

Flex Picture eBook Builder - Simplifier la création de livres électroniques accessibles

Danya Gharbieh¹, Maximilian Punz¹, Klaus Miesenberger¹, et Valentin Salinas-Lopez¹

Institut Integriert Studieren, Altenbergerstr. 69, 4040 Linz, Autriche
{danya.gharbieh, maximilian.punz, Klaus.Miesenberger,
valentin.salinas_lopez}@jku.at

Résumé. Le projet FPB s'appuie sur les dernières avancées d'EPUB3 pour transformer la création de matériel d'apprentissage numérique inclusif. Bien que le support d'EPUB3 pour les médias riches offre un immense potentiel pour l'éducation accessible, l'adoption généralisée est entravée par une courbe d'apprentissage abrupte et la nature chronophage du développement. Cet article présente l'approche de FPB pour rationaliser la création de publications EPUB3 interactives et adaptables. FPB permet la création de livres numériques illustrés qui s'adaptent dynamiquement aux besoins individuels des enfants, favorisant ainsi l'égalité d'accès à l'éducation de base.

Mots-clés : Publication numérique accessible - EPUB3 - Illustrations numériques accessibles

1 Introduction

Cet article présente une nouvelle approche visant à améliorer l'accessibilité et la facilité d'utilisation des publications numériques grâce à des éléments non textuels adaptables. Le projet Flex Picture eBook (FPB), financé par l'UE, vise à révolutionner l'adaptabilité des publications numériques, en particulier des livres illustrés pour enfants. En testant un prototype de publication avec des enfants handicapés, leurs soignants et des éducateurs, le projet FPB vise à améliorer l'inclusion dans l'édition numérique.

Nous présentons une suite logicielle conçue pour rationaliser la création. Elle offre des outils innovants dans deux domaines essentiels :

- Création d'illustrations accessibles : Aider les auteurs à produire des illustrations qui sont intrinsèquement conçues pour s'adapter à un éventail de niveaux de compréhension.
- Intégration EPUB3 : Fournir des mécanismes permettant d'incorporer de manière transparente des illustrations complexes et en couches dans des publications EPUB3 conformes, afin d'assurer la compatibilité avec les appareils électroniques grand public et les technologies d'assistance les plus courantes.

En s'attaquant à ces défis distincts mais interconnectés, le projet FPB ouvre la voie à l'intégration de l'accessibilité dans le secteur de l'édition.

2 L'état de l'art

De nombreux outils de traitement de texte tels que *Google Docs*[3] permettent désormais d'exporter des fichiers texte vers le format EPUB. Toutefois, ce format ne permet aux utilisateurs que de convertir des fichiers statiques et exclut la possibilité de gérer les données de l'utilisateur, ce qui le rend inadapté à la création de livres électroniques "Flex Picture".

Actuellement, ce type de livres électroniques doit être fabriqué à la main, ce qui pose de nombreuses difficultés susceptibles de dissuader les auteurs et les éditeurs de distribuer leurs œuvres dans ce nouveau format accessible. Le principal problème réside dans la nature technique du type de fichier EPUB[1]. Il s'agit en fait de fichiers ZIP spécialement formatés, contenant des pages XHTML. Tout le texte doit être saisi manuellement dans ces fichiers XHTML et stylisé à l'aide de CSS, tandis que les fonctionnalités interactives doivent être codées en JavaScript. D'une certaine manière, chaque page peut être considérée comme un site web individuel qui doit être programmé manuellement. Cela pose un sérieux défi technique aux éditeurs, dont la majorité n'a pas de formation en programmation.

