Microsoft has reviewed the November 7, 2013 User Agent Accessibility Guidelines (UAAG) 2.0 last call working draft. Microsoft is appreciative of the hard work done by the working group. But it is our opinion that the draft as it stands is not ready to advance to last call status and that fundamental rework is necessary before it can be reconsidered for another last call publication. Due to the flaws that we have found, we do not believe it is productive to provide detail feedback. Instead, we will provide feedback at a higher level and highlight examples where suitable. We advise the working group to carefully consider our suggestions and make the necessary fundamental changes to UAAG to make it a useful set of guidelines.

# Lack of distinction between content, browsers, assistive technologies, and operating systems

The most fundamental problem we see is the lack of distinction between the content, browsers, assistive technologies, and operating systems. This problem manifests itself throughout the draft.

## Definition of “User agent” should not include “web-based user agent”

We are strongly opposed to the interpretation that web content such as online translation services, email services, and document authoring services should be included as user agents. These services, when consumed via the web, are web content and should consult WCAG 2.0, not UAAG 2.0 for accessibility guidance. Adding UAAG 2.0 to the mix of applicable guidance to web content already applicable with WCAG 2.0 makes it confusing for content producers to determine which W3C accessibility guideline to follow. It also makes the already muddled UAAG 2.0 even more so by diluting focus from its primary goal of outlining accessibility guidance for web browsers. Microsoft urges the working group to revise the definition of user agent accordingly and to remove all draft language related to the notion that web services can be considered user agents.

## Separation of assistive technologies from user agents

While UAAG 2.0 no longer explicitly includes the AT into its definition of user agent, it is still obvious that AT is targeted as an applicable entity for UAAG implementation. Guideline 1.6, for example, is clearly written for AT instead of browsers. It defies common sense to create one accessibility guideline for AT and browsers. This greatly distracts the primary focus of UAAG, making it less relevant and less suitable for consideration by AT and browser makers. The working group should make it clear that UAAG is not intended for AT regardless of whether AT performs functions that are typically provided by browsers and that UAAG needs to be revised accordingly.

## Separation between browsers and OS

While it is understood that the separation between browsers features and OS features can be blurry at times, there are functions that are generally expected to be provided by the OS to facilitate accessibility. We do not expect the working group to delineate the responsibility of OS and browsers since it is contextual and fluid. But we expect the working group to recognize such separation of responsibility. For example, success criterion 1.4.1 (Text Scale, Color, Font) should be separated into two criteria—one simply asking the browsers to follow the OS text size and such at level A and another to provide its own options if the OS fails to provide text options at level AA.

## Delineate between content and browser

There should be clear separation of the guidelines that are applicable to the browser itself (such as 3.2.2 or 2.3.4) and how it handles web content (such as 3.2.3 and 2.2.1). In theory, success criteria applicable to handling of web content should always contain the term “rendered content” to make this distinction clear. But this is, unfortunately, not done with any consistency. For example, guideline 2.2 are clearly applicable to browser treatment of rendered content, but the term never appears in its four success criteria. This makes it difficult for the audience to interpret the intention of the guidelines.

## Examples in the implementation document do not make distinction of content, browser, assistive technologies, and OS

We recognize the value of personalizing the examples, but these examples are not implementation examples. They are use case scenarios with no explanation of what is expected of the browser as oppose to that of the content, the OS, and the AT. It is understandable that average users do not understand the roles and responsibilities of content, browsers, assistive technologies, and OS. But the working group needs to provide more sufficient context for a technical audience. We believe this problem is a reflection of the lack of focus in UAAG. Take, for example, the second example from 1.2.2. (Maria uses a screen reader. When a table lacks marked up header rows, the user agent gives her the option to have the first row treated as the table header row.) It is entirely unclear as to what is expected to be done from the example. It is possible for the reader to interpret that:

1. the browser is supposed to alter the content without authorization from the content author to change the structure of the table, or
2. the browser is supposed to alter what it passes through to the accessibility API, or
3. the AT is supposed to figure out the table header row from the accessibility API and present it to the user as such, or
4. the AT is supposed to interrogate the HTML code, determine the header, ignore the information from an accessibility API, and present the header to the user

Since there are so many ways to interpret the example and the lack of scope of what a user agent is, it is essentially not implementable in reality. Microsoft asks the working group to reexamine the fundamental nature of what a user agent is and rewrite UAAG from top to bottom with a more precise understanding of user agents and what can be done to make them more accessible.

# Setting realistic Level A, AA, and AAA success criteria

Assuming the working group agrees to narrow the scope of UAAG, we then ask that the delineation between the levels to be reexamined. It is our position that a browser meeting level A should be able to:

1. interact with web content that meets WCAG 2.0, and
2. facilitate programmatic access of the content and its user interface to and from AT, and
3. follow the accessibility settings from the OS, and
4. provide an user interface that is generally accessible, such as enabling keyboard control on its own features when operating on an environment where keyboard control is available, and
5. nothing more

Level AA should consist of success criteria that aid users in case of common content failures and predictable solutions are available.

Level AAA should consist of success criteria that aid users in case of content failures and where implementable solutions are available.

Microsoft appreciates the aspiration of the working group. But it must be understood that the final deliverable must be grounded on reality and that implementations are necessary for UAAG to be considered for recommendation. We encourage the working group to channel its aspiration and creativity elsewhere where developers can consult for future-generation-ideas instead of cluttering UAAG with unclear, untested, or unrealistic success criteria.

# Think beyond desktop computers

Another fundamental flaw in UAAG is that it is unclear whether it reflects the reality that user agents are commonly available in mobile phones, tablet, wearable, and television form factors where keyboards, if available, are used for text entry instead of navigation, for example. Indeed, WCAG 2.0 did not account for this in 2008 because these form factors were brand new at the time. But UAAG should be better prepared to address mobile accessibility in 2014. We encourage the working group to closely examine non-desktop environments and consider whether there are unique constraints or requirements as a result.

# Realistic expectation of the end users

We found some of the success criteria related to the use of alternate style sheet to be counterproductive. First, custom-created style sheets are far too complex of a solution for the vast majority of the users. It will more likely result in unintended errors than otherwise. Second, it is not a scalable solution. It is unrealistic to expect users to create a portfolio of style sheets for all the websites and to maintain them as the sites change over time. We believe this is an outdated approach that is no longer relevant in today’s web environment and urge the working group to reconsider.

# Subjectivity must be eliminated

We found some success criteria to be too subjective to be tested with any reliability. For example, success criterion 2.3.1 (The user can navigate directly to important elements in rendered content.) contains two subjective terms that makes it essentially untestable—directly and important. It is highly debatable as to which navigation methods are considered as “directly” navigable. It is also impossible to objectively determine what an important element is. We also found it dubious that the working group expects a user agent to make such a subjective decision when humans would have a difficult time making such decision. Thus, even if a user agent attempts to determine what is important, it would still be highly debatable as to whether it has done so successfully. We ask the working group to meticulously remove subjectivity from UAAG.