

inlb

Institut Nazareth
& Louis-Braille



WAI-ARIA : nouvelle frontière de l'accessibilité



Walter de Abreu Cybis, Dr. Ing.
Jean-Marie D'Amour, M. Ed.

23 nov. 2012

L'Institut Nazareth et Louis-Braille

- § Seul centre de réadaptation spécialisé uniquement en déficience visuelle au Québec
- § Existe depuis plus de 150 ans



Notre mission

§ Contribuer de façon avant-gardiste à développer l'autonomie et à favoriser la pleine participation sociale de la personne ayant une déficience visuelle.



Pour l'accessibilité du Web



- § Nous avons une approche multi-clientèles, pour toutes les personnes handicapées **ou non**
- ▶ Plusieurs de nos clients ont plus d'une limitation
 - ▶ Il serait contreproductif de travailler pour une seule clientèle

Test des agents utilisateurs

Prise en charge de la spécification WAI-ARIA

1. Introduction
2. Les tests
3. Conclusions
4. Questions

Test des agents utilisateurs

Introduction



Spécification WAI-ARIA

WAI – Web Accessibility Initiative

ARIA - Accessible Rich Internet Applications

Ensemble de déclarations concernant les rôles et les attributs des composants dynamiques enrichis des pages Web (à ajouter au code HTML)

è Mieux informer l'utilisateur sur la sémantique des contenus dynamiques des pages Web

è Lui permettre d'interagir avec ces contenus à l'aide du clavier

Test des agents utilisateurs

Introduction



Pourquoi WAI-ARIA?

Dans les applications Web, les techniques HTML pour l'accessibilité ne suffisent plus.

- è Pour des interfaces enrichies, il faut une information sémantique enrichie

Deux clientèles sont particulièrement visées

- è Les utilisateurs d'un lecteur d'écran

- è Les personnes qui naviguent au clavier

Test des agents utilisateurs

Introduction

Spécification WAI-ARIA

Lecteur d'écran

Exemple : bouton à glissoire

§ role="slider"
§ aria-valuemin="0"
§ aria-valuemax="100"
§ aria-valuenow="50"

§ modifie son comportement
§ annonce aux utilisateurs
qu'ils sont en train
d'interagir avec un bouton à
glissoire qui est à la moitié
de sa valeur maximale

Test des agents utilisateurs

Introduction

Spécification WAI-ARIA

72 rôles en 4 catégories

- § abstraits
- § composants d'interface
- § structure des documents
- § repères

32 attributs en 3 catégories

- § composants d'interface
- § glisser-déposer
- § relations.

Javascripts

- § navigation au clavier
- § contrôle du focus
- § priorité des boîtes de dialogue modales
- § transformations dynamiques

Test des agents utilisateurs

Introduction



Spécification WAI-ARIA

Ce travail n'est pas toujours simple!

Le résultat n'est pas optimal ni homogène!

Les agents utilisateur sont plus ou moins performantes à rendre accessible une interface dynamique

è L'équipe de l'accessibilité des technologies à l'INLB a réalisé une batterie de tests pour connaître plus en détail cette réalité...

Test des agents utilisateurs

Les tests



1. Sélection des cas de test
2. Montage d'un banc d'essai
3. Définition des rôles et attributs à tester
4. Définition des critères de réussite
5. Réalisation des tests
6. Analyse des résultats

Test des agents utilisateurs

1. Sélection des cas de tests

- ▶ OpenAjax Alliance- Accessibility [1]
- ▶ iCita - Illinois Center for Information Technology and Web Accessibility [2].

1. zone de liste déroulante (Combobox)
2. panneau d'onglet (Tabpanels),
3. accordéon
4. barre de menu
5. repères (landmarks)
6. barre d'outils (toolbar)
7. arborescence (tree)
8. tableau interactif (grid)
9. bulle d'aide (tooltip)
10. région déployable (expandble)
11. région vive (live region)

