $\beta AD(C) = \beta PC(C)$ et $\beta EC(C) = \varnothing$ (sous-arbre marron)
- $\beta EC(C) = \beta PC(C)$ et $\beta AD(C) = \varnothing$ (sous-arbre rose)

Chapitre 5

Expérimentations

5.1 Contextes d'usage 95
5.1.1 Relecture de contributions étudiantes 95
5.1.2 Qualification de la banque de questions de Faq2Sciences 97
5.2 Outils proposés 98
5.2.1 Existant 99
5.2.2 Modélisation 101
5.2.3 Outil de différentiel 102
5.3 Retours d'usage 104
5.3.1 Contributions étudiantes 104
5.3.2 Banque de questions 107
5.4 Évaluation 109
5.4.1 Positionnement épistémologique 109
5.4.2 Bilan de notre recherche 110

Ce chapitre a pour but de présenter les propositions que nous avons pu instrumenter et confronter à un usage réel. Nous préciserons les deux contextes abordés, la relecture de contributions étudiantes d'une part et la qualification d'une banque de questions d'autre part. Après avoir rappelé l'existant technologique, nous détaillerons les outils mis au point pour ces contextes, et plus particulièrement un prototype d'affichage de différentiel basé sur un algorithme développé par Kelis. Enfin, nous donnerons les retours d'usage de ces expérimentations, à partir desquels nous discuterons de l'évaluation de nos travaux.

5.1 Contextes d'usage

5.1.1 Relecture de contributions étudiantes

Dans le cadre des cours NF29 *(https://stph.scenari-community.org/nf29/index.html)* et API04 *(https://*

stph.scenari-community.org/api04/index.html) enseignés à l'UTC par Stéphane Crozat, il est demandé à chaque étudiant de réaliser, sous la forme d'un module Opale, l'un des deux types de contribution suivants :

- fiche de lecture pédagogique (analyse d'un texte, mise en perspective, liste de citations, glossaire, questionnaire, exercices de réflexion...) ;
- cours de technologie (partie théorique et partie pratique/application pouvant prendre la forme d'un tutoriel, d'un TP à réaliser...).

Les modules font l'objet d'un exposé oral avec support (format web diaporama) et d'une publication sur un site web (format web classique). Il est conseillé aux étudiants d'utiliser les filtres de version courte/standard afin notamment de ne pas surcharger le diaporama.

Revue pair à pair

Chaque étudiant participe également à la revue et l'évaluation du travail (rédaction et présentation) d'un autre étudiant, et se place pour cela dans une posture critique. En outre, chaque étudiant procède à une auto-évaluation de son travail. Pour ce faire, les étudiants sont amenés à annoter les contenus des modules :

Figure 72 - Commentaires dans l'éditeur Scenari suite à la revue pair à pair.

Relecture par l'enseignant

Après la revue pair à pair, l'enseignant fait une revue du module et des commentaires afin de les valider, de les compléter ou de les nuancer (posture critique et/ou régulatrice). Une phase d'amélioration est alors proposée à l'auteur du module, à l'issue de laquelle l'enseignant procède à une relecture finale. Par ailleurs, certains sujets de contribution étant récurrents et améliorables d'une année à l'autre, l'enseignant peut laisser des commentaires sous la forme d'aide-mémoires pour les étudiants de l'année suivante.

> Dans les deux cas d'amélioration, l'enseignant a besoin d'une fonction de différentiel pour distinguer les nouveaux contenus de ceux qu'il a déjà lus, et ainsi relire plus efficacement. Or, utilisé dans la vue d'édition de Scenari déjà sujette à la fragmentation, au filtrage des contenus et à l'annotation, le différentiel mène a une « saturation de ce qui est possible pour relire, a fortiori le texte de quelqu'un d'autre » (*sic.*).

5.1.2 Qualification de la banque de questions de Faq2Sciences

Le projet Faq2Sciences a impliqué la mise en commun de quiz réalisés par plusieurs universités au sein d'une banque nationale de questions (environ 1200 quiz en 2015).

Les tests de positionnement ont été réalisés à partir de cette banque de questions (QCU et QCM Opale) et d'un modèle dérivé de Topaze ajoutant notamment les deux types de fragment suivants : le groupe de quiz et l'étape de groupes de quiz (composée de plusieurs groupes de quiz).

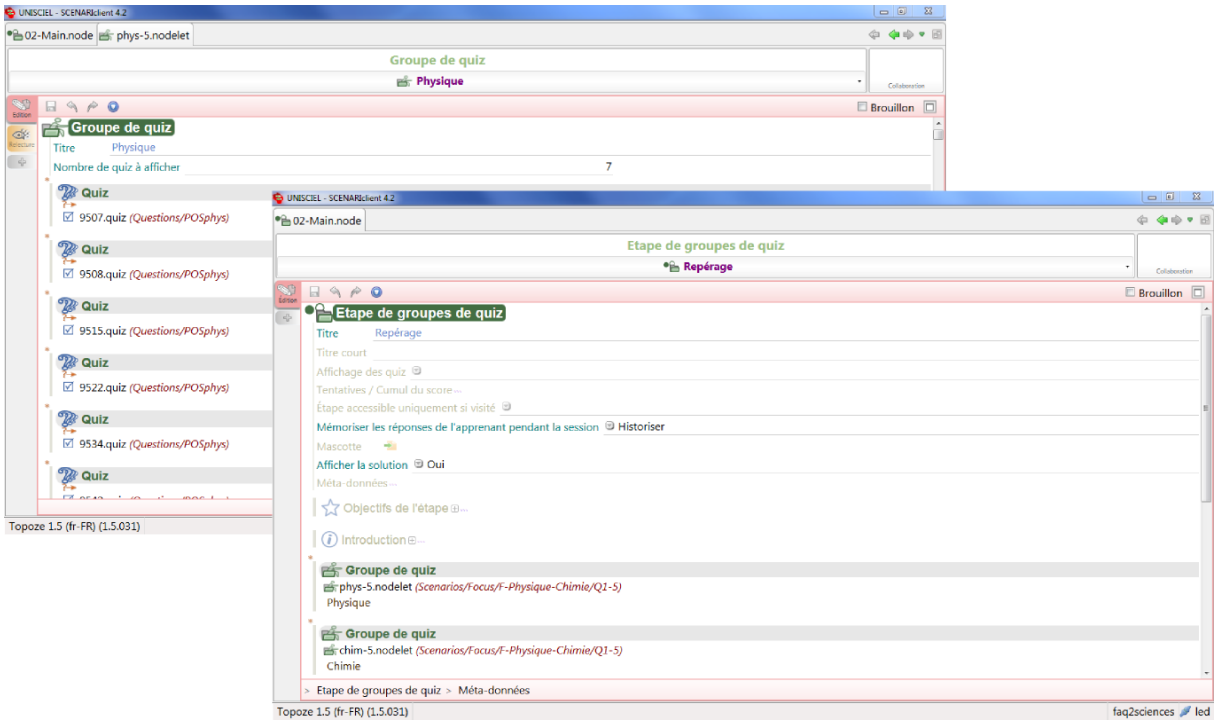


Figure 73 - Groupe de quiz et étape de groupes de quiz.

Dans la publication web du test de positionnement, l'apprenant répond à chaque question sur une page différente, avant de valider l'étape pour voir son score ainsi que les solutions et explications associées aux questions. Afin que l'apprenant puisse refaire l'étape sans tomber sur les mêmes questions, ces dernières sont tirées aléatoirement grâce à un paramètre renseigné par l'auteur dans les groupes de quiz (métadonnée "Nombre de quiz à afficher" dans la figure ci-dessus). Enfin, le parcours à travers les différentes étapes du test est personnalisé en fonction des scores de l'apprenant (cf. enchaînements conditionnés dans Topaze (p. 23)) : par exemple, si son score en physique est trop faible lors de l'étape dite de "repérage" (c'est-à-dire comportant également des questions de mathématiques, de chimie, etc.), un lien vers une étape d'approfondissement sur cette thématique lui sera proposé.

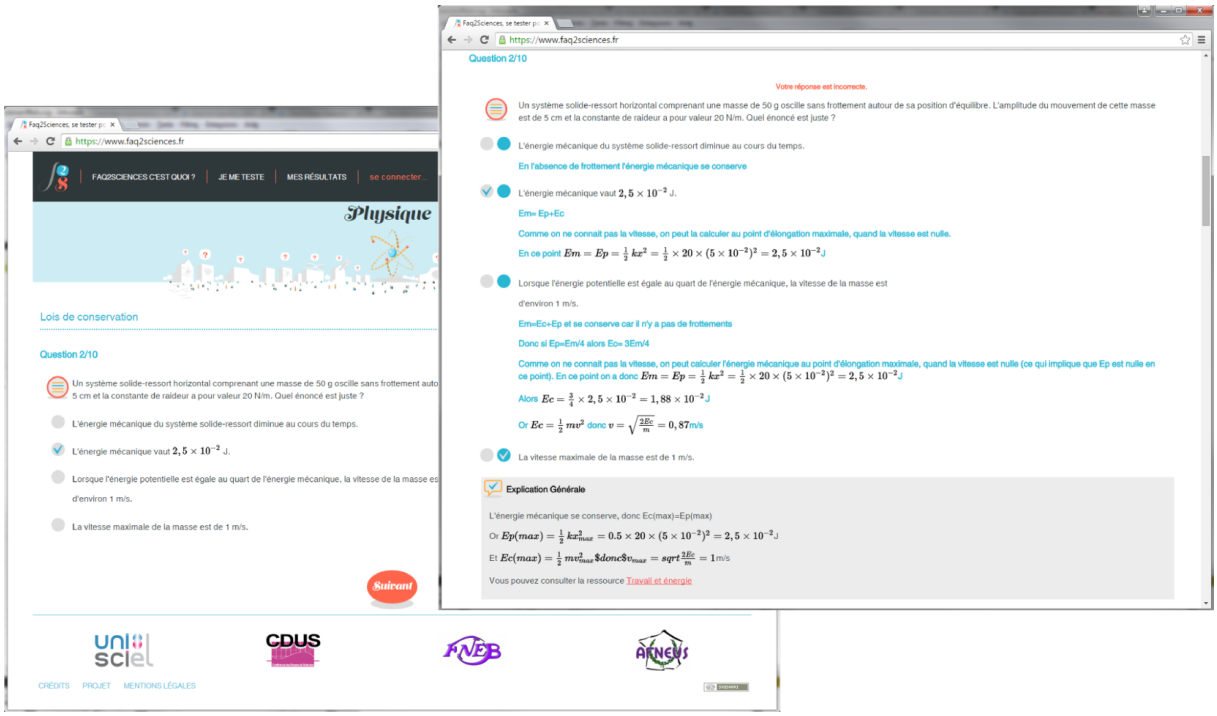


Figure 74 - Publication d'un test de positionnement : questionnaire interactif avec sélection aléatoire.

Revue des questions

En plus de la scénarisation des quiz au sein des tests de positionnement, un enjeu de ce projet a été de qualifier et d'améliorer les questions. Cette revue a impliqué six enseignants et six étudiants pour la relecture et l'annotation des quiz (afin de soulever des erreurs, des imprécisions dans les énoncés, solutions ou explications...), ainsi que deux personnes (le chef de projet et la documentaliste) chargées de l'intégration des modifications suggérées par les annotations dans l'environnement d'édition. Les questions ont été gérées à l'aide d'un cycle de vie comportant trois états : "validé", "rejeté" et "brouillon".

La publication du test de positionnement ne convient pas à la revue des questions, car cette dernière ne demande pas aux relecteurs de se mettre dans la position d'un apprenant. En effet dans cette publication, les aspects suivants apparaissent superflus :

- nécessité de répondre aux questions et de valider ses réponses pour afficher les solutions et explications ;
- variabilité des questions présentées (tirage aléatoire), au détriment de l'exhaustivité requise en relecture ;
- personnalisation du parcours en fonction des scores.

5.2 Outils proposés

Dans cette section seront présentés les outils que nous avons proposés pour instrumenter la relecture dans les contextes d'usage abordés précédemment. Le principal besoin soulevé est celui d'une forme documentaire disposant des fonctions nécessaires à la relecture (annotation et différentiel), ces fonctions étant peu ou pas utilisables dans les formes existantes.

Le premier axe de ce travail a concerné la modélisation de la forme de relecture. En nous appuyant

sur l'existant (une prévisualisation mono-page avec commentaires), nous avons pu tester une partie de la stratégie de linéarisation présentée dans nos propositions. Le second axe a consisté à développer un outil de différentiel (librairie Javascript d'affichage des différences) à intégrer à la forme de relecture.

5.2.1 Existant

Cette section présente l'extension OpaleGenPdf *(http://scenari-platform.org/addons/co/Opale35GenPdf.html)*, sur laquelle se sont basés nos développements dans le cadre des contributions étudiantes.

Publication print (X)HTML

Les évolutions du standard CSS (versions 2.1 puis 3) ont introduit des options de formatage spécifiques aux publications paginées : taille et orientation des pages (exemple : "A4 portrait"), numérotation des pages, compteurs d'éléments HTML (pour le chapitrage), sauts de page, entêtes et pied de page, sélecteur de pages paires ou impaires... Ces options sont implémentées par différentes librairies (Flying Saucer, Prince XML...) pour permettre de générer des documents PDF à partir d'un fichier (X)HTML et d'une feuille de style CSS.

Utilisant le moteur Flying Saucer, l'extension OpaleGenPdf est basée sur un générateur web (pour obtenir une sortie XHTML) et une transformation "mono-page", c'est-à-dire produisant un seul fichier XHTML pour publier tout le contenu du module :

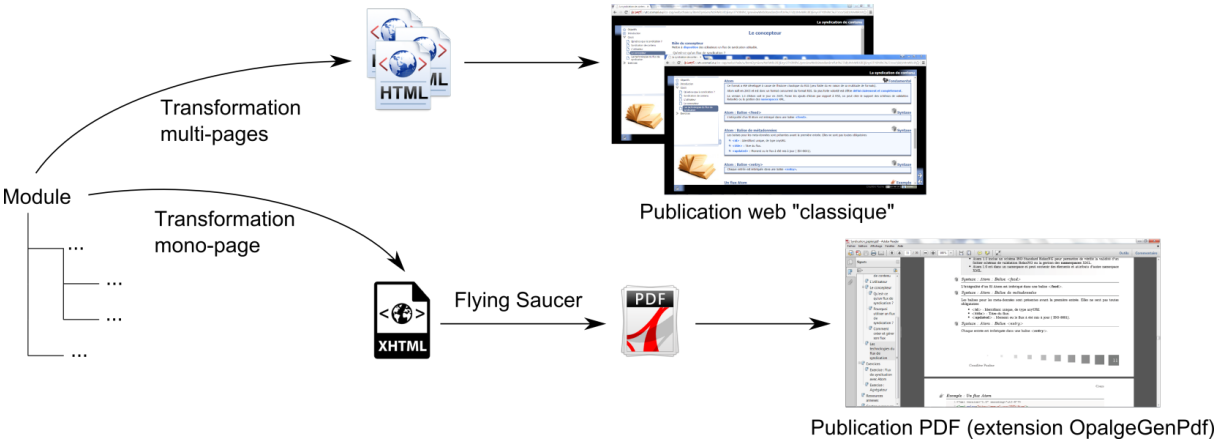


Figure 75 - Transformations multi-pages et mono-page dans Opale.

D'un point de vue technique, la linéarisation peut être mise en œuvre à travers une transformation mono-page des sources XML.

