

Rapport projet de synthèse NumWeb

Sofiane Hamiti  
Nicolas Reitz

Table des matières

[I. Introduction 3](#_Toc317768043)

[II. L’existant 3](#_Toc317768044)

[III. Accessibilité du web 5](#_Toc317768045)

[IV. Critique de Vimperator 6](#_Toc317768046)

[V. Notre démarche 7](#_Toc317768047)

[A. Qu’est-on susceptible de faire sur internet ? 8](#_Toc317768048)

[B. Comment résoudre ces différents problèmes ? 8](#_Toc317768049)

[1. La problématique commune à toutes les tâches : le clic 8](#_Toc317768050)

[2. Navigation 9](#_Toc317768051)

[3. Téléchargement 11](#_Toc317768052)

[*4.* Vidéo et Musique 11](#_Toc317768053)

[5. Copié/Collé 12](#_Toc317768054)

[*6.* Impression 12](#_Toc317768055)

[C. Technologie pour la mise en œuvre du plug-in 12](#_Toc317768056)

[1. XUL 12](#_Toc317768057)

[2. JavaScript 13](#_Toc317768058)

[3. Ajax 13](#_Toc317768059)

[D. Mise en œuvre 13](#_Toc317768060)

[1. Interface du prototype 13](#_Toc317768061)

[2. Fonctionnement du prototype 15](#_Toc317768062)

[3. Le CGI (Common Gateway Interface) 15](#_Toc317768063)

[4. Relation prototype/CGI 15](#_Toc317768064)

[E. Tests utilisateurs 15](#_Toc317768065)

[F. Conclusion 15](#_Toc317768066)

# Introduction

Les utilisateurs handicapés physiques peuvent se heurter à des difficultés pour accéder au contenu du World Wide Web. Ces problèmes peuvent toutefois se résoudre grâce à des aides techniques afin de faciliter l'interaction et l'accès au contenu de la page et de permettre aussi de profiter au maximum des outils offerts par le navigateur utilisé.

Notre volonté est de développer une de ces aides techniques afin que les personnes ayant un handicap physique puissent accéder au contenu du World Wide Web de façon autonome. Le but est d’associer un clavier à balayage et un capteur de type tout-ou-rien afin de manipuler la page web.

Malgré la volonté de rendre cet outil accessible au plus grand nombre, à notre stade du développement, le clic reste indispensable. Le prototype reste donc accessible uniquement aux personnes capables de le créer.

Le développement utilise plusieurs technologies. Le développement se base sur le navigateur web Firefox, avec lequel il est possible de connecter des plug-ins codés en XUL. XUL est un langage d’interface graphique dont le dynamisme est apporté via du JavaScript. De plus nous avons utilisé la combinaison d’Ajax et de script CGI afin de réaliser des actions spécifiques tel que la simulation de clic.

# L’existant

Pour des personnes atteintes d’un déficit moteur sévère conjugué à des troubles de la parole, une solution couramment adoptée pour les aider à communiquer avec leur entourage consiste à réaliser un défilement lumineux automatique sur un écran d'ordinateur ou sur un écran autonome. Ce défilement contrôlé par un capteur de type tout-ou-rien adapté à la personne permet de sélectionner un item (icône, mot, lettre, ...) parmi la matrice d'items proposée en fonction de l'action (lire, écrire, etc.).

Le principal défaut d'une communication à balayage est sa lenteur d'action : le délai de défilement élémentaire entre deux items peut varier de 200 ms à plusieurs secondes selon les possibilités physiques et cognitives de la personne.

La majeure partie des recherches dans ce domaine traite du problème de la saisie de texte avec l'usage d'un clavier virtuel optimisé. Dans le cas d'un langage de communication alphabétique de nombreuses études ont cherché à améliorer cet état de fait en diminuant le nombre moyen de sélections nécessaires pour composer une phrase. On peut également chercher à diminuer le temps d'accès à un item en les reconfigurant dynamiquement après chaque validation de façon à présenter les items les plus probables en début de défilement. D'autres travaillent sur le temps d'action que met la personne à valider un item en vue d'adapter automatique le temps de défilement.

Avec les méthodes à balayage, se pose également le problème du nombre d’items sélectionnables qui soient rapidement accessibles. En effet dans le cas de l'utilisation d'un logiciel de navigation sur le Web, on peut se trouver face à un grand nombre de liens ou l'accès par défilement est très lent, incommode voire même impossible si le site web ne respecte pas les normes d'accessibilité de pages web.

