6.2 Keyboard Input (Developing to Exploratory)

Background

- Update 10/8: Meeting on Tuesdays 10:00 Eastern starting 10/15 through to 11/19.
- Update (9/6): Keyboard subgroup meeting via Zoom on Wednesdays 11:00 Eastern starting 8/14.
    - For invite, please email bailey at access hyphen board dot u s
    -
- Process
- Subgroup Handbook
- Keyboard Support Scratchpad
- Consolidated Outcomes
    - **Usable**
        - **Comparable keyboard effort** - The number of input commands required to complete a task using the keyboard is similar to the number of input commands when using other input modalities.
        - **Consistent keyboard interaction** - Keyboard interface interactions are consistent.
        - **Keyboard commands** - Application keyboard commands do not conflict with platform commands, and the user is informed of non-standard commands.
    - **Operable (potential prerequisites)**
        - **Keyboard only** - All functionality can be performed through the keyboard interface only, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.
        - **No keyboard trap** - If keyboard focus can be moved to an interactive component, then the keyboard focus can be moved away from that component.
-

Could we have one outcome that encapsulates Comparable keyboard effort, Consistent keyboard interaction, Keyboard only, and No keyboard trap? “Keyboard only” being the core or umbrella expectation. “Keyboard only” may be a less challenging start. Custom keyboard command and No keyboard trap are the most tangible.

Work specifically on “Keyboard Commands”?

Then brainstorm methods that will give endpoints to a decision tree.

Additional Background

- Closest WCAG 2.x Guideline
  - Guideline 2.1: Make all functionality available from the keyboard.
- Parent category from consolidated outcomes
  - 6 Input / Operation

Outcome: 6.2.1 Comparable keyboard effort

The number of input commands required to complete a task using the keyboard is similar to the number of input commands when using other input modalities.

Goal

The input effort required to complete a task should be equivalent regardless of the input modality being used.

What to do

Complete tasks using only keyboard commands.   Assess the number of commands required to complete tasks using only keyboard commands. Assess the effort needed to complete tasks using other modalities (such as using a mouse). Assess if the level of effort is similar between keyboard and other modality.

User Need(s)

- As a user of a keyboard, the complexity and number of input commands required to complete a task should be comparable with those of a user of other methods of input.
- As a keyboard user, I should be provided with a method of skipping repetitive navigation links
- As a keyboard user, I need to be able to access content in an equivalent way as other input modalities.
- As a user of a keyboard, I should be able to complete actions in a comparable amount of time to other users.
- As a keyboard user, timing should not prevent me from completing actions.

Outcome: 6.2.2 Consistent keyboard interaction

Keyboard interface interactions are consistent.

Goal

The keyboard experience when using web content is what I have come to expect. If the keyboard interaction is not conventional, then I am informed ahead of time about the work-arounds needed

What to do

You would know you have met the outcome if the content requires only use of tab, shift-tab, arrow keys, spacebar, enter, and escape keys for navigation. Alternatively, instructions would detail keyboard use of the content, and would not use keystrokes required by assistive technology.

You would know you have met the outcome if the content requires only the fundamental keyboard conventions. Instructions would detail keyboard use of the content. Keystrokes would not use keystrokes required by assistive technology.

User Need(s)

- As a keyboard user, I always know where I am in the interface.
- As a user of assistive technology, I need my keyboard commands to remain consistent.
- As a keyboard user, I should not get unintentionally stuck.
- If I implement a keyboard device within my site/app, that control must respond correctly to standard keyboard commands.
- As a keyboard user, information on available shortcuts is provided in context and in a way I can perceive.
- As a keyboard user, I shouldn't have my expected keyboard commands disabled.
- As a keyboard user, I am informed of non-standard commands before I need to make use of one
- As a keyboard user, when I reach a custom implementation of a common component, all keyboard commands that exist are implemented

Outcome: 6.2.3 Custom keyboard commands

Application keyboard commands do not conflict with platform commands, and the user is informed of non-standard commands.

Goal

Keyboard interaction including navigation and operation, is intuitive for users familiar with the platform.

What to do

- As an author, any keyboard commands implemented in content do not conflict with keyboard usage of the platform.
- As an author, documentation for any keyboard commands implemented in content are readily available to users.
- An author would know that they have met this outcome if they have not added keyboard shortcut or accelerator keys to their content.

- An author would know that they have met this outcome if all added keyboard shortcuts are documented and readily available in the content.
- Content authors who add keyboard shortcuts must be well familiar with using a platform via only the keyboard.

User Need(s)

For each application keyboard command, there is not an overlap with platform commands.

Strongly related to:

- As a keyboard user, keyboard shortcuts and assistive technologies do not interfere with each other.
- As a keyboard user, the keyboard commands that I need to press don't interfere with my operating system and/or assistive technology
- As a keyboard user, I am informed of non-standard commands before I need to make use of one
- As a keyboard user, information on available shortcuts is provided in context and in a way I can perceive.
- If I implement a keyboard device within my site/app, that control must respond correctly to standard keyboard commands.
- As a keyboard user, when I reach a custom implementation of a common component, all keyboard commands that exist are implemented

Outcome: 6.2.4 Keyboard only

All functionality can be performed through the keyboard interface only, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.

Goal

All user flows can be completed by keyboard only.