Prévisualisation mono-page avec commentaires

En 2013, Kelis a été sollicité par l'ENAC pour un projet de médiatisation sous Opale. Une médiatisation consiste à intégrer dans une chaîne éditoriale des ressources documentaires existantes, ici des cours destinés à des pilotes de ligne, afin d'en ouvrir les possibilités de rééditorialisation absentes de l'outil de production d'origine, ici un éditeur bureautique.

Durant ce projet, la médiatisation a été effectuée suivant un processus collaboratif impliquant une cellule de production (auteurs formés à Opale) et des relecteurs. Pour ces derniers, une nouvelle prévisualisation avec commentaires a été modélisée, s'appuyant non pas sur la transformation multi-pages de la publication web classique, mais sur la transformation mono-page de la publication print XHTML :

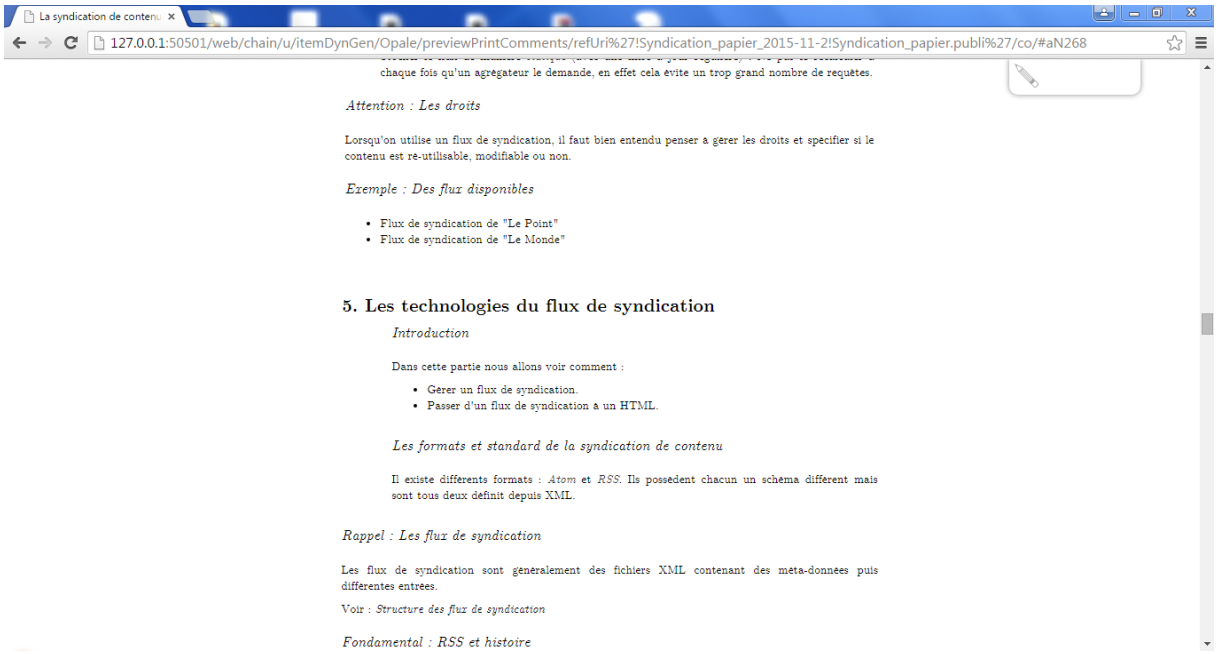


Figure 76 - Prévisualisation mono-page.

Cette prévisualisation permet également d'afficher les commentaires un par un, en surbrillance, via des boutons de navigation :

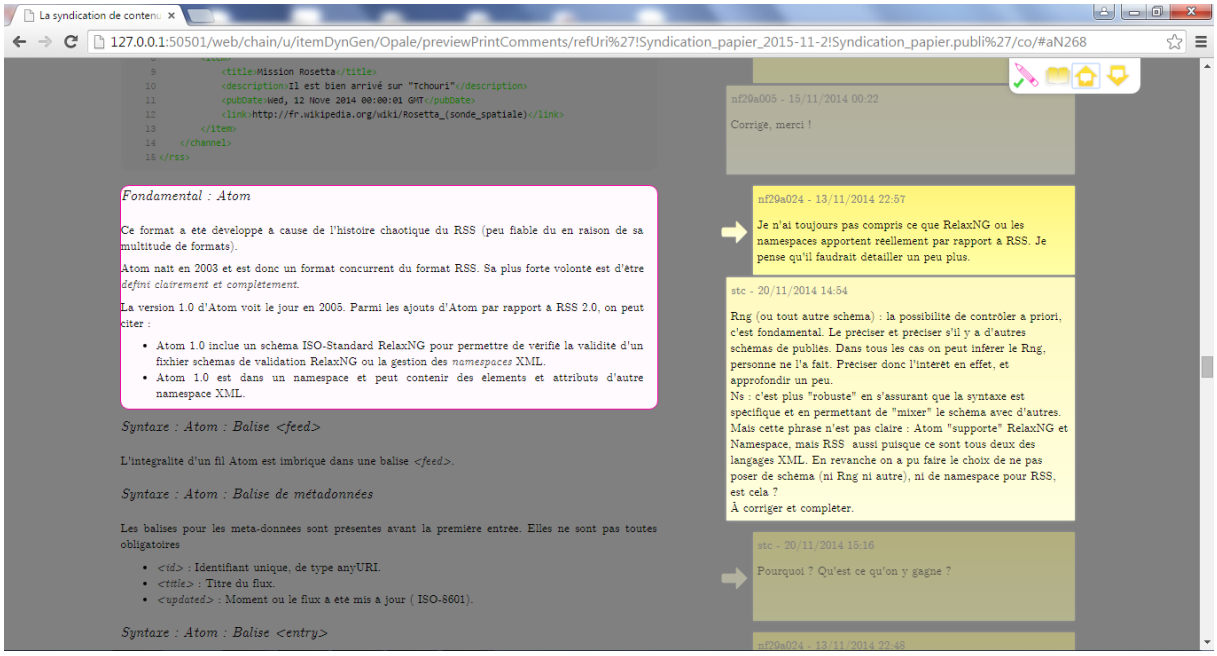


Figure 77 - Prévisualisation mono-page avec navigation à travers les commentaires.

Les relecteurs ont ainsi pu travailler avec une forme documentaire plus proche de l'éditeur bureautique qu'ils utilisaient à l'origine, c'est-à-dire une forme linéaire donnant une vue synoptique de l'ensemble du document et des commentaires associés, quel que soit le niveau du plan. Notons cependant les deux points suivants :

- les renvois vers des grains ont été traités dans la logique d'une publication papier, c'est-à-dire par l'ajout d'annexes en fin de document ;
- les entrées de glossaires, références bibliographiques et autres éléments matérialisés par des incises dans la publication web ont été maintenus tels quels.

5.2.2 Modélisation

Contributions étudiantes

Pour l'instrumentation de cet usage, nous avons repris la prévisualisation mono-page avec commentaires modélisée pour l'ENAC (qui a par ailleurs été intégrée dans l'extension OpaleGenPdf).

Notre contribution technique a ici consisté essentiellement en la mise au point de l'outil de différentiel et son intégration à cette prévisualisation.

Banque de questions

Pour cet usage, une étape de modélisation a été nécessaire car la chaîne éditoriale ne disposait d'aucune prévisualisation permettant de relire les questions. Nous avons ajouté une prévisualisation avec commentaires non pas sur l'ensemble du test de positionnement, mais sur les groupes de quiz pris individuellement. En effet, les différents enchaînements entre étapes de groupes de quiz ne faisaient pas partie des éléments à relire.

Dans cette prévisualisation, nous avons modifié certains aspects de la transformation utilisée par la publication web. La linéarisation a consisté à afficher :

- toutes les questions du groupe, sans sélection aléatoire, sur une seule et même page (éliminant ainsi le lien de validation du groupe de quiz) ;
- les solutions et explications, consécutivement aux énoncés, en lieu et place des cases à cocher/boutons radio.

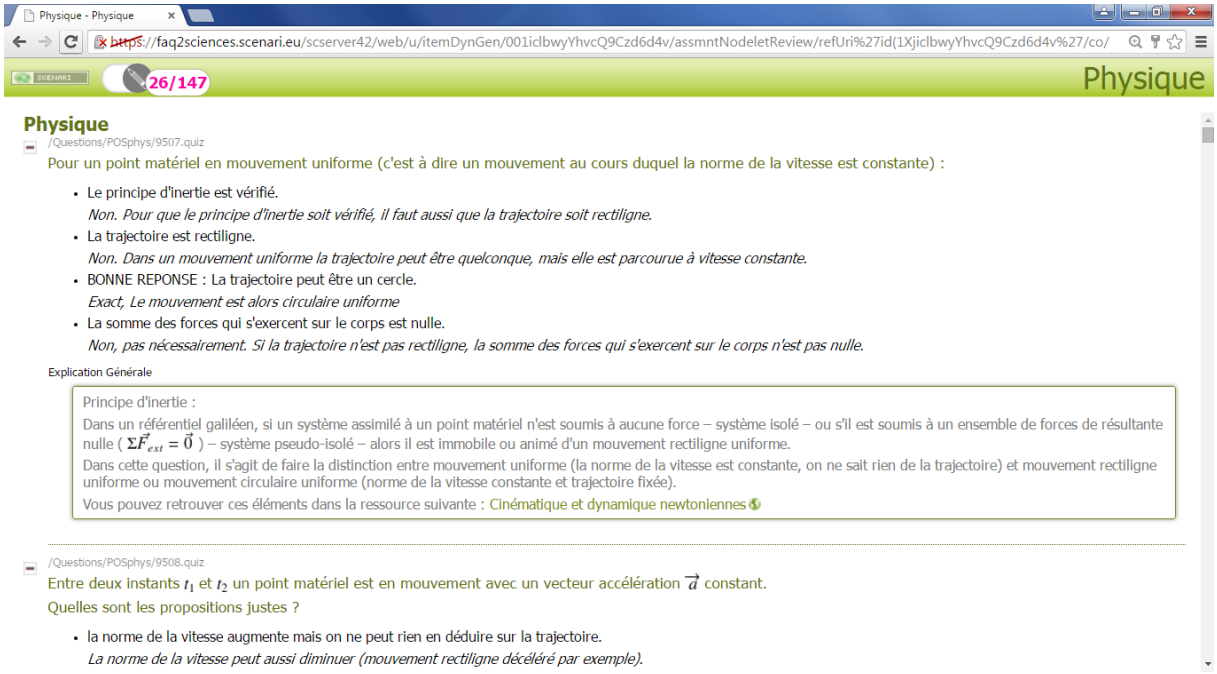


Figure 78 - Prévisualisation d'un groupe de quiz linéarisé.

De plus, nous avons ajouté et paramétré le système de commentaires dans cette prévisualisation :

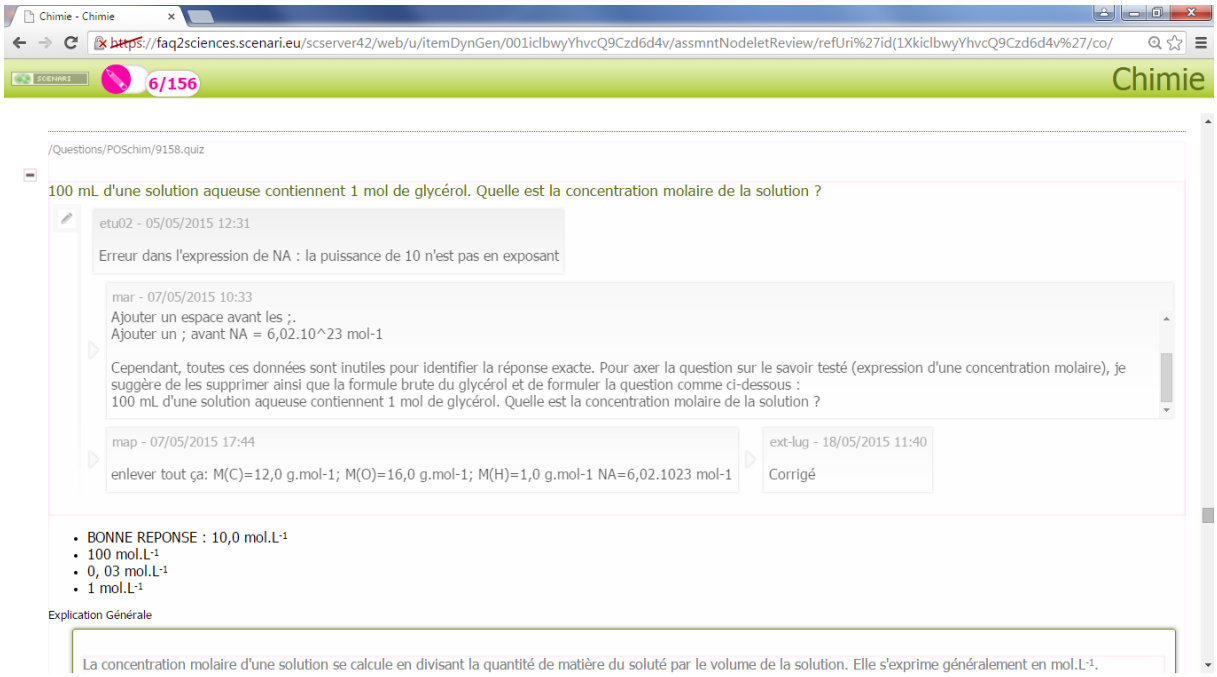


Figure 79 - Prévisualisation d'un groupe de quiz linéarisé avec commentaires.

En outre, l'URI de chaque question a été ajoutée au dessus de l'intitulé, afin de pouvoir identifier dans Scenari le fragment concerné par les commentaires.

5.2.3 Outil de différentiel

Versioning dans Scenari

La version 4 de Scenari ajoute aux chaînes éditoriales une dimension collaborative de gestion des fragments, rendue possible par un stockage s'appuyant sur une base de données orientée graphe, plutôt que sur un simple système de fichiers (Crozat, 2012b). Cette version de Scenari propose notamment l'historisation automatique de chaque fragment ainsi que le *versioning* manuel d'un fragment et de son sous-réseau (*ibid.*). Dans la première fonction, un système de purge permet de limiter le nombre d'entrées d'historique pour un fragment (créées à chaque enregistrement) ; dans la seconde, la version fige non-seulement l'état du fragment, mais aussi de tous les fragments de son sous-réseau (il est ainsi possible de ré-appliquer une transformation complète sur ces fragments).

Nous nous appuyons sur les versions créées manuellement au niveau des fragments "racine" (module et groupes de quiz) au moment où les contenus n'ont pas encore été modifiés. Dans la prévisualisation, une fois que le relecteur a activé la fonction de différentiel, ces versions apparaissent dans une liste déroulante :

Figure 80 - Sélection d'une ancienne version du module pour la comparaison.

Après la sélection d'une version, deux flux XHTML sont comparés à l'aide d'un algorithme de diff XML, développé par Kelis, s'exécutant côté serveur via une *servlet* Java. Le premier flux correspond à la prévisualisation "courante" (celle qui s'affiche dans le navigateur avec les contenus à jours) ; le second flux correspond à la prévisualisation de la version sélectionnée par le relecteur, chargée dans une *iframe* cachée. Les flux XHTML ne contiennent que les éléments pertinents pour la comparaison : les flux sont par exemple expurgés des éléments invariants tels que les scripts, les éléments du template de la page, etc.. Les résultats retournés par le diff sont ensuite traités côté client : des manipulations du DOM de la prévisualisation courante sont effectuées afin d'afficher les différences.

