



PERITEXT

Publication
Multimodale
Orientée
Contexte

Une introduction

Sommaire

Introductionss	3
Le format de données peritext	10
Des documents de très bonne qualité technique et éditoriale	17
Adaptation et modularité.....	20
Contextualiseurs.....	22
Contextualiseur bib	24
Contextualiseur glossary	26
Contextualiseur webpage	27
Contextualiseur embed	30
Contextualiseur vidéo	33
Contextualiseur dicto	35
Contextualiseur table	37
Contextualiseur vegalite	38
Contextualiseur data-presentation	41
Contextualiseur codefiles	45
Contextualiseur p5 (processing).....	47
Gabarits	49
Générateurs.....	50
Éditeur Ovide	52
Contribuer à peritext.....	53
À propos	54
Références	56
Glossaire.....	64
Index des auteurs	65
Notes.....	66

Introductionss

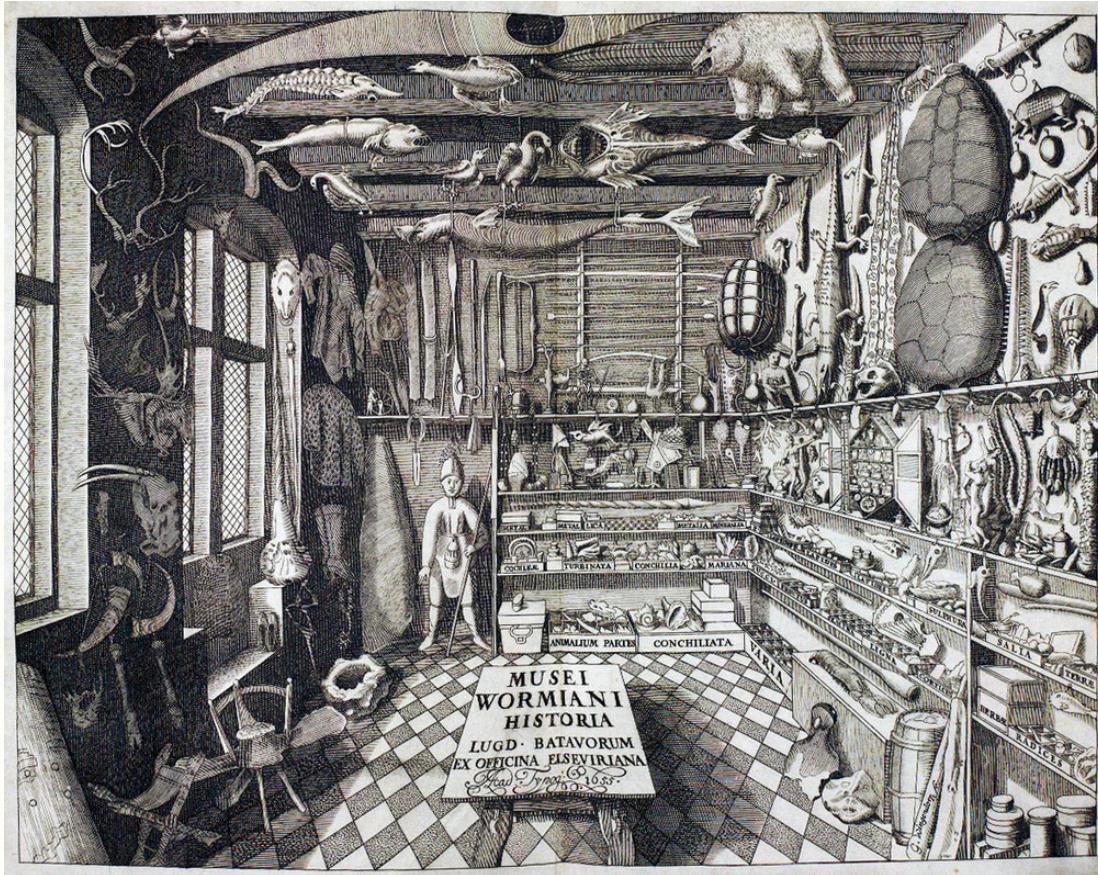
Ce site est dédié à la présentation (et la démonstration) de l'écosystème Peritext, un projet ouvert et libre développé pour supporter des pratiques de publication académique tirant davantage parti du web et des documents divers convoqués dans les recherches, et autorisant une publication à haute qualité graphique et éditoriale sur toute la palette des formats de communication utilisés par les chercheurs aujourd'hui.

En quoi consiste Peritext ? pour commencer, à produire le site que vous lisez en ce moment !

Pourquoi Peritext

En amont comme en aval de l'acte de publication, les textes produits et échangés par les chercheurs universitaires convoquent de manière de plus en plus intime les environnements numériques (au premier rang desquels le web) dans lesquels ils sont inscrits.

On assiste au surgissement d'une variété d'objets hétéroclites qui participent de manière de plus en plus active aux processus de recherches. En témoignent des exemples aussi divers que la prolifération d'enregistrements audiovisuels en ligne produits ou analysés dans un contexte scientifique, l'utilisation des réseaux sociaux et de twitter comme élément de corpus ou de restitution des débats en cours, la mobilisation d'archives et de dépôts de sources numérisées rendant possible leur présentation sous forme diagrammatique ou tabulaire.



Le cabinet d'Ole Worm

Reproduction de la gravure conservée à Modène représentant un cabinet de curiosités

Source: *Bibliothèque Estense, Modène.*

Cela dit, dans les formats les plus répandus de l'édition scientifique, les traces du web et autres ressources numériques peinent encore à sortir de la marge des pages : on les remarquera subrepticement au détour d'une note de bas de page (sous la forme d'un discret hyperlien), ou bien timidement reproduites dans une annexe secondaire, à la fin d'une entrée bibliographique ou dans la légende d'une figure, etc. Ces connexions à des éléments externes désignent des ressources riches, dont le contenu contribue aux arguments développés, souvent disponibles librement par

ailleurs. Elles restent pourtant souvent sur le seuil des échanges et des arguments, notamment du fait qu'elles s'effacent des documents une fois traduites et formatées selon les conventions de publication instituées (articles, monographies, actes, mémoires, thèses, ...).

D'un autre côté, l'essor de la publication en ligne, du mouvement de l'accès ouvert au résultats de la recherche, et de nouvelles pratiques de communication de la part des chercheurs eux-même qui vont de l'usage des réseaux sociaux à l'expérimentation de diverses formes de publication en ligne, ont instauré un nouveau régime de production et de circulation des documents académiques où ces derniers se voient discutés, critiqués, reformatés et éditorialisés par-delà une multitude de mondes scientifiques, sociaux et médiatiques.

Il est donc nécessaire d'inventer de nouvelles manières de connecter les modes de production des documents académiques avec les contextes du web. Cette recontextualisation doit opérer dans les deux sens : les environnements numériques et toutes les ressources et formes qu'ils proposent doivent être plus significativement convocables dans les documents académiques, et les documents académiques doivent être plus significativement contextualisables dans les environnements numériques. C'est là le projet à la base de Peritext.

Pour quels contextes Peritext est-il fait (et pas fait) ?

Peritext ne s'adresse pas à toutes les pratiques de publication académique.

Peritext pourrait être pertinent pour vous si vous remplissez au moins deux des conditions suivantes :

- Vos recherches convoquent de manière très importante des documents extérieurs (tableaux de données, médias, sites web, ...) et votre activité consiste notamment à les analyser et les commenter en profondeur

- Vous désirez publier au-delà des formats traditionnels associés à la promotion et la reconnaissance des chercheurs, notamment en ayant une activité sur le web
- Vous êtes attaché à la qualité technique des documents que vous produisez, notamment en termes de métadonnées et d'indexation
- Vous êtes attaché à la qualité graphique et éditoriale des documents que vous produisez en tant que chercheur, que ce soit vous ou un membre de votre équipe qui s'en occupe

D'un autre côté, Peritext en tant que tel ne prend pas en charge les dimensions suivantes du travail de communication académique :

- la dimension collective & réticulaire du travail de communication scientifique (commentaires, réactions, collaboration, ...). Tout cela se passe autour d'un document peritext, mais pas en son sein. Ce point ne devrait pas évoluer dans le futur du projet, et l'utilisation d'outils tiers tels qu'Hypothes.is ou disq.us sera favorisé si besoin.
- l'expérimentation de structures rhétoriques qui se départissent de l'organisation traditionnelle des documents académiques, faits de sections (chapitres, parties, ...) se suivant de manière linéaire, et d'un hypertexte assez simple permettant de produire une table des matières, un glossaire, et une liste de référence. Ce point est soumis à une probable évolution ultérieure.
- l'expérimentation de formes de publication académique qui ne prennent pas le texte pour médium principal et structurant pour l'argumentation.