What to do

This outcome would be met if you are able to complete all tasks by only the use of keyboard actions for both navigation and input. Exception is permissible where the action of completing a task is not only dependent upon the task having been completed but also requires specific user movement (e.g. requirement that the user must drag an object to follow specified path).

User Need(s)

- 4 As a keyboard user, I should be able to perform all actions using a keyboard exclusively
- 5 As a keyboard user, I should not get unintentionally stuck.
- 7 As a keyboard user, timing should not prevent me from completing actions.
- 9 As a keyboard user, I always know where I am in the interface

- 14 As a keyboard user, I should be provided with a method of skipping repetitive navigation links
- 19 As a keyboard user, I am not expected to use a certain viewport in order to use a keyboard
- 22 As a keyboard user, I should be able to access all scrollable regions of a view/app so I can read all text within these regions
- 23 As a keyboard user, I need to be able to access content in an equivalent way as other input modalities.
- 2 As a user of a keyboard, I should be able to complete actions in a comparable amount of time to other users.
- 3 As a user of a keyboard, the complexity and number of input commands required to complete a task should be comparable with those of a user of other methods of input.
- 1 As a user of assistive technology, I need my keyboard commands to remain consistent.
- 6 As a keyboard user, keyboard shortcuts and assistive technologies do not interfere with each other.
- 8 As a keyboard user, I should not be required to press multiple keys at the same time to complete an action
- 10 As a keyboard user, I am informed of non-standard commands before I need to make use of one
- 11 As a keyboard user, information on available shortcuts is provided in context and in a way I can perceive.
- 12 As a keyboard user, the keyboard commands that I need to press don’t interfere with my operating system and/or assistive technology
- 13 As a keyboard user, my choice of keyboard is fully supported
- 15 As a keyboard user, I would like my user agent to provide keyboard shortcuts to quickly navigate around an interface.
- 20 As a keyboard user, applications should be flexible in allowing me to use the keyboard commands I want to use (e.g. through personalisation)
- 21 As a keyboard user, I shouldn't have my expected keyboard commands disabled.
- 24 As a keyboard user, when I reach a custom implementation of a common component, all keyboard commands that exist are implemented
- 16 If I implement a keyboard device within my site/app, that control must respond correctly to standard keyboard commands.

Decision Tree for 6.2.4 Keyboard only

1. Check if tabbing can cycle through all UIE and return back to where started.
    a. Check if the page has any UIE that can be focused and activated with pointer input but not with the keyboard. If this is the case, the outcome fails.
    b. Interact with each UIE, for example, activate it to change state, go to other pages, execute functions, or provide sample text input. Check if this is possible with the keyboard. If not, the outcome fails.
    c. What is content requirement for exposing interactive features so they are usable via AT? (for example “forms mode”)

        i. May be part of method. Relates to AT as UA.
    d. Shift-tab also needs to be tested
2. For elements that can be activated or followed (execute a link) or submitting a button
    a. Enter or spacebar work to trigger / activate that element
3. For complex interactive elements
    a. Tab list
        i. Can arrows be used for interaction?
    b. Select list (combo boxes)
        i. Can arrows be used for interaction?
    c. Grids (interactive)
    d. Slider - arrow keys vs. using thumbs
        i. Expected behaviour of these controls
    e. Media controls, toggle switches
    f. Radio buttons - expected keyboard commands to select and move on to next area outside of choice
4. Actionable elements that receive focus, ~~can be activated with space bar, enter key or other documented keys~~
    a. Disabled vs. readonly ? Nature of an element involves activation , how is this processed?
        i. Readonly - focus but not editable
        ii. Disabled - wouldn't receive focus
5. Are there UI elements which are responsive to mouse but not keyboard?
    a. Importance of focus and activation
        i. Works with keyboard only is it AT provider need to update a keyboard only issue on vendor side? Does this fail 2.1.1 or 4.1.2 or otherwise?
            1. Dependent on code vs. AT issue.
6. Check viewport at reduced size.
    a. With responsive design, on smaller screens, it is common for navigation elements to collapse into a "hamburger menu". If applicable, can this menu be navigated by keyboard?
    b. One technique for different viewport sizes, which might be otherwise overlooked (even when default viewport size).
7. Focus order also impacts reading order, overlaps with keyboard (2.x 2.4.3 focus order)

Outcome: 6.2.5 No keyboard trap

If keyboard focus can be moved to an interactive component, then the keyboard focus can be moved away from that component.

Goal

Users are able to navigate content using only the keyboard without getting stuck. Users are able to access components and content and identify the exit (or) ways to exit clearly using keyboard-only.

What to do

Ensure that standard keyboard operation may be used for navigating through content, including moving focus from components. When non-standard keyboard operation is needed for navigation or to move focus from components, those are documented before being encountered.

User Need(s)

- As a keyboard user, I should not get unintentionally stuck.
- As a keyboard user, timing should not prevent me from completing actions.
- As a keyboard user, scrolling is not trapped with screen zooming
- As a keyboard user, I should be able to identify the exit from the active component e.g. modal, menu
- As a keyboard user, I could use consistent (standard) keyboard commands to exit out of the active component e.g. 'ESC command to close a modal'
- As a keyboard user, keyboard shortcuts and assistive technologies do not interfere with each other.
- As a keyboard user, I am informed of non-standard commands before I need to make use of one
- As a keyboard user, I shouldn't have my expected keyboard commands disabled.
- As a keyboard user, when I reach a custom implementation of a common component, all keyboard commands that exist are implemented.