Test des agents utilisateurs

1. Sélection des cas de tests

[-] Crust

Select Crust

Deep Dish

Thick and cheesy

Thick and spicy

Thin

[+] Veggies

[+] Carnivore

[+] Delivery

Test des agents utilisateurs

2. Montage d'un banc d'essai

- ▶ PC (Windows XP et Windows 7)
- ▶ Mac Mountain Lion
- ▶ iOS 6

- § NVDA
- § JAWS 14
- § JAWS 13
- § JAWS 12
- § JAWS 11
- § VoiceOver

- § FireFox
- § Internet Explorer 8
- § Internet Explorer 9
- § Safari – Mountain Lion
- § Safari - iOS

Test des agents utilisateurs

3. Rôles et attributs testés

Rôles testés

§ application	§ tree, treeitem
§ combobox	§ group
§ tab et tablist	§ region
§ navigation	§ grid
§ menu, menubar, menuitem ...	§ button
§ tab, tablist, tabpanel	§ toolbar
§ landmark	§ timer
	§ log
	§ presentation

Test des agents utilisateurs

3. Rôles et attributs testés

Attributs (et valeurs) testés

- § autocomplete (inline)
- § expanded (true/false)
- § activedescendent
(value)
- § hidden (true/false)
- § controls (value)
- § checked (true/false)
- § pressed (true/false)
- § live (assertive/polite)
- § atomic (true/false)
- § relevant (option)

Test des agents utilisateurs

4. Critères de réussite

52 critères de réussite

Exemple

#	Rôles et attributs	Critère de réussite
1	role=application	1- l'agent utilisateur annonce "Application" lorsque l'utilisateur se déplace sur un composant de la zone d'application (avec les flèches, la touche de tabulation ou de navigation rapide); 2- l'interface réagit comme application, i.e. que les raccourcis clavier des lecteurs d'écran en mode navigation cessent de fonctionner et que l'utilisateur navigue avec les raccourcis au clavier prévus par le concepteur des exemples testés;
2	role=combobox	1- l'agent utilisateur annonce "zone de liste déroulante" ou "zone de liste déroulante avec édition" lorsque l'utilisateur se déplace sur la liste avec les flèches, la touche de tabulation ou de navigation rapide
3	aria-autocomplete="inline"	1- l'agent utilisateur annonce que le champ fait de l'auto-complétion; 2- l'agent utilisateur annonce le nom de l'option correspondant à une saisie initiée.
4	aria-expanded="true/false" (role="combobox")	1- l'agent utilisateur annonce que le composant est ouvert ou fermé (ou dans des termes équivalents...)
5	aria-selected="true/false"	1- l'agent utilisateur lit l'option "l'index" sélectionnée (qui a le focus des actions) dès le début.
6	role=application	1- l'agent utilisateur dit "Application" lorsqu'on est sur un composant de la zone d'application (avec les flèches, la touche de tabulation ou de navigation rapide); 2- l'interface réagit comme application, i.e. que les raccourcis clavier des lecteurs d'écran en mode navigation cessent de fonctionner et que l'utilisateur navigue avec les touches prévues par le concepteur des exemples testés
7	role="tablist" + role="tab"	1- l'agent utilisateur annonce "onglet" ou une expression équivalente ("Tab"); 2- l'agent utilisateur annonce combien d'onglets existent et quelle est la position de l'onglet sélectionné;
8	aria-hidden	1- l'agent utilisateur n'annonce pas les composants cachés
9	role="tab" + aria-controls + role="tabpanel"	1- l'agent utilisateur présente le bon panneau
10	role="navigation"	1- l'utilisateur peut se déplacer directement à la zone avec les touches de commande des lecteurs d'écran; 2- l'agent utilisateur annonce la zone lorsqu'on accède à un de ses composants la première fois (par les flèches et par la commande); 3- l'agent utilisateur affiche la zone dans les listes de repères (landmarks)
11	role="search"	idem ci-dessus
12	role="button"	idem ci-dessus
13	role="form"	idem ci-dessus
14	role="document"	idem ci-dessus
15	role="contentinfo"	idem ci-dessus
16	role="menuitem"	l'agent utilisateur annonce "barre de menu"
17	role="menu"	1- l'agent utilisateur annonce "menu" ou "sous-menu" et son nom (et vice et versa); 2- l'agent utilisateur annonce sa position relative dans la barre de menu;
18	role="menubar"	1- l'agent utilisateur annonce "option de menu"; 2- l'agent utilisateur annonce son nom (et vice et versa); 3- l'agent utilisateur annonce sa position relative dans le groupe;
19	aria-checked	1- l'agent utilisateur annonce l'état coché ou non coché d'une option;
20	role="menuradio" et aria-checked	1- l'agent utilisateur annonce le nom de l'option son type "option de menu radio" et son état coché ou non coché;
21	role="menucheckbox" et aria-checked	1- l'agent utilisateur annonce le nom de l'option son type "option de menu checkbox" et son état coché ou non coché;
22	role="tree"	1- l'agent utilisateur annonce "arborescence" et son nom dès qu'on y déplace la première fois sur n'importe quel élément de l'arborescence;
23	role="treeitem"	1- l'agent utilisateur annonce "item d'arborescence" et son nom; 2- l'agent utilisateur annonce son niveau dans l'arborescence; 3- l'agent utilisateur annonce sa position relative dans le groupe; 4- lorsqu'on arrive sur l'arborescence et que l'item a déjà le focus, on annonce le nom et le type de l'item ainsi que sa position dans l'arborescence et le nom de la branche à laquelle il appartient;
24	role="group"	1- l'agent utilisateur annonce un groupe d'arborescence; 2- l'agent utilisateur annonce combien d'éléments sort dans le groupe;
25	aria-expanded="true/false" et role="group"	1- l'agent utilisateur annonce le nom du groupe et son état "ouvert" ou « fermé »
26	role="region"	1- l'agent utilisateur lit le texte de la région;
27	role="grid"	1- l'agent utilisateur annonce « tableau » et son nom;
28	role="button"	1- l'agent utilisateur annonce un bouton et son état sélectionné;
29	role="presentation"	1- l'agent utilisateur n'annonce pas la sémantique native du composant, dans ce cas, une case à cocher (input type="checkbox")
30	aria-checked	1- l'agent utilisateur annonce l'état coché ou non coché de l'élément
31	role="button"	1- l'agent utilisateur annonce la bulle d'aide et lit son contenu;
32	aria-hidden	1- l'agent utilisateur n'annonce pas la bulle lorsqu'elle est repliée (lecture en continu);
33	aria-required	1- l'agent utilisateur annonce que l'entrée est obligatoire;
34	role="button"	1- l'agent utilisateur annonce « barre d'outils »
35	aria-labelledby	1- l'agent utilisateur lit le libellé;
36	aria-pressed	1- l'agent utilisateur lit l'état pressé de l'objet;
37	role="timer" / aria-live="assertive" / aria-atomic="true" / aria-relevant="all"	1- l'agent utilisateur lit le contenu dynamique présenté dans les champs; 2- l'agent utilisateur annonce le type de dynamique (ajout ou suppression)
38	role="log" / aria-live="polite" / aria-atomic="true" / aria-relevant="all"	1- l'agent utilisateur lit le contenu dynamique ajouté dans les champs; 2- l'agent utilisateur annonce les suppressions