Qu'est-ce que Peritext ?

Peritext est d'abord une "spécification technique" pour la description de documents académiques visant à terme à devenir un format ouvert et interopérable. Autrement dit, il s'agit d'un mode d'emploi expliquant comment décrire le contenu d'un document académique d'une façon

conventionnelle & prévisible. Ce dernier décrit comment les données d'un récit écrit par un chercheur avec un outil compatible doivent être structurées, et contribue à produire un *format de données* particulier, le *format Peritext*.

Autour du *format Peritext* s'articulent un ensemble de modules technologiques et des applications directement utilisables qui en actualisent les potentialités et lui donnent du sens¹. Ces modules technologiques ont tous en commun le fait qu'ils visent à aider à automatiser toutes les tâches relevant des normes techniques et éditoriales liées à la publication académique contemporaine, tout en ménageant beaucoup de place et de souplesse pour l'affichage de contenus non-conventionnels et leur design en accord avec l'argument fait par le texte.

Une écriture et une édition multi-supports

Le premier atout du format peritext est que sa manière de décrire un document sous la forme d'une représentation intermédiaire unique permet de produire une diversité de documents de type "codex" (pdf, epub) et de type "site web" à partir d'un même contenu.

Peritext rentre donc dans la catégorie de technologies intitulées "WRPE" ("Write Once, Publish Everywhere"²) qui consistent à produire une diversité de documents à partir d'un même texte source. Ainsi, avec Peritext les auteur.e.s peuvent écrire une seule fois leur article, chapitre ou livre, et disposer en retour et simultanément de diverses contextualisations de ce contenu pour les différents supports (site web, pdf, epub, ...).

Cela dit, Peritext se distingue une première fois de ses voisins dans la mesure où il permet également aux utilisateurs de se faire "éditeurs-designers" et disposer de divers moyens très détaillés de mettre en forme leur récit pour chacun des supports disponibles, en fonction de leur corpus, de leur argument et de leur projet intellectuel.

Un modèle unifié et souple pour gérer les arguments centrés sur des documents et autres éléments extérieurs

Outre la dimension multi-supports³ qui le caractérise une première fois, Peritext tire une seconde spécificité de sa vocation à soutenir la convocation de nombreux éléments extérieurs dans la structure argumentative d'un texte académique (références, données, documents audiovisuels, etc.). Pour ce faire, il repose sur un format de données particulier centré sur la séparation entre les ressources convoquées dans un récit et les différentes formes que ces dernières peuvent prendre dans le cadre de l'argumentation opérée par un texte.

Pour en savoir plus, rendez-vous à la section "Le format de données peritext".

Respect des normes techniques et éditoriales de la publication académique

Sur le plan académique, peritext prend en charge une série de nécessités spécifiques à la publication académique.

Pour en savoir plus, rendez-vous à la section "des documents de très bonne qualité technique et éditoriale".

Adaptation et modularité

Peritext est pensé selon une logique de "boîte ouvrable", c'est-à-dire que ses différents modules fonctionnent avec des paramètres par défaut mais sont par ailleurs profondément paramétrables par les utilisateurs qui le souhaitent.

Outre les paramètres exposés par chacun des modules, peritext prend

en charge une diversité de langages permettant de préciser et de modeler les contenus avec beaucoup de liberté, comme par exemple le langage markdown pour certains contextualiseurs et les langages de mise en forme que sont css et css print qui peuvent être utilisés de manière distincte pour chacun des types de supports produits à partir d'un contenu.

Enfin, il est construit selon une logique modulaire qui permet beaucoup de souplesse et de variations dans l'implémentation d'un écosystème éditorial spécifique (en termes de supports visés, de types de documents pris en charge, ...).

Pour en savoir plus, rendez-vous à la section "adaptation et modularité".

Économie de moyens technique

La représentation d'un document peritext est faite au format json, ce qui veut dire qu'elle ne nécessite pas de base de données et peut être stockée dans un simple fichier. De la même manière, les modules composant peritext ont vocation à être assemblés à l'intérieur d'une application javascript, pouvant être déployée sur quantité de services gratuits.

Enfin, le générateur de site web actuel produit un répertoire de fichiers html simples, qui peuvent être exportés sur un serveur ftp, ou sur un service d'hébergement gratuit tels que github pages ou surge.

En conclusion, même s'il peut être intégré dans une infrastructure ou une plateforme de grande envergure, l'écosystème peritext est d'abord pensé pour des contextes d'implémentations légers, décentralisés et peu coûteux en équipement.

Pour en savoir plus, rendez-vous par exemple à la section "générateurs".

Le format de données peritext

Le schéma de données du format peritext est dédié à la description de "récits" académiques, qui peuvent correspondre à des formats traditionnels aussi variés que :

- un article de revue académique
- une monographie
- des actes de colloque
- un numéro de revue académique

... ou à des formes ne correspondant pas à un contexte d'édition académique établi.

Techniquement, peritext se fonde sur le format json, qui permet de décrire un objet selon une structuration complexe faite d'éléments imbriqués, et de stocker cette description dans de simples chaînes de caractères et autres fichiers de texte.

La spécification exacte du format peritext n'est actuellement pas complètement fixée¹, elle est néanmoins définie dans un premier module du projet intitulé peritext-core [\(30\)](#) ².

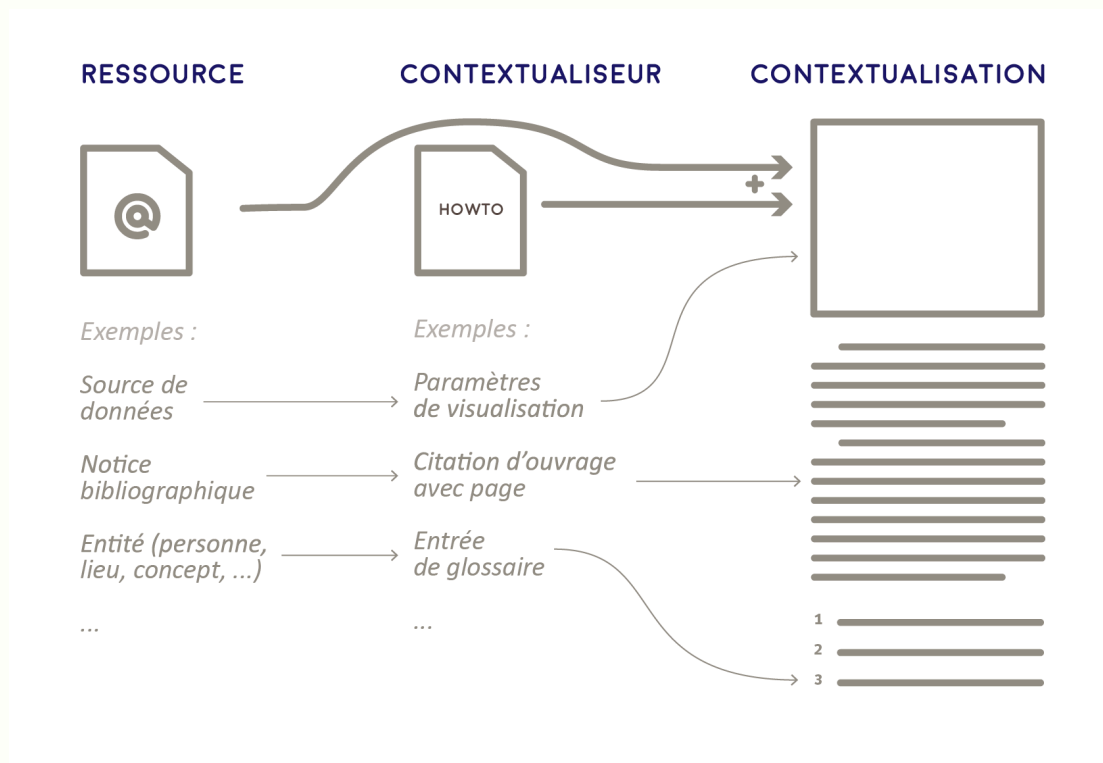
Les paragraphes qui suivent en décrivent et justifient les grandes lignes. Pour une idée plus précise du schéma, voir le répertoire peritext-core ou sa documentation technique [\(31\)](#).

