Édition collaborative *local-first* de documents LaTeX

Thématiques	Interaction humain-machine, génie logiciel, systèmes distribués
Niveau	Master 2 ou dernière année d'école d'ingénieur
Durée	6 mois
Gratification	Environ 600 euros/mois (4,35 euros/heure)
Date de début	Début 2025
Lieu du stage	Équipe ex)situ du laboratoire LISN de l'Université Paris-Saclay
Encadrants	Camille Gobert et Michel Beaudouin-Lafon
Poursuite en thèse	Possibilité de candidater pour une bourse (selon candidat)

Ce stage de M2 s'inscrit dans le projet OnePub, qui propose de repenser l'interaction et la collaboration dans les chaînes éditoriales dans le but de transformer la manière dont on conçoit des ouvrages publiés dans de multiples formats (papier, web, ePub…) à partir d'une source unique.

Contexte

Dans le cadre du projet OnePub, nous développons un prototype logiciel permettant de concevoir des documents, tels que des articles scientifiques ou des manuels scolaires, qui remet en question l'organisation des chaînes éditoriales traditionnelles. En effet, l'édition et la production d'ouvrage nécessite de nombreux intervenants, des outils propriétaires, différents formats de fichiers, et de nombreux allers-retours et re-saisie d'informations entre les intervenants. L'idée du projet OnePub est de travailler sur une source de données unique, synchronisée entre tous les utilisateurs intervenant sur l'ouvrage (Figure 1), à travers des outils d'édition adaptés à chaque type d'utilisateur. Notre prototype permet à plusieurs utilisateurs de travailler simultanément (à la façon d'un outil comme *Google Docs*), y compris lorsque l'on n'est pas connecté à internet, en phase avec le mouvement *local-first software* (Kleppmann et al., 2019). Il permet en outre d'exporter l'ouvrage dans de multiples formats, de manière à s'adapter aux contraintes et à tirer parti des opportunités offertes par chaque format (paginé ou non, interactif ou non…), avec une considération particulière pour la génération d'une version accessible du document.

Notre implémentation actuelle est principalement écrite en TypeScript et repose sur les technologies du web. Le contenu du document est rédigé à l'aide du langage AsciiDoc, qui ressemble à Markdown, tandis que le style est décrit à l'aide du langage CSS. À partir de ces sources nous générons différentes versions d'un même ouvrage, telles qu'une version HTML pour le web et une version PDF pour l'impression. Nous encodons ces données à l'aide de structures de données propices à la collaboration synchrone et à la gestion de versions appelées *Conflict-free Replicated Data Types* (CRDTs), qui permettent aussi bien de synchroniser chaque modification en temps réel entre tous les utilisateurs connectés que de travailler hors-ligne et ne partager ses modifications qu'à sa prochaine reconnexion. À l'inverse des données entrées par les utilisateurs, qui sont synchronisées via les CRDTs, les fichiers compilés, tels que les sorties HTML et PDF du document, sont recalculés localement. Le logiciel maintient pour cela un graphe de dépendances entre fichiers, permettant d'automatiser la mise à jour des fichiers affectés par la modification d'un autre fichier.

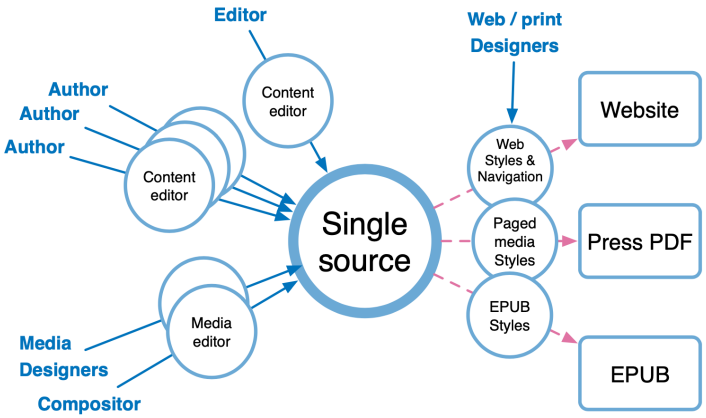

Figure 1 – Schéma de la collaboration entre différents acteurs envisagée dans le projet OnePub.