Nous avons choisi d'afficher le différentiel à deux niveaux : dans le plan et dans le contenu :

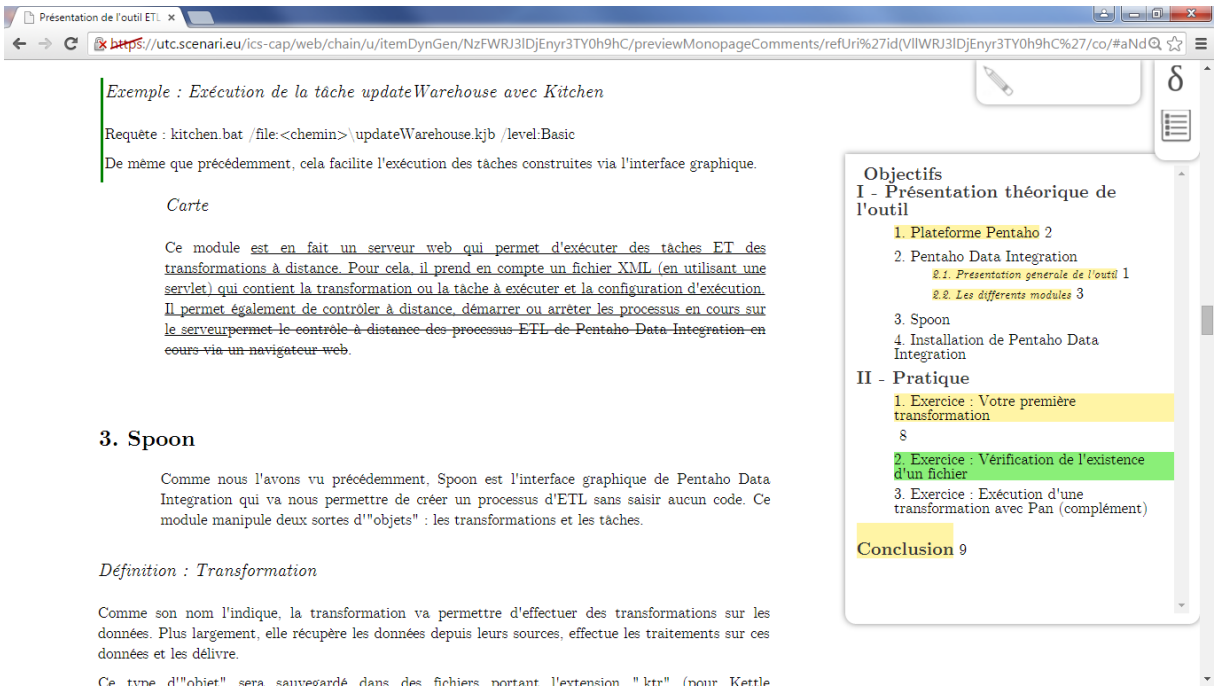


Figure 81 - Affichage du différentiel aux niveaux du plan et du contenu.

Dans le plan (disponible en marge permanente), les parties ajoutées sont de couleur verte tandis que les parties dont le contenu a été modifié sont de couleur jaune. Pour ces dernières, le nombre de différences à l'intérieur de la partie est précisé à droite de son titre. Les modifications de contenu sont affichées à l'aide de textes barrés (suppressions) ou soulignés (ajouts), ou par une marge rouge ou verte si la différence concerne un "bloc entier" (un paragraphe, une balise *div*, etc.). Un mode de navigation "précédant/suivant" (activable via l'icône se situant sous le delta en haut à droite) a également été ajouté afin de pouvoir afficher les différences une à une en surbrillance.

Nous avons choisi de ne pas afficher les parties supprimées dans le contenu, jugeant que cela aurait surchargé l'interface et délinéarisé davantage la lecture (typiquement, dans le cas d'une partie volumineuse supprimée). Par conséquent, ces parties ne sont pas indiquées dans le plan.

Enfin, notre logique d'affichage n'est pas exhaustive par rapport à la typologie de différences pouvant être renvoyées par l'algorithme de diff XML. Par exemple nous ne traitons pas, à l'heure actuelle, des différences portant sur du contenu mixte (typiquement, l'"enveloppement" d'un texte par une balise *strong*, *span*, etc.). Pour ces différences non-traitées, nous donnons la possibilité au relecteur d'afficher l'ancienne version du contenu, afin qu'il puisse faire la comparaison lui-même. Dans l'exemple ci-dessous, la différence non-traitée concerne l'ajout d'une balise autour du texte "objet signifiant" :

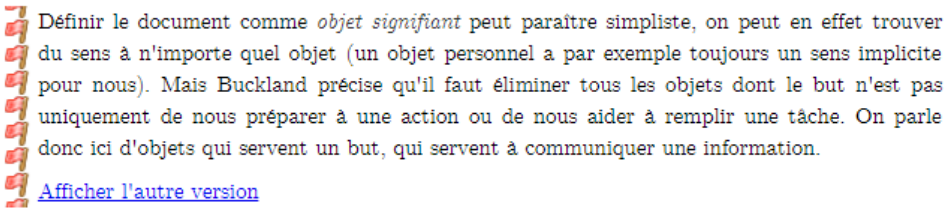


Figure 82 - Affichage d'une différence "non-traitée" : lien permettant d'afficher l'ancienne version.

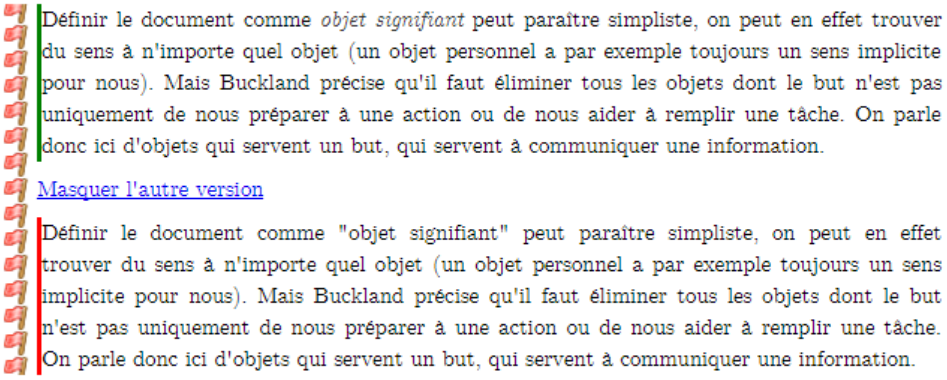


Figure 83 - Affichage d'une différence "non-traitée" : vis-à-vis des deux versions.

5.3 Retours d'usage

5.3.1 Contributions étudiantes

La forme de relecture a été testée au cours des cycles de validation de trois modules début 2015. Nous présentons les retours d'usage de cette expérimentation en nous appuyant sur les *verbatim* de l'enseignant ayant réalisé la relecture.

Le point de départ de cette expérimentation était la nécessité d'utiliser le différentiel afin de relire plus efficacement la nouvelle version du module : « le diff est essentiel, sans je ne peux pas faire le

job ». Cependant : « le diff dans l'éditeur [Scenari] ne convient pas, c'est trop compliqué à lire (trop d'info dans l'éditeur, trop chargé graphiquement, trop fragmenté... avec le diff je pense qu'on arrive à la saturation de ce qui est possible pour relire, a fortiori le texte de quelqu'un d'autre) ». Ce constat nous amène à formuler l'hypothèse selon laquelle le différentiel est plus utilisable dans une forme linéarisée, moins saturée fonctionnellement parlant et donnant une meilleure vue d'ensemble du contenu, par rapport à l'éditeur Scenari.

Cependant, en ne disposant plus de la fonction d'édition du contenu, l'enseignant n'a pas pu corriger les fautes d'orthographes (relecture de forme), ce qu'il a pourtant l'habitude de faire quand il relit dans l'éditeur. Ce point suggère d'ajouter un correcteur orthographique à la forme de relecture, dont les propositions de correction pourraient être intégrées directement aux sources XML par une simple sélection, sans avoir à passer par l'éditeur. Plus largement, on peut s'interroger sur le fait de rendre la forme de relecture complètement éditable, typiquement dans le cas où le relecteur voudrait corriger "lui-même" un mot pour pallier les limites éventuelles du correcteur orthographique. Cependant cette solution nous semble assez discutable : entraînant la possibilité de faire des modifications de contenu allant au-delà de la simple correction orthographique, elle n'est pas sans risque vis-à-vis de la cohérence éditoriale du contenu modifié, qui peut figurer dans un fragment appartenant à plusieurs contextes de rééditorialisation (p. 34).

Le second axe d'analyse de ces retours d'usage concerne le rôle qu'a joué la linéarisation dans la relecture parallèle du contenu et des commentaires. Pour cela, nous avons besoin d'expliciter deux notions relatives aux commentaires : l'ancrage et la portée. L'ancrage d'un commentaire désigne le niveau de contenu auquel il est *techniquement* associé (et où le commentaire sera par conséquent affiché), ce niveau pouvant être plus ou moins profond : balise pédagogique, grain, division, module entier... La portée désigne la "quantité de contenu" auquel le commentaire est *sémantiquement* associé : on dira d'un commentaire qu'il a une portée limitée s'il concerne peu de contenu au niveau où il est ancré (par exemple, un commentaire sur un grain peut ne concerner que son titre), ou étendue dans le cas inverse (le commentaire porte sur tout ou partie du grain).

Dans le cadre de cette expérimentation, nous avons observé que l'enseignant devait souvent relire des commentaires d'ancrage global (peu profond) et de portée étendue. En effet, il était par exemple demandé aux étudiants *reviewers* de faire des commentaires généraux sur le module (donc ancrés à ce niveau). Pour cette relecture, le mode d'affichage des commentaires "un à un" (p. 100) s'est avéré être une solution « pertinente », car elle permet à un commentaire de continuer à apparaître dans la marge tandis que le relecteur poursuit le défilement vertical du contenu (« je check un commentaire général et plus loin dans le texte ce qui a été fait »). Autrement dit, la désynchronisation du défilement des commentaires (boutons précédant/suivant) par rapport au défilement du contenu linéarisé (par *scrolling*) a facilité la tâche de l'enseignant.

Pour finir, plusieurs remarques ont été émises quant aux choix ergonomiques du différentiel :

- L'affichage des parties ajoutées et modifiées dans le plan devrait être répété au niveau des contenus (par exemple avec des marges de même couleur), afin de mieux faire la correspondance entre ces deux niveaux de différentiel.
- Le choix de ne pas afficher les parties supprimées dans le plan et dans le contenu a été remis en cause : en effet, dans le cas où un plan a été lourdement modifié, il est important de pouvoir les visualiser. Deux pistes ont été proposées :
    1. la possibilité d'inverser les versions comparées (ie. les ajouts deviennent des suppressions et vice versa) ;
    2. un affichage optionnel des parties supprimées dans le contenu (bloc replié par défaut).
- L'affichage des différences "non-traitées", signalées par des drapeaux rouges, n'a pas été bien compris : une légende explicitant les différents éléments graphiques du différentiel est nécessaire.
- Il aurait été utile de pouvoir visualiser explicitement les déplacements de contenu, qui faute d'un affichage dédié, ne se remarquent qu'implicitement à travers l'ajout et la suppression de ce

même contenu à différents endroits du document. Cela vaut aussi dans le cas où deux parties ont beaucoup de texte en commun (déplacement suivi de quelques modifications). Ces remarques suggèrent une adaptation de l'algorithme de diff pour détecter ce type de différences.

Pour l'un des modules relus, nous avons noté une remarque intéressante : « Il faut visualiser les versions courte et standard, sinon on ne comprend pas les diffs, dans ce module ». On observe en effet que les différences entre les deux versions de ce module sont essentiellement dues à l'utilisation des filtres dans la seconde version :

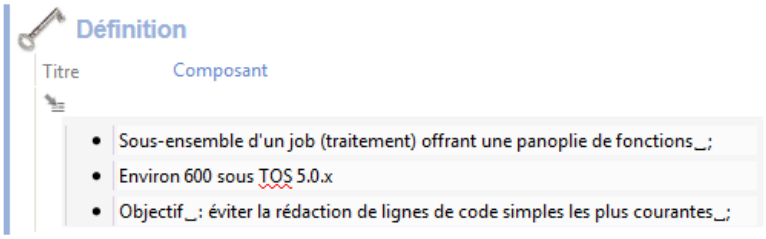


Figure 84 - Extrait de contenu (première version).

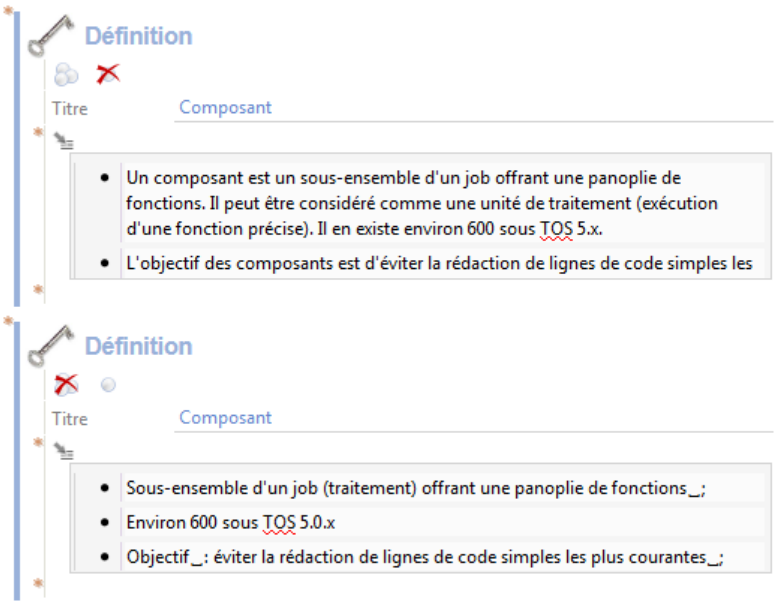


Figure 85 - Extrait de contenu (seconde version, avec usage des filtres).

Ainsi, le différentiel paraît ici peu approprié (notons qu'il s'agit, pour les deux versions comparées, de la version standard du contenu) :

Définition : Composant

- Un composant est un s~~S~~ous-ensemble d'un job ~~(traitement)~~ offrant une panoplie de fonctions. Il peut être considéré comme une unité de traitement (exécution d'une fonction précise). Il en existe environ 600 sous TOS 5.x.~~;~~
- L'objectif des composants est d'~~Objectif : é~~viter la rédaction de lignes de code simples les plus courantes : (exemple : mettre en majuscule tout une chaîne de caractères)

- Environ 600 sous TOS 5.0.x

Figure 86 - Différentiel entre les deux contenus (version standard).

Pour ce type de réécriture, une forme de relecture avec tabulation des versions courte et standard semble plus adaptée que le différentiel.

Enfin, un nouveau cas de relecture est apparu fin 2015, portant sur un module rédigé à partir de deux contributions des années précédentes (par exemple, les fiches de lecture "What is a document" et "What is a digital document" ont été fusionnées en "What is a (digital) document"). Faute d'instrumentation pour ce genre de cas, qui requerrait un différentiel "trois voies" (*three-way diff*), deux relectures ont dû être menées (une par ancienne version). Notre testeur nous a rapporté la solution qu'il a construite pour effectuer cette relecture, avec quelques pistes d'amélioration : « Une idée (simple) pour la triple comparaison, ce que j'ai fait finalement, c'est d'afficher les deux comparaisons en vis à vis. Il pourrait y avoir une option de synchronisation optionnelle des plans. C'est multipliable à N versions au besoin avec un système d'onglets. Là j'ai utilisé deux tabs du navigateurs, comme j'ai un grand écran (ou si j'avais eu un double écran), on peut envisager deux versions dans 1 écran, puis des onglets pour la version 4+. »

5.3.2 Banque de questions

La forme de relecture a été utilisée par les enseignants et étudiants sur une période de deux mois (Printemps 2015) consacrée à la qualification de la banque de questions. L'outil de différentiel n'a en revanche pas été utilisé suite aux modifications. Nous avons observé un panel de 1094 questions (chiffres de décembre 2015), destinées aux tests de positionnement pour les spécialités Maths, Physique, Chimie et SVT.

cycle de vie	**questions commentées**	**total (questions commentées ou non)**
état "validé"	465	1008
état "rejeté"	35	69
état "brouillon"	3	3
sans état	0	14
total	503	1094

Tableau 7 - Nombre de questions commentées sur l'ensemble de la base de questions.