Caractéristiques générales du format peritext

Le format peritext est d'abord structuré autour des aspects suivants :

- un récit Peritext est décrit par une série de métadonnées qui décrivent son titre et sous-titre, ses auteurs, etc.
- un récit Peritext est structuré en *sections*. Une section peut être une partie, un chapitre, une introduction, des remerciements, ... les sections sont assemblées dans un ordre linéaire pour construire le *sommaire* du récit. Elles peuvent disposer d'un niveau hiérarchique qui permet de définir les sections les plus générales et les sections d'importance moindre.
- chaque section dispose de ses propres métadonnées (titre, auteurs, ...) et d'un contenu structuré autour d'un texte principal et d'une série de notes.
- les contenus d'une section ou d'une note en particulier sont aussi représentés sous la forme de blocs de textes annotés avec des styles et des entités spécifiques.³(28)
- outre les sections et leur contenu, au niveau d'un récit peritext sont également définis des paramètres (settings) qui définissent des informations relatives à l'affichage du récit telles que : le gabarit à utiliser, les règles de citation, des paramètres tels que la position des notes, et enfin le code css à injecter pour chacun des types de supports
- enfin, un récit peritext est décrit par une série d'objets relatifs à la contextualisation d'éléments extérieurs dans l'argument : les ressources, les contextualiseurs et les contextualisations

Caractéristiques spécifiques liées à l'argumentation fondée sur les documents : le modèle RCC



Le modèle de document Peritext

Peritext tire sa spécificité de sa vocation à soutenir la convocation de nombreux éléments extérieurs dans la structure argumentative d'un texte académique (références, données, documents audiovisuels, etc.).

Pour ce faire, dans un récit Peritext, la convocation d'éléments extérieurs dans l'argument du récit s'effectue via trois types d'objets :

- **les ressources** décrivent tous les éléments à convoquer dans le texte, selon un **registre documentaire** indifférent à la forme qu'ils prendront dans le récit. Les ressources peuvent désigner des éléments très variés dans leur nature (références, données, médias, mais aussi entités de glossaire telles que personnes, lieux, thèmes, ou encore morceaux de code, ...)
- **les contextualiseurs** décrivent un ensemble de paramètres

permettant d'afficher une ressource d'une certaine manière. Ils agissent sur un **registre rhétorique** pour préciser comment la ressource doit être affichée.

- **les contextualisations** enfin sont la conjonction d'une ressource et d'un contextualiseur à un endroit précis de l'argument. Elles sont par ailleurs assorties d'un titre et d'une légende.

Le modèle de données de Peritext est donc intitulé RCC car il sépare la liste des Ressources utilisées dans le documents, les Contextualisations de ces ressources à des endroits spécifiques et les Contextualiseurs qui les font s'afficher d'une manière spécifique.

Démonstration du modèle RCC

Soit une ressource de type "table", c'est-à-dire un simple tableau de données.

Il est d'abord possible de la contextualiser au moyen d'un contextualiseur de type "table", qui est le contextualiseur par défaut de ce type de ressource. Ce dernier consiste à représenter la ressource sous la forme d'un tableau. En l'utilisant pour contextualiser la ressource dans le paragraphe qui suit, on obtient :

a	b
A	2
B	4
C	6
D	8
E	2

a	b
F	3
G	5

Dataset très simple

Maintenant, il est possible que dans un récit donné, l'auteur souhaite à nouveau faire mention de cette ressource sans pour autant souhaiter l'afficher à nouveau comme un encombrant tableau. Pour ce faire, il pourra par exemple utiliser le contextualiseur "bib" qui contextualise la ressource sous la forme d'une référence bibliographique. C'est ce qui va être fait à la fin de ce paragraphe⁽²⁶⁾.

Enfin, mettons que notre auteur décide maintenant de discuter de la comparaison des valeurs numériques décrites dans les données de cette ressource. Il utilisera alors un troisième contextualiseur de type "vegalite", qui permet de construire des visualisations de données. C'est ce qui va être fait dans le prochain paragraphe :

Dataset très simple

C'est ainsi que le modèle RCC implémenté par le format Peritext permet :

- de représenter les éléments convoqués dans un argument de plusieurs manières différentes, selon les besoins de l'argumentation
- de paramétrer finement la manière dont on veut voir affichée une ressource grâce aux paramètres des contextualiseurs et à l'injection

de code css via l'éditeur

- de maintenir une rigueur documentaire qui pourra être exploitée ensuite pour l'indexation du document et pour la génération d'une interface utilisateur exploitant le système hypertexte ainsi créé. En effet, malgré la diversité des diverses formes de contextualisations pouvant être appliquées à une ressource donnée, le lien avec cette dernière restera toujours présent, permettant par exemple de produire une liste de mention ou une liant ressources et contextualisations. Par exemple, dans le présent gabarit de présentation, il est à noter qu'en cliquant sur l'icône en dessous de chacune des contextualisations il est possible de prévisualiser et d'accéder à toutes les autres contextualisations de la même ressource dans le récit.

Remarque sur les principes du format peritext et sa relation à la forme

Dans une application d'édition suivant le modèle de données Peritext, l'auteur a donc non seulement la main sur les éléments extérieurs qu'il veut citer dans son texte, mais également sur la manière de les afficher et sur un ensemble de paramètres lui permettant de modéliser précisément l'usage qu'il veut en faire dans son argument (par exemple : préciser un passage de texte, mettre en lumière une certaine dimension d'un jeu de données, spécifier un cadrage particulier sur une visualisation, etc.). **Il a ainsi ainsi la possibilité de développer un fil argumentatif pleinement articulé autour de la mobilisation des ressources qu'il convoque.**

Peritext tente de se départir de la traditionnelle distinction entre les modèles de description centrés sur les contenus (*What You See is What You Mean*⁴) et ceux centrés sur la forme (*What You See is What You Get*⁵). Il s'agit plutôt de modéliser les *intentions argumentatives* attachées à un moment particulier du texte⁶, plutôt qu'une "forme finie" ou un "contenu sans forme". La notion de *représentation intermédiaire* joue ici un rôle crucial, comme nom de cet état intermédiaire des contenus décrits selon le format

Peritext.

Des documents de très bonne qualité technique et éditoriale

Le modèle RCC contenu dans Peritext (voir la section "le format de données Peritext") induit une haute qualité de description des ressources impliquées dans la production des documents peritext.

Gestion des références bibliographiques

Peritext s'adosse au standard international Citation Style Language [\(21\)](#) pour gérer les références. Il est ainsi capable de gérer la plupart des formats et langues de citation en utilisation. Les modules de l'écosystème permettent par ailleurs de prendre en charge n'importe quel type de ressource Peritext sous la forme d'une citation bibliographique.

L'éditeur démonstrateur Ovide (voir section afférente) propose par ailleurs une interface d'import de références au format BibTeX et une interface expérimentale d'ajout de références par interface graphique.

Métadonnées à l'échelle d'un récit et de ses sections

Dans les gabarits par défaut de l'écosystème de Peritext, les métadonnées renseignées au niveau des récits et des sections d'un document peritext sont mises à profit de la manière la plus complète possible.

Ainsi, les fichiers de sortie html de peritext présentent des métadonnées encodées au format DublinCore, Twitter, Open Graph & schema.org.

Métadonnées à l'échelle des éléments

À l'échelle des éléments affichés par les générateurs de Peritext (et notamment les contextualisations de ressource), Peritext entend fournir le meilleur degré de description machine possible.

Pour ce faire, chaque mention de ressource inclue l'imbrication automatique d'un composant COiNS [\(20\)](#) permettant par exemple de l'enregistrer avec le plugin Zotero.

Est prévue dans la feuille de route des modules une implémentation ultérieure des normes d'encodage sémantique au niveau des composants html eux-mêmes (au moyen du format RDFa) afin de permettre une lecture optimale d'un document peritext par une machine.