Unassigned User Needs

User needs from previous scratchpad that were *not* brought over from the User Need section of older scratchpad. These might belong to another Outcome or Guideline.

- As a keyboard user, I am warned about changes in context before they occur. [17]
- As a keyboard user, I am warned about changes in focus before they occur. [18]
- As a user, I don't want a keyboard shortcut to be the only way to do something. (Or, there should not be only one way to do something.) [25]

Decision Tree for 6.2.5 No keyboard trap

1. Can tab through entire page, not getting stopped on UI element
    a. Even if shift-tab to get out, traditionally this failure under 2.x 2.4.2.focus order
    b. Might count as Keyboard Trap for 3.x
2. With 2.x "key trap" appears only after entering some content.
3. With 2.x, after skipping entry (leave empty), focus resets to top of page or returns focus to element.

Methods

Sample Method Decision Tree (text alternatives for images)

1. Does the organization have a style guide which includes guidance on all types of text-alternatives, and a policy that editors are required to follow the style guide?
   1.1. Yes, See Text Alternative Style Guide Assertion (Enhanced), continue
   1.2. No, continue

2. Would removing the image impact how people understand the page?
   2.1. No, see Decorative Image Methods (Enhanced) [stop]
   2.2. Yes, continue

3. Can a concise and plain-text description provide an equivalent to the image?
   3.1. No, see Complex Image Methods (Baseline) [stop]
   3.2. Yes, continue

4. Is the image a control?
   4.1. Yes, see Image as Control Methods (Baseline) [stop]
   4.2. No, continue

5. Is the image one of a set of images which provide multiple views of the same subject matter?
   5.1. Yes, see Multiple Image Views Methods (Baseline) [stop]
   5.2. No, see Image Description Methods (Baseline)

6. Is the image used for conveying information (e.g. a content image)?
   6.1. See Image Description Method (Baseline) [stop]

7. Any other image scenario
   7.1. See Alternative method [stop]

Method Decision Tree for 6.2.3 Custom keyboard command

6.2.3 Custom keyboard commands (take 2, 8/14)

For each type of action where the content sets a keyboard command.:

1. Does the content owner document unique commands
   a. Yes
   b. No
2. Does the content allow a setting for changing keyboard commands?
   a. Yes “Override content level command”, Stop
   b. No, continue.

3. Does the content allow a setting for disabling all keyboard commands?
    a. Yes [method]
    b. No.
4. Does the platform allow means control over keyboard command.
    a. Yes [method]
    b. No, see Alternative Means

Alternative means:

1. Content does not meet this requirement.
2. Accessibility supported might allow Assistive Technology has a “pass thru” function.
3.

Method list

[Method Name]

Level

[One of Prerequisite/Baseline/Enhanced]

Method

[Description of method, not technology specific]

Techniques

[List of technology/platform specific techniques.]

- ([Platforms]) [Method title]

- (HTML, EPUB) Using alt attributes so the image is ignored. - Example

Override Platform Level Command

Level

[One of Prerequisite/Baseline/Enhanced]
Baseline

Method

[Description of method, not technology specific]

Techniques

[List of technology/platform specific techniques.]

- ([Platforms]) [Method title]
- (HTML, EPUB) Using alt attributes so the image is ignored. - Example

Document Unique Command

Level

[One of Prerequisite/Baseline/Enhanced]
Baseline

Method

[Description of method, not technology specific]

Techniques

[List of technology/platform specific techniques.]

- ([Platforms]) [Method title]
- (HTML, EPUB) Using alt attributes so the image is ignored. - Example

[Assertion name]

Level

[One of Prerequisite/Baseline/Enhanced]

Test

- [Example] The organization has a documented style guide which includes guidance on text-alternatives, and a policy that editors are required to follow the style guide.
-

Alternative Means

What to do when no method meets outcome for Custom keyboard commands

If none of the above methods match the scenario you are faced with, you can meet the outcome in another way. It must meet the outcome and general information under “what to do”.

Meeting Minutes

Use this for decisions, in-process scratchpads, and other rough notes.

9 July 2024

For pre-breakout portion of call:

Questions Bruce had before subgroup activity breakouts.

Q1: Who can grant edit access? Can everyone at least make suggestions?

> AC: The sub-group facilitator can grant access, and that is best to do for people in the sub-group (i.e. members of AG).

Q:2 How much deference to current numbering from consolidated outcomes? Can we reorder outcomes?

> AC: Please keep the number of the main outcome that’s being worked on, please discuss if you want to move onto a different outcome.

Q3: Consolidated outcomes doc provides outcome names and outcome text. Goal is to be “Plain english sentence of what the outcome should achieve” but is that much different than the outcome text?

> AC: Goal does allow for a bit more text / explanation, the outcome should be concise. If we end up with very similar things, we can drop the goal. For example, keyboard focus had:
>
> **Outcome:** The keyboard focus must be visually indicated [commanding, normative]
>
> **Goal:** Make it easier to spot the keyboard focus. [colloquial]

Q4: Is the model example (placeholder text) *only one* outcome from 5.1.1? (i.e., Image Alternatives: Images have equivalent text alternatives)

- If so, will we (keyboard) need *five* method decision trees? (Bruce thinks so.)
- If we have multiple method decision trees, do we need multiple *sets* of methods? (Bruce thinks *maybe* not; we can see how it goes…)

AC: I think the image alternative outcomes will be collapsed into one or two. Each outcome should have 1 decision tree if there is more than one method.
We could share methods between outcomes, although it hasn't come up yet.