Les 503 questions commentées comportent un total de 881 fils de discussions. Une analyse "manuelle" du contenu de ces commentaires nous a permis de recenser plusieurs mots-clés permettant de décrire l'activité des enseignants et étudiants lors de cette phase de relecture :

Mot-clé	Fréquence (en nombre de fils de discussion)	Description
remplacer	92	commentaires dans lesquels le relecteur précise la modification à intégrer
ajouter	58	
supprimer	17	
équipe technique	41	commentaires adressés à l'équipe technique, portant typiquement sur une image à agrandir, un formule Latex à revoir, etc.
faute(s) de frappe faute(s) d'orthographe	25	correction orthographique
exposant	23	commentaires soulignant qu'un nombre devrait être écrit en exposant (indice, puissance...)
corrigé	84	commentaires confirmant l'intégration d'une modification (appelant en général à la clôture du fil de discussion)
explication	41	commentaires portant sur les explications locales (associées aux réponses individuelles) ou bien globales (associées à l'ensemble de la question)

Tableau 8 - Analyse du contenu des commentaires par mots-clés.

Nous classons les six premiers mots-clés dans la catégorie des commentaires "performatifs", c'est-à-dire des commentaires décrivant explicitement l'action à entreprendre sur le contenu. Ces mots-clés sont représentés dans *au moins* 92 fils de discussion (en effet, plusieurs de ces mots-clés peuvent apparaître dans un même fil de discussion), soit dans une part significative. D'après nous, cela peut expliquer le fait que le différentiel n'ait pas été nécessaire dans cette expérimentation : la trace du fil de discussion (demande explicite de modification, généralement suivie d'un commentaire "corrigé") est suffisante pour valider la modification.

Les fils de discussion comportant le mot-clé "explication" (41 au total) nous intéressent particulièrement, car ils justifient selon nous le choix de modélisation consistant à afficher les explications à la suite des questions, et non après validation du questionnaire. Concernant l'affichage des réponses, nous n'avons pas trouvé de mot-clé permettant d'évaluer la quantité de commentaires concernés. Un exemple isolé nous a néanmoins permis de constater l'intérêt de ce choix : dans un QCM où une seule bonne réponse était indiquée, deux étudiants ont commenté l'énoncé ("Parmi les propositions suivantes, laquelle ou lesquelles sont vraies ?") en suggérant de le modifier ou bien de transformer la question en un QCU.

Concernant l'autre élément de modélisation que nous avions mis en avant, à savoir la suppression de la sélection aléatoire des questions dans la forme de relecture, les données que nous avons ne nous permettent pas d'évaluer quantitativement cet aspect (ie. savoir si cela a permis de commenter plus de questions) : la seule donnée que nous avons est celle du nombre de questions commentées par rapport au nombre total de questions du panel observé, soit un taux d'environ 46%. D'un point de vue plus qualitatif en revanche, il semble assez évident que cette solution permet un gain de temps par rapport au fait de conserver la sélection aléatoire : celle-ci aurait en effet amené les enseignants et étudiants à relire plusieurs fois les groupes de quiz afin d'être certains d'avoir couvert toutes les questions.

5.4 Évaluation

5.4.1 Positionnement épistémologique

Nos travaux s'inscrivent dans le cadre d'une recherche technologique dont l'enjeu est d'une part de faire émerger des formes documentaires en prise avec la tendance technique du numérique, et d'autre part d'expérimenter ces formes dans des situations d'usage.

D'après Theureau (2009), la recherche technologique se fonde sur une épistémologie dans laquelle la science entretient une *relation organique* avec la technique (rappelant ainsi la définition de la technologie par Koyré). Cette relation organique permet de créer un lien bidirectionnel entre la recherche scientifique et l'ingénierie : d'une part l'ingénierie met en œuvre les solutions qui émergent de la recherche scientifique et les confronte aux usages ; d'autre part, la recherche scientifique est relancée par les observations faites du côté des usages.

En s'appuyant sur les travaux de Theureau, Arribe (2014) pose les bases du programme de recherche technologique "rééditorialisation documentaire", ayant la rééditorialisation pour objet théorique et les chaînes éditoriales pour objets technologiques, c'est-à-dire les objets à travers lesquel la rééditorialisation peut être expérimentée. L'enjeu de ce programme est d'« anticiper les évolutions des formes d'écriture induites par le support numérique » afin de mieux outiller la rééditorialisation, supposée comme étant « plus [efficace] pour la rédaction et la maintenance de fonds documentaires importants » (*ibid.*).

C'est dans ce cadre épistémologique que nous positionnons nos travaux, en élargissant l'enjeu du programme aux formes documentaires en général (pour l'écriture mais aussi la lecture), afin d'y inclure les formes de relecture que nous proposons. Notons que si notre recherche n'a pas directement pour objet la rééditorialisation, elle vise idéalement à mieux outiller le processus de relecture dans les chaînes éditoriales, et par conséquent à permettre l'usage des techniques de rééditorialisation malgré leur impact sur la relecture (p. 34).

Le programme "rééditorialisation documentaire" s'appuie sur l'hypothèse philosophique, développée par Leroi-Gourhan, Simondon et Stiegler, selon laquelle les objets techniques évoluent à travers un dynamisme qui leur est propre.

Nous avons vu en effet chez Leroi-Gourhan (p. 40) que la part de l'homme dans l'invention technique était à mettre en regard de la *tendance* selon laquelle la matière s'organise, à partir de ses lois internes, en un objet technique.

Chez Simondon, la tendance évacue le déterminisme humain dans l'évolution de l'objet technique, à travers le processus de *concrétisation* : étant au départ « la traduction physique d'un système intellectuel », l'objet se concrétise lorsqu'il « tend vers la cohérence interne, vers la fermeture du système des causes et des effets qui s'exercent circulairement à l'intérieur de son enceinte, et de plus incorpore une partie du monde naturel qui intervient comme condition de fonctionnement, et fait ainsi partie du système des causes et des effets » (Simondon, 1958, p. 56). La concrétisation est illustrée par Simondon notamment à travers l'exemple de la turbine de Guimbal. Le problème d'origine consistait à insérer la turbine avec une génératrice dans la conduite forcée d'une centrale hydroélectrique. Pour cela, la taille de la génératrice devait être diminuée, rendant alors impossible l'évacuation de la chaleur à plein régime, du moins à l'air libre. L'idée de Guimbal fut de plonger la génératrice dans un carter rempli d'huile sous pression et reliée à la turbine, l'ensemble étant actionné par l'eau. L'huile agitée par le fonctionnement de la génératrice permet de dégager la chaleur vers l'extérieur du carter, avant que celle-ci ne soit évacuée par l'eau. L'eau et l'huile, plurifonctionnelles dans cet ensemble, constituent ce que Simondon nomme un *milieu associé*, conditionnant le fonctionnement de l'objet technique en même temps qu'il est créé par lui.

Pour Simondon, le rôle de l'homme n'est plus d'inventer mais d'*anticiper* le dynamisme d'évolution

de l'objet technique vers le stade concret, tel que le rappelle Stiegler : « [Ce dynamisme], bien que n'étant plus soumis à l'intention humaine, requiert néanmoins la dynamique opératrice de l'anticipation. L'objet, qui n'est pas produit par l'homme, a néanmoins besoin de lui en tant qu'il anticipe [...] » (Stiegler, 1994a, p. 95). C'est précisément cette anticipation que l'on retrouve au cœur du programme "rééditorialisation documentaire".

En suivant la théorie du support [p.126] proposée par Bachimont (2004), qui postule que tout objet technique est l'inscription matérielle d'une connaissance et que toute connaissance est d'origine technique, il s'ensuit que le dynamisme propre d'évolution des objets techniques s'applique également aux documents. En effet, ces derniers appartiennent à la classe technique des inscriptions sémiotiques, qui sont considérées pour ce qu'elles représentent et non pour ce qu'elles sont, à la différence des inscriptions instrumentales, qui prescrivent un savoir-faire par leurs structures matérielles (*ibid.*). Puisque d'après cette théorie, « les propriétés matérielles du support d'inscription conditionnent l'intelligibilité de l'inscription » (*ibid.*), c'est en faisant travailler la tendance technique du numérique que l'on pourra anticiper l'évolution du document avec ce nouveau support, et en particulier l'émergence de nouvelles formes documentaires visant à être confrontées à des situations d'usage.

5.4.2 Bilan de notre recherche

Dans le cadre épistémologique où nous nous situons, l'évaluation des propositions théoriques se fait en deux temps : le premier est celui de l'expérience de laboratoire, lors de laquelle des prototypes sont testés pour un usage réel certes, mais prenant place dans un contexte dit "protégé" ; le second est celui de l'évaluation par les usages en contexte dit "non-protégé" (ou contexte industriel), et nécessite un outillage technologique avancé (au-delà de simples prototypes) afin de mobiliser ces formes documentaires dans les usages "quotidiens" d'une organisation. **La validation effective des propositions théoriques ne peut intervenir qu'à travers l'évaluation par les usages.** En effet d'après Arribe, c'est en étant mobilisées dans ces contextes non-protégés que les formes documentaires sont réfutables, ce qui d'après Popper est la condition de validité d'une hypothèse scientifique. Une forme documentaire "réfutée" est à comprendre comme inadaptée au contexte d'usage, ce qui permet de remettre en cause les propositions théoriques qui en sont à l'origine : il s'agit de la "relance" de la science par l'ingénierie et les usages tel qu'envisagée dans la recherche technologique selon Theureau.

Sur l'ensemble de nos propositions théoriques, nous n'avons pu expérimenter que la linéarisation des documents "multi-pages" (relecture de contributions étudiantes) et celle des questionnaires interactifs (qualification de la banque de questions de Faq2Sciences). Les expérimentations que nous avons menées ont eu lieu en contexte protégé, en lien avec des membres de notre équipe de recherche. Nous ne pouvons donc pas entreprendre d'évaluation par les usages de nos propositions à ce stade.

Cependant, des perspectives d'évaluation sont ouvertes en contexte industriel (clients de Kelis), où des formes de relecture ont été modélisées en reprenant des principes similaires aux propositions que nous avons expérimentées dans nos travaux, notamment à l'AFPA, où les relecteurs disposent d'une forme *mono-page*, et à l'IFCAM, où la prévisualisation utilisée par les experts métiers (p. 25) a évolué notamment en affichant *directement* les résultats des exercices du module, sans interactivité. Ces deux contextes posent des questions intéressantes pour la suite de nos travaux. À l'AFPA, des filtres "stagiaire" et "formateur" sont utilisés pour décliner les documents de formation. La forme de relecture affiche l'exhaustivité des contenus (sans tenir compte des filtres) dans une seule et même linéarité. Une étude sur les usages de cette forme nous permettrait de voir si cette solution est satisfaisante ou bien si elle peut être améliorée grâce à la tabulation, typiquement lorsqu'un contenu "stagiaire" s'oppose à un contenu "formateur". Par ailleurs le cas de l'IFCAM est intéressant à analyser : si les exercices sont affichés sous forme linéaire, ils restent toutefois accessibles dans leur forme interactive via un lien "Tester l'exercice..." (l'exercice s'ouvre alors en sur-fenêtre). La validation de l'interactivité des exercices n'est-elle pas également un enjeu de la relecture ? Est-ce là le signe qu'une forme de relecture ne devrait pas chercher à rompre totalement avec la forme finale qu'elle est censé valider ?

Conclusion

Dans ce mémoire, nous nous sommes intéressé à la question de la relecture des documents numériques en prenant les chaînes éditoriales comme cadre d'étude. Oscillant entre instabilité, interactivité et rééditorialisation, les formes documentaires qu'elles proposent ne sont pas adaptées à la relecture. Nous avons alors suivi le principe du polymorphisme afin de proposer des formes dédiées à cette activité.

L'analyse de l'état de l'art nous a montré que les relectures de fond et de forme étaient outillées à travers les fonctions d'annotation et de correction automatique, et que le différentiel permettait de répondre à la propriété d'instabilité du document numérique. Les deux autres propriétés mobilisées dans la problématique, à savoir l'interactivité et la rééditorialisation, ont motivé la proposition de deux grands axes de conception de formes de relecture.

Premièrement, la linéarisation vise à redonner aux contenus une linéarité matérielle leur permettant d'être relus de façon exhaustive. Nous avons détaillé cette stratégie à travers des solutions répondant à différents sous-problèmes posés par l'interactivité. Une transformation mono-page permet ainsi de rétablir la finitude synoptique d'un document de structure linéaire, tandis qu'un document multilinéaire peut être linéarisé à travers la construction par le relecteur d'un parcours primaire, en référence auquel les autres parcours sont ensuite exprimés. Les contenus interactifs (augmentation, parenthèse, navigation) peuvent être soit réintégrés, soit affichés dans une zone dédiée en marge permanente ou bien en sur-fenêtre, en permettant au relecteur de mémoriser la trace de sa relecture lorsque ces contenus sont référencés plusieurs fois au sein du document. Les contenus calculés supposent quant à eux une projection statique, affichant l'ensemble des résultats potentiels de l'interaction, tel que pour les questionnaires de type QCU ou QCM, ou bien schématisant les structures conditionnelles en jeu lorsqu'elles sont plus complexes.

Deuxièmement, la tabulation cherche à permettre la relecture en parallèle de plusieurs contextes de rééditorialisation d'un document. En prenant le cas de la déclinaison dont nous avons analysé quelques propriétés formelles, nous avons proposé l'ébauche d'un algorithme de tabulation, restant toutefois à préciser et compléter.

Sur le plan expérimental, nous avons confronté deux de nos propositions théoriques à des contextes d'usage. Le premier contexte est celui de la relecture de contributions étudiantes et nous a permis de tester une forme linéarisée d'un document "multi-pages" intégrant un outil de différentiel. Cette expérimentation a mis en lumière le rôle joué par la linéarité du contenu par rapport à la relecture de ce dernier "en parallèle" avec celle des commentaires. Le différentiel s'est révélé plus utilisable du fait de l'allègement fonctionnel de la forme de relecture (pas de fragmentation, pas d'édition...) par rapport à l'éditeur Scenari. Dans le second contexte, concernant la qualification d'une banque de questions, nous avons mis en œuvre la linéarisation de questionnaires interactifs. Des éléments tant quantitatifs (nombre de questions commentées) que qualitatifs (commentaires sur les explications des solutions) suggèrent l'efficacité du dispositif proposé.

Ayant eu lieu dans des contextes "protégés", ces expérimentations n'ont pas fait l'objet d'une évaluation par les usages, ce qui ne permet pas à l'heure actuelle de valider ces deux propositions dans un cadre de recherche technologique tel que le nôtre. Des mobilisations actuelles (comme à l'AFPA ou à l'IFCAM par exemple) et futures de ce type de forme de relecture en contexte industriel permettront néanmoins de confirmer ou d'infirmer les hypothèses sous-tendant ces propositions.