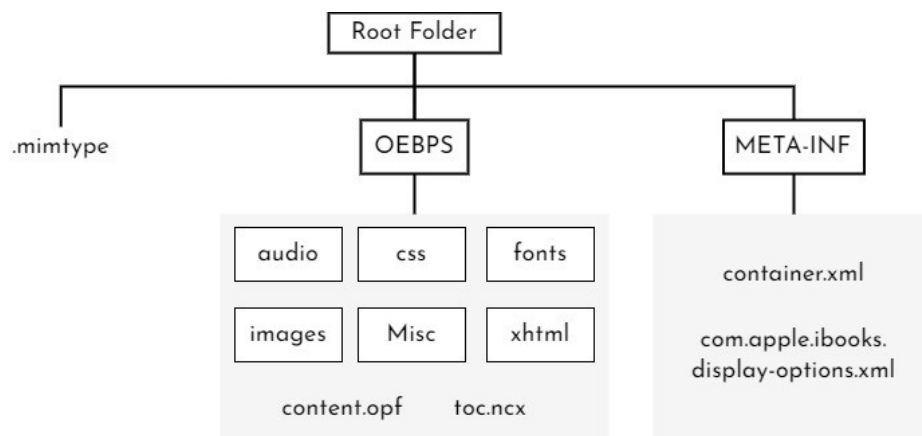


Fig. 1. La structure des fichiers du format EPUB. Le mimetype indique au lecteur électronique qu'il s'agit d'une publication numérique, tandis que META INF contient des fichiers qui pointent vers les documents racines. Le répertoire OEBPS contient les dossiers dans lesquels sont stockés tous les fichiers du livre (classés dans les sous-dossiers correspondants).

Le format EPUB comporte également des directives très strictes qui doivent être respectées. Si une seule condition n'est pas respectée ou si un paramètre est modifié de manière inattendue, l'EPUB ne peut pas être affiché sur un lecteur d'écran compatible[8]. Cette fragilité s'étend à de multiples dimensions, telles que la manière dont les fichiers sont disposés dans les conteneurs. Il existe une structure de dossier obligatoire qui dicte quel type de fichier doit être placé dans quel répertoire (voir fig. 1). Toutes ces structures de fichiers doivent être créées, et même des erreurs subtiles telles qu'une mauvaise orthographe du nom d'un dossier peuvent être préjudiciables. Le format nécessite également quelques fichiers différents qui indiquent le format

Ces fichiers doivent également être écrits de manière très particulière, ce qui nécessite un certain nombre de balises de script et de spécifications pour fonctionner correctement. Ces fichiers doivent également être écrits d'une manière très particulière, nécessitant un certain nombre de balises de script et de spécifications pour fonctionner correctement. Les IDE et les traitements de texte, dans lesquels ces fichiers sont généralement créés, n'indiquent pas à l'utilisateur les balises manquantes ou les options autorisées, ce qui rend le travail sur ces fichiers très opaque et peut souvent conduire à une grande frustration.

Il existe des logiciels spécialisés pour tester les EPUB, tels que *EPUB Checker* [4], qui permettent à l'utilisateur de valider ses fichiers. Leur but est d'attirer l'attention sur les défauts présents dans les documents et de les informer des problèmes spécifiques. Cela rend le processus de création un peu plus transparent, en donnant aux utilisateurs une liste d'erreurs qu'ils peuvent corriger. Cependant, pour rendre les fichiers conformes à la norme, l'utilisation d'un validateur nécessite toujours des connaissances techniques et peut prendre beaucoup de temps.

Des logiciels tels que *Sigil* [6] aident à naviguer dans ce labyrinthe, en permettant aux utilisateurs de générer les structures de répertoires et les fichiers nécessaires. Mais leur utilisation reste assez exigeante sur le plan technique, ce qui la rend inaccessible aux personnes qui n'ont pas d'expérience dans les technologies web. En effet, comme dans les exemples d'autres processeurs de texte, la sortie de ces fichiers est statique, et le code Javascript supplémentaire nécessaire à la création des livres électroniques Flex Picture doit être implémenté manuellement.

Le logiciel que nous proposons devrait abaisser la barrière à l'entrée qui accompagne la création de livres électroniques Flex, en permettant aux utilisateurs de générer automatiquement des EPUB conformes qui incluent la fonctionnalité Flex-eBook, l'utilisateur devant interagir le moins possible avec le codage, tout en lui donnant accès à l'éventail complet des fonctionnalités qui accompagnent le format.

3 Méthodologie

La section ci-dessous explique la méthodologie utilisée pour créer le Flex Picture eBook Builder. Le processus de création a été divisé en deux parties distinctes, mais interconnectées : L'interface utilisateur et le système qui crée les EPUBS eux-mêmes.