Systèmes hypertexte à double-sens

Peritext est attaché à une utilisation optimale de la technologie hypertexte, initialement pensée pour les documents académiques [\(29\)](#) . Sans explorer trop avant les possibilités rhétoriques et organisationnelles offertes par cette technologie dans le contexte académique ¹, Peritext tente d'optimiser les formes d'hypertexte existant dans les documents traditionnels (liste de références, glossaire, index d'auteurs) en les opérationnalisant techniquement comme des liens à double sens à tous les endroits où ils sont implémentés.

Ainsi, par exemple, une référence à une entrée de glossaire permettra toujours d'accéder à la liste de toutes les mentions de cette entrée, et cette liste permettra dans l'autre sens d'accéder à la localisation de chacune de ces mentions.

Peritext reprend plus ou moins ici certaines idées éprouvées dans de précédentes expérimentations en Humanités Numériques telles que "An Inquiry Into Modes Of Existence" [\(9\)](#) mais également d'expériences plus anciennes telles que le projet Xanadu qui envisageait un système hypertexte permettant de parcourir les liens dans les deux sens².

Pour voir une concrétisation de ce principe, rendez-vous dans les sections "glossaire" et "références" de ce site web, qui permettent d'accéder à la liste des mentions associées aux différentes ressources citées dans le texte.

Par ailleurs, pour information et comme démonstration d'une opérationnalisation possible de ce principe des liens à double sens, il est également possible de cliquer sur les citations présentes dans cette page pour voir apparaître sur la colonne de droite une liste des mentions des ressources associées ailleurs dans le document.

Adaptation et modularité

Peritext est un projet ouvert, pensé selon une logique modulaire permettant une multitude de variations et d'adaptations autour d'un même modèle de données interopérable, celui du *format peritext*.

Le projet présente divers niveaux d'ouverture pour son adaptation aux besoins de chacun, qui correspondent aussi à différents niveaux d'expertise.

Niveau d'adaptation 1 : mise en forme paramétrique et css

L'adaptation de la forme des documents peritext est d'abord permise par un ensemble d'options et de paramètres associés aux documents (exemple: position des notes) et des contextualiseurs utilisés dans un récit.

Il est par ailleurs possible d'écrire du code css spécifique pour chaque document, et de différencier les styles pour les supports de type “web” et pour les supports de type “codex”. Concernant les supports de type “codex”, les générateurs en place actuellement dans l'écosystème gèrent la norme css print (3) .

Ce premier niveau est pris en charge notamment par les éditeurs de contenus de Peritext, comme par exemple Ovide (voir la section afférente).

Niveau d'adaptation 2a : gabarits

Un gabarit (ou template) décrit selon quelle forme interpréter les données incluses dans un récit au format peritext.

À partir du modèle de document peritext, il est possible de développer de nouveaux gabarits de mise en forme (sous la forme de composants React) ou de développer des alternatives aux existants.

Voir la section [gabarits](#) pour en savoir plus.

Niveau d'adaptation 2b : contextualiseurs

Les générateurs et les éditeurs de l'écosystème peritext utilisent des modules contextualiseurs pour décider comment afficher une contextualisation donnée pour un type de support donné.

Chaque contextualiseur peut exposer jusqu'à 4 composants distincts suivant le support et la localisation de la contextualisation : comme paragraphe en mode web, dans un paragraphe en mode web, comme paragraphe en mode codex, dans un paragraphe en mode codex.

La logique modulaire des contextualiseurs permet d'implémenter des configurations spécifiques de Peritext en fonction des besoins de chacun (par exemple : plus orienté vers le partage de codes sources, plus orienté vers les médias vidéos et audio, plus orienté vers la visualisation des données, ...).

Il est possible d'adapter Peritext en développant de nouveaux contextualiseurs ou de développer des alternatives aux existants.

Rendez-vous à la section [contextualiseurs](#) pour en savoir plus.

Niveau d'adaptation 3 : générateurs

Peritext produit des documents (pdf, epub, site web) au moyen de modules intitulés [générateurs](#) qui consomment un récit peritext, des [contextualiseurs](#) et des [gabarits](#) pour produire un fichier utilisable et partageable. À ce niveau d'adaptation, il est possible de faire à peu près n'importe quoi à partir d'un document encodé au format peritext en écrivant un nouveau générateur.

Rendez-vous à la section [générateurs](#) pour en savoir plus.

Contextualiseurs

Selon la logique RCC, les générateurs de documents articulés autour de Peritext consomment des *contextualiseurs* qui peuvent prendre en charge des ressources courantes dans la publication académique (référence bibliographique, image, élément de glossaire) mais aussi une variété d'autres éléments interactifs (vidéo, production dicto, production quinoa, visualisation vega, ...).

Les modules contextualiseurs décrivent comment afficher une ressource dans un contexte particulier et selon des paramètres spécifiques. Ce sont des sortes de "mini-programmes" qui lisent une ressource et produisent une mise en forme spécifique en fonction des paramètres qui leur sont données.

Il est important de comprendre que les contextualiseurs ne sont pas forcément attachés à un type de ressource. Certains contextualiseurs sont associés à un seul type de ressource (exemple: vidéo), d'autres peuvent être appliqués à plusieurs types de ressources différents (exemple: référence bibliographique), ce qui fait une des forces de Peritext.

Techniquement, ces bibliothèques exposent un ou plusieurs composants react ainsi que des feuilles de style.

Liste d'idées pour des contributions portant de nouveaux contextualiseurs

- unity : un contextualiseur jouant un code source unity
- three.js : un contextualiseur jouant une géométrie 3D au moyen de la technologie WebGL
- pdf : un contextualiseur affichant un pdf à partir d'une url au moyen d'un module de lecture personnalisé

- facebook: rendre un post facebook selon une mise en forme de qualité ou personnalisée
- twitter : rendre un tweet ou une liste de tweets selon des paramètres personnalisés

Contextualiseur bib

Le contextualiseur "bib" (pour bibliographique) affiche une ressource sous la forme d'une référence bibliographique.

Celle-ci peut être une citation courte, c'est-à-dire dans la forme prise dans le corps des textes, comme ici (16) .

Elle peut aussi être longue, c'est-à-dire dans la forme prise normalement dans les bibliographies, comme suit :

1.
RICCI, Donato. *An Account of Digital Humanities from the AIME Project* [en ligne]. 2016. Revue échappées. Disponible à l'adresse : <http://echappees.esapyrenees.fr/content/2-numeros/3-numero3/5-an-account-of-digital-humanities-from-the-aime-project/echappees-an-account-of-digital-humanities-from-the-aime-project.pdf>

An Account of Digital Humanities from the AIME Project

Le formatage exact de la citation dépend du style et de la locale de citation définie pour le récit Peritext en entier (par exemple : ISO-61, APA, Chicago, ...). Le contextualiseur se met à jour en fonction des données de formatage qui lui sont fournies en amont, ce qui fait que toutes les citations d'un document sont toujours harmonisées et à jour.

Paramétrisation du contextualiseur bib

Il est possible de paramétrer le contextualiseur bib en indiquant pour une citation particulière un préfixe, un suffixe et/ou un localisateur comme un numéro de page. Voici un exemple (16, p. 2 version papier) .

Extensibilité du contextualiseur bib

Outre la contextualisation de relevés bibliographiques, le contextualiseur

bib peut être utilisé sur n'importe quel type de ressource grâce à un module qui convertit les métadonnées des ressources dans un format lisible par celui-ci¹.

Voici par exemple la contextualisation bib d'une ressource de type vidéo (17) ou une autre d'une ressource de type site web (2) .

Contextualiseur glossary

Le contextualiseur glossary permet de signaler des entrées de glossaire dans un récit peritext.

Voici un exemple de mention de Michel-Ange .

Il est possible d'utiliser des alias par exemple en citant la même ressource que dans le paragraphe précédent mais en l'appelant Michel-Angelo .

Rendez-vous dans la section "glossaire" de ce document.

Contextualiseur webpage

Le contextualiseur webpage affiche la page web associée à une ressource.

Il peut s'agir d'un hyperlien, d'une vignette du site ou encore d'un iframe (site imbriqué) du site.