On trouve des solutions logiciels qui utilisent un clavier virtuel dédié au contrôle du navigateur (clic souris, touche de tabulation, flèches de directions, ...). Ces solutions émulent généralement les déplacements du curseur de la souris en permettant de contrôler son déplacement afin de faire "comme si l'utilisateur contrôlait sa souris".

On trouve sur le marché des solutions pour piloter les déplacements de la souris via l'usage de caméra (IR, webcam) qui permettent de contrôler le curseur par un suivi de regard ou mouvement de tête. Sans parler des réglages techniques qui peuvent être fastidieux dans certains cas (éclairage de la pièce, sensibilité des déplacements), l'inconvénient majeur à notre sens est que le fait de contrôler les déplacements du curseur ne suffit pas il faut aussi comme sur une souris avoir la possibilité de cliquer. Généralement avec ce type de technologie le clic se fait par une pause à l'endroit où se situe le curseur ce qui demande de la précision et une certaine concentration à l'usage. De plus le curseur suit le regard ou bien la tête de l'utilisateur et de ce fait il n'y a pas vraiment de repos pour la lecture puisque l'arrêt du curseur est compris comme la volonté de cliquer ce qui peut poser problème si l'arrêt se fait sur un item/liens sélectionnable.

L'analyse de l'usage d'un utilisateur face à son écran peut se diviser grossièrement en deux grandes phases : le temps de lecture et le temps d'action sachant que pour certaines personnes le déplacement du curseur est une aide à la lecture et non la volonté de cliquer où se situe le curseur.

C'est dans cette idée de mixer l'usage d'un clavier virtuel contrôlable par accès directe ou bien par défilement, de respecter les phases de lecture et d'action, de permettre d'avoir un grand nombre d'items sélectionnables rapidement que se situe le travail présenté dans ce papier.

On se propose d'introduire le concept de marquage/numérotation à la volée des informations à l'écran qui ouvre la voie à un accès direct avec un clavier virtuel. En effet le fait de numéroter chaque item donne à l'utilisateur la possibilité d'interagir rapidement via son capteur de commande et ce de manière localisée quel que soit son type proportionnel ou Tout ou rien. Il peut être plus rapide d'écrire "67" sur un clavier numérique que de sélectionner l'item 67 dans une liste, ceci dépendant bien entendu de la technique de balayage utilisée.

Notre objectif est de fournir une méthode générique d'interaction utilisable dans le cadre des interfaces d'aide à la communication basée sur une interaction via un clavier numérique qui permet de contrôler un nombre important d'item en limitant les déplacements du curseur de contrôle de l'interface ou bien les défilements dans le cadre des interfaces à balayage. Le prototype d’une application dédiée à l'utilisation du navigateur Firefox est présenté.

# Accessibilité du web

Actuellement, la majorité des sites internet est accessible uniquement au moyen de navigateurs graphiques traditionnels. Selon l’importance du handicap physique, moteur ou neurophysiologique leur accès est difficile voire impossible.

Pour répondre à cela, la WAI (Web Accessibility Initiative) du W3C (World Wide Web Consortium) propose un ensemble de directives destiné aux créateurs de sites Internet, afin de rendre leurs sites plus accessibles aux personnes handicapées. Ces directives s'appliquent au niveau du code HTML et demandent une réflexion sur la structure et l'organisation des données du site.

Cependant, l'essor des services et applications en ligne reposant sur l'utilisation croissante de technologies hybrides telles que JavaScript ou AJAX complique les choses. La récente Accessible Rich Internet Applications Suite (ARIA) vise à mettre en place le cadre normatif nécessaire à l'accessibilité des applications Web dynamiques.

Dans le cas où le handicap moteur entraîne une imprécision dans les gestes et donc une difficulté à utiliser pleinement certains périphériques externes comme le clavier ou la souris, l’usage de la touche tabulation (ou équivalent) permet de sélectionner séquentiellement les éléments cliquables du document. Ceci est toutefois inopérant si le site ne respecte pas les directives du WAI avec en particulier l’utilisation du JavaScript.

D’autres stratégies existent pour soulager la tâche motrice de l’utilisateur comme par exemple le clic automatique par survol de la souris. L’utilisateur déplace le curseur souris et le clic est obtenu à la position du curseur après un certain délai sans bouger.

Dans le cas d’un handicap moteur plus important, il est possible d’éviter une bonne partie des tâches motrices avec l’usage de logiciels de reconnaissance vocale. Mais il est plus courant de se porter sur des solutions à base de défilement. Le défilement automatique des liens de la page affichée est la plus courante. CompuWeb-Access propose également un défilement automatique du texte de la page et eSSENTIAL Accessibility™ propose de nombreuses variantes d’accès aux éléments de la page : Clic automatique, Recherche manuelle / Recherche automatique, Souris XY, Souris de direction, Souris radar et Détection de mouvements mains libres à base de caméra web.