De futurs travaux sont nécessaires afin d'implémenter et tester les autres propositions qui n'ont été présentées qu'à l'état de maquette dans ce mémoire. L'usage du modèle Opale pour les contributions étudiantes pourrait faire l'objet de nouvelles expérimentations en contexte protégé, notamment pour la tabulation (relecture du module à la fois dans sa version courte et standard) et pour la zone de contenus référencés (entrées de glossaire, références bibliographiques...). Le modèle Juriguide utilisé à l'UCANSS permettrait quant à lui de tester ces mêmes propositions en contexte industriel (tabulation

des trois déclinaisons d'une fiche d'une part ; zone des articles de loi référencés d'autre part). Enfin, de nouvelles recherches sont nécessaires afin de proposer des formes de relecture pour les techniques de réutilisation par transclusion et de dérivation. Pour cette dernière, il faudrait étudier une éventuelle application de la tabulation.

Pour conclure en revenant sur la question de la philologie par laquelle nous avons introduit ce mémoire, nous pensons qu'elle se pose effectivement au niveau du document numérique, de par sa variance intrinsèque, comme nous avons pu le voir à travers les propriétés qui le rendent plus difficile à relire et à valider dans ses différents parcours interactifs, versions et rééditorialisations. Aujourd'hui, la philologie exploite les outils numériques pour l'édition savante des textes anciens (outils de visualisation et de comparaison de variantes). En revanche, il nous semble que la question se pose en d'autres termes pour les documents numériques. Plutôt que d'établir une version de référence à partir d'un ensemble de variantes ne s'accordant pas en tout point du texte (philologie classique pour les textes anciens), il s'agirait ici de relire et valider le document numérique *dans sa variance* à partir d'une version de référence. En particulier, une forme de relecture peut être vue comme une version de référence synthétisant *temporairement* la variance (par linéarisation, tabulation, etc.), sans pour autant l'éliminer de façon définitive.

Bibliographie

- André, J., Furuta, R., Quint, V.. *Structured documents.* Cambridge University Press, 1989. Vol. 2.
- Arribe, T., Crozat, S., Bachimont, B., Spinelli, S.. « Chaînes éditoriales numériques : allier efficacité et variabilité grâce à des primitives documentaires. ». *Actes du colloque CIDE.15 "Métiers de l'information, des bibliothèques et des archives à l'ère de la différentiation numérique", Tunis, Tunisie, 1-3 novembre 2012, 2012.*
- Arribe, T.. *Conception des chaînes éditoriales : documentariser l'activité et structurer le graphe documentaire pour améliorer la maîtrise de la rééditorialisation.* Thèse, Université de Technologie de Compiègne, 2014.
- Arsenault, B., Baudin, C., Bouyahi, A., Gravel, S., Laroche, M. E.. *Principes directeurs en révision professionnelle.*. Réviseurs Canada, 2014.
- Aubert, O., Prié, Y.. « Advene: active reading through hypervideo ». *Proceedings of the sixteenth ACM conference on Hypertext and hypermedia, 2005.* ACM. 235-244.
- Bachimont, B., Cailleau, I., Crozat, S., Majada, M., Spinelli, S.. « Outils auteurs : approche industrielle versus approche artisanale ». *ARIADNE, Lyon, 2002.*
- Bachimont, B., Crozat, S.. « Instrumentation numérique des documents: pour une séparation fonds/forme ». *Information-Interaction-Intelligence*, 2004. 4(1).
- Bachimont, B.. « Dossier patient et lecture hypertextuelle ». *Les cahiers du numérique*, 2001. 2(2), 105-123.
- Bachimont, B.. *Arts et sciences du numérique : ingénierie des connaissances et critique de la raison computationnelle.* Mémoire de HDR, Université de Technologie de Compiègne, 2004.
- Bachimont, B.. *Ingénierie des connaissances et des contenus.* Hermès, 2007.
- Barabucci, G.. *A universal delta model.* PhD thesis, Universita di Bologna, 2013.
- Barcellini F., Grosse C., Albert C., Saint-Dizier P.. « LELIE: a tool dedicated to procedure and requirement authoring ». *Proceedings of the Second Workshop on Computational Linguistics and Writing (CLW 2012): Linguistic and Cognitive Aspects of Document Creation and Document Engineering, 2012.* Association for Computational Linguistics. 35-38.
- Bézivin J.. « On the unification power of models ». *Software & Systems Modeling*, 2005. Volume 4, Issue 2, pp. 171-188. DOI : DOI : 10.1007/s10270-005-0079-0
- Bottini, T.. *Instrumenter la lecture critique multimédia.* Thèse, Université de Technologie de Compiègne, 2010.
- Bouchardon, S.. *Littérature numérique : le récit interactif*. Hermès science publications, 2010.
- Briet, S.. *Qu'est-ce que la documentation ?.* Éditions documentaires et industrielles, 1951. http://martinetl.free.fr/suzannebriet/questcequeladocumentation/
- Brissaud, S.. *La lecture angoissée ou la mort du correcteur.* 1998.
- Buckland, M.. « What is a "document"? ». *JASIS*, 1997. 48(9), 804-809.
- Buckland, M.. « What is a digital document ». *Document numérique*, 1998. 2(2), 221-230.
- Bush, V.. « As we may think ». *The atlantic monthly*, 1945. vol.176 n°1, 101–108.

- Cerquiglini B.. « Vingt ans après ». *Genesis. Manuscrits–Recherche–Invention*, 2010. 30, 15-17.
- Cerquiglini B.. *Éloge de la variante. Histoire critique de la philologie.*. Paris, Minuit, 1989.
- Chevalier, F., Dragicevic, P., Bezerianos, A., Fekete, J. D.. « Using text animated transitions to support navigation in document histories ». *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2010.* ACM. 683-692.
- Collins-Sussman, B. and Fitzpatrick, B. and Pilato, M.. *Version control with Subversion.* O'Reilly Media, Inc., 2004.
- Crozat, S., Bachimont, B., Cailleau, I., Bouchardon, S., Gaillard, L.. « Éléments pour une théorie opérationnelle de l'écriture numérique ». *Document numérique*, 2011. 14(3), 9-33.
- Crozat, S., Bachimont, B.. « Réinterroger les structures documentaires: de la numérisation à l'informatisation ». *Information-Interaction-Intelligence*, 2004. 4(1).
- Crozat, S.. « C2M : Chaînes éditoriales collaboratives multimédia ». *Actes du colloque CIDE.15 "Métiers de l'information, des bibliothèques et des archives à l'ère de la différentiation numérique", Tunis, Tunisie, 1-3 novembre 2012, 2012b.*
- Crozat, S.. « Chaînes éditoriales et rééditorialisation de contenus numériques ». *Séminaire IST Inria : le document numérique à l'heure du web de données, 2012a.* ADBS. 179-220.
- Crozat, S.. « Les tropismes du numérique ». *Hypermédias et pratiques numériques: Actes de H2PTM'15*, 2015a.
- Crozat, S.. *As we may... Réflexions autour du support numérique.* 2015b. http://aswemay.fr/
- Crozat, S.. *Scenari - La chaîne éditoriale libre : Structurer et publier textes, images et son.*. Eyrolles, 2007.
- Damerau F. J.. « A Technique for Computer Detection and Correction of Spelling Errors ». *Commun. ACM*, 1964. 7, 171-176.
- Fraser, N.. « Differential synchronization ». *Proceedings of the 9th ACM symposium on Document engineering, 2009.* ACM. 13-20.
- Gebers Freitas E.. *Environnement numérique de lecture : instrumentation de l'activité de lecture savante sur support numérique.* Thèse, Université de Technologie de Compiègne, 2008.
- Giffard, A.. *Pour une critique pharmacologique de la lecture numérique.* 2013. http://alaingiffard.blogs.com/culture/2013/01/pour-une-critique-pharmacologique-de-la-lecture-n
- Gonzales-Aguilar, A., Ramírez-Posada, M., Crozat, S.. « Scenari-Opale: cadena editorial digital para la producción de contenidos e-learning ». *El profesional de la información*, 2012. 21(4), 433-438.
- Goody, J.. *La raison graphique : la domestication de la pensée sauvage.* Minuit, Paris, 1979.
- Hirst, G., Budanitsky, A.. « Correcting real-word spelling errors by restoring lexical cohesion ». *Natural Language Engineering*, 2005. 87-111.
- Hunt, J.-W., MacIlroy, M. D.. « An algorithm for differential file comparison ». *Bell Laboratories*, 1976.
- Josse, I.. « Crowdsourcing facing Cultural heritage of printed texts: the platform Correct (Co-operative text correction and enrichment) ». *Archiving Conference, 2014.* Society for Imaging Science and Technology. Vol. 2014, No. 1. 169-173.
- Kang, J., Saint-Dizier, P.. « Discourse structure analysis for requirement mining ». *International Journal of Knowledge Content Development & Technology*, 2013. 3(2), 43-65.

- Kang, J., Saint-Dizier, P.. « Une expérience d'un déploiement industriel de LELIE: une relecture intelligente des exigences ». *INFORSID, 2015.*
- Kukich K.. « Techniques for automatically correcting words in text ». *ACM Computing Surveys (CSUR)*, 1992. 24(4), 377-439.
- Lagarrigue, M., Rossant, F., Pierrot, A., Gardes, J., Maldivi, C., Petit, E.. « Assessing the quality of digital re-publishing of textual documents through the follow-up of a correction protocol by crowdsourcing ». *International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM), 2014.* IEEE. 1-5.
- Leroi-Gourhan, A.. *L'homme et la matière.* Albin Michel, 1943.
- Levenshtein, V. I.. « Binary codes capable of correcting deletions, insertions, and reversals ». *Soviet physics doklady*, 1966. 10-8, 707-710.
- Mays, E., Damerau, F. J., Mercer, R. L.. « Context based spelling correction ». *Information Processing & Management*, 1991. 27(5), 517-522.
- Myers, E. W.. « An O(ND) difference algorithm and its variations ». *Algorithmica*, 1986. 1(1-4), 251-266.
- Nelson, T. H.. *Computer Lib: Dream Machines.* Tempus Books of Microsoft Press Redmond, 1987.
- Nelson, T. H.. *Literary Machines.* Mindful Press, 1981.
- O'Hara, K.. *Towards a typology of reading goals.* Xerox Research Centre Europe, Cambridge Laboratory, UK, 1996.
- Pédauque, R. T. (collectif). *Document : forme, signe et médium, les re-formulations du numérique.*. 2003. http://archivesic.ccsd.cnrs.fr/sic_00000511
- Pédauque, R. T. (collectif). *Document et modernités.* 2006. http://archivesic.ccsd.cnrs.fr/sic_00001741
- Pfaender, F.. *Spatialisation de l'information.* Thèse, Université de Technologie de Compiègne, 2009.
- Power, R., Scott, D., Evans, R.. « What You See Is What You Meant: direct knowledge editing with natural language feedback ». *ECAI, 1998.* 98. 677-681.
- Prié, Y.. *Vers une phénoménologie des inscriptions numériques. Dynamique de l'activité et des structures informationnelles dans les systèmes d'interprétation.* Thèse, Mémoire de HDR, Université Claude Bernard-Lyon I, 2011.
- Richard B., Prié Y., Calabretto S.. « Lecture active de documents audiovisuels : organisation de connaissances personnelles par la structuration d'annotations ». *18es Journées Francophones d'Ingénierie des Connaissances, 2007.*
- Rönnau, S., Philipp, G., Borghoff, U. M.. « Efficient change control of XML documents ». *Proceedings of the 9th ACM symposium on Document engineering, 2009.* ACM. 3-12.
- Rothenberg J.. « The nature of modeling ». *Widman L E, Loparo K A, Nielsen N R. Artificial intelligence, simulation & modeling. John Wiley & Sons, Inc.*, 1989. pp. 75-92.
- Roussey, J.-Y., Piolat, A.. « Critical reading effort during text revision ». *European Journal of Cognitive Psychology*, 2008. 20(4), 765-792.
- Saint-Dizier, P.. « Processing natural language arguments with the platform ». *Argument & Computation*, 2012. 3(1), 49-82.

- Schilit, B. N., Golovchinsky, G., Price, M. N.. « Beyond paper: supporting active reading with free form digital ink annotations ». *Proceedings of the SIGCHI conference on Human factors in computing systems, 1998.* ACM Press/Addison-Wesley Publishing Co.. 249-256.
- Shneiderman, B.. « Tree visualization with tree-maps: 2-d space-filling approach ». *ACM Transactions on graphics (TOG)*, 1992. 11(1), 92-99.
- Simondon, G.. *Du mode d'existence des objets techniques.* Aubier, 1958.
- Stiegler, B.. « Annotation, navigation, édition électroniques : vers une géographie de la connaissance ». *Le texte et l'ordinateur*, 1995. http://arsindustrialis.org/node/1937
- Stiegler, B.. « Machines à écrire et matières à penser ». *Genesis*, 1994b. 25-49.
- Stiegler, B.. *La technique et le temps : la faute d'Epiméthée (Vol. 1).*. Galilée/Cité des sciences et de l'industrie, 1994a.
- Stiegler, B.. *Pharmakon, pharmacologie.* Ars Industrialis. Association internationale pour une politique industrielle des technologies de l'esprit. 2012. http://www.arsindustrialis.org/pharmakon
- Tai, K. C.. « The tree-to-tree correction problem ». *Journal of the ACM (JACM)*, 1979. 26(3), 422-433.
- Theureau J.. *Le cours d'action : méthode réfléchie. Collection Travail & activité humaine* : Octares Éditions, 2009. ISBN : 978-2-915346-64-0
- Van Deemter, K., Power, R.. « Authoring multimedia documents using WYSIWYM editing ». *Proceedings of the 18th conference on Computational linguistics, 2000.* Volume 1. 222-228.
- Vandendorpe C.. *Du papyrus à l'hypertexte : Essai sur les mutations du texte et de la lecture.* Paris : La découverte., 1999.
- Viégas, F. B., Wattenberg, M., Dave, K.. « Studying cooperation and conflict between authors with history flow visualizations ». *Proceedings of the SIGCHI conference on Human factors in computing systems, 2004.* ACM. 575-582.
- Vion-Dury, J. Y.. « A generic calculus of XML editing deltas ». *In Proceedings of the 11th ACM symposium on Document engineering, 2011.* ACM. 113-120.
- Virbel, J.. « Annotation dynamique et lecture expérimentale : vers une nouvelle glose ? ». *Littérature*, 1994. 96, 91-105.
- Wilcox-O'Hearn, A., Hirst, G., Budanitsky, A.. « Real-word spelling correction with trigrams: A reconsideration of the Mays, Damerau, and Mercer model ». *Computational Linguistics and Intelligent Text Processing*, 2008. 605-616.
- Wilkinson, D. M, Huberman, B. A.. « Assessing the value of coooperation in wikipedia ». *First Monday*, 2007. http://arxiv.org/pdf/cs/0702140.pdf
- Zacklad M., Lewkowicz M., Boujut J. F., Darses F., Détienne F.. « Formes et gestion des annotations numériques collectives en ingénierie collaborative ». *Actes des journées Ingénierie des Connaissances, 2003.*
- Zacklad M.. « Annotation : attention, association, contribution ». *In : Annotations dans les Documents pour l'Action.* Hermes science publications, 2007. 29-46.
- Zacklad M.. « Processus de documentarisation dans les Documents pour l'Action (DopA) : statut des annotations et technologies de la coopération associées (nouvelle version corrigée) ». *Le numérique : Impact sur le cycle de vie du document pour une analyse interdisciplinaire, Montréal (Québec), 2005.* Éditions de l'ENSSIB.