Page embarquée



Site web hypotheses

Extensibilité du contextualiseur webpage

Outre les ressources webpage, le contextualiseur webpage peut être utilisé sur toutes les ressources qui indiquent une url dans leur métadonnées, ou les ressources bibliographiques qui indiquent une url.

Exemple d'une ressource "dicto" affichée avec le contextualiseur webpage :



Vectors - une analyse

Analyse par étiquetage des différents articles du journal Vectors

Exemple d'une ressource référence bibliographique affichée avec le contextualiseur webpage (le lien pointe vers fichier pdf, qui sera donc affiché avec les moyens du navigateur) :

An Account of Digital Humanities from the AIME Project

Exemple d'une ressource pointant vers une visualisation tableau :

Une visualisation hébergée sur tableau

Contextualiseur embed

Le contextualiseur "embed" prend en charge les ressources de type "embed", qui ne sont rien d'autre que des portions de code html directement intégrées dans les contenus¹.

Exemple d'un embed soundcloud :

Un exemple d'embed soundcloud

Un exemple d'embed de code svg, construit avec l'outil raw graphs :



Exemple d'embed de code svg tiré de raw graphs

Stanford professor of Computer Science, Marc Levoy, offers a free course on digital photography. <https://t.co/7H1BGQoWlX> pic.twitter.com/IOM2AefKqo

— manovich (@manovich) [30 août 2017](#)

Exemple d'embed basique d'un tweet

Exemple d'embed d'un post facebook :

Cette publication Facebook n'est plus disponible. Elle a peut-être été supprimée ou les paramètres de confidentialité de la publication ont peut-être été modifiés.

Exemple d'embed d'un post facebook

Pages d'aide 

Exemple d'embed tiré du service tableau :

Contextualiseur vidéo

Le contextualiseur vidéo joue une ressource sous la forme d'un player vidéo.

Les sources prises en charge pour l'instant sont youtube, vimeo, et une url pointant vers une vidéo html5.

En mode codex, le contextualiseur affiche une vignette pointant vers l'url de la vidéo.

En voici un exemple :



Objectified. (English)

Autor: Gary Hustwit Sinopsis: Reflexión sobre el papel de los objetos que nos rodean en nuestra vida cotidiana y las personas que los diseñan. Bibliografía: - Industrial Design A-Z <http://amzn.to/2pIIMJ0> - Design of the 20th Century <http://amzn.to/2qfTEz6> - Bauhaus <http://amzn.to/2pIf0Ju> - 1000 Chairs <http://amzn.to/2rbmnEI> - 1000 Lights <http://amzn.to/2qD96bS> - Hooked: How to Build Habit-Forming Products <http://amzn.to/2qD9KpO> - Value Proposition Design: How to Create Products and Services Customers Want <http://amzn.to/2r3LFYY> - Product Design and Development <http://amzn.to/2rbkXu0> - Everything Industrial Designers Need to Know Every Day <http://amzn.to/2qfy7Ha> - Material Innovation: Product Design <http://amzn.to/2pIzX27> - Making It: Manufacturing Techniques for Product Design <http://amzn.to/2qCVCNb> - Drawing for Product Designers <http://amzn.to/2rblcoU> - The Industrial Design Reader <http://amzn.to/2pIFrK3>

Source: Neojaponismo (youtube: <https://www.youtube.com/watch?v=TyofGn8fiUU>)

Contextualiseur dicto

À propos de Dicto

Dicto est un format de données, un éditeur et un composant de player développés par Robin de Mourat & Donato Ricci dans le cadre d'analyses de transcriptions¹.

Utiliser le contextualiseur dicto



Ressource dicto représentée avec le contextualiseur dicto

Analyse par étiquetage des différents articles du journal Vectors

Non-exclusivité du contextualiseur dicto

Une ressource de type dicto est naturellement prise en charge par le contextualiseur dicto. Cela dit, elle peut par ailleurs aussi être prise en charge par le contextualiseur "video" :



Ressource dicto représentée en vidéo simple

Analyse par étiquetage des différents articles du journal Vectors

Contextualiseur table

Le Contextualiseur de type table affiche une ressource de type table (exemple: tirée d'un fichier csv) sous la forme d'un tableau traditionnel.

En version codex, il produit un tableau simple.

En version web, il produit un tableau interactif (pagination, recherche, ...)

En voici un exemple :

a	b
A	2
B	4
C	6
D	8
E	2
F	3
G	5

Dataset très simple

Contextualiseur vegalite

vega (et son petit frère allégé vega lite) est une grammaire de description de visualisations de données interactives et statiques. Il est développé par l'équipe Interactive Data Lab au sein de l'Université de Washington [\(18\)](#) .

Le contextualiseur vegalite permet de contextualiser une ressource de type table sous la forme d'une visualisation de données paramétrée selon la grammaire vegalite.

Utiliser le contextualiseur vega

Soit la ressource suivante :

a	b
A	2
B	4
C	6
D	8
E	2
F	3
G	5

Dataset très simple

Elle est par défaut affichée via le contextualiseur table. La voici désormais affichée avec le contextualiseur vega :

Dataset très simple

Par défaut, le contextualiseur cherche un état minimal pour afficher la table sous la forme d'une visualisation, pas très intéressant donc.

Ajoutons-lui maintenant le paramètre de spécification suivant :

```
{
  "description": "A simple bar chart with embedded data.",
  "mark": "bar",
  "encoding": {
    "x": {"field": "a", "type": "ordinal"},
    "y": {"field": "b", "type": "quantitative"}
  }
}
```

On obtient :

Dataset très simple

Apprendre à utiliser la grammaire vegalite

Les possibilités offertes par vega sont très importantes et couvrent une grande partie des besoins courants en termes de visualisation de données. La documentation proposée sur le site de vega permet de s'en faire une première idée :

Vega lite

Pour prévisualiser et tester une spécification vegalite avant de l'intégrer dans un contextualiseur, rendez-vous sur l'outil de développement proposé par ses créateurs :

Outil et de prévisualisation d'édition de la grammaire vega

Contextualiseur data-presentation

Les présentations de données

Le concept de présentation de données désigne des documents qui consistent à proposer aux lecteurs une visite guidée dans une visualisation. Le lecteur parcourt des slides, qui sont associées à une vue particulière dans la visualisation associée.

Le format des présentations de données est actuellement développé par le médialab de sciences po [\(6\)](#) dans le cadre du programme FORCCAST [\(8\)](#) .

Un premier exemple de présentation de données peut être expérimenté via l'outil ManyLines [\(22\)](#).

Un outil plus récent permettant de construire les présentations qui suivent est appelé à être rendu public courant 2018.

Utiliser le contextualiseur data-presentation

Un exemple de présentation de réseau :

Mon beau réseau roi des forêts

.

Slide 1

Conformément à ce que nous avons décrit ci-dessus comme la première tâche d'une

histoire de l'annotation, nous commencerons par un repérage des formes sous lesquelles se présente la note à l'époque de l'Humanisme. Afin de ne pas préjuger du résultat de l'enquête, nous ne ferons pas *a priori* de séparation entre notes marginales manuscrites et notes marginales imprimées en écartant de notre champ d'investigation l'une ou l'autre de ces espèces. Nous les examinerons donc successivement pour voir s'il y a finalement lieu ou non d'établir une distinction ; après quoi nous examinerons une troisième forme matérielle de production de la note, celle de sa publication en recueils, phénomène intellectuel et éditorial très important aux XVe et XVIe siècles.

Slide 2

Conformément à ce que nous avons décrit ci-dessus comme la première tâche d'une histoire de l'annotation, nous commencerons par un repérage des formes sous lesquelles se présente la note à l'époque de l'Humanisme. Afin de ne pas préjuger du résultat de l'enquête, nous ne ferons pas *a priori* de séparation entre notes marginales manuscrites et notes marginales imprimées en écartant de notre champ d'investigation l'une ou l'autre de ces espèces. Nous les examinerons donc successivement pour voir s'il y a finalement lieu ou non d'établir une distinction ; après quoi nous examinerons une troisième forme matérielle de production de la note, celle de sa publication en recueils, phénomène intellectuel et éditorial très important aux XVe et XVIe siècles.