Alors que CompuWeb-Access et HandiLog demandent une intégration de leur technologie directement dans le site, eSSENTIAL Accessibility™ s’appuie sur un navigateur propriétaire. Cette solution apparait discutable dans la mesure où ce navigateur reste perfectible et éloigné des standards du marché. De plus, malgré les nombreuses techniques de sélection proposées, l’accès aux éléments de la page reste long et parfois difficile voire impossible.

De plus le travail proposé pourra être facilement adapté à tous les outils de la fondation Mozilla à l’initiative de Firefox. Pour faciliter l’accès aux éléments de la page, nous avons basé notre travail sur l’extension Vimperator décrite dans la partie suivante.

# Critique de Vimperator

Dans la recherche de l’existant, nous avons trouvés deux papiers dont l’aboutissement est fortement ressemblant à Vimperator. Ces papiers exposent notamment un algorithme de fonctionnement général et le second sur la problématique d’optimisation de labélisation des items.

Le but de Vimperator est de pouvoir contrôler Firefox sans utiliser d’autres périphériques que le clavier. Le fonctionnement de Vimperator au niveau de manipulation s’inspire du célèbre éditeur de texte Vim. Vimperator est notamment à l’origine de Pentadactyl, outil plus orienté pour développeur.

Vimperator permet notamment la navigation dans les onglets, dans la page web, la recherche de texte, navigation dans les liens hypertextes. La navigation par lien hypertexte fonctionne via l’association d’un label à un lien.

L’imprime écran ci-dessous montre le site de l’APF (www.apf.fr)



Ci-dessous le même avec la numérotation des liens fait par Vimperator. L’utilisateur a activé cette fonctionnalité à l’aide de la touche « f ». L’accès au lien est alors possible en saisissant le numéro. Un système de filtre permet de réduire le nombre de label en fonction de ce qui est déjà saisi. Aussi bien lors de la navigation par numéro que par recherche alphanumérique.



Toutefois, certaines limites existent. La numérotation est effectuée par rapport à la page affichée et commence à 1. En cas de défilement, un élément de la page changera donc de numéro. Même si Vimperator fonctionne très bien dans la majorité des cas, certaines difficultés peuvent apparaître avec du code JavaScript ou certaines feuilles de style. Par exemple, sur le site www.apf.fr, il est impossible d’accéder aux menus déroulants : « S’INFORMER », « NOUS AIDER » et « S’IMPLIQUER ». Ils sont numérotés mais ils nécessitent le pointage avec la souris pour déclencher le script JavaScript. De même, les éléments (à gauche de PACTE APF) sont correctement numérotés de 18 à 22 et donc accessible directement mais les numéros n’apparaissent pas correctement à l’écran. Ces défauts rendent donc l’usage de la souris encore nécessaire.

# Notre démarche

Deux de nos objectifs sont de minimiser les modifications sur le plug-in Vimperator, et d’utiliser un minimum le plug-in Vimperator. C’est pourquoi nous avons développé notre propre plugin avec ses propres fonctionnalités mais aussi une reprise de fonctionnalités de Vimperator (notamment pour la numérotation des liens).

## Qu’est-on susceptible de faire sur internet ?

Afin de savoir quels actions est susceptible de faire une personne sur internet nous avons procédés par association d’idées en se basant sur les trois sens les plus usité dans l’interaction avec l’ordinateur (Toucher, Ouïe, Vue).

## Comment résoudre ces différents problèmes ?

### La problématique commune à toutes les tâches : le clic

La difficulté même d’une personne handicapée physique est de pouvoir cliquer à un endroit précis. En effet cette modalité permet toutes les actions relatives à l’utilisation d’un ordinateur. Pour pouvoir palier à ce problème il faut que l’utilisateur soit capable de fournir une entrée binaire, couplée à un système de balayage l’utilisateur est capable d’exécuter une action précise.

Reste à définir comment. L’utilisation d’une souris classique peut se révéler être une bonne approche dans un premier temps. Néanmoins la détection d’un clique peut entrer en conflit avec le système, en effet que ce passe-t-il si l’utilisateur déplace la souris au moment du clic : la perte du focus et/ou une interaction non souhaité (fermeture de fenêtre, déclenchement d’un lien hypertexte, etc.).

Nous avons donc opté pour l’utilisation d’un Joystick. Les avantages de l’utilisation de ce type de périphérique est de pouvoir exploiter facilement une seule entrée qui plus est binaire.