Annexes

Annexe 1

Modélisation des chaînes éditoriales

SCENARIbuilder

SCENARIbuilder est l'environnement de conception des modèles documentaires Scenari. Il permet de paramétrer les primitives du méta-modèle de chaînes éditoriales (*modeling*) et de compiler un modèle documentaire sous la forme d'une archive de code source (*wsppack*) qui peut ensuite être installée dans SCENARIchain (Arribe *et al.*, 2012 ; Arribe, 2014).

Nous allons illustrer les différentes primitives de SCENARIbuilder en décrivant les étapes de conception d'une chaîne éditoriale très simple :

- des sections composées d'un ensemble de blocs de texte et de sous-sections, ces dernières étant éventuellement des fragments autonomes (et donc réutilisables au sein de plusieurs sections) ;
- une publication HTML de la section sous la forme d'un site web composé d'une page par section, chaque page étant accessible via un menu arborescent.

La primitive *wspDefinition* permet de déterminer les types de fragments qui pourront être instanciés par l'auteur, ainsi que les éventuelles publications qui leur seront associées. Dans notre exemple, la chaîne éditoriale permettra de créer uniquement des fragments de type section (*section.model*), et de publier ces derniers sous la forme d'un site web (*site.generator*) :

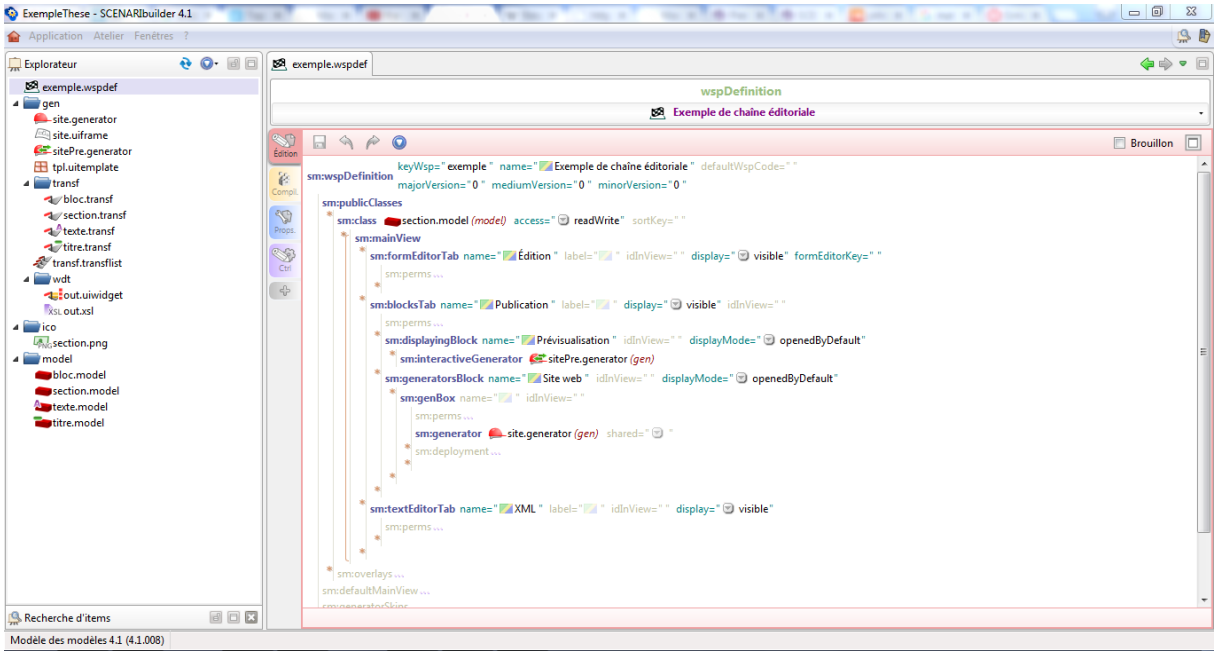


Figure 87 - exemple.wspdef (wspDefinition).

Primitives de document

La structuration des fragments est définie à partir des primitives de document telles que la *compositionPrim* :

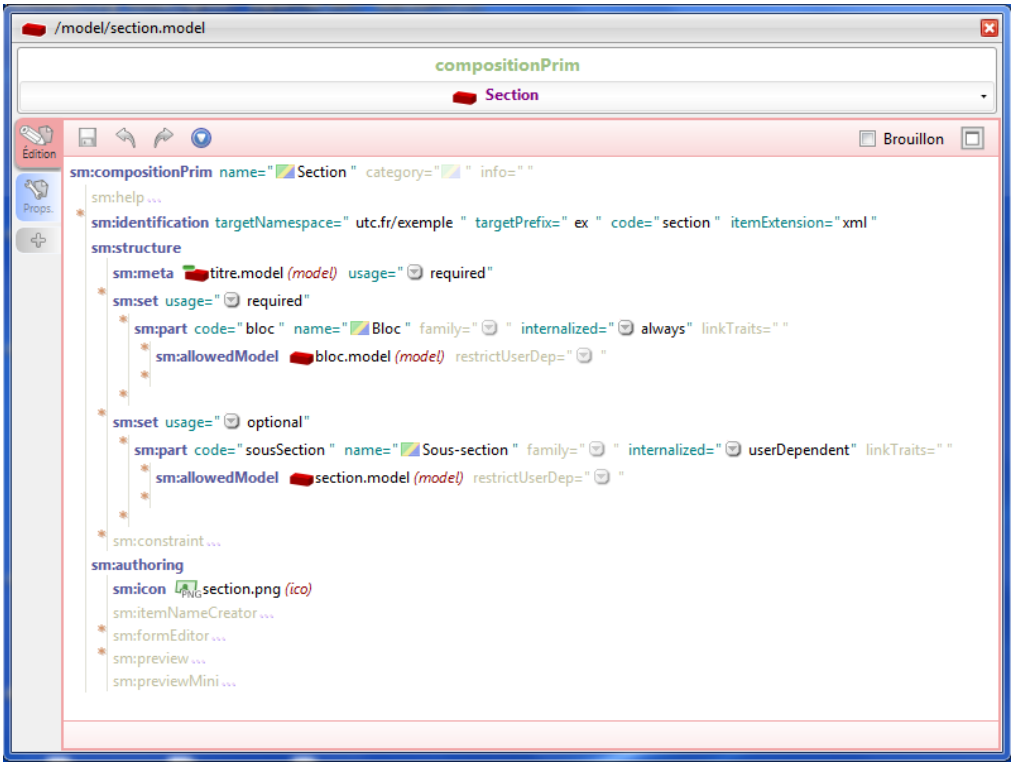


Figure 88 - section.model (compositionPrim).

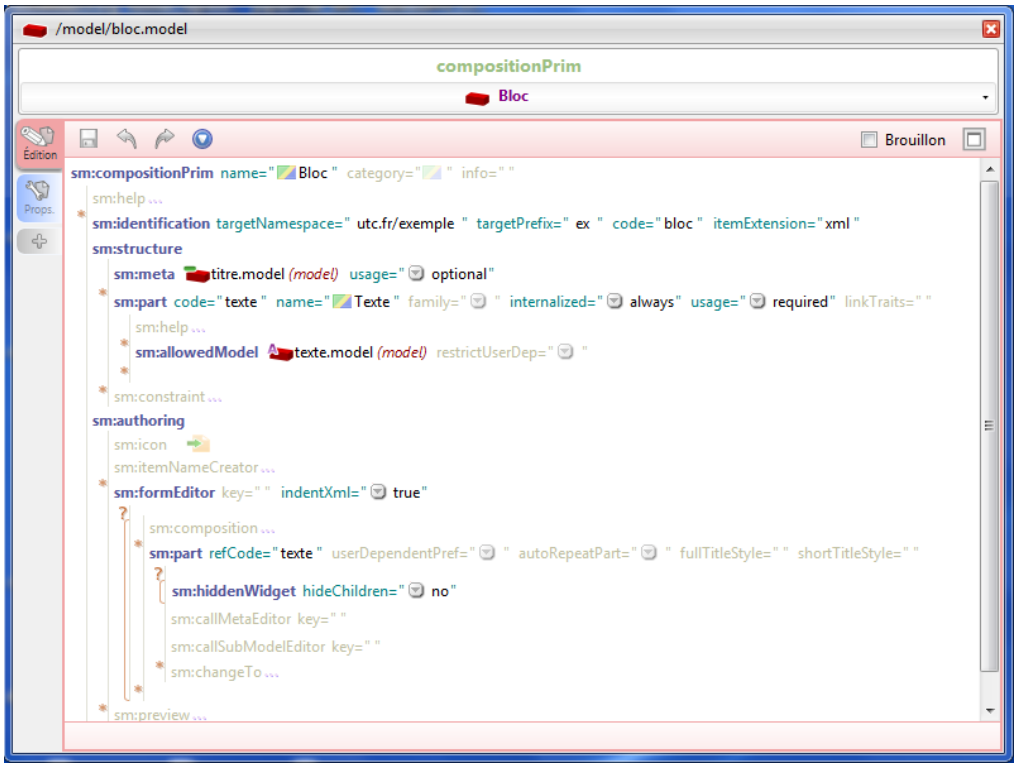


Figure 89 - bloc.model (compositionPrim).

Un *set* permet de rendre répétable la structure spécifiée (*part*) : ainsi, on pourra avoir plusieurs blocs

et plusieurs sous-sections dans une section. L'attribut *usage* permet de préciser si la structure est obligatoire (*required*) ou facultative (*optional*). L'attribut *internalized* permet quant à lui de préciser si la structure sous-jacente est incluse dans le fichier XML courant (*always*), si elle constitue au contraire un fichier XML autonome lié par référence (*never*), ou bien si l'auteur a le choix entre ces deux alternatives (*userDependant*). Enfin, le texte d'un bloc est défini par une *textPrim* (*text.model*), tandis que les sections et les blocs sont titrés grâce à une *titlePrim* (*titre.model*). Des métadonnées plus élaborées peuvent être modélisées avec une *dataFormPrim*.

Générateurs

Les générateurs sont les primitives définissant les publications pour un format donné (HTML, OD...). Par exemple, un générateur web (*webSiteGenerator*) permet de publier les fragments XML au format HTML :

Figure 90 - site.generator (webSiteGenerator).

Parmi les différents éléments du générateur web, on note : une liste de *templates* de page ("squelette" HTML des pages du site) ; un dossier de *skin* (feuilles de style CSS) ; une liste de *transformers* (primitives de transformation, détaillées ci-dessous).

Primitives de transformation

Pour chaque primitive de document, il existe une primitive de transformation associée, relativement au générateur (*compositionXhtmlTransf*, *dataFormXhtmlTransf*, etc.). L'objet d'un *transformer* est de paramétrer la transformation XSL de la structure spécifiée (*bloc.model* dans le *transformer* ci-dessous) :

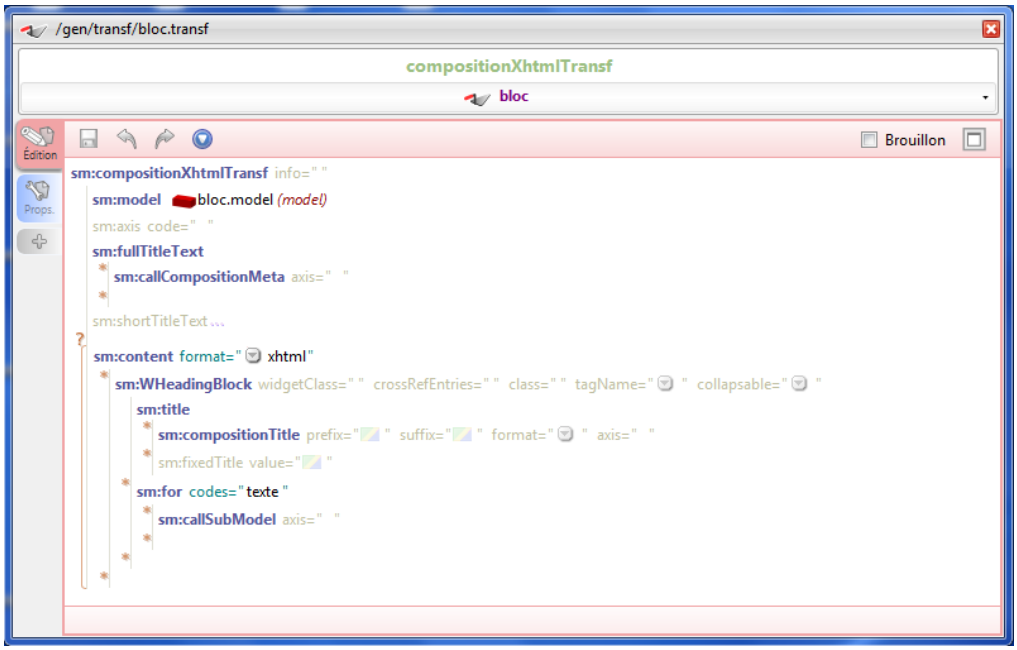


Figure 91 - bloc.transf (compositionXhtmlTransf).

Le contenu d'un bloc est publié dans une structure HTML comportant notamment un *heading* (*h1*, *h2*, *h3*... en fonction de la profondeur) pour le titre. L'instruction *sm:callSubModel* permet d'appeler le *transformer* de la structure de niveau inférieur (ici, *texte.transf*).

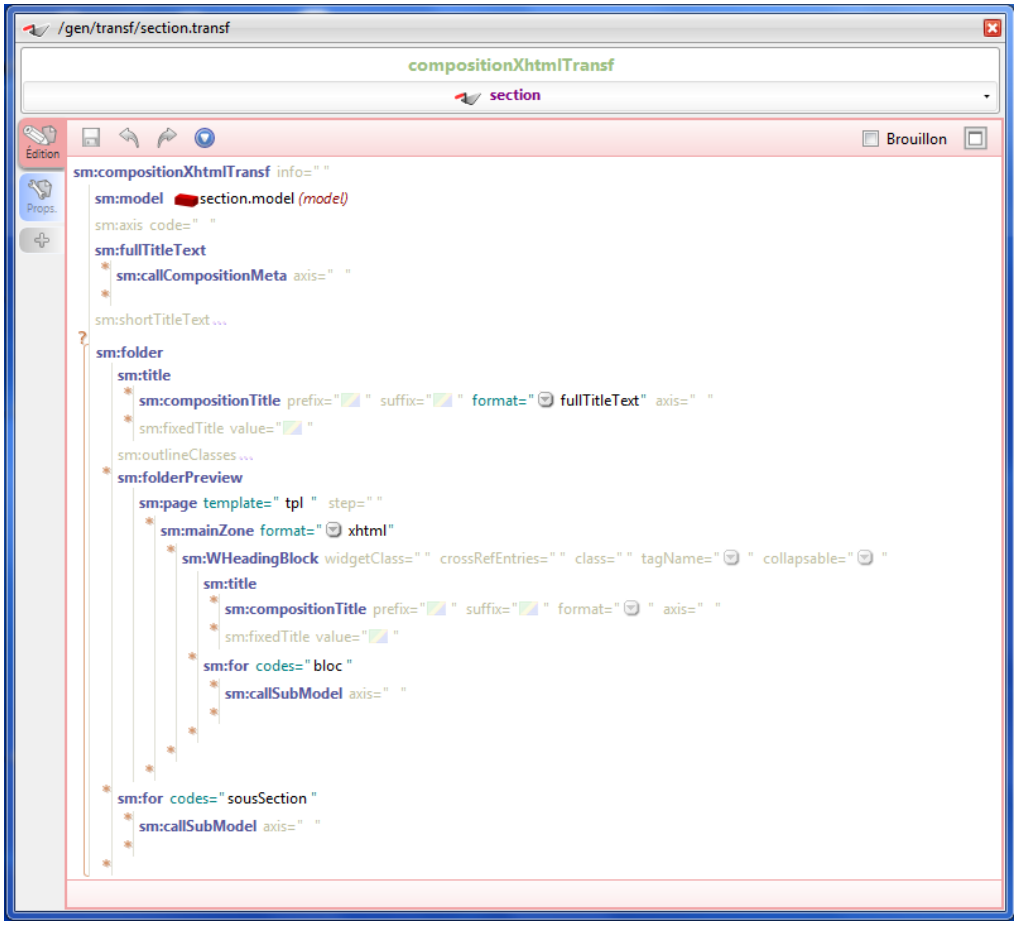


Figure 92 - section.transf (compositionXhtmlTransf).