Slide 3

Conformément à ce que nous avons décrit ci-dessus comme la première tâche d'une histoire de l'annotation, nous commencerons par un repérage des formes sous lesquelles se présente la note à l'époque de l'Humanisme. Afin de ne pas préjuger du résultat de l'enquête, nous ne ferons pas *a priori* de séparation entre notes marginales manuscrites et notes marginales imprimées en écartant de notre champ d'investigation l'une ou l'autre de ces espèces. Nous les examinerons donc successivement pour voir s'il y a finalement lieu ou non d'établir une distinction ; après quoi nous examinerons une troisième forme matérielle de production de la note, celle de sa publication en recueils, phénomène intellectuel et éditorial très important aux XVe et XVIe siècles.

Mon beau réseau roi des forêts

Un exemple de présentation de frise chronologique :

Histoire de la visualisation

Robin.

Histoire de la visualisation

Cette présentation traite de l'histoire de la visualisation

Un exemple de présentation de carte :

Histoire du Fort et des combats de 1870-1871

Robin.

Histoire du Fort et des combats de 1870-1871

Une histoire du fort d'Issy

Une exemple de présentation fondée sur un fichier svg :

Exemple de présentation SVG

Robin.

Exemple de présentation SVG

D'autres formats et types de présentation de données sont appelés à apparaître au fil du développement de ce projet voisin.

Contextualiseur codefiles

Le contextualiseur fichiers de code affiche une ressource de type "fichiers de code" (codefiles) sous la forme d'une interface interactive ou statique.

Voici un exemple de contextualisation codefiles :

second.css

```
.ovide-InlineCitation
{
  margin: 0;
  padding: $gutter-small 0 $gutter-small 0;

  cursor: pointer;
  -webkit-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
  user-select: none;
}
```

my_file.js

```
import React from 'react';

export default ({
  resource: {
    data: {
      code={}
    }
  },
  // contextualizer,
  // contextualization
}) => {
  return (
    <div className="peritext-contextualizer-static peritext-contextualizer-code">
      {
        Object.keys(code)
          .map(fileName => {
            return <div key={fileName} className="file-display">
              {Object.keys(code).length > 1 && <h2>{fileName}</h2>}
              <pre>
```

```
        <code>
            {code[fileName]}
        </code>
    </pre>
</div>
    ))
}
</div>
)
```

Exemple de ressource codefiles

Contextualiseur p5 (processing)

Le contextualiseur p5.js interprète une ressource de type "fichiers de code" pour en faire une application p5.js.

Prenons la ressource de type "fichiers de code" suivante :

```
function sketch (p) {  
  var rotation = 0;  
  
  p.setup = function () {  
    p.createCanvas(200, 200, p.WEBGL);  
  };  
  
  p.draw = function () {  
    rotation += 0.01;  
    p.background(100);  
    p.noStroke();  
    p.push();  
  
    p.rotateY(rotation);  
    p.box(100);  
    p.pop();  
  };  
};
```

Ressource processing js

Choisissons maintenant le contextualiseur p5 pour l'afficher :

Ressource processing js

On obtient un "canvas" dans lequel est représenté un programme processing interprété à partir du code de la ressource.

Pour en savoir plus sur les capacités de p5.js, rendez-vous sur le site

officiel du projet(27) .

Gabarits

Les **gabarits** définissent la manière selon laquelle doivent être affichés les contenus d'un récit. Ils concernent à la fois le code html/css de chaque portion de contenu, et l'organisation même des pages/sections pour les documents codex ou des routes pour les documents web.

Par exemple, le présent document est mis en forme au moyen des gabarits *peritext-template-dynamic-garlic* et *peritext-template-static-garlic*.

Gabarits existants

Dynamic Garlic

Il s'agit d'un **gabarit** de présentation web très simple, fortement calqué sur la structure rhétorique d'un livre académique traditionnel. En ce qui concerne les contenus, il présente une colonne principale déroulant les contenus de manière linéaire, et une colonne optionnelle permettant de naviguer dans les différentes contextualisations d'une même ressource.

Il présente également un ensemble de "vues de coupe" dans le document calquées sur les formats imprimés traditionnels, comme une bibliographie/liste de ressources et un glossaire interactifs permettant de prévisualiser et de naviguer dans les contextualisations d'une même ressource.

Static Garlic

Il s'agit d'un **gabarit** pour générateurs codex très simple plutôt pensé pour les pdf interactifs et les epub, utilisant les polices Roboto et Merryweather et présentant une table des matières interactives, un glossaire, un index des auteurs, et des références interactives.

Générateurs

Les générateurs produisent un fichier pour un support donné à partir d'un récit peritext et d'un ensemble d'options¹.

Ils se divisent en deux grandes familles :

- les générateurs de type codex sont dédiés à tous les formats de sortie structurés autour de pages à tourner. Techniquement, ces formats de sortie n'intègrent pas de code javascript
- les générateurs de type web qui produisent d'une manière ou d'une autre un site web ou application assimilée (application mobile, application bureau, ...)

Générateur pdf (codex)

Produit un fichier pdf interactif et imprimable, avec le moteur PrinceXML⁽²⁾ qui permet de gérer toutes les règles CSS print².

Générateur epub (codex)

Produit un fichier epub pour liseuses, prenant en charge les hyperliens internes et acceptant du css personnalisé.

Générateur next (web)

Produit un site web statique (série de fichiers html) au moyen du framework next⁽⁴⁾ . Le framework next.js a la particularité de générer des applications *isomorphiques* : le contenu de chaque page est écrit dans le code html (pour l'usage des machines et l'indexation), mais le comportement et l'affichage est géré via javascript pour un maximum d'interactivité et de subtilité dans la présentation.

Liste d'idées pour des contributions au niveau des générateurs

- Générateur de fichiers markdown (devra d'une manière ou d'une autre représenter les contextualisations)
- Générateur de fichiers LaTeX
- Générateur d'un fichier .doc
- Générateur d'un fichier xml/idml (compatible avec InDesign)
- Générateur de site monopage adapté aux récits de type article
- Générateur de document audio via une API de text-to-speech
- Générateur de document "spatialisé" avec Unity ou mobilizing.js
- ...

Éditeur Ovide

Ovide est le premier éditeur développé pour l'écosystème Peritext. Il est dérivé de manière extensive de l'éditeur Quinoa, en développement au sein du médialab sciences [po\(6\)](#) et lui aussi développé en partie autour du format peritext.

Il a vocation à faire office de "preuve de concept" plutôt que d'application en soi, étant donné que sa configuration technique est davantage tournée vers l'expérimentation que vers une utilisation à grande échelle comme un outil "sur étagère".

Les données de l'application sont stockées sur le navigateur de l'utilisateur, et il peut les télécharger à tout moment.

L'éditeur propose une interface d'édition graphique, mais prend aussi en charge la syntaxe markdown [\(5\)](#) comme raccourci.

L'éditeur permet d'écrire un récit peritext et de l'exporter aux formats epub, pdf, et html, via des générateurs installés sur un serveur avec lequel il doit être connecté.

Rendez-vous sur la version alpha de l'application [\(7\)](#) pour la découvrir :

Éditeur Ovide

Contribuer à peritext

L'intégralité du code de peritext est placée sous licences libres LGPL & CECCIL-B. Il peut donc être modifié et utilisé hors exploitations commerciales.

Pour l'instant, les modèles de données associés aux ressources, contextualiseurs, gabarits de peritext ne sont pas stabilisés, car une utilisation prolongée est nécessaire pour les gérer et les fixer de manière intelligente¹. Il en est de même pour l'API des modules, qui évolue en fonction de l'expérience tirée des premières implémentations du projet.

Ainsi, les possibilités de contribution à peritext sont amenées à s'élargir petit à petit, une fois le projet stabilisé et apte à fonctionner comme une plateforme pour de nombreuses initiatives.

Créer/hacker un contextualiseur

... doc à écrire

Créer/hacker un gabarit/template

... doc à écrire

Créer/hacker un générateur

... doc à écrire

À propos

À propos

Peritext est un projet libre développé dans un contexte de recherche en design et esthétique. Il a bien sûr également beaucoup à voir avec les sciences de l'information et de la communication, les sciences de la documentation et l'informatique.