Notes

Attendance

- Bruce
- Detlev
- Azlan
- LoriO
- Theo
- Chuck
- Marco

For subgroup conversation:

- Check that you can make suggestions.
    - One goal for today is practice with template.
- Is prose for Goal any different than "the full outcome text from the list"?
    - That is, did we already start with "Plain english sentence of what the outcome should achieve"?
- What is "item of scope" and is it the same for all five outcomes?
    - Some things are inputs controls
    - Set of Web pages
    - Level of effort (6.2.1), item of scope is process
    - For Keyboard trap, viewport
        - Might be more granular, active area, pop-up

Area of the screen where the focus is, could have non model dialogItem of scope options:
Application Keyboard Command

We will need multiple decision trees.
Methods might be common to multiple outcomes.

Actions for asynchronous work

- Theo: Will work on a decision tree for non-conflict commands
- Bruce: Will review the 25 user needs from scratchpad and determine what hasn't been assigned. Table of user needs (25 rows) versus our 5 outcomes. [done]
    - Bruce suggests starting with simpler outcomes, saving Keyboard Only for last.
- Unassigned/Future: Turn existing guidance (WCAG 2) into methods.

- Theo: Taking existing user needs, putting them into a table. 2nd column will outline other outcomes beyond the drivers that may better support the user's needs. Related outcomes.
- Have not started our methods, methods can be ported from understanding documents or other sources.
- Have not done the exercise of writing out the goals and the what to do’s.

Items for reporting back on AG call

- Replicated outcomes (so we have 5)
- “Item of scope”:
    - Inputs
    - Process
    - Viewport or page
- Duplication versus overlap
- Filled in document with user needs from scratchpad to help support next week’s work
- Asynchronous work can be adding methods,
- Anticipated issues:
    - What is user agent responsibility?
    - What is author's responsibility?
    - Which are technology / platform dependent ?

Next week: If Theo has a decision tree (for not having conflicting keystrokes), that would be a good place to start and see if we can’t expand the decision tree and address “bigger” challenges.

Suggestion from Alastair to maybe only pick one of five. Decision tree exercise should be *after* having worked on methods.

Table: 25 User Needs versus 5 outcomes

	User Need	6.2.1 comp. keyboard effort	6.2.2 consistent interaction	6.2.3 (unique) kb commands	6.2.4 keyboard only	6.2.5 no keyboard trap
1	As a user of assistive technology, I need my keyboard commands to remain consistent.		yes		yes	
2	As a user of a keyboard, I should be able to complete actions in a comparable amount of time to other users.	yes			yes	

	User Need	6.2.1 comp. keyboard effort	6.2.2 consistent interaction	6.2.3 (unique) kb commands	6.2.4 keyboard only	6.2.5 no keyboard trap
3	As a user of a keyboard, the complexity and number of input commands required to complete a task should be comparable with those of a user of other methods of input.	yes			yes	
4	As a keyboard user, I should be able to perform all actions using a keyboard exclusively.				yes*	
5	As a keyboard user, I should not get unintentionally stuck.		yes		yes	yes
6	As a keyboard user, keyboard shortcuts and assistive technologies do not interfere with each other.			yes	yes	yes
7	As a keyboard user, timing should not prevent me from completing actions.	yes			yes	yes
8	As a keyboard user, I should not be required to press multiple keys at the same time to complete an action.				yes*	
9	As a keyboard user, I always know where I am in the interface.		yes		yes	
10	As a keyboard user, I am informed of non-standard commands before I need to make use of one.		yes	yes	yes	yes
11	As a keyboard user, information on available shortcuts is provided in context and in a way I can perceive.		yes	yes	yes	
12	As a keyboard user, the keyboard			yes	yes	

	User Need	6.2.1 comp. keyboard effort	6.2.2 consistent interaction	6.2.3 (unique) kb commands	6.2.4 keyboard only	6.2.5 no keyboard trap
	commands that I need to press don’t interfere with my operating system and/or assistive technology.					
13	As a keyboard user, my choice of keyboard is fully supported.				yes*	
14	As a keyboard user, I should be provided with a method of skipping repetitive navigation links.	yes			yes	
15	As a keyboard user, I would like my user agent to provide keyboard shortcuts to quickly navigate around an interface.				yes*	
16	If I implement a keyboard device within my site/app, that control must respond correctly to standard keyboard commands.		yes	yes	yes	
17	@@ As a keyboard user, I am warned about changes in context before they occur.					
18	@@ As a keyboard user, I am warned about changes in focus before they occur.					
19	As a keyboard user, I am not expected to use a certain viewport in order to use a keyboard.				yes*	
20	As a keyboard user, applications should be flexible in allowing me to use the keyboard commands I want to use (e.g. through personalisation).				yes*	
21	As a keyboard user, I shouldn't		yes		yes	yes

	User Need	6.2.1 comp. keyboard effort	6.2.2 consistent interaction	6.2.3 (unique) kb commands	6.2.4 keyboard only	6.2.5 no keyboard trap
	have my expected keyboard commands disabled.					
22	As a keyboard user, I should be able to access all scrollable regions of a view/app so I can read all text within these regions.				yes*	
23	As a keyboard user, I need to be able to access content in an equivalent way as other input modalities.	yes			yes	
24	As a keyboard user, when I reach a custom implementation of a common component, all keyboard commands that exist are implemented.		yes	yes	yes	yes
25	@@ As a user, I don’t want a keyboard shortcut to be the only way to do something (or, there should not be only one way to do something).					
tally		5	8	6	22	6

16 July 2024

Use this for decisions, in-process scratchpads, and other rough notes.