1 – l'agent utilisateur annonce "Application" lorsque l'utilisateur se rend sur un composant de la zone d'application (en arrivant avec les flèches, la touche de tabulation ou de navigation rapide);

2 - l'interface réagit comme application, i.e. les raccourcis clavier des lecteurs d'écran en mode navigation cessent de fonctionner et l'utilisateur navigue avec les raccourcis au clavier prévus par le concepteur des exemples testés;

Test des agents utilisateurs

4. Critères de réussite

52 critères de réussite

#	Rôles et attributs	Critère de réussite
1	role=application	1 - l'agent utilisateur annonce "Application" lorsque l'utilisateur se déplace sur un composant de la zone d'application (avec les flèches, la touche de tabulation ou de navigation rapide); 2 - l'interface réagit comme application, i.e. que les raccourcis clavier des lecteurs d'écran en mode navigation cessent de fonctionner et que l'utilisateur navigue avec les raccourcis au clavier prévus par le concepteur des exemples testés
2	role=combobox	1 - l'agent utilisateur annonce "zone de liste déroulante" ou "zone de liste déroulante avec édition" lorsque l'utilisateur se déplace sur la liste avec les flèches, la touche de tabulation ou de navigation rapide
3	aria-autocomplete="inline"	1 - l'agent utilisateur annonce que le champ fait de l'auto-complétion; 2 - l'agent utilisateur annonce le nom de l'option correspondant à une saisie initiée.
4	aria-expanded="true/false" (role="combobox")	1 - l'agent utilisateur annonce que le composant est ouvert ou fermé (ou dans des termes équivalents...)
5	aria-activedescendant	1 - l'agent utilisateur lit l'option "liens" sélectionnée (qui a le focus des actions) des le début
6	role=application	1 - l'agent utilisateur dit "Application" lorsqu'on est sur un composant de la zone d'application (avec les flèches, la touche de tabulation ou de navigation rapide); 2 - l'interface réagit comme application, i.e. que les raccourcis clavier des lecteurs d'écran en mode navigation cessent de fonctionner et que l'utilisateur navigue avec les touches prévues par le concepteur des exemples testés
7	role="tablist" + role="tab"	1 - l'agent utilisateur annonce "onglet" ou une expression équivalente ("Tab"); 2 - l'agent utilisateur annonce combien d'onglets existent et quelle est la position de l'onglet sélectionné.
8	aria-hidden	1 - l'agent utilisateur n'annonce pas les composants cachés
9	role="tab" + aria-controls + role="tabpanel"	1 - l'agent utilisateur présente le bon panneau
10	role="navigation"	1 - l'utilisateur peut se déplacer directement à la zone avec les touches de commande des lecteurs d'écran; 2 - l'agent utilisateur annonce la zone lorsqu'on accède à un de ses composants la première fois (par les flèches et par la commande); 3 - l'agent utilisateur affiche la zone dans les listes de repères (landmarks)
11	role="search"	idem ci-dessus
12	role="banner"	idem ci-dessus
13	role="main"	idem ci-dessus
14	role="document"	idem ci-dessus
15	role="contentinfo"	idem ci-dessus
16	role="menubar"	l'agent utilisateur annonce "barre de menu"
17	role="menu"	1 - l'agent utilisateur annonce "menu" ou "sous-menu" et son nom (et vice et versa); 2 - l'agent utilisateur annonce sa position relative dans la barre de menu;
18	role="menuItem"	1 - l'agent utilisateur annonce "option de menu"; 2 - l'agent utilisateur annonce son nom (et vice et versa); 3 - l'agent utilisateur annonce sa position relative dans le groupe;
19	aria-checked	1 - l'agent utilisateur annonce l'état coché ou non coché d'une option;
20	role="menuradio" et aria-checked	1 - l'agent utilisateur annonce le nom de l'option son type "option de menu radio" et son état coché ou non coché
21	role="menuitemcheckbox" et aria-checked	1 - l'agent utilisateur annonce le nom de l'option son type "option de menu checkbox" et son état coché ou non coché
22	role="tree"	1 - l'agent utilisateur annonce "arborescence" et son nom dès qu'on s'y déplace la première fois sur n'importe quel élément de l'arborescence;
23	role="treeitem"	1 - l'agent utilisateur annonce "item d'arborescence" et son nom; 2 - l'agent utilisateur annonce son niveau dans l'arborescence; 3 - l'agent utilisateur annonce sa position relative dans le groupe; 4 - lorsqu'on arrive sur l'arborescence et que l'item a déjà le focus, on annonce le nom et le type de l'item ainsi que sa position dans l'arborescence et le nom de la branche à laquelle il appartient
24	role="group"	1 - l'agent utilisateur annonce un groupe d'arborescence; 2 - l'agent utilisateur annonce combien d'éléments sont dans le groupe;
25	aria-expanded="true/false" et role="group"	1 - l'agent utilisateur annonce le nom du groupe et son état « ouvert » ou « fermé »
26	role="region"	1 - l'agent utilisateur lit le texte de la région
27	role="grid"	1 - l'agent utilisateur annonce « tableau » et son nom;
28	role="button"	1 - l'agent utilisateur annonce un bouton et son état sélectionné
29	role="presentation"	1 - l'agent utilisateur n'annonce pas la sémantique native du composant, dans ce cas, une case à cocher (input type="checkbox")
30	aria-checked	1 - l'agent utilisateur annonce l'état coché ou non coché de l'élément
31	role="button"	1 - l'agent utilisateur annonce la bulle d'aide et si son contenu
32	aria-hidden	1 - l'agent utilisateur n'annonce pas la bulle lorsqu'elle est repliée (reclure en continu)
33	aria-required	1 - l'agent utilisateur annonce que l'entrée est obligatoire
34	role="button"	1 - l'agent utilisateur annonce « faire d'outil »
35	aria-labelledby	1 - l'agent utilisateur lit le libellé
36	aria-pressed	1 - l'agent utilisateur lit l'état pressé de l'objet
37	role="timer" + aria-live="assertive" + aria-atomic="true" + aria-relevant="all"	1 - l'agent utilisateur lit le contenu dynamique présent dans les champs; 2 - l'agent utilisateur annonce le type de dynamique (ajout ou suppression)
38	role="log" + aria-live="polite" + aria-atomic="true" + aria-relevant="all"	1 - l'agent utilisateur lit le contenu dynamique ajouté dans les champs; 2 - l'agent utilisateur annonce les suppressions