Une page HTML est créée (*sm:page*) suivant le *template* "tpl", pour y accueillir le contenu des blocs de la section (*bloc.transf* est appelé via *sm:callSubModel*). Concernant les sous-sections, chacune d'entre elles occasionne la création d'une nouvelle page puisque ce *transformer* s'appelle récursivement.

Compilation du modèle

Une fois compilé via le *wspDefinition* (*exemple.wspdef*), le modèle documentaire est utilisable par l'auteur dans SCENARIchain (contenu inspiré de Wikipédia *(https://fr.wikipedia.org/wiki/The_Grand_Budapest_Hotel)*) :

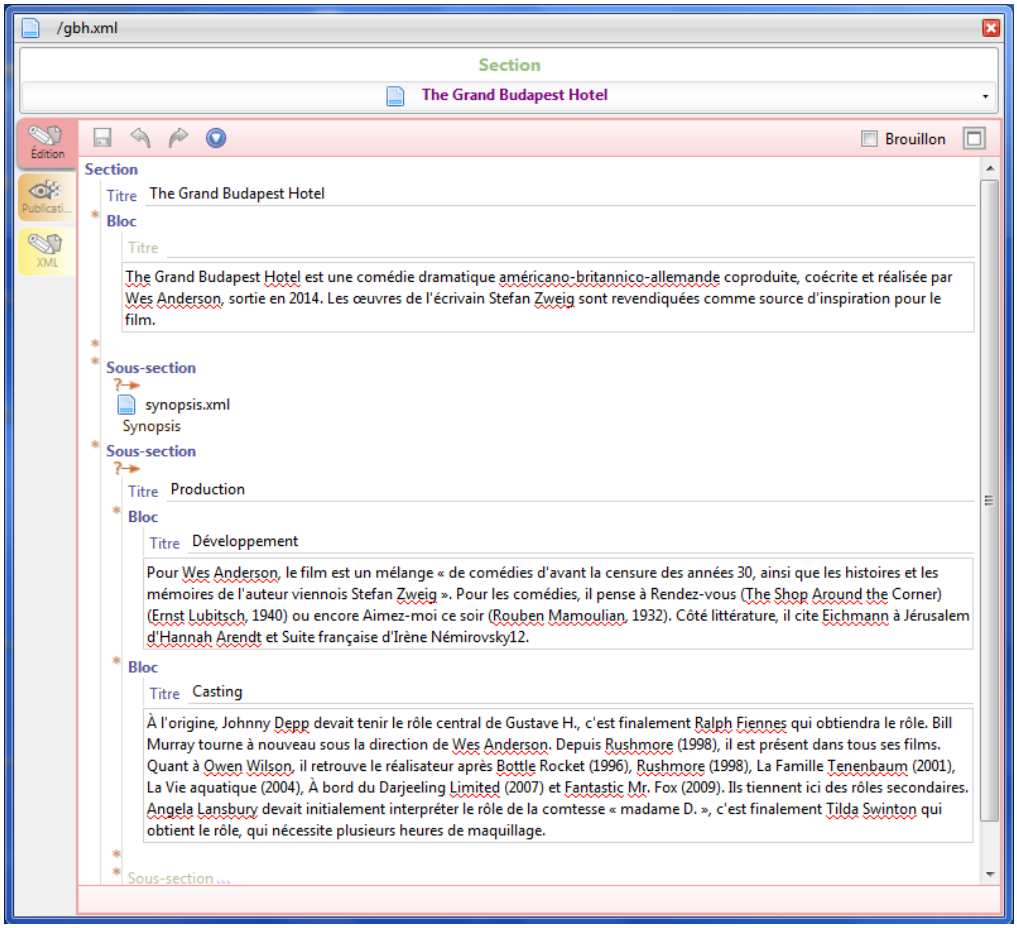


Figure 93 - gbh.xml (section).

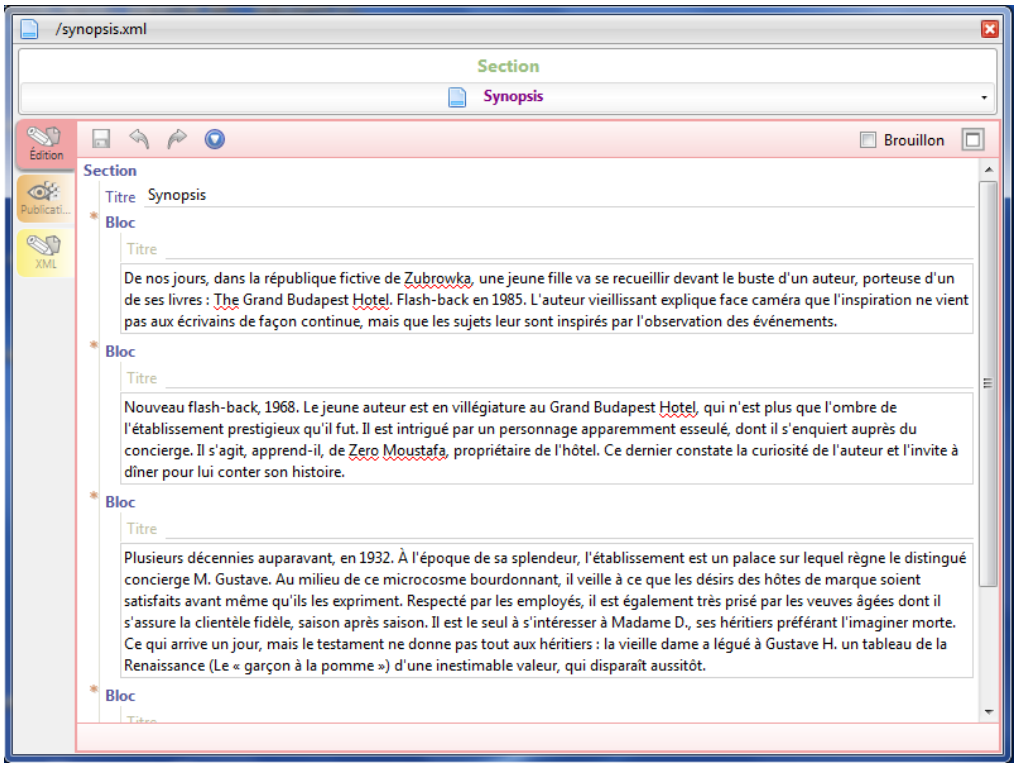


Figure 94 - synopsis.xml (section).

Le fragment *gbh.xml* peut ensuite être publié au format web :

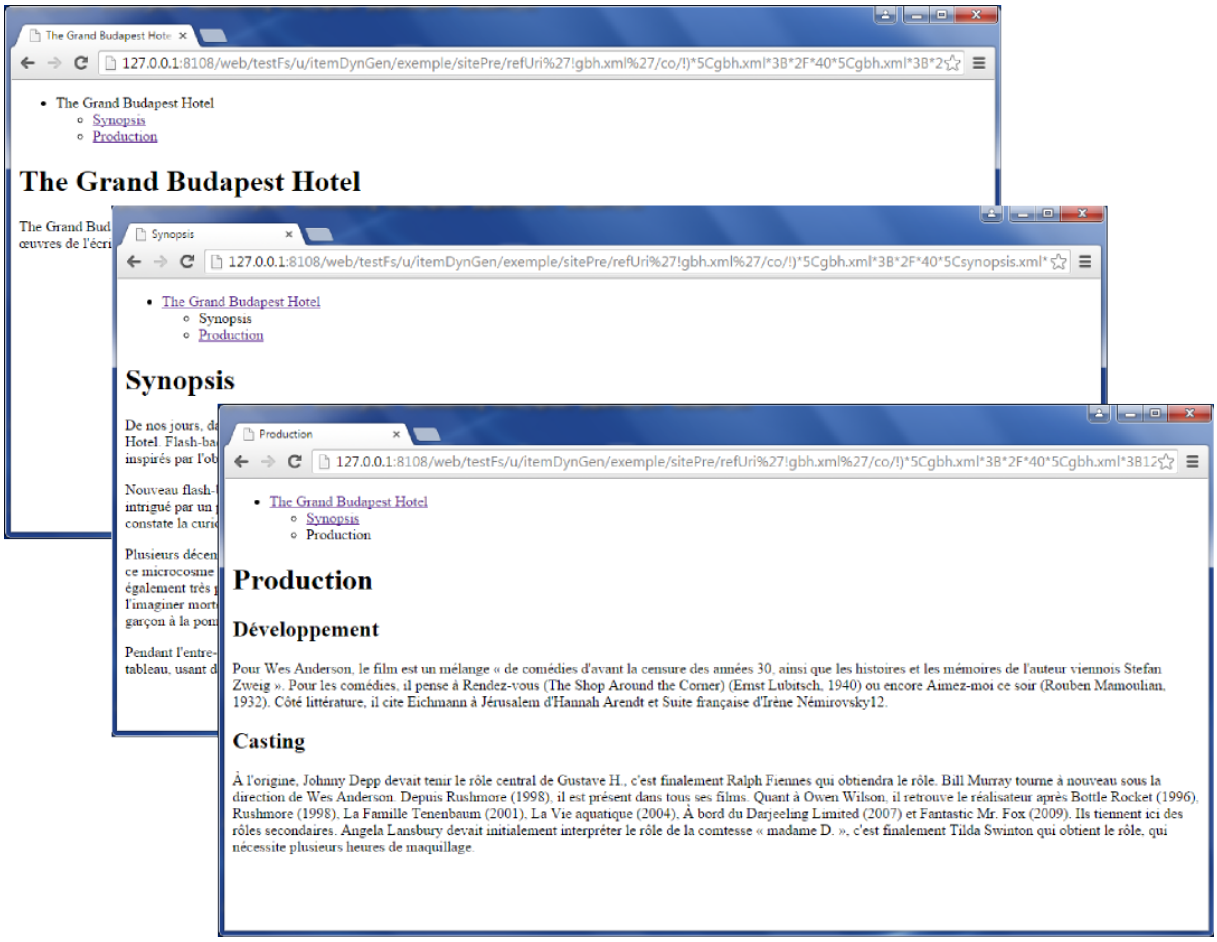


Figure 95 - Site web généré.

Ingénierie dirigée par les modèles

Un modèle est une représentation simplifiée de la réalité (Rothenberg, 1989). Un méta-modèle est ce à quoi se conforme un modèle (Bézivin, 2005). Par exemple, une carte géographique est un modèle puisqu'elle donne une vue simplifiée de la surface terrestre à l'échelle 1:1. De plus, la carte est "écrite" dans le langage graphique défini par sa légende, qui est donc le méta-modèle de la carte (*ibid.*).

En développement logiciel, l'ingénierie dirigée par les modèles consiste à mettre au point un algorithme de transformation permettant de générer un système à partir d'un modèle (représentation simplifiée du système). Cet algorithme ne peut fonctionner que si le modèle est conforme à un méta-modèle définissant l'ensemble des modélisations possibles du système.

Les modèles documentaires des chaînes éditoriales Scenari sont conçus selon l'approche de l'ingénierie dirigée par les modèles (Arribe, 2014). La conformité des différentes chaînes éditoriales au même méta-modèle (primitives) leur permet de partager des fonctions génériques (éditeur WYSIWYM, fonctionnalités de gestion, etc.). Autrement dit, le code spécifique de la chaîne éditoriale s'exécute en s'articulant au code générique de SCENARIchain.

La conception de chaînes éditoriales suivant l'approche de SCENARIbuilder permet finalement de combiner variabilité et efficacité (Arribe *et al.*, 2012) :

- les primitives sont génériques ;
- depuis le développement de SCENARIbuilder, les coûts de conception d'une chaîne éditoriale Scenari ont été divisé par dix.

Annexe 2

Théorie du support

La théorie du support (Bachimont, 2004) postule que tout objet technique est l'inscription matérielle d'une connaissance et que toute connaissance est d'origine technique.

Connaissance

Pour Bachimont : « Une connaissance est la capacité d'exercer une action pour atteindre un but. » (2004, p. 65). Bien qu'elle se rapporte *in fine* à l'action, cette définition n'insiste pas tant sur l'application directe d'une connaissance en action que sur le fait que la connaissance exprime la possibilité de réaliser l'action de manière non-seulement différée, mais surtout *répétable*. Liée à l'action, la connaissance est source de modification de ce sur quoi elle agit. Bachimont oppose ainsi les connaissances pratiques, exprimant un savoir-faire modifiant le monde matériel, aux connaissances théoriques, pour lesquelles les modifications s'opèrent dans le monde des représentations.

Inscription matérielle de connaissance

Par ailleurs : « est technique tout ce qui, par sa structure matérielle, prescrit et commande la réalisation d'actions possibles. » (*ibid.*, p. 70). Puisque la connaissance est précisément la capacité de réaliser une action, il s'ensuit qu'un objet technique est l'*inscription matérielle d'une connaissance*.

Les actions prescrites par les objets techniques ont pour origine les *saillances* présentées par les structures matérielles de l'environnement. Bachimont précise cependant que l'environnement ne contient pas les connaissances ni ne détermine les actions, celles-ci pouvant toujours ne pas être effectuées ou bien l'être autrement. Il ne s'agit donc pas de déterminisme mais de *conditionnement* de l'action par les structures matérielles. En mémorisant et en prescrivant l'action, ces structures se rendent disponibles pour la répétition de l'action, autre caractéristique fondamentale de la connaissance. Bachimont illustre ce point avec l'exemple d'une cisaille électrique qui, par la structure de sa poignée, prescrit et mémorise le geste à appliquer pour une utilisation sans danger, déchargeant ainsi l'utilisateur d'éloigner consciemment ses mains des lames cisaillantes.

Inscriptions instrumentales et inscriptions sémiotiques

Dans la lignée de l'opposition entre connaissances pratiques et théoriques, Bachimont distingue trois types de savoir qui renvoient à différents types d'inscriptions et d'objets associés :

- Le savoir-faire est associé à la figure de l'outil, qui est inscription du geste qu'il prescrit.
- Le savoir-produire est associé à la figure de la machine, qui est inscription du processus qu'elle reproduit. En outre, le savoir-produire peut être vu comme le savoir-faire étendu au geste automatisé par la machine.
- Le savoir-penser est associé à la figure du document, qui est inscription de la connaissance qu'il reformule.

Le savoir-faire (comprenant le savoir-produire) et le savoir-penser constituent alors deux classes

techniques d'inscription : les inscriptions instrumentales et les inscriptions sémiotiques. Les inscriptions sémiotiques ont la particularité d'être considérées pour ce qu'elles représentent et non pour ce qu'elles sont : Bachimont parle ainsi d'objets matériels *intentionnels*.

Genèse technique de la connaissance

La seconde thèse défendue par la théorie du support est que toute connaissance est d'origine technique. En effet, c'est parce que l'objet technique permet de prescrire une action qu'il mémorise la connaissance liée à cette action : l'objet technique est ainsi une mémoire externe.