### Navigation

Nous avons distingué trois moyens de naviguer via :

* des liens préenregistrés : les favoris
* Vimperator pour naviguer de page en page
* des balayages écran et simulation de cliques pour activer les liens hypertextes (nous rentrerons dans les détails dans la partie V.D.3

#### Les liens

Vimperator gère les liens via l’utilisation de la touche « f ». La réflexion que nous avons eue à ce niveau est la portée de numérotation des liens dans la page. Par défaut Vimperator numérote les liens visibles sur l’écran à un instant T. Il est intéressant d’avoir une numérotation sur l’intégralité de la page, en effet de cette manière permet à l’utilisateur de faire des interactions intermédiaires (monter, descendre dans la page) avant de visiter une nouvelle page.



#### Gestion des Favoris et des préférences utilisateurs

L’utilisation des favoris est une fonctionnalité dont une grande majorité des usagers du web se servent. Nous avons trouvé diverses manières quant à ce problème. Dans un premier temps, nous avons pensé à l’utilisation de fichier, grâce à ceci l’utilisateur pourra éditer certaines préférences (vitesse de balayage, etc.), nous l’avons dans un premier temps adapté à la gestion des favoris. Par la suite nous avons trouvé comment manipuler les favoris de Firefox directement.

##### Via l’utilisation de fichier

Nous proposons plusieurs solutions à ce niveau. Nous avons regardé plusieurs méthodes afin d’avoir un aperçu des différentes possibilités que nous offre les technologies du web et de Firefox.

###### Avec Firefox

En utilisant cette méthode on peut lire, écrire, manipuler des fichiers.

###### Avec FileReader et FileWriter

Une autre solution est de passer via ces objets qui permettent respectivement la lecture et l’écriture directement en JavaScript sans passer par Firefox. Ces objets sont issus de HTML5. Dans l’écriture de code cette méthode est plus lisible et plus intuitive que via Firefox.

###### Avec Ajax

Dans la mesure où nous utilisons un serveur local pour la lecture d’un script servant au balayage, nous pouvons aussi utiliser l’Ajax pour récupérer des données de préférence. Cela nécessite bien sûr d’avoir les fichiers sur le serveur local.

##### Via l’utilisation de la base de données SQLite de Firefox

Firefox utilise une base de données SQLite comme base de données interne. Sa manipulation est possible à travers deux objets : le premier servant pour la lecture, écriture, modification des informations sur un lien donné. Le second pour récupérer l’ensemble des liens suivant certains critères (répertoire, nom, etc.).

#### Le balayage

Nous exploitons le balayage de l’écran dans le cas ou Vimperator n’est plus suffisant. Comme cité auparavant dans le cas des menus déroulants, lorsqu’une fenêtre modale s’ouvre, ou simplement si l’utilisateur souhaite déclencher une action qui n’est ni associé à un élément formulaire ou lien.

##### Ecran



Ce type de balayage est utile dans le cas où l’utilisateur aurait à manipuler l’interface de Firefox par exemple. Les traits noirs sont les curseurs utilisés pour le balayage.

##### Fenêtre

Ce type de balayage est plus adapté au contenu web, en effet ce balayage est spécifique au contenu ce qui représente un gain de temps par rapport à un balayage écran.

### Téléchargement

La navigation jusqu’au téléchargement du contenu ne pose pas de problème en effet Vimperator rempli très bien ce rôle. Pour ce qui est du choix de répertoire de destination on pourrait utiliser un balayage écran.

### Vidéo et Musique

Il existe divers moyens d’insérer une vidéo dans une page par exemple avec une balise non standard : <embed> ou la balise standardisée <object> qui permet d’incorporer du contenu dans une page HTML. Ce contenu pouvant être du HTML, ou de la vidéo. Avec l’arrivé du HTML5, une nouvelle méthode se développe avec l’utilisation de la balise <video>, cette balise permet une manipulation plus aisée de la vidéo.

Du fait que certaines balises puissent ne pas être interprétées, on peut se retrouver dans le cas d’une imbrication de ces balises afin que le navigateur soit capable de la lire. Ce type d’imbrication, pouvant devenir vite compliqué, ne simplifie pas le cas de la manipulation de vidéo. Par contre, avec le système de balayage expliqué dans une partie précédente il est facile de lancer une vidéo, qui nécessite une « activation » manuelle. En effet il est fréquent que les vidéos se lance automatiquement sous des sites comme youtube.

### Copié/Collé

L’utilisation d’un objet propre à Firefox devait permettre cette manipulation. Nous avons testé un code disponible sur le site [www.developer.mozilla.org](http://www.developer.mozilla.org) qui semble ne pas fonctionner. Après de plus ample recherche il semble que Firefox bloque cette fonctionnalité dans les dernières des raisons pour cause de sécurité.