Notes

Attendance

- Chuck
- Azlan
- LoriO
- Sarah Horton
- Detlev Fischer
- Theo

Actions for asynchronous work

- Discussion about prerequisites
  - If this method/outcome is not present, there is no accessibility (issue will likely prevent affected users from completing task)
  - Rationale for rating severity, can review work from Issue Severity

23 July 2024

Use this for decisions, in-process scratchpads, and other rough notes.

Notes

Attendance

- Bruce
- Azlan
- Marco
- Jen
- Poornima Subramanian

Agenda

- Going forward, Bruce will try to provide an agenda (in this doc) before we talk.
  - Probably will *not* be sending out minutes.
  - But did everyone get Bruce's email sent 7/11 ?
  - Confirm email address as needed.
- Review agenda, adding as necessary.
- Review doc insertions/changes since last week.
- Review actions for asynchronous work from last week.
- Method versus What to Do.

Actions for asynchronous work

- Bruce to send out minutes.
  - Email Jen her action item.
  - Ask Theo to write Goal and What to Do.
- Bruce to format 6.2.3 method as method in this document.

Items for reporting back to larger AG call

- Made progress on Goals and What to do
- One What to re-formatted as Methods
- We rename 6.2.3 to Custom keyboard command
- Any suggestions for plain language for "User Flows"?

30 July 2024

Use this for decisions, in-process scratchpads, and other rough notes.

Notes

Attendance

- Bruce has to leave AG call at 12:20
- Azlan
- Marco has to leave at 12:10 EST
- Jen sends regrets
- Lori O sends regrets
- Sarah Horton
- Chris Loiselle
- Chuck regets, to facilitation another subgroup

Agenda

- Anyone missing last week's notes? (Bruce sent 7/24).
- Review agenda, adding as necessary.
- Review any substantive asynchronous in this doc *after* last Tuesday.
    - Bruce did some editorial formatting.
        - Decision Tree for 6.2.3 Custom keyboard commands done. Methods need filling in, but that seems straight forward.
    - Jen drafted Goal and What to do for 6.2.1
    - Bruce drafted Goal and What to do for 6.2.2
- Discuss work plan going forward (pending email from AG chairs)
    - Look for time slot going forward.
- Review actions for asynchronous work from last week.

Actions for asynchronous work

- Bruce will send out minutes (done)

Items for reporting back to larger AG call

- Need more time.
- Any suggestions for plain language for "User Flows"?
- **Wednesday 11:00 Boston to noon going forward, August 7th.**

14 August 2024

Use this for decisions, in-process scratchpads, and other rough notes.

Notes

Attendance

- Bruce
- Marco
- Jen
- Chris

Agenda

- Recap status of work from last few weeks
- Is the time slot (Weds, 11 am U.S. Boston, same as AG) good enough?
    - Bruce had to cancel last week (8/14) and will be can't make it next week (8/21).
- Review agenda, adding as necessary.
- Review actions for asynchronous work from last week.
- Discussed Google Doc as sufficient tool

Actions for asynchronous work

- Bruce to send out minutes
- Bruces to check chairs expectations.
- Bruce to email version
- Bruce to note WCAG2 gap allowing complex keyboard shortcuts. Authors can invent shortcuts that override common keyboard shortcuts. (Theo's ^Z story.)
- Split documentation of keyboard shortcuts as separate method/requirement of 6.2.3.

Items for reporting back to larger AG call (8/20)

- Five outcomes, 6.2.1 Comparable keyboard effort, 6.2.2 Consistent keyboard interaction, 6.2.3 Keyboard commands [are unique], 6.2.4 Keyboard only, 6.2.5 No keyboard trap
    - Meeting Wednesday 11:00 Boston
- Subgroup progress to date:
    - Mapping user needs to each outcome (1st meeting, lots of duplicates as to be expected)
    - Goal and What to Do for each outcome (2nd/3rd meeting)
    - Work for next several meeting is methods and decision trees
- 6.2.3 Custom keyboard command
    - Renamed from "Keyboard commands" to "Custom keyboard commands" to be more descriptive
    - Reoriented to only be about content. It is not feasible for keyboard platform commands to be comprehensively documented.
    - Decision tree close to done

20 August 2024

Use this for decisions, in-process scratchpads, and other rough notes..

Notes

Detlev shared example of keyboard trap, where after opening form and start typing. State of UI element prevented further keyboard interaction. (E.g. started typing name, and could not leave name field.) UI elements just taking focus is not enough.   Another example is selections (e.g. checking a check box) opens up more fields.

You to Everyone 11:53 AM
https://docs.google.com/document/d/1uE2WCxPmvNopdCbuZQm_-cGyEdxEouRmZ8UUIlyutoU/edit#heading=h.7120wskf56qi

Graham to Everyone 11:54 AM
hmmm seems I cant use Mic
yeah should be ok

Lori Oakley to Everyone 11:59 AM
I he to drop for another meeting, I'll be back shortly

Graham to Everyone 11:59 AM
my gut reaction is no, because otherwise we could argue nearly all things could be fixed with "x specific software"
I have used screen readers for years for testing etc. I still don't know how to activate "pass-through", are end users who rely on AT daily likely to know how to activate this?

