

READ : une plateforme pour l'évaluation de méthodes d'accès non-visuelles aux documents numériques

Laurent Sorin
Université de Toulouse; UPS &
UT2J; IRIT
118 route de Narbonne
31062 Toulouse
(+33)5.61.55.77.02
sorin@irit.fr

Julie Lemarié
Université de Toulouse; UT2J;
CLLE
5 allées Antonio Machado
31100 Toulouse
(+33)5.61.50.35.41
lemarie@univ-tlse2.fr

Mustapha Mojahid
Université de Toulouse; UPS;
IRIT
118 route de Narbonne
31062 Toulouse
(+33)5.61.55.63.18
mojahid@irit.fr

RESUME

READ (**R**estitution de l'**A**rchitecture des **D**ocuments) est un logiciel permettant un accès non-visuel *augmenté* à des documents balisés. Cette plateforme a été initialement développée pour évaluer différentes méthodes de restitution de la mise en forme des textes [8]. La flexibilité de l'architecture de READ permet d'implémenter facilement de nouvelles modalités d'entrée et de sortie, ainsi que de nouvelles fonctionnalités ; l'application sera mise à disposition prochainement en open source pour réutilisation non-commerciale. Le but principal est de permettre l'implémentation et l'évaluation rapide de méthodes d'accès non-visuels aux documents, domaine dans lequel un effort de recherche certain est attendu [7]. Une démonstration permettra aux participants de la conférence d'observer les fonctionnalités implémentées et d'avoir un aperçu des possibilités d'extension de l'application.

Mots clés

Accessibilité numérique, interactions non-visuelles, plateforme de recherche, open source.

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces - Ergonomics, Input devices and strategies, Auditory (non-speech) feedback, Prototyping.

K.4.2 [Computers and Society]: Social Issues- Assistive technologies for persons with disabilities.

1. INTRODUCTION

READ a été initialement développé dans le cadre d'un projet visant à évaluer différentes méthodes de restitution des informations et fonctions de lecture qu'offre la mise en forme visuelle des textes [8], afin d'améliorer l'accessibilité des documents numériques. Le terme *restitution* fait référence à la possibilité pour les déficients-visuels d'accéder aux mêmes informations que les lecteurs voyants et avec une efficacité optimale. En ce qui concerne l'accessibilité des documents numériques,

un certains nombre de récents travaux montre que la problématique est large et d'actualité.

Par exemple les lecteurs d'écrans, l'outil principal utilisé par les déficients-visuels pour accéder à l'information numérique, présentent une utilisation souvent frustrante et peu efficace [6], même quand le contenu est supposé accessible d'après les normes en vigueur [7].

De plus, la linéarité des modalités audio et tactile est souvent mentionnée comme une cause prépondérante des problèmes observés. Plusieurs initiatives en recherche tentent en effet de proposer un accès non-linéaire par exemple avec des pistes audio simultanées [5] ou encore une vue condensée de l'information avec des résumés de textes [4]. La multi-modalité est également considérée comme une solution potentielle aux problèmes de surcharges informationnelles liées à la linéarité de ces canaux [3]. On retrouve également ce type d'approche pour l'accès aux contenus complexes comme les mathématiques [1].

L'évaluation de ce type de paradigme nécessite de développer des outils dédiés, ou de modifier des outils open source existants comme certains lecteurs d'écrans. Cependant, de tels outils présentent une architecture complexe car devant gérer l'interaction avec le système d'exploitation ; READ fournit une alternative simplifiée dans le cadre de l'accessibilité des documents. L'objectif est donc de fournir une base réutilisable pour ce type d'implémentations, requérant un minimum de développements. READ fournit pour cela un environnement Windows dans lequel les modalités d'entrée et de sorties sont paramétrables. Nous souhaitons publier cet outil le plus rapidement possible sous licence CC BY-NC-SA¹.

2. ARCHITECTURE DU SYSTEME

L'objectif général était de développer un outil entièrement configurable à l'architecture modulaire, basée sur du XML² et le rajout de plugins.

-Spécifications techniques : READ a été développé en C# pour profiter de la librairie native permettant de contrôler les voix SAPI³. N'importe quel ordinateur utilisant .NET 3.5 (ou plus récent) peut utiliser READ. La configuration peut être définie dans les fichiers XML

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s). IHM '15, Oct 27-30 2015, Toulouse, France

¹ Creative Commons Attribution (by) NonCommercial ShareAlike

² Extensible Markup Language

³ Speech Application Programming Interface

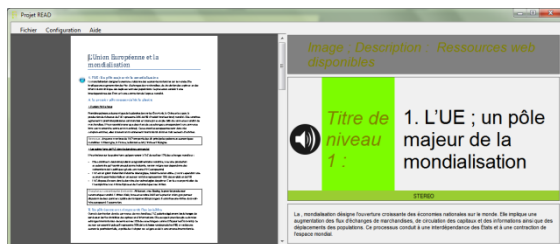


Figure 1. Capture d'écran de READ

correspondants (format choisit pour sa simplicité et son interopérabilité).

-Architecture globale : L'application utilise un paradigme de blocs et d'événements de lecture : le document est segmenté en blocs d'informations, chaque bloc comprenant un ou plusieurs événements pouvant être joués séquentiellement ou en parallèle.