3.1 Construire une interface utilisateur accessible

La création d'une interface utilisateur accessible est cruciale pour garantir que tous les utilisateurs, y compris les personnes handicapées, puissent interagir avec les produits numériques de manière efficace et confortable. Pour rendre le logiciel accessible au plus grand nombre, nous avons décidé d'utiliser Electron[2] comme cadre de développement, car il permet de créer des applications indépendantes de la plate-forme. Dans la section ci-dessous, vous trouverez des explications détaillées sur tous les écrans présentés à l'utilisateur.

Projet Il s'agit du point de départ d'un parcours de publication du BFP. Ici, l'utilisateur a la possibilité de gérer des projets existants ou d'en commencer de nouveaux. (voir fig. 2)

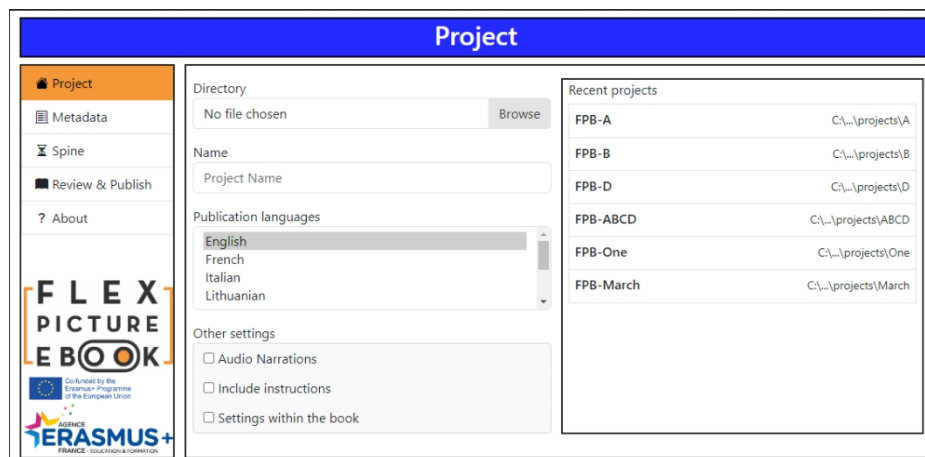


Fig. 2. Écran de projet

- Gérer les projets existants : utilisez la section "Projets récents" située sur le côté droit de l'interface. Cette fonction permet d'accéder facilement aux projets précédents.
- Commencer un nouveau projet : utiliser l'espace principal pour :
 1. Répertoire : naviguez pour sélectionner le répertoire dans lequel les fichiers du projet seront stockés.
 2. Nom : donnez un nom descriptif au projet.
 3. Langues de publication : spécifiez les langues dans lesquelles les EPUB seront créés.
 4. Autres paramètres : personnalisez les paramètres du livre, indiquez s'il comprendra des options telles que des narrations audio, du contenu pédagogique et si les paramètres seront inclus.

Métadonnées L'utilisateur fournit ici les informations essentielles qui décrivent son BFP. (voir fig. 3)

- Ajouter des métadonnées : choisissez un type de métadonnées dans "Métadonnées disponibles" à gauche, puis cliquez sur "Ajouter" pour afficher les champs obligatoires dans "Métadonnées ajoutées" à droite.
- Supprimer les métadonnées : choisissez un élément dans la liste "Métadonnées ajoutées" à droite, puis cliquez sur "Supprimer" pour renvoyer les champs dans la liste des métadonnées disponibles à gauche.
- Ajout de types de métadonnées spécifiques à une langue : certains types de métadonnées nécessitant une saisie multilingue (par exemple le titre) disposeront d'espaces de table modifiables pour chacune des langues de publication sélectionnées précédemment.
- Types requis : L'application met en évidence les types requis et invite l'utilisateur final à les compléter avant de poursuivre.
- Section Info : affiche des informations sur le type de métadonnées sélectionné et sur la manière de les définir correctement.