Graham to Everyone 12:16 PM
Although this falls into "keyboard trap" probably, some custom elements retain focus within them (I am thinking Code editors), ESC should take you to the NEXT focusable item on the page, I have been trapped a few times where ESC takes you before and so you can't tab past it.

The issue is it isn't a "trap" by standard definitions.
If they are not tabbable then they are not focusable, so odds are they will not work for a screen reader anyway.
This isn't a "decision tree" item, it is more a testing methodology item. The same rules apply at all screen sizes.

Graham to Everyone 12:23 PM
@Bruce Bailey (USAB, he/him) can you open the chat please, nobody is seeing the messages I am writing and I need someone to be my voice today! thanks.
Tabbing is super important for those with dexterity / accuracy impairments such as Cerebral Palsy or Parkinsons, so the Tab test is essential anyway to facilitate those users.

Graham to Everyone 12:31 PM
Please see above message about code editors as this is related.
lease see above message about code editors as this is related.
The problem is I didn't add context as was responding to the conversation at the time so not sure how much of the above makes sense! haha

Graham to Everyone 12:36 PM
Shift-Tab - not necessarily, I have "skip links" throughout a document to skip long tables etc. I would not expect them to show on shift-tab necessarily.
And then that raises the question of should a "skip link" be available at the BOTTOM of a long table (of links for example) to skip it if going in reverse?

Graham 12:42 PM
There needs to be an element of "user expectation". If you cannot get passed something it MUST be a fail unless there is a clearly explained / visible way to get past that item.

We also need to consider people with cognitivie impariments, and therefore "expected behaviour" is important
OK, I think there needs to be a discussion about usage of chat at a group level. If we ever get a contributor who has a severe speech impairment / is mute their experience will be awful and not accessible.

Attendance

- Azlan
- Bruce
- Detlev
- Lori
- Poornma
- Graham

Items for reporting back to AG:
Q: Is scope strictly web or is it broader?
A: Trying to keep outcomes and decision trees generic.
Methods expected to be technology specific

21 August 2024

Use this for decisions, in-process scratchpads, and other rough notes.
Notes added to inline comments.

28 August 2024

Use this for decision, in-progress scratpads, and other rough notes.

Attendance

Bruce Azlan Chris Jen Lori Giacomo Marco

Agenda

- Recap
    - Work since last week?
    - Bruce cleaned editorial, accepted deletions insertions for Decision Trees for 6.2.3 Custom keyboard commands and 6.2.4 Keyboard only (thanks Detlev).
- Briefly discuss if heading structure for this doc should be redone?
- Suggestion for 8/28: Outline how 6.2.4 Keyboard only relates to other Outcomes?

From top, back in July:

- **Usable**
    - **6.2.1 Comparable keyboard effort** - The number of input commands required to complete a task using the keyboard is similar to the number of input commands when using other input modalities.
    - **6.2.2 Consistent keyboard interaction** - Keyboard interface interactions are consistent.
    - **6.2.3 Custom keyboard commands** - Application keyboard commands do not conflict with platform commands, and the user is informed of non-standard commands.
- **Operable (potential prerequisites)**
    - **6.2.4 Keyboard only** - All functionality can be performed through the keyboard interface only, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.
    - **6.2.5 No keyboard trap** - If keyboard focus can be moved to an interactive component, then the keyboard focus can be moved away from that component.

- We think we are okay with current ordering and grouping as initially provided.
- Keyboard trap, because such blocker (non-interference to use WCAG2 terms) can't be part of Keyboard only.
- No keyboard trap is not something which can tested independently of Keyboard only.
- A separate audit for No keyboard trap would make it too easy to miss keyboard traps.
- A keyboard trap will block further audit
    - Is this a full stop?
    - Can auditor continue for other keyboard accessibility?
        - Keyboard trap might full block further testing. Trap would need to document in workflow, as critical failure. May be more than one keyboard trap!

    - From practical view, what is unit testing? Org using auditor would not want audit to stop!
- Separate decision trees between what author/UX designer/developer does versus audit?
    - Developer needs same rules as auditor.
    - Decision trees seem to work for a design/develop perspective.

Actions for asynchronous work

- Bruce to add numbers and tiles to duplicative titles.
- Chris will add notes to this doc from Word document circulated last week. (done)
- Need glossary term for standard navigation keys.
- Bruce to clean up doc more.

More notes on glossary term:

- Standard or usual for app like Excel different from Open Office.
- Browsers and operating systems have default controls.
    - What are default for the application?
    - Use of Excel within a web browser? What is usual/common there?
    - What are all users aware of?
        - First time Excel user probably uses arrow keys rather than enter and tab.
        - Might need two levels of this?
            - Think of how your mother uses the computer!

From chat: 11:17:51 From Chris Loiselle - Oracle to Everyone:
https://www.w3.org/WAI/standards-guidelines/wcag/wcag3-intro/#for-your-review for reference
11:46:48 From Chris Loiselle - Oracle to Everyone:
https://www.w3.org/WAI/ARIA/apg/practices/keyboard-interface/ uses A primary keyboard navigation convention common across all platforms ... might want to improvise off of their wordsmithing.
11:58:13 From Giacomo Petri to Everyone:
Fundamental Keyboard Navigation Conventions
11:58:16 From Giacomo Petri to Everyone:
From aria practices

Regrets from Jen for 9/3.