Certaines "relations" entre blocs peuvent être définies. Par exemple des relations liées au discours, comme le fait qu'un bloc puisse constituer une illustration d'un ou plusieurs autres blocs. Enfin, plusieurs modes d'accès peuvent être définis, chacun avec une configuration spécifique.

-Fichiers de configuration : les fichiers spécifient comme chaque balise XML du document d'origine sera transformée en blocs d'événements (ex. avant chaque <p>, jouer un fichier son). N'importe quel vocabulaire peut être utilisé du moment qu'il correspond à celui utilisé dans le document.

On peut également définir plusieurs types d'événements (lire du texte avec une voix de synthèse, jouer un son, etc.) ayant des paramètres spécifiques (ex. lire du texte avec une certaine voix, à une certaine hauteur, avec une certaine vitesse).

-Modalités d'entrée/sortie et fonctions définies : Actuellement, READ peut utiliser n'importe quelle voix SAPI et contrôler ses paramètres avec des balises SSML⁴. Des sons .WAV (ex. icônes audio) peuvent être joués, et n'importe quel son ou parole peut être spatialisée, c'est-à-dire que l'on simule une source sonore dans l'espace (voir [2] pour un exemple d'application). READ fournit également une visualisation des événements en cours de lecture (voir Figure 1). La navigation peut se faire séquentiellement bloc à bloc ou alors en fonction des relations entre bloc. Enfin, chaque action système et utilisateur est enregistrée dans un fichier texte pour les besoins de l'analyse.

-Rajout d'extensions: Pour rajouter de nouveaux périphériques ou modalités d'entrée ou de sortie, ou encore de nouvelles fonctions de navigation, des plugins peuvent être facilement rajoutés à l'application en plaçant les fichiers C# correspondants dans le répertoire dédié de l'application. Plusieurs interfaces de programmation sont à disposition pour que les plugins puissent interagir avec l'application.

3. CONCLUSION

Cette plateforme permet d'implémenter et de combiner différentes modalités d'entrée et de sortie afin d'implémenter différentes méthodes d'accès multimodal ou non-linéaire aux documents. READ pourrait ainsi être utilisé pour évaluer l'impact de ces méthodes sur les

performances de navigation ou encore l'apprentissage, dans le cadre de la recherche autour de l'accès non-visuel aux documents.

La principale limite de READ réside cependant dans la nécessité de devoir éditer manuellement les fichiers XML de configuration, les fichiers de relation et le ou les documents utilisés. Ce processus peut être relativement long, ce qui nous semble une contrepartie acceptable dans un cadre de la recherche.

Nous avons récemment utilisé READ pour évaluer différentes méthodes de restitution de la mise en forme des documents avec 60 participants non-déficients et 12 participants non-voyants⁵. Même si ces expérimentations ne visaient pas à évaluer le système en lui-même mais les méthodes implémentées, il est à noter que sur les 40h de tests le système a été stable et dépourvu d'erreurs. De plus, certaines méthodes implémentées ont observé des scores d'utilisabilité en moyenne de 90 sur 100 (mesurés avec un questionnaire SUS⁶).

4. REMERCIEMENTS

Ces travaux de recherche ont été financés par le PRES-Toulouse et la Région Midi-Pyrénées.

5. BIBLIOGRAPHIE

- [1] Bates, E. and Fitzpatrick, D. Spoken mathematics using prosody, earcons and spearcons. *ICCHP 2010*, (2010), 407–414.
- [2] Cofino, J., Barreto, A., Abyarjoo, F., and Ortega, F.R. Sonically-Enhanced Tabular Screen-Reading. *International Technology and Persons with Disabilities Conference*, (2014).
- [3] El-glaly, Y.N. and Quek, F. Digital Reading Support for The Blind by Multimodal Interaction. *ICMI '14*, (2014), 439–446.
- [4] Gadde, P. and Bolchini, D. From Screen Reading to Aural Glancing : Towards Instant Access to Key Page Sections. *ASSETS '14*, (2014), 67–74.
- [5] Guerreiro, J. and Gonçalves, D. Text-to-Speeches : Evaluating the Perception of Concurrent Speech by Blind People. *ASSETS '14*, (2014), 169–176.
- [6] Lazar, J., Allen, A., Kleinman, J., and Malarkey, C. What Frustrates Screen Reader Users on the Web: A Study of 100 Blind Users. *International Journal of Human-Computer Interaction* 22, 3 (2007), 247–269.
- [7] Power, C., Freire, A., Petrie, H., and Swallow, D. Guidelines are only half of the story: accessibility problems encountered by blind users on the web. *SIGCHI Conference on Human Factors in Computing Systems*, ACM (2012), 433–442.
- [8] Sorin, L., Mojahid, M., Aussenac-gilles, N., and Lemarié, J. Improving the accessibility of digital documents for blind users : contributions of the Textual Architecture Model. *HCI International 2013*, Springer-Verlag Berlin Heidelberg (2013), 423–431.

⁵ The results will be published soon ; this experiment aim was to test text formatting restitution

⁶ Standard Usability Scale

⁴ Speech Synthesis Markup Language