Histoire du projet / remerciements

- 2013-2016 : Robin de Mourat obtient un contrat doctoral de 3 ans de la part du Ministère de l'Enseignement Supérieur et de la Recherche via l'ENS Cachan pour effectuer un doctorat auprès de Nicolas Thély à l'université Rennes 2, dans le cadre du groupe de recherche MONADE [\(23\)](#) de la chaire Humanités Numériques nouvellement créée. Une grande partie du temps consacré au montage de Peritext est redevable de ce contrat et de cet environnement.
- Janvier 2014 – Mars 2015: phase d'observation participante du projet Enquête sur les Modes d'Existence [\(9\)](#) , projet philosophique soutenu par un écosystème d'instances imprimées et web – qui débouchera sur des entretiens commentés et une étude de cas [\(10\)](#). Une grande partie des concepts centraux de Peritext découle de cette rencontre et des nombreux échanges qu'elle a occasionné.
- 2015 : rédaction par Donato Ricci & Robin de Mourat d'une premier retour en termes de design à propos du projet EME, dont émerge la centralité du concept de *contexte* dans les enjeux actuels de la publication académique [\(15\)](#)
- Juin 2015 : workshop "Open AIME", dédié à la confrontation de l'écosystème AIME (de ses composantes techniques et principes de design) aux cas d'étude d'autres chercheurs [\(13\)](#)

- Janvier 2016 : Publication collective de l'article "Clues, Anomalies, Understanding"⁽¹⁴⁾ qui propose un site web utilisant un prototype du futur peritext fondé sur une syntaxe markdown+json, alors intitulé modulo ⁽¹¹⁾
- Été 2016 : Première spécification de Peritext & implémentation de Peritext ⁽¹²⁾
- Septembre 2016 : présentation du projet Peritext à la Maison des Sciences de l'Homme de Lille
- 2016-2017 : développement de Peritext dans le cadre du projet Quinoa pris en charge par le médialab Sciences Po ⁽⁶⁾ pour le programme FORCCAST ⁽⁸⁾
- Mai-Septembre 2017 : exposition du projet au festival international de design graphique de Chaumont lors de l'exposition "une cartographie de la recherche en design graphique" ⁽²⁴⁾
- Juin-Août 2017 : refactorisation de tout le projet comme un écosystème de modules, développement de l'éditeur Ovide, préparation progressive du projet à une ouverture collective

Références

1.

DE MOURAT, Robin. Vectors - une analyse [en ligne]. Disponible à l'adresse : <http://dicto-formats.herokuapp.com/share/transcription/vectors-une-analyse?display=fullscreen&showtags=1&showrailway=1&leftcolumn=50>
Analyse par étiquetage des différents articles du journal Vectors

Mentions: [p. 35](#), [p. 36](#), [p. 28](#)

2.

Site web hypotheses. [en ligne]. Disponible à l'adresse : <http://hypotheses.org/>

Mentions: [p. 27](#)

3.

DE MOURAT, Robin et RICCI, Donato. Dicto - site web. [en ligne]. 2017. Disponible à l'adresse : <http://dicto-playground.herokuapp.com/>
Site web de démonstration de l'outil Dicto

Mentions: [p. 68](#)

4.

Objectified. (English). [en ligne]. Disponible à l'adresse : <https://www.youtube.com/watch?v=TyofGn8fiUU>
Autor: Gary Hustwit

Mentions: [p. 33](#), [p. 25](#)

5.

Dataset très simple.

Mentions: [p. 37](#), [p. 38](#), [p. 39](#), [p. 39](#), [p. 13](#), [p. 14](#), [p. 14](#)

6.

Prince XML. [en ligne]. Disponible à l'adresse : <https://www.princexml.com/>

Mentions: p. 50, p. 25

7.
Spécification W3C pour le css print. [en ligne]. Disponible à l'adresse :
<https://www.w3.org/TR/css-print/>

Mentions: p. 68, p. 20

8.
Framework javascript next.js. [en ligne]. Disponible à l'adresse :
<https://zeit.co/blog/next3>

Mentions: p. 50

9.
Markdown – wikipedia. [en ligne]. Disponible à l'adresse :
<https://fr.wikipedia.org/wiki/Markdown>

Mentions: p. 52

10.
Médialab Sciences Po. [en ligne]. Disponible à l'adresse :
<http://www.medialab.sciences-po.fr/fr/>

Mentions: p. 52, p. 55, p. 41

11.
DE MOURAT, Robin. Éditeur Ovide. [en ligne]. Disponible à l'adresse :
<http://ovide.surge.sh/>

Mentions: p. 52, p. 52

12.
Programme FORCCAST. [en ligne]. Disponible à l'adresse :
<http://controverses.org/>

Mentions: p. 55, p. 41

13.
An Inquiry Into Modes Of Existence. [en ligne]. Disponible à l'adresse :
<http://modesofexistence.org/>

Mentions: p. 54, p. 18

14.

AIME, expérimentation de quelques instruments d'écoute. [en ligne].
Disponible à l'adresse : <http://modesofexistence.org/-recherche-en-design-re-penser-les-normes-de-la-production-scientifique/>

Mentions: p. 54

15.

Un exemple d'embed soundcloud.

Mentions: p. 30

16.

Exemple d'embed de code svg tiré de raw graphs.

Mentions: p. 31

17.

Exemple d'embed basique d'un tweet.

Mentions: p. 31

18.

Exemple d'embed d'un post facebook.

Mentions: p. 32

19.

Clues, Anomalies, Understanding – digital companion. [en ligne]. 2015.
Disponible à l'adresse : <http://modesofexistence.org/anomalies/>

Mentions: p. 55

20.

Première implémentation de Peritext. [en ligne]. Disponible à l'adresse :
<https://github.com/peritext/peritext>

Mentions: p. 55

21.

Compte-rendu du workshop « open aime ». [en ligne]. 2015. Disponible à l'adresse : <http://modesofexistence.org/open-aime-workshop-11-12-june-paris/>

Mentions: [p. 54](#)

22.

RICCI , Donato, DE MOURAT , Robin, LECLERQ, Christophe et LATOUR, Bruno. *Clues, Anomalies, Understanding. Detecting underlying assumptions and expected practices in the Digital Humanities through the AIME project.* [en ligne]. 2015. Visible Language. Disponible à l'adresse : <http://visiblelanguagejournal.com/issue/172/article/1172>

Mentions: [p. 55](#)

23.

RICCI , Donato, DE MOURAT, Robin et BOULANGER , Pierre-Laurent. *AIME: opening the context of a Humanities inquiry.* [en ligne]. 2015. Disponible à l'adresse : <http://echappees.esapyrenees.fr/content/2-numeros/3-numero3/5-an-account-of-digital-humanities-from-the-aime-project/echappees-an-account-of-digital-humanities-from-the-aime-project.pdf>

Mentions: [p. 54](#)

24.

RICCI, Donato. *An Account of Digital Humanities from the AIME Project* [en ligne]. 2016. Revue échappées. Disponible à l'adresse : <http://echappees.esapyrenees.fr/content/2-numeros/3-numero3/5-an-account-of-digital-humanities-from-the-aime-project/echappees-an-account-of-digital-humanities-from-the-aime-project.pdf>

Mentions: [p. 28](#), [p. 24](#), [p. 24](#), [p. 24](#)

25.

Outil et de prévisualisation d'édition de la grammaire vega. [en ligne]. Disponible à l'adresse : <https://vega.github.io/editor/#/edited>

Mentions: [p. 40](#)

26.

Vega lite. [en ligne]. Disponible à l'adresse : <https://vega.github.io/vega-lite/docs/>

Mentions: [p. 38](#), [p. 39](#)

27.

Exemple de ressource codefiles.

Mentions: [p. 45](#)

28.

Mon beau réseau roi des forêts.

Mentions: [p. 41](#)

29.

Scalar. [en ligne]. Disponible à l'adresse : <http://scalar.usc.edu/>

Mentions: [p. 67](#)

30.

Ted Nelson demonstrates Xanadu Space. [en ligne]. Disponible à l'adresse : https://www.youtube.com/watch?v=En_2T7KH6RA
Ted Nelson présente le logiciel Xanadu Space qui spatialise en trois dimensions un système hypertexte.

Mentions: [p. 67](#)

31.

La norme OpenURL et la technologie Context Objects in Span. [en ligne]. Disponible à l'adresse : <http://blog.stephanepouyllau.org/116>

Mentions: [p. 18](#)

32.