Exemple

1 – l'utilisateur peut ouvrir et de fermer une zone de liste déroulante à l'aide du clavier;
2 – l'utilisateur doit être capable de sélectionner les items au moyen des touches d'édition et des flèches du clavier

Test des agents utilisateurs

5. Réalisation des tests



Exemples

Démonstration de la Barre d'outils

Test des agents utilisateurs

5. Résultats

Positions	Agents utilisateur		Performance (notes sur 100)
	Lecteurs d'écran	Navigateurs	
1	NVDA	FF	95,6
2	JAWS 14	FF	78,1
3	JAWS 13	FF	77,0
4	VO	Safari - M. Lion	73,6
5	JAWS 12	FF	72,4
6	JAWS 13	IE8	69,8
7	JAWS 14	IE9	69,5
8	JAWS 14	IE8	68,6
9	JAWS 12	IE8	67,0

Test des agents utilisateurs

5. Résultats

Positions	Agents utilisateur		Performance (notes sur 100)
	Lecteurs d'écran	Navigateurs	
10	JAWS 13	IE9	67,6
11	NVDA	IE9	63,0
12	NVDA	IE8	63,0
13	JAWS 12	IE9	60,8
14	VO	Safari-iOS 5	58,3
15	JAWS 11	FF	57,3
16	JAWS 11	IE8	45,1
17	JAWS 11	IE9	41,2
Moyenne générale			66,3

Test des agents utilisateurs

5. Résultats

Lecteurs d'écran	Performance (notes sur 100)
NVDA	73,9
JAWS 14	72,1
JAWS 13	71,5
JAWS 12	66,7
VO	65,9

Navigateurs	Performance (notes sur 100)
FF	75,6
Safari - M.Lion	65,9
IE8	61,2
IE9	58,2

Test des agents utilisateurs

5. Résultats

Composante d'interface	Performance moyenne (notes sur 100)	Pires performances (note <50,0%)
Bulle d'aide (Tooltip)	92,1	Aucune
Accordéon	86,4	Jaws11-IE8/IE9
Repères (Landmark)	74,3	Jaws11-IE8/IE9/FF
Barre d'outils	72,9	NVDA-IE8/IE9
Panneau d'onglet	69,5	Jaws11/12-IE8/IE9

Test des agents utilisateurs

Conclusions

§ Les solutions parfaites se font rares !

