

AJAX Live Regions & Accessibility

by Charles L. Chen

Overview

- Introduction
- WAI-ARIA Live Region Markup
- Examples and Demos with Fire Vox
- Future Work
- Additional Resources
- Questions and Answers

Introduction

- What is AJAX?
 - AJAX: Aynchronous JavaScript And XML
 - AJAX allows for dynamic updates to a web page without doing a refresh
 - Real world events: headlines, stocks, game stats
 - Response to user actions: warnings, prompts

Introduction

- Why is AJAX a problem for accessibility?
 - What is a web page?
 - Traditional model: Static document
 - AJAX model: Dynamic application
 - Should users be informed of the change, and if so, when?
 - Some changes may be trivial and should not be spoken at all
 - Some changes should be spoken immediately; others should only be spoken when convenient

Introduction

- How can we solve the “AJAX” problem?
 - Strictly speaking, AJAX is not the problem; the problem is dynamic content in live regions
 - The question: How should AT handle a change to part of a web page?
 - The answer: It depends on what that part of the web page is.

Introduction

- How can we solve the “AJAX” problem?
 - Certain types of changes happen to certain parts of the web page
 - Game stat updates will happen in the part of the web page that is for displaying game stats; headline updates will happen in a different part
 - JavaScript code usually refers to a particular region (div, span, etc.) by its ID and adds/removes/changes content there

Introduction

- How can we solve the “AJAX” problem?
 - Who knows the purpose of each part of the page better than the web developer?
 - Users should have full control over how live regions behave
 - Web developers should be empowered with setting reasonable defaults for the live regions on their pages

WAI-ARIA Live Region Markup

- WAI-ARIA
 - Web Accessibility Initiative
 - Accessible Rich Internet Applications
 - W3C
- Markup live regions with information about how they should be treated by AT
 - These are the defaults; users can override these settings if they choose

WAI-ARIA Live Region Markup

- `live=POLITENESS_SETTING`
- `atomic=BOOLEAN`
- `relevant=[LIST_OF_CHANGES]`
- `controls=[IDLIST]`
- `labelledby=[IDLIST]`
- `describedby=[IDLIST]`

live=POLITENESS_SETTING

- off (default)
 - Do not speak this region
- polite
 - Speak this region when the user is idle
- assertive
 - Speak this region as soon as possible
- rude
 - Speak this region RIGHT NOW

atomic=BOOLEAN

- true
 - This whole region must be spoken when any of its nodes are changed; the individual changes cannot stand on their own
- false (default)
 - Speak only the node that changes; there is enough context for the individual changes to make sense

relevant=[LIST_OF_CHANGES]

- additions (default)
 - Nodes added to the region are important and should be spoken
- removals
 - Nodes removed from the region are important and should be spoken
- text (default)
 - Nodes changed in the region are important and should be spoken
- interim*
 - All additions and changes are relevant, even if they were removed/changed before they could be spoken
 - *Note: This is a proposed property and has not been officially accepted yet.

Examples and Demos

- Multiple live regions with different politeness
- Atomic vs non-atomic
- Interim vs non-interim
- Form with warnings
- Scoreboard
- Chatroom

Future Work

- Causality
 - No way to distinguish a change caused by a world event vs a change that is a response to user action
- Higher levels of abstraction for web developers
 - Default live properties for various widgets based on ARIA roles
- Style changes
 - Display/Visibility changes are style changes and are NOT fired as DOM events

Additional Resources

- W3C WAI-ARIA
 - <http://www.w3.org/TR/aria-state/>
- ARIA Live Regions Report
 - http://developer.mozilla.org/en/docs/AJAX:WAI_ARIA_Live_Regions
- Accessible Live Region Test Cases
 - <http://accessibleajax.clcworld.net/>
- Fire Vox
 - <http://firevox.clcworld.net/>

Questions and Answers

- Any questions?

Thank you for your time