04 September 2024

Use this for decision, in-progress scratpads, and other rough notes.

Attendance

Bruce Chris Lori. Giacomo Regrets from Jen and Theo.

Agenda

- Welcome, Introductions
- Recap since last week?
- AG call yesterday focussed on Assertions, Bruce would like that to be focus for today.
  - Assertions per outcome, or for group of outcomes?

Notes

- Assertions should be specific to outcomes. General assertions will be addressed by larger group.
- Statement of fact that org/author did something.
- An ACR (from VPAT) is an assertion. (general)
- We use third-party testing. (general)
- We assert that we have keyboard shortcuts documented for our developers.
- Example from ALT group was existence of style guide.
  - Style guide would have to be vetted for completeness/accuracy.
  - See presentation materials 9/4 AG call.
- (Generic) Assertions need to apply to plug-ins used by website.
  - Website owners may have limited control over the 3rd party service – and also requirement to use 3rd party service (from parent org, legally, only product available).
- (Generic) Customer requirement for accessibility statement, including details of testing.
- We as an organisation have tested using ANDI for keyboard accessibility.
  - ANDI would help keyboard testing by https://www.section508.gov/training/web-software/andi-training-videos/focus-element/ , however it is what company asserts on how they've used ANDI [what is an assertion for WCAG3]
  - How the org uses the tool
    - We have identified tool to do outcome X/Y/Z
- There are no keyboard traps.
- We validite using ANDI that keyboard focus can be moved to all interactive components, and that the keyboard focus can be moved away from that component.
- (Generic) Org. states what they are doing to come into compliance with an audit.
  - Can be treated as specific (to an SC) because web content audit has specific defects identified.
  - Developer (responding to audit) makes assertions about what they are doing to fix problems.
- See WAI statement generator: https://www.w3.org/WAI/planning/statements/generator/#create-efforts
- Another example: Welcome to the Accessibility Statement Generator built by Nomensa
- We tested the web content page, using XYZ methodes, and achieved ABC results.
  - Bug statements (defects) are tasks that need to get done.
- Example.: All of our non-decorative images have appropriate alt text.
- Enforcement:

    - Who is responsible for testing assertions? (Won't be W3C. Likely to be regulators.)

Noting that focusable is very intertwined with keyboard accessibility.

For reference (during call):

6.2.1 Comparable keyboard effort - The number of input commands required to complete a task using the keyboard is similar to the number of input commands when using other input modalities.

6.2.2 Consistent keyboard interaction - Keyboard interface interactions are consistent.

6.2.3 Custom keyboard commands - Application keyboard commands do not conflict with platform commands, and the user is informed of non-standard commands.

6.2.4 Keyboard only - All functionality can be performed through the keyboard interface only, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.

6.2.5 No keyboard trap - If keyboard focus can be moved to an interactive component, then the keyboard focus can be moved away from that component.

10 September 2024

Notes for AG call.

- Group meeting
    - Weekly, for an hour, Wednesdays 11:00 Eastern.
    - Bruce is not available tomorrow, 9/11. Jen's hosting.
- Five related outcomes:
    - 6.2.1 Comparable keyboard effort
    - 6.2.2 Consistent keyboard interaction
    - 6.2.3 Custom keyboard commands
    - 6.2.4 Keyboard only
    - 6.2.5 No keyboard trap
- First exercise, mapping User Needs
    - See table above, 22 of 25 users needs versus 5 outcomes.
        - Missing only 3 from the initial set.
    - Keyboard only hits all 22.
- Comparable keyboard effort
    - Have not started.
- Consistent keyboard interaction

    - Briefly started, did not get much traction, but might be easier now.
- Custom keyboard commands
    - Thrashed for bit, unpossible to document all OS/platform keyboard commands.
    - Solved by turning the outcome around!
    - Make the web app document any keyboard shortcuts it makes itself.
- Keyboard only
    - Something we have lots of experience from 2.x with success criterion 2.1.1, and decision tree common along.
    - See 6.2.4 decision tree above.
- No keyboard trap
    - Decision tree and methods are concrete enough, but…
    - Won't catch some keyboard traps without testing for keyboard accessibility.
        - Example, entering text in form field exposes the keyboard trap – when merely tabbing through form did not.
    - Is this outcome a parallel concurrent process with Keyboard only?
        - Part of "All functionality of the content is operable through a keyboard interface" includes not hitting a keyboard trap.
- Works parallels AG calls
    - So last week, 9/4 focus was on assertions.
- Assertions
    - Did not make much progress on outcome-specific assertions.
        - Could have outcome-specific for test methodology used.
        - We have identified tool X for do meeting outcomes X/Y/Z,
    - Example, ANDI good for keyboard testing.
        - Could provide assertions related to its features.
        - Would be hard to disambiguate from testing for keyboard from testing for focus.
    - ACR from VPAT feels like an outcome-oriented assertion list.
    - From group conversation, probably won't return to assertions because they seem like a whole-of-WCAG3 issue.
- Unresolved question:
    - How to document dependance on AT features? (E.g., forms mode (and analogs) in JAWS, NVDA, Dragon.)
- Next steps?
    - Outline dependencies of 6.2.4 on others?
    - New clean document?
    - Nice to have notes at the end of this document, but decision trees are hard to find with current formatting.