### Impression

Il est possible de faire l’impression d’une page via JavaScript ou encore via le balayage écran.

## Technologie pour la mise en œuvre du plug-in

### XUL

XUL est un langage permettant la mise en œuvre d’une interface graphique, au même titre que le HTML. A titre de comparaison.

XUL

<window xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">

<box>

<description>Hello world!</description>

</box>

</window>

HTML

<html>

<body>

<p>

Hello World!

</p>

</body>

</html>

### JavaScript

L’utilisation de JavaScript permet d’amener le dynamisme à XUL au même titre qu’il le fait pour le HTML.

### Ajax

Dans le cadre de ce projet, l’Ajax est employé afin d’exécuter des scripts coté serveur. Afin d’éviter une mise en application compliqué du JavaScript nous utilisons la librairie JQuery.

## Mise en œuvre

Le temps étant assez court pour la réalisation du projet, l’ergonomie du clavier n’a pas été étudiée en profondeur pendant le développement puisque ce n’était pas notre objectif premier. Nous voulions délivrer un prototype avec le maximum de fonctionnalités indispensables à la navigation (liens, favoris, navigation dans la page et dans les onglets, permettre de fermer une fenêtre pop-up).

### Interface du prototype

La page web se présente de la façon suivante :

* A gauche de la page se trouve le clavier de navigation et à droite la page web concernée
* Le balayage des boutons s’effectue de gauche à droite et le clic devra être effectué dès lors que le bouton associé à la fonction souhaitée sera rouge

/////////////////////////////////////////////////////////////////////////////////////////////////

Apercu de la page d’accueil

/////////////////////////////////////////////////////////////////////////////////////////////////

Le bouton onglet va nous permettre de naviguer dans les onglets de la fenêtre courante.

/////////////////////////////////////////////////////////////////////////////////////////////////

Apercu de onglet

/////////////////////////////////////////////////////////////////////////////////////////////////

Les boutons ‘gauche’ et ‘droite’ sont respectivement pour naviguer vers l’onglet de gauche ou de droite, lorsque l’onglet à l’extrémité droite est rencontré, l’onglet suivant sera le premier et vice versa. Il est également possible d’ouvrir ou de fermer un onglet via les boutons ‘ouvrir’ ou ‘fermer’. L’ouverture s’effectuera à la suite des onglets il faudra malgré l’ouverture, naviguer vers ce dernier.

Le bouton ‘clavier’ va comme son nom l’indique nous présenter le clavier du plug-in. Celui-ci est une simple image de clavier AZERTY sur laquelle un numéro est associé à chaque caractère. Le clavier numérique en-dessous de l’image permettra à l’utilisateur de sélectionner le caractère souhaité.

/////////////////////////////////////////////////////////////////////////////////////////////////

Apercu de onglet

/////////////////////////////////////////////////////////////////////////////////////////////////

Le bouton ‘favoris’ permettra bien évidemment d’avoir accès aux boutons correspondant aux pages sélectionnées comme favoris par l’utilisateur. Le bouton aura comme libellé le nom du site et le clic aura pour action d’ouvrir le site dans un nouvel onglet. Il est aussi possible de rajouter ou de supprimer un favori de la liste et du fichier par la même occasion. Le fichier est stocké dans les fichiers temporaires en locale.

/////////////////////////////////////////////////////////////////////////////////////////////////

Apercu de favoris

/////////////////////////////////////////////////////////////////////////////////////////////////

Le bouton lien va permettre la coloration et la numérotation de tous les liens visibles ou non de la page ainsi que l’apparition d’un clavier numérique qui permettra la sélection du lien souhaité.

/////////////////////////////////////////////////////////////////////////////////////////////////

Apercu de liens

/////////////////////////////////////////////////////////////////////////////////////////////////

Le bouton ‘balayage ‘ est utile dans le cas où l’utilisateur souhaiterait cliquer à un endroit précis de la page ou sur l’écran le choix devra se faire dans le menu qui s’ouvre. Le balayage sur la page permettra par exemple de manipuler une vidéo alors que le balayage écran permettra lui de fermer une fenêtre système qui se serait ouverte pendant la navigation.

/////////////////////////////////////////////////////////////////////////////////////////////////

Apercu debalayage

/////////////////////////////////////////////////////////////////////////////////////////////////

### Fonctionnement du prototype

### Le CGI (Common Gateway Interface)

### Relation prototype/CGI

## Tests utilisateurs

## Conclusion