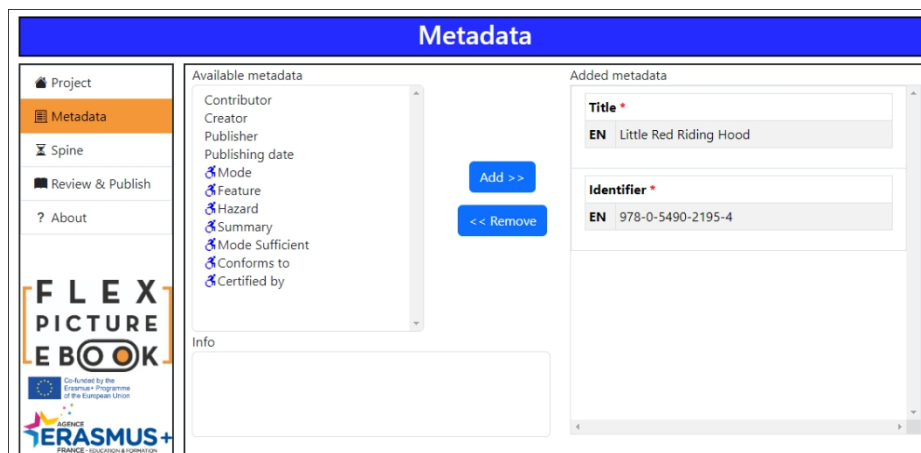


Fig. 3. Écran de métadonnées

Spine Dans cette section, tous les éléments de contenu peuvent être ajoutés et arranger les pages du FPB. (voir fig. 4)

- Ajouter et ordonner des pages dans la publication, à l'exception de la couverture (première page) et du crédit (dernière page).
- Section du contenu de chaque page :
 1. Texte : saisissez le contenu du texte de la page dans le tableau de texte modifiable pour chacune des langues de publication.
 2. Narrations : téléchargez des fichiers de narration audio pour chacune des langues de publication.
 3. Images et scénarios :
 - (a) Image : Parcourir pour sélectionner le fichier XHTML contenant l'illustration adaptée (SVG).
 - (b) Style : L'application identifiera les ressources externes liées telles que les fichiers CSS et JavaScript et invitera l'utilisateur à spécifier leur emplacement pour une intégration et un rendu corrects dans le projet.
 - (c) Vérifier les fichiers liés et éviter la duplication inutile des ressources utilisées sur plusieurs pages.
 4. Texte alternatif : saisissez le texte alternatif de l'image dans le tableau modifiable pour chacune des langues de publication.
 5. Supprimer : Supprime la totalité de la page sélectionnée, à l'exception de la couverture (première page) et du crédit (dernière page).

Révision et publication Dans la dernière section, les BFP sont finalisés et préparés pour la distribution. (voir fig. 5)

- Nom du fichier : Définissez le nom du fichier de publication. Par défaut, il contiendra la même valeur que celle saisie dans le champ "Nom" de l'écran du projet.