Mais la mémoire peut aussi être interne dans le cas où la connaissance est portée par le corps biologique ou corps propre. Bachimont propose donc d'envisager le corps propre comme un cas particulier d'objet technique, et la pensée comme une dynamique de réinscription ou reformulation, par la conscience, des inscriptions ayant pour support un objet technique externe en inscriptions corporelles. Ainsi, la lecture est une reformulation d'inscriptions sémiotiques en inscriptions corporelles, l'écriture étant la dynamique inverse. Pour que le corps propre puisse être considéré comme le support de cette réinscription, il faut qu'il soit détaché de la conscience, comme l'est un objet technique situé dans l'environnement matériel externe au corps propre (et à la conscience). Ce dernier se distingue néanmoins des supports matériels externes dans le fait que d'une part, il est lié à l'évolution biologique du corps, entraînant ainsi de nombreuses transformations et par là même de nombreuses réinscriptions (Bachimont illustre cela avec l'image du *palimpseste*) ; et d'autre part, il est privé dans le sens où seule la conscience associée à ce corps peut accéder aux inscriptions dont il est le support.

Annexe 3

La linéarité étudiée à travers un cas d'usage

À travers l'œuvre *Un conte à votre façon* de Raymond Queneau, Bouchardon (2010) illustre trois caractéristiques de la linéarité : la structure du récit, l'organisation matérielle et la (dis)continuité de lecture.

Dans cette œuvre de structure réticulaire (qualifiée d'hypertexte papier ou de "proto-hypertexte"), chaque fragment propose au lecteur de décider de la suite du récit en le laissant choisir le fragment suivant parmi deux possibles :

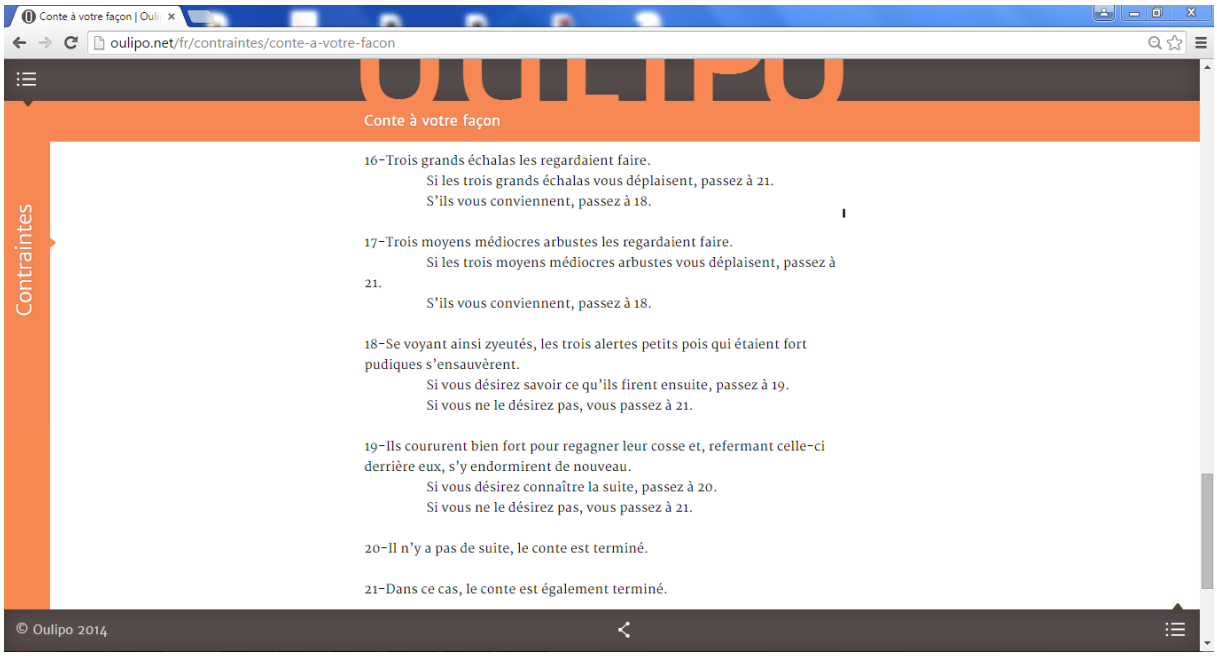


Figure 96 - Un conte à votre façon (source : http://oulipo.net/fr/contraintes/conte-a-votre-facon).

Dans la version papier d'origine, les fragments sont disposés les uns à la suite des autres à une moyenne de quatre fragments par page (Bouchardon, 2010, p. 77). Ainsi, après le fragment 16 par exemple, le lecteur peut choisir d'aller en 18 ou 21, ou bien de lire 17 avant 18 puis 19 et 20 avant 21. Dans le premier cas, il s'agit d'une lecture discontinue en accord avec la structure réticulaire du récit ; dans le second, on a une lecture continue en accord avec la linéarité matérielle du support.

Cette œuvre a eu deux adaptations numériques, par Antoine Denize et Bernard Magné (*Machines à écrire*) d'une part, et de par Gérard Dalmon d'autre part. Ces deux versions interactives proposent différentes façons de transposer les "liens" entre fragments :

- Dans la version de Denize et Magné, des indices visuels et sonores orientent le lecteur, après chaque fragment, vers un fragment ou un autre. Les fragments suivis s'écrivent les uns après les

autres sur un cahier (utilisé comme métaphore visuelle).

- Dans la version de Dalmon, le lecteur choisit le prochain fragment en cliquant sur un lien "Oui" ou "Non". Dès lors, le nouveau fragment se substitue à l'ancien (une page équivaut à un fragment).

Il s'ensuit que le support numérique ne permet pas ici d'appréhender *Un conte à votre façon* dans son ensemble. En effet, dans le premier cas les fragments non-suivis n'apparaissent à aucun moment sur le cahier, et dans le second, un fragment n'est vu que si le lien permettant d'y accéder est suivi. Pour Bouchardon, cette différence est fondamentale (p. 78) :

- « Pour le lecteur, le fait de cliquer sur une partie de l'écran pour passer au fragment suivant est [...] très différent du fait de *sauter* du texte. Dans la version papier, le lecteur, qui a une appréhension de la globalité du texte du conte, est conscient de ce qui ne sera pas lu, de ce qui est *perdu* dans la lecture [...]. Il éprouve un sentiment de discontinuité dans sa lecture. »
- « Toute autre est l'impression éprouvée par le lecteur des deux versions interactives : que chaque fragment se substitue au précédent ou qu'il vienne s'écrire à la suite du précédent, le lecteur aura - paradoxalement - l'impression d'une continuité dans l'adaptation de ce récit non-linéaire [...]. »
- Bouchardon souligne par ailleurs qu'il y a plus d'effort de la part du lecteur dans la version papier que dans les versions interactives. En effet dans la première, il s'agit de chercher un fragment distant potentiellement de plusieurs pages, et non d'un simple "tourne-page". Inversement, le lien hypertexte joue pleinement son rôle de *prise en charge* dans les versions interactives (la machine traite elle-même la navigation vers l'unité narrative suivante).

Annexe 4

Recherche d'un plus long chemin dans un graphe

Le plus long chemin d'un graphe peut être calculé à l'aide de l'algorithme de Bellman. Pour aborder ce problème, nous nous appuierons sur un exemple d'utilisation de la méthode *potentiel-tâche* utilisée en gestion de projet industriel. Cet exemple, bien que n'ayant rien à voir avec notre cadre documentaire, nous permettra d'illustrer plus facilement la notion de plus long chemin à l'aide d'un cas d'application assez classique.

Sources

- Méthode potentiel-tâche : http://ressources.aunege.fr/nuxeo/site/esupversions/2b1c56b6-109d-488a-94a3-3ea525f8beef
- Algorithme de Bellman : http://www7.inra.fr/mia/T/schiex/Export/Sup2013.pdf

Méthode potentiel-tâche

La méthode potentiel-tâche permet notamment de déterminer la durée théorique d'un projet à partir des durées de réalisation des tâches de ce projet ainsi que des contraintes de succession entre ces tâches. Soit l'exemple suivant, à partir duquel nous allons appliquer cette méthode :

Tâche	Label	Durée (en semaines)	Tâches préalables
Fondation et maçonnerie	A	7	aucune
Plan des aménagements intérieurs	B	8	aucune
Toiture	C	2	A
Installations électriques et sanitaires	D	4	A et B
Façade	E	3	C et D
Peintures intérieures	F	1	C et D

Tableau 9 - Projet de construction d'un bâtiment.

Le graphe ci-dessous est construit à partir des données du tableau des tâches. Les nœuds du graphe

représentent les tâches, tandis que les liens indiquent les contraintes de succession entre tâches. Chaque lien est valué par la durée de la tâche de son nœud d'origine. Enfin, deux tâches fictives représentant respectivement le début (α) et la fin (ω) du projet sont ajoutées au graphe (les liens partant du nœud α sont valués à 0).

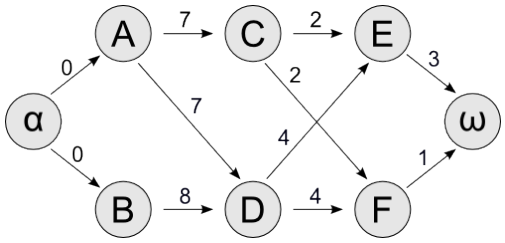


Figure 97 - Graphe utilisé pour la méthode potentiel-tâche.

Pour chaque tâche, on cherche à connaître ses dates de réalisation au plus tôt et au plus tard, afin de savoir si un retard sur cette tâche est critique par rapport au projet global. Ces dates sont exprimées relativement à la date de début du projet, fixée à 0. Par exemple, si une tâche a pour date au plus tôt 8 et pour date au plus tard 10, elle a une marge de deux semaines en cas de retard de réalisation. En revanche, si les dates au plus tôt et au plus tard sont égales, cette tâche est critique car tout retard aura un impact sur la durée globale du projet. Pour déterminer la date au plus tôt d'une tâche, on procède par comparaison de la somme des durées sur chaque chemin permettant d'accéder à cette tâche depuis la tâche α. En effet, parmi les différents chemins possibles entre deux mêmes tâches, c'est celui correspondant à la durée maximale qui doit l'emporter. Par exemple :

- la date au plus tôt de A est 0 car il n'y a qu'un seul chemin pour l'atteindre depuis α, valant 0 (*idem* pour B) ;
- la date au plus tôt de C est 7 car il n'y a également qu'un seul chemin pour l'atteindre, valant 7 (0 + 7) ;
- la date au plus tôt de D est 8, en suivant le chemin α-B-D (0 + 8 = 8) plutôt que α-A-D (0 + 7 = 7) ;
- la date au plus tôt de F est 12, en suivant le chemin α-B-D-F (0 + 8 + 4 = 12) plutôt que α-A-D-F (0 + 7 + 4 = 11) ou encore α-A-C-F (0 + 7 + 2 = 9) ;
- ... et ainsi de suite en parcourant tous les nœuds : E et finalement ω ont pour date au plus tôt 12 et 15 ;
- ainsi, le projet global aura une durée de 15 semaines (s'il n'y a pas de retard).

Notons que la date au plus tôt d'une tâche *i*, notée *ti*, est définie formellement comme suit : $ti = Max(tj + dj)$, avec :

- *j* appartenant à l'ensemble des nœuds précédant *i* ;
- *dj* étant la durée de *j* ;
- la tâche initiale *t0* ayant pour date au plus tôt 0.

Le problème illustré par cet exemple correspond en fait à la recherche d'un chemin le plus long (par somme des valeurs des liens) entre deux nœuds d'un graphe. En effet, les dates au plus tôt de chaque tâche correspondent à la longueur du chemin le plus long entre α et le nœud représentant cette tâche. Si l'on parcourt le graphe à l'envers (de ω à α) en choisissant à chaque fois la nœud précédant de sorte que la somme de sa date au plus tôt et de sa durée soit égale à la date au plus tôt du nœud courant, on obtient le chemin le plus long dans le graphe : de ω, on va en E (12 + 3 = 15) ; de E, on va en D (8 + 4 = 12)... jusqu'à obtenir α-B-D-E-ω comme plus long chemin. Notons, sans nous attarder sur ce point, que les dates au plus tard de chaque tâche peuvent également être déterminées par un parcours inverse du graphe.

Algorithme de Bellman

L'algorithme de Bellman permet de rechercher les plus courts chemins entre deux nœuds d'un graphe orienté *sans circuit* (GOSC) et dont les arcs sont pondérés. Il est classiquement appliqué pour des problèmes d'optimisation : par exemple, on cherche à minimiser le coût du trajet entre deux villes entre lesquelles plusieurs itinéraires sont possibles (les arcs du réseau routier sont par exemple pondérés en fonction du nombre de kilomètres, du prix des péages, etc.). C'est l'adaptation de cet algorithme à la recherche des plus *longs* chemins qui nous intéresse ici.

Cet algorithme repose sur le principe de *relaxation*, que l'on peut expliquer comme suit. Soient :

- *p[ ]* le tableau des poids de chaque arc ;
- *d[ ]* le tableau de distance de chaque sommet à S0, initialisé avec d[S0] = 0 et d[Si] = -∞ pour tout i différent de 0 ;
- *pred[ ]* le tableau des prédécesseurs de chaque sommet sur le plus long chemin, initialisé à *null* pour tous les sommets.

```
relaxation(u, v) {
  si d[u] + p[u,v] > d[v] alors d[v] = d[u] + p[u,v]
  pred[v] = u
}
```

L'idée de l'algorithme est la suivante :

- étant donné un sommet *u*, la procédure *relaxation(u, v)* est appelée pour tous les sommets *v* adjacents à *u* ;
- le traitement précédent est répété sur chaque sommet ordonné suivant un tri topologique (c'est-à-dire qu'un sommet sera toujours traité avant ses successeurs) ;
- une fois tous les nœuds traités, on peut reconstruire le(s) plus long(s) chemin(s) en suivant les prédécesseurs depuis le sommet final dans le tableau *pred[ ]*.

Notons que si les sommets ne sont pas ordonnés selon un tri topologique, le traitement devra être répété jusqu'à ce que la relaxation ne fasse plus effet.

Nous illustrons le déroulement de cet algorithme à l'aide du graphe et du tableau ci-dessous. Dans le tableau, les valeurs successives sont séparées par des slashs. On remarque que la distance du sommet A au sommet E a été relaxée deux fois. Le chemin le plus long de ce graphe est donc A-B-D-E.

Le tableau ci-dessous donne le résultat de l'exécution de l'algorithme sur ce graphe. Les valeurs successives des distances et des prédécesseurs sont séparées par des slashs. On remarque que la distance du sommet A au sommet E a été relaxée deux fois. Le chemin le plus long de ce graphe est donc A-B-D-E.

Sommet	Distance	Prédécesseur
A	-∞/0	*null*
B	-∞/4	*null*/A
C	-∞/1	*null*/A
D	-∞/7	*null*/B
E	-∞/7/11	*null*/B/E

Tableau 10 - Exemple d'application de l'algorithme de Bellman.

Application au graphe d'un document multilinéaire

Pour appliquer l'algorithme de Bellman dans notre contexte, il est nécessaire de valuer les liens entre les nœuds du graphe : une méthode triviale consiste à attribuer la valeur 1 à tous les liens. Il

convient également d'ajouter une étape fictive ω liée par toutes les étapes finales (20 et 21 dans le graphe d'*Un conte à votre façon*), ainsi que de supprimer les liens provoquant des circuits (le lien allant de 8 à 7 dans notre exemple). De cette façon, nous pouvons calculer les distances maximales de l'étape 1 à toutes les autres étapes puis en déduire un ou plusieurs plus longs chemins de 1 à ω, en nous basant sur la même démarche que celle de l'exemple de la méthode potentiel-tâche.