Objectifs du stage

L'objectif de ce stage est d'adapter le prototype afin de le rendre compatible avec le langage et l'écosystème LaTeX, qui sont massivement utilisés dans l'édition technique et académique (Knauff et Nejasmic, 2014). Outre le fait de démontrer la versatilité de l'architecture logicielle de OnePub, cette adaptation offrirait, à terme, une alternative libre et *local-first* à d'autres environnements d'édition collaborative de documents tels que Overleaf et Typst. Les solutions envisagées pourront s'inspirer de travaux antérieurs réalisés dans l'équipe ex)situ, mais nous sommes ouverts aux alternatives !

Dans un premier temps, l'objectif consiste en l'ajout de primitives permettant de compiler un ensemble de fichiers (code LaTeX, images, etc.) qui utilisent le graphe de dépendance entre fichiers afin de pouvoir automatiser la génération du PDF produit par un compilateur LaTeX lorsqu'une des dépendances est modifiée. Ce travail pourra s'inspirer de l'état de l'art sur la compilation incrémentale, de prototypes de recherche tels que FileWeaver (Gori et al., 2020), ainsi que de travaux en cours cherchant à atteindre des objectifs similaire, tels que Jacquard de Ink & Switch.

Dans un second temps, l'objectif consiste en l'amélioration de l'interface d'édition (Figure 2) afin de la doter d'outils spécifiques à l'interaction avec un document produit à l'aide de LaTeX. Cette étape est à même d'être adaptée en fonction du profil et des intérêts du candidat. Une possibilité serait d'adapter l'approche proposée par *i*-LaTeX (Gobert et Beaudouin-Lafon, 2022), qui permet à l'utilisateur de visualiser et de modifier le code LaTeX du document à l'aide de widgets affichés par dessus le PDF généré.

Compétences recherchées

- Une bonne connaissance de JavaScript ou TypeScript, et plus généralement des technologies de développement web *front-end* (HTML, CSS, composants web, Rollup/Vite, etc.)
- Une connaissance du langage LaTeX et de son outillage (Latexmk, SyncTeX, etc.) est un plus.
- Une connaissance des systèmes Unix, et notamment des systèmes de fichiers et de processus.
- Une bonne culture en informatique, avec un intérêt prononcé pour l'interaction humain-machine, le développement logiciel, les systèmes distribués et/ou les outils d'édition numérique.
- Une aisance avec l'anglais parlé et écrit.

Figure 2 – Interface graphique du prototype montrant un exemple d'ouvrage conçu avec OnePub : un résumé de thèse adapté depuis une version PDF (extraite de la liste des thèses soutenues à l'ONERA en 2023, pp. 50–51). Sont illustrés, de gauche à droite : (1) la liste des fichiers faisant partie des sources partagées de l'ouvrage ; (2) le contenu du document au format AsciiDoc, representé sous forme de texte brut ; et (3) le rendu du document au format HTML, paginé à l'aide de la bibliothèque Paged.js.

Bibliographie

Gobert et Beaudouin-Lafon, 2022 DOI : 10.1145/3491102.3517494

Camille Gobert et Michel Beaudouin-Lafon. 2022. i-LaTeX : Manipulating Transitional Representations between LaTeX Code and Generated Documents. *CHI Conference on Human Factors in Computing Systems*. CHI '22. ACM, p. 1-16.

Gori et al., 2020 DOI : 10.1145/3379337.3415830

Julien Gori, Han L. Han et Michel Beaudouin-Lafon. 2020. FileWeaver : Flexible File Management with Automatic Dependency Tracking. *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. UIST '20. ACM, p. 22-34.

Kleppmann et al., 2019 DOI : 10.1145/3359591.3359737

Martin Kleppmann, Adam Wiggins, Peter van Hardenberg et Mark McGranaghan. 2019. Local-First Software : You Own Your Data, in Spite of the Cloud. *Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*. Onward ! 2019. ACM, p. 154-178.

Knauff et Nejasmic, 2014 DOI : 10.1371/journal.pone.0115069

Markus Knauff et Jelica Nejasmic. 2014. An Efficiency Comparison of Document Preparation Systems Used in Academic Research and Development. *PLOS ONE* 9.12, p. 1-12.