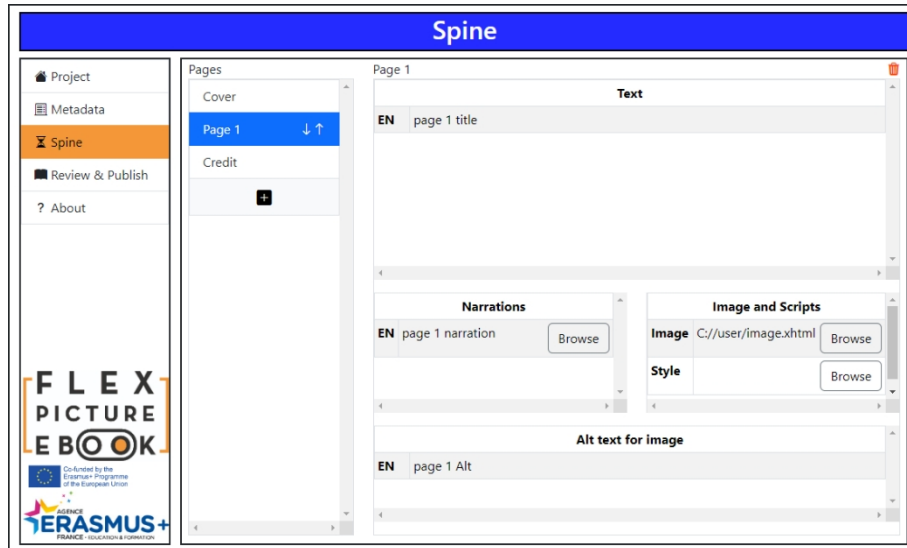


Fig. 4. Écran de la colonne vertébrale

- Lancer le livre électronique publié : Cochez la case pour lancer immédiatement le livre électronique (à l'aide de Thorium).
- Générer : Créer la structure des fichiers intermédiaires et les fichiers de métadonnées. Tout le contenu généré sera placé dans un ensemble de sous-dossiers (un par langue de publication) dans le répertoire racine du projet. Cela permet à l'utilisateur d'examiner et de modifier les fichiers qui constitueront les BFP si nécessaire.
- Révision : Valider la structure du fichier généré par rapport aux normes EPUB3 à l'aide de l'outil EPUB checker.
- Publier : Produit un EPUB à partir des fichiers générés pour chaque langue de publication prête à être distribuée.

3.2 Gestion des fichiers et création automatisée de livres électroniques

L'un des défis les plus intéressants que nous ayons eu à relever a été l'importation automatique des fichiers nécessaires au fonctionnement de l'interactivité. Les images sont enregistrées sous forme de SVG dans des fichiers XHTML et ont souvent besoin d'un code Javascript supplémentaire, ou d'un style et d'un formatage CSS, pour être affichées correctement. Afin de simplifier au maximum le processus pour l'utilisateur, nous avons décidé d'importer automatiquement ces fichiers dans notre programme. Pour ce faire, le document XHTML est analysé ligne par ligne à la recherche d'instructions d'importation. En utilisant la manipulation de chaînes de caractères, les endroits exacts où ces déclarations pointent sont ensuite déterminés. La plupart du temps, ces instructions spécifient un emplacement relatif à celui du fichier sélectionné, ce qui signifie que la position absolue dans le système de fichiers doit être déterminée pour importer la dépendance. Après une vérification rapide, les chemins sont ajoutés à la liste des documents.

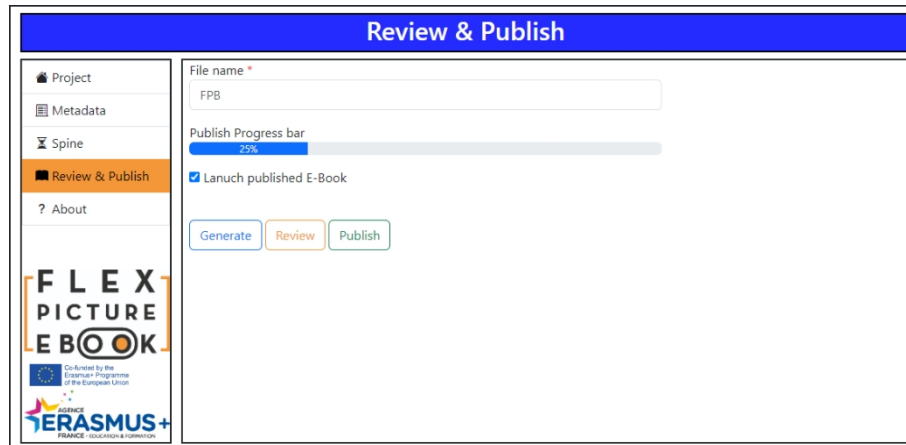


Fig. 5. Écran de révision et de publication

qui est intégré dans l'EPUB. Si plusieurs fichiers pointent vers la même dépendance (ce qui signifie qu'un fichier a été réutilisé dans plusieurs XHTML), le système ne l'importe qu'une seule fois. Si une dépendance n'a pas été trouvée, l'utilisateur est invité à la sélectionner manuellement.

Avant la création de l'EPUB, tous ces documents sont copiés dans les dossiers correspondants. Pour que tout fonctionne correctement, les déclarations d'importation des XHTML doivent pointer vers ces nouveaux emplacements, ce que notre système résout en remplaçant les anciens chemins par les nouveaux chemins relatifs. Comme chaque EPUB a la même structure de fichier, la création des chemins relatifs peut être standardisée.