Notes *from* AG call

- Alastair noted: If everything can be objectively tested, that's ok!
- We need not return to assertions.

11 September 2024

Use this for decision, in-progress scratpads, and other rough notes.
Jen Hosting, regrets from Bruce.

- See comments from Chris added to doc 9/11.

17 September 2024

This area is used for decisions, in-progress scratpads, and other rough notes.

Agenda

- Welcome and recap
    - Comments in document scribed by Chris on 9/11.
    - Detlev mentioned on Tuesday touchpoint on AG call that “Keyboard Only” is an umbrella outcome for the others
- TPAC next week, who’s going, are we meeting?
- Look at the preview build for Focus Appearance as model for Keyboard Only
- 6.2.1 Comparable keyboard effort, what do?

Notes

- Attendance: Bruce, Chris, Lori. Giacomo. Regrets from Jen.
- From last week: testing with ANDI, and discussion if testing with AT is subset?
    - Testing with only the keyboard is default for anyone using SR.
        - Where does that fit into decision tree?
    - See Chris’ comments on 6.2.4 keyboard only decision tree.
- Recap from AG meeting yesterday, DetLev gave update
    - Explainer Google Doc and Editor’s Draft
        - Look for (glossary) terms in particular – especially relevant to subgroups.
    - WCAG3 draft for public comment to be published soon after TPAC
        - WAI WCAG 3 Introduction is a stable URL for status
    - All on the AG call agreed that the concrete examples were very helpful
    - Some work overlaps with WCAG2 issues (single pointer in particular)
- TPAC
    - AG will be working on WCAG3 for hours
        - Public schedule is available, also by logging in to your W3C account
    - Free registration for Invited Experts should re-opening Monday 9/23
        - Remote-only attendance to AG calls is also free.
            - Email w3t-tpregister@w3.org (or an AG chair) if you have questions or difficulties.
        - Looking forward to the retrospective presentation
        - Joint meeting between APA, AG, and ARIA
        - Keyboard subgroup will not meet next week
- 6.2.1 Comparable keyboard effort

    - Group agrees to continuing deferring
    - Single pointer (from 2.2) might warrant revisiting when we come back to comparable effort analysis.
- Bruce drafted a goal and what to do for Custom Keyboard Commands
- Keyboard only as main – or only – Outcome?
    - No objections.
- Discussed Focus Appearance Google Doc and preview build as model for Keyboard Input.
    - Methods for Focus Appearance page includes Decision Tree and list of methods
        - Methods available links from page
    - Note parenthetical status of Enhanced, baseline, and prerequisites
        - List these in order of importance vs. the current mixed list out in numbered formatting.
    - Within the decision tree, links bring users to sub page vs. anchor links for methods.
        - Is that intended?
    - Discussed use of stop vs. no , continue ?
        - It is only baseline requirements at the moment that would have a stop
    - Giacomo shared Google doc version of Focus Appearance
- Google Docs not working great for our collaboration.
    - Word export is working well for post-meeting but is read-only.
- AG calls have been oriented towards repo and comments
    - Works okay for narrow topics
    - Not an appealing option for majority of AG participants for drafting
    - The github repo also hosts the WCAG3 wiki but that is probably not appealing to anyone but Bruce.
- Word online is not familiar to most subgroup members on the call.
- Slack might be an option.
    - Okay for Giacomo, Chris, Lori
    - Bruce resisting Slack and other social media
    - Accessibility is of some concern

Action Items

- Bruce will send out email with Word version attached.
- Bruce will send out email with “goal” and “what to do” for Custom Keyboard Commands.
- Bruce draft on a decision tree and/or methods modelled after Focus Appearance, location and format TBD
- Everyone please keep thinking on topic for mechanics of collaboration.
- Next call is 10/2.

15 October 2024

This area is used for decisions, in-progress scratpads, and other rough notes.

Agenda

- Welcome and recap
- The recent SURVEY for participation in upcoming subgroups might help this subgroup pick meeting time..
- Look at the preview build for Focus Appearance as model for Keyboard Only
- Updated Goal for 6.2.2 Consistent keyboard interaction to:
  - Keyboard interaction including navigation and operation, is intuitive for users familiar with the platform.

Notes

Regrets from Theo, Gregg, Azlan
Attendance: Bruce, Giacomo, Chris, Jen

Consensus that Focus Appearance decision tree is a good model to emulate.

Some reference links:

- Current WCAG 3 repo on GitHub including issues / discussions / PRs
  - PR112 for PR112 includes build preview of How To for Focus Appearance (and Text Alternatives
- Previous (now obsolete) Silver repo on GitHub but that has an
  - index page to recent (September 11, 2024) editors drafts
- Accessibility Guidelines "Silver" (w3c.github.io)
- w3c/silver: Accessibility Guidelines "Silver" (github.com)
- Chris suggested ARIA-AT community group work on enabling AT Interoperability being relevant to our conversation of how keyboard compatibility includes voice recognition and keyboard emulation via switch input.

Regrets from Jen for next couple of weeks.

Action items:

- Bruce to ask for new Google doc to facilitate drafting.
- Bruce to clean up notes, send out email notes.