CitationStyles.org. [en ligne]. Disponible à l'adresse : <http://citationstyles.org/>

Mentions: [p. 17](#)

33.

Ressource *processing js*.

Mentions: p. 47, p. 47

34.

ManyLines. [en ligne]. Disponible à l'adresse : <http://tools.medialab.sciences-po.fr/manylines>

Mentions: p. 41

35.

DE MOURAT, Robin. *Exemple de présentation SVG*.

Mentions: p. 43

36.

DE MOURAT, Robin. *Histoire du Fort et des combats de 1870-1871*.
Une histoire du fort d'Issy

Mentions: p. 43

37.

DE MOURAT, Robin. *Histoire de la visualisation*.
Cette présentation traite de l'histoire de la visualisation

Mentions: p. 43

38.

Le modèle de document Peritext.

Mentions: p. 12

39.

Projet MONADE. [en ligne]. Disponible à l'adresse : https://www.mshb.fr/projets_mshb/monade/2289/

Mentions: p. 54

40.

LE SIGNE. *Une cartographie de la recherche en design graphique - Biennale de design graphique 2017*. [en ligne]. Disponible à l'adresse : <http://www.centrenationaldugraphisme.fr/fr/cig/page/biennale-de->

design-graphique-2017/expositions/une-cartographie-de-la-recherche-en-design-graphique

Mentions: p. 55

41.

Pandoc. [en ligne]. Disponible à l'adresse : <https://pandoc.org/>

Mentions: p. 66

42.

Une visualisation hébergée sur tableau. [en ligne]. Disponible à l'adresse : https://public.tableau.com/views/MM34-5MillenniaofSolarEclipses/SolarEclipse?:embed=y&:toolbar=no&:loadOrderID=0&:display__count=yes

Mentions: p. 28

43.

Site officiel de p5.js. [en ligne]. Disponible à l'adresse : <https://p5js.org/>

Mentions: p. 48

44.

Conventions de conversion de données pour le projet draft-js. [en ligne]. Disponible à l'adresse : <https://draftjs.org/docs/api-reference-data-conversion.html>

Mentions: p. 11, p. 66

45.

BUSH, Vannevar et OTHERS. As we may think. *The atlantic monthly*. 1945. Vol. 176, n° 1, pp. 101-108.

Mentions: p. 18

46.

Le cabinet d'Ole Worm. 1655.

Reproduction de la gravure conservée à Modène représentant un cabinet de curiosités

Mentions: p. 4

47.

DE MOURAT, Robin. Répertoire peritext-core sur github. [en ligne]. Disponible à l'adresse : <https://github.com/peritext/peritext-core>

Mentions: p. 10

48.

Peritext core – documentation technique. [en ligne]. Disponible à l'adresse : <https://peritext.github.io/peritext-core/>

Mentions: p. 10

49.

INTERNET ENGINEERING TASK FORCE. Spécification JSON Schema. [en ligne]. Disponible à l'adresse : <http://json-schema.org/>

Mentions: p. 67

Glossaire

contextualiseur

Mentions: [p. 21](#), [p. 21](#), [p. 21](#)

CSS

Mentions: [p. 20](#), [p. 20](#)

gabarit

Mentions: [p. 21](#), [p. 20](#), [p. 21](#), [p. 49](#), [p. 49](#), [p. 49](#)

générateur

Mentions: [p. 21](#), [p. 21](#), [p. 21](#)

Michel-Ange

Mentions: [p. 26](#), [p. 26](#)

Index des auteurs

Pierre-Laurent Boulanger

Mentions: [p. 54](#)

Vannevar Bush

Mentions: [p. 18](#)

Robin De Mourat

Mentions: [p. 35](#), [p. 36](#), [p. 68](#), [p. 52](#), [p. 52](#), [p. 55](#), [p. 54](#), [p. 28](#), [p. 43](#), [p. 43](#), [p. 43](#), [p. 10](#)

Internet Engineering Task Force

Mentions: [p. 67](#)

Bruno Latour

Mentions: [p. 55](#)

Le Signe

Mentions: [p. 55](#)

Christophe Leclercq

Mentions: [p. 55](#)

Donato Ricci

Mentions: [p. 68](#), [p. 55](#), [p. 54](#), [p. 28](#), [p. 24](#), [p. 24](#), [p. 24](#)

Notes

- ¹ Tous les modules sont écrits en Javascript et utilisent notamment sur la bibliothèque de rendu open source "react", qui permet de produire indifféremment du code pour des applications client et des applications serveurs, et pour des environnements interactifs ou et des environnements statiques (type générateur de pdf)
- ² "Écrire une seule fois, publier partout.". Voir par exemple la technologie Pandoc [\(25\)](#) dans ce registre.
- ³ On parle aussi parfois de "publication multimodale".
- ⁴ "Ce que vous voyez est ce que vous voulez dire". Voir des formats comme
- ⁵ "Ce que vous voyez est ce que vous obtenez" (à l'impression). Voir des formats comme word dans cette catégorie.
- ⁶ Ce type d'annotation s'adosse à la spécification open source draft-raw développé par l'équipe de développement de l'entreprise Facebook, qui sert à décrire au format json des contenus textuels "riches" pour des applications utilisant la technologie de rendu html intitulée react, qui est utilisée dans tous les modules de l'écosystème peritext. Voir le site du projet draft-js pour plus d'informations [\(28\)](#) .
- ⁷ On propose le terme "What you see is what you argue" ("ce que vous voyez est ce que vous argumentez") pour décrire cette troisième voie qui par se référence à la rhétorique ne sépare pas la question de la forme de la question du sens.
- ⁸ C'est pourquoi tous les modules du projets sont pour l'instant à l'état de version "pre-alpha", c'est-à-dire qu'ils sont instables & non

utilisables pour des situations réelles car appelés à évoluer de manière non-compatible avec leur version actuelle.

- 9 Peritext-core est notamment dédié à décrire le format peritext au moyen de la spécification json-schema développée par l'Internet Engineering Task Force (32)
- 10 Voir le projet Scalar (19) , qui expérimente des formes alternatives de structuration des arguments académiques en tirant parti des technologies de base de données relationnelles.

11



Ted Nelson demonstrates Xanadu Space

Ted Nelson présente le logiciel Xanadu Space qui spatialise en trois dimensions un

système hypertexte.

Source: *photonhunter* (youtube: https://www.youtube.com/watch?v=En_2T7KH6RA)

- 12 Il s'agit du format csl-json.
- 13 Note: le code javascript de balises script injectées selon cette méthode ne sont pas prises en compte.
- 14 Voir une démonstration d'un format de sortie possible pour dicto (1) .
- 15 Note: prince xml est la seule technologie non libre utilisée dans un module peritext. Le choix de Prince XML est motivé par sa prise en charge hors pairs de la spécification w3c pour le css print(3) . Les autres technologies de conversion pdf en javascript, basées sur webkit ou phantom, ne prennent pas (ou très mal en charge) ces spécifications.
- 16 *Techniquement, ces bibliothèques exposent une fonction qui consomme un récit au format json et des modules de contextualisation et de gabarit, pour produire en sortie un fichier utilisable (e.g. ".pdf").*
- 17 Il faudra notamment à terme :
- écrire un schéma de description des métadonnées des contextualiseurs
 - décider où stocker les schémas des types de ressources et leur modèle (dans les contextualiseurs par défaut ? dans des modules séparés ?)
 - trouver la meilleure manière de gérer la définition des "plans de sites" en ce qui concerne les couples gabarit web / générateur de sites web

intégrer dans le modèle la possibilité de spécifier des liens
internes au récit (vers une section / vers une contextualisation)

Peritext - publication multimodale orientée contexte

Une introduction

Robin De Mourat

En amont comme en aval de l'acte de publication, les textes produits et échangés par les chercheurs universitaires convoquent de manière de plus en plus intime les environnements numériques (au premier rang desquels le web) dans lesquels ils sont inscrits.

Il est donc nécessaire d'inventer de nouvelles manières de connecter les modes de production des documents académiques avec les contextes du web. Cette recontextualisation doit opérer dans les deux sens : les environnements numériques et toutes les ressources et formes qu'ils proposent doivent être plus significativement convocables dans les documents académiques, et les documents académiques doivent être plus significativement contextualisables dans les environnements numériques. C'est là le projet à la base de Peritext.