Notre système génère un certain nombre de fichiers, tels que le menu affiché au début de l'EPUB, qui sont générés dynamiquement à partir de modèles. Ces modèles sont ensuite remplis en fonction des métadonnées spécifiées dans la partie frontale. Cela nous permet de créer rapidement des BFP dans plusieurs langues prédéfinies. L'un des problèmes qui subsistent est le fait que pour chaque langue, le modèle doit être traduit manuellement, ce qui représente une énorme charge de travail pour nous. Bien que la plupart des fichiers soient indépendants de la langue, des éléments tels que les menus de configuration nécessitent toujours la traduction des noms des boutons. La plupart des autres modèles sont générés de manière totalement dynamique, mais les menus sont (pour la plupart) des fichiers XHTML pré-écrits, dont le contenu est modifié en fonction des spécifications de l'utilisateur. Cela est dû à la taille relativement importante des fichiers de menu, qui a rendu cette approche plus efficace lors de la production. Toutefois, cela peut constituer un obstacle en termes de traduction, car le contenu textuel est enchevêtré dans le code HTML. Une solution qui sera explorée consiste à créer les menus à partir de modèles indépendants de la langue et à lire le texte à partir de fichiers qui ne contiennent que les noms des éléments, ce qui permet à ces fichiers d'être traduits par un service de traduction automatique tel que DeepL[5]. Ces fichiers devraient toujours être vérifiés par des natifs de la langue en question, mais cela permettrait de gagner beaucoup de temps et de réduire les coûts potentiels.

Le système est conçu de telle manière que si tous les champs obligatoires sont remplis, le programme exporte un EPUB fonctionnel, qui peut être validé sans problème. Le résultat du programme contient deux parties : L'EPUB et un dossier contenant tous les fichiers bruts, permettant à l'utilisateur de les parcourir et de s'assurer qu'ils contiennent les bonnes informations. Comme indiqué précédemment, l'EPUB lui-même est un ZIP contenant tous ces documents, dont l'extension de fichier a été changée en *.epub*. Cependant, il y a d'autres éléments qui pourraient potentiellement faire échouer notre programme : Premièrement, le format exige que le ZIP soit dans un ordre particulier, à savoir que le fichier *.mimetype* (qui indique au lecteur électronique que le dossier en question est bien un EPUB), se trouve en première position. Deuxièmement, la compression inhérente à un fichier ZIP ne peut pas être appliquée de la même manière à tous les documents, en particulier le fichier *.mimetype* doit être laissé non compressé et non crypté[7]. Ordonner les éléments à l'intérieur d'un fichier ZIP ou compresser les éléments de manière non uniforme peut s'avérer assez difficile sans logiciel spécialisé, c'est pourquoi nous avons veillé à ce que le processus de rendu soit aussi automatisé que possible.

4 Travaux futurs

Notre logiciel devrait réduire la charge de travail et les connaissances techniques requises pour créer des livres électroniques Flex et abaisser la barrière à l'entrée de manière à ce que le format se généralise. La création d'un outil permettant aux illustrateurs d'exporter directement vers le format requis et de simplifier le processus d'abstraction manuelle des images est l'un des moyens de simplifier davantage ce processus. Dans les prochaines étapes, cet outil sera implémenté en tant que plugin pour Adobe Illustrator, ce qui, nous l'espérons, rendra la tâche de création d'images plus facile et plus accessible.

Remerciements. Flex Picture eBook a reçu un financement de la Commission européenne et de l'Agence Erasmus+ France dans le cadre de l'action KA220-SCH Partenariats de coopération dans l'enseignement scolaire. Convention de subvention n° : 2022-1-FR01-KA220- SCH-000088072.

Références

1. Consortium Daisy : Epub (2024), <https://daisy.org/activities/standards/epub/>
2. Electron : cadre Electron (2024), <https://www.electronjs.org/docs/latest/>
3. Google : Google docs (2024), <https://docs.google.com/>
4. pagina : Epub-checker (2024), <https://pagina.gmbh/startseite/leistungen/publishing-softwareloesungen/epub-checker/>
5. SE, D. : Deepl (2024), <https://www.deepl.com/de/translator>
6. Sigil : Sigil - éditeur epub (2024), <https://sigil-ebook.com/>
7. W3C : Exigences relatives aux fichiers zip Epub (2023), <https://www.w3.org/TR/epub-33/#sec- zip-container-zipreqs>
8. Groupe de travail EPUB 3 du W3C : Epub 3.3 (2023), <https://www.w3.org/TR/epub-33>