ACCESSIBILITY ON DEMAND

The Accessibility Tree - How ARIA Works

CSUN 2015 Presentation: Bryan Garaventa, SSB BART Group, Inc.
Presentation materials download: http://whatsock.com/csun/2015

What is ARIA

ARIA stands for Accessible Rich Internet Applications, and is a technical specification published by the World Wide Web Consortium (W3C) Web Accessibility Initiative (WAI).

The purpose of the ARIA spec is to map simulated web controls to equivalent control types on the platform Accessibility API, and to provide various mechanisms to enhance the accessibility of web technologies.

Operating System vendors, browser vendors, and Assistive Technology vendors, then follow the *ARIA spec to ensure that control mappings are supported properly in each.*

It's important to understand that all ARIA Roles and supporting States and Properties map to their equivalent Roles, States and Properties within the Accessibility API on that Operating System, which is why the same ARIA markup when correctly implemented is valid across all devices and platforms.

How ARIA Works

There are three primary levels that dictate how ARIA works:

1. The platform Accessibility API and Accessibility Tree on the Operating System
2. The Accessibility Tree constructed by browsers
3. How Assistive Technologies interface with the browser Accessibility Tree

The OS Accessibility Tree

All mainstream Operating Systems include an Accessibility API, which is used to define the functional characteristics of common control types within an object based environment, the Graphical User Interface (GUI).

When the GUI is constructed so that users can interact with it, every common control type inherits specific Roles, States and Properties from the Accessibility API, which Assistive Technologies can then use to interface with.

The hierarchy of these common control type objects is known as the Accessibility Tree.

It helps to imagine this as an upside-down tree, where the trunk (at the top) is the OS Accessibility API, branches connected to the trunk are individual software applications, twigs connected to each branch are application view panes, and leaves connected to each twig are individual software controls that can be interacted with.

When a leaf (common control type object) is added to a twig (application view pane), it automatically maps to all of the equivalent Roles, States and Properties for that control type in the trunk (OS Accessibility API).

The Browser Accessibility Tree

When a web page is loaded by a browser, it first builds a Document Object Model (DOM), then, using this information, it builds an Accessibility Tree that maps to the platform Accessibility API.

As the browser tracks DOM changes, these changes are automatically updated in the Accessibility Tree, and events are fired so that Assistive Technologies can listen for and convey these changes appropriately to users.

This is where ARIA comes into play.

When ARIA is used in the DOM, such as setting interactive widget Roles, States and Properties, the browser modifies the Accessibility Tree accordingly, tying previously unmapped or incorrectly mapped elements into the platform Accessibility API so that Assistive Technologies can interface with them.

All of the specifications that browsers must follow in order to do this, are documented in the User Agent Implementation Guide (UAIG).

Front end engineers then use the Roles Model and ARIA Supported States and Properties specifications to synchronize markup design patterns with supported functionality in the UAIG.

All three of these specification documents are normative; meaning they are "required for conformance" by the W3C.

Assistive Technologies and the Accessibility Tree

When spec conformant Assistive Technologies interact with the GUI on the operating system, they are interfacing with objects in the Accessibility Tree using the Roles States and Properties available on each object.

This is true within web technologies as well, where Assistive Technologies interface with the Accessibility Tree constructed by the browser.

Since Accessibility Tree objects on the OS and Accessibility Tree objects within the browser map to the same control types on the platform Accessibility API, Assistive Technologies can interface with objects in both of these environments equally; providing the same level of feedback for related control types.

This is how Assistive Technologies, such as screen readers like JAWS, NVDA and VoiceOver, recognize and interact with ARIA control types and related functionality; by interfacing directly with the Accessibility Tree.