100 %



50 %



0 %



Cas de test – rôles et attributs WAI-ARIA

Agents utilisateurs

comparant.html	Notes par attribut	NIDA 7,29			JAWS 11 4,70			JAWS 12 6,47			JAWS 13 7,15			JAWS 14 7,21			VO 6,59		
		IE8	IE9	FF	IE8	IE9	FF	IE8	IE9	FF	IE8	IE9	FF	IE8	IE9	FF	Safari-Linux	Safari-iOS5	
Noter global:		6,30	6,30	9,86	4,81	4,12	5,73	6,70	6,08	7,24	6,98	6,76	7,70	6,84	6,95	7,81	7,36	5,83	
Test 1		5,7	7,2	7,2	10	6	2	5,2	4	7,2	8	4	4	8,8	2	2	10	7,2	2,4
role="combobox"	Crit	5,3	10	10	10	4	0	4	10	6	0	0	10	0	0	10	10	10	10
aria-autocomplete="inline"		2,9	6	6	10	6	0	6	0	6	6	0	0	4	0	0	10	6	0
aria-expanded	Crit	4,1	0	0	10	10	0	10	0	10	10	0	0	10	0	0	10	0	0
recevoir clavier Critère de réussite		7,1	10	10	10	0	0	0	10	0	8	10	10	10	10	10	10	10	2
aria-activedescendant		8,2	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	0
Test 2		7,3	6,0	6,0	10,0	6,0	6,0	7,0	6,0	6,0	7,3	7,5	7,5	8,0	9,0	9,0	9,0	8,5	5,0
role="application"	C	5,1	6	6	10	4	4	2	4	4	2	4	4	2	10	10	10	4	0
role="tablist" * role="tab"		7,4	8	8	10	6	6	6	6	6	10	6	6	10	6	6	6	10	10
aria-hidden	Crit	8,2	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
recevoir clavier Critère de réussite		7,9	10	10	10	4	4	10	4	4	8	10	10	10	10	10	10	10	8
Test 3		8,8	9,0	9,0	10,0	6,0	6,0	10,0	6,0	7,0	10,0	9,0	9,0	10,0	9,0	10,0	10,0	10,0	10,0
role="tab" * aria-controls * role="tablist"		9,1	8	8	10	8	8	10	8	10	10	8	8	10	8	10	10	10	10
recevoir clavier Critère de réussite		8,6	10	10	10	4	4	10	4	4	10	10	10	10	10	10	10	10	10
Test 4		7,9	10,0	10,0	10,0	0,0	0,0	0,0	8,0	8,0	8,0	10,0	10,0	10,0	10,0	10,0	10,0	10,0	10,0
role="navigation"	Crit	7,9	10	10	10	0	0	0	8	8	8	10	10	10	10	10	10	10	10
role="arch"	Critère de	7,9	10	10	10	0	0	0	8	8	8	10	10	10	10	10	10	10	10
role="banner"	Critère de	7,9	10	10	10	0	0	0	8	8	8	10	10	10	10	10	10	10	10
role="main"	Critère de	7,9	10	10	10	0	0	0	8	8	8	10	10	10	10	10	10	10	10
role="document"	Crit	7,9	10	10	10	0	0	0	8	8	8	10	10	10	10	10	10	10	10
role="contentinfo"	Crit	7,9	10	10	10	0	0	0	8	8	8	10	10	10	10	10	10	10	10
Test 5		5,2	5,4	5,4	7,1	4,9	2,9	4,2	4,9	4,9	6,3	6,3	4,9	6,3	4,9	4,9	6,2	5,4	4,0
role="menubar"	Critère	2,9	10	10	10	0	0	0	10	0	10	10	10	10	10	10	10	10	0
role="menu"	Critère de	6,2	8	8	10	6	0	0	6	6	6	6	6	6	6	6	6	10	10
role="menubutton"	Critère	7,4	10	10	10	8	0	0	8	8	8	8	8	8	8	8	8	8	0
aria-checked	Critère	8,2	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	0
role="menubutton" et aria-checked		0,6	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
role="menubutton" et aria-checked		0,6	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
recevoir clavier Critère de réussite		10,0	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
Test 6		5,7	5,2	5,2	10	5,2	6	8	6,8	4,8	8	2	2	7,6	7,6	7,6	7,6	2,8	1,2
role="tree"	Critère	4,1	6	6	10	8	10	8	10	8	10	10	10	8	8	8	8	8	0
role="treeitem"	Crit	4,7	10	10	10	0	2	2	10	6	4	10	0	8	8	8	8	2	2
role="group"	Critère	4,1	0	0	10	0	0	10	0	0	10	0	0	10	10	10	10	0	0
role="group" aria-expanded="true"		7,8	10	10	10	10	10	10	10	10	10	0	0	10	10	10	10	2	0
recevoir clavier Critère de réussite		8,0	10	10	10	8	8	10	8	8	10	0	0	10	10	10	10	10	4
Test 7		4,4	10,0	10,0	10,0	0,0	0,0	0,0	8,0	8,0	0,0	8,0	8,0	0,0	0,0	0,0	0,0	10,0	10,0
role="region" aria-expanded="true"		4,4	10	10	10	0	0	0	8	8	0	8	8	0	0	0	0	10	10
Test 8		4,4	4,0	4,0	10,0	2,0	2,0	4,0	2,0	4,0	4,0	4,0	4,0	4,0	3,0	3,0	6,0	4,0	0,0
role="grid"	C	4,1	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	0
role="button"	C	8,2	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	0
role="presentation"		4,1	0	0	10	0	0	10	0	0	10	0	0	10	10	10	10	0	0
aria-checked		1,8	10	10	10	0	0	0	0	0	0	0	0	10	10	10	10	0	0
recevoir clavier Critère de réussite		2,9	0	0	10	0	0	0	0	0	0	0	0	10	10	10	10	10	0
Test 9		9,4	10,0	10,0	10,0	5,0	7,5	10,0	10,0	10,0	10,0	10,0	10,0	10,0	10,0	10,0	10,0	10,0	6,5
role="tablist"	C	8,2	10	10	10	0	10	10	10	10	10	10	10	10	10	10	10	10	0
aria-hidden	C	10,0	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
aria-pressed		8,6	10	10	10	0	0	10	10	10	10	10	10	10	10	10	10	10	6
recevoir clavier Critère de réussite		10,0	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
Test 10		7,8	2,5	2,5	10,0	6,5	6,0	6,5	10,0	8,0	10,0	10,0	8,0	10,0	10,0	10,0	10,0	5,0	7,0
role="toolbar"		6,1	0	0	10	0	0	0	10	0	10	10	10	10	10	10	10	5	0
aria-labelledby Critère de réussite		8,2	0	0	10	10	0	10	10	0	10	10	0	10	10	10	10	5	10
aria-pressed Critère de réussite		7,9	10	10	10	10	0	10	10	8	10	10	10	10	10	10	10	10	10
recevoir clavier Critère de réussite		8,9	10	10	10	6	6	6	10	8	10	8	10	8	10	10	10	10	8
Test 11		6,8	0,0	0,0	8,0	8,0	7,0	8,0	8,0	7,0	8,0	6,0	7,0	10,0	5,0	5,0	7,0	8,0	8,0
role="timer" aria-live="assertive" aria-atomic="true"		6,9	0	0	8	8	8	8	8	8	8	6	8	8	6	6	10	8	8
role="img" aria-label="palette" aria-activedescendant="img1"		6,0	0	0	8	8	6	8	8	6	8	6	8	6	10	4	4	8	8

Test des agents utilisateurs

Conclusions



- § Les résultats suggèrent aux personnes aveugles et malvoyants d'utiliser NVDA, FireFox et NVDA-FireFox.

- è Étude portant sur un échantillon limité de rôles et attributs WAI-ARIA
 - ▶ 27 rôles et 15 attributs
 - ▶ 11 cas de tests
 - ▶ 17 agents utilisateurs
 - ▶ À une date donnée

Test des agents utilisateurs

Conclusions



- § WAI-ARIA est un outil puissant et indispensable pour l'ergonomie des interfaces Web enrichies
 - ▶ Pour les utilisateurs du clavier et
 - ▶ Pour les utilisateurs d'un lecteur d'écran

Test des agents utilisateurs

Conclusions



§ WAI-ARIA est encore loin d'une prise en charge fiable et cohérente

- ▶ D'un navigateur à l'autre
- ▶ D'un lecteur d'écran à l'autre
- ▶ D'un composant d'interface à l'autre

Test des agents utilisateurs

Conclusions



- § WAI-ARIA est utilisable dès maintenant et indispensable pour l'adaptation d'applications Web enrichies dans un environnement contrôlé
 - ▶ Avec un navigateur connu et
 - ▶ Avec un lecteur d'écran connu

Test des agents utilisateurs

Conclusions



§ WAI-ARIA n'est pas prêt pour une utilisation sur un site Web public à moins de l'utiliser

- ▶ Comme enrichissement dans une approche d'amélioration progressive
- ▶ Pour certains composants spécifiques largement pris en charge

OU

- ▶ Avec une version de rechange basée sur l'utilisation du HTML standard

Fin

Merci!

La parole est à vous...



Institut Nazareth
& Louis-Braille

VOTRE référence en déficience visuelle

Centre de réadaptation en déficience visuelle agréé par le Conseil québécois d'agrément