

直感的 UI をスクリーンリーダーで利用する場合の ユーザーの学習コストについて

Web 会議ツールを中心に

植村 要

中央大学 〒192-0393 東京都八王子市東中野 742-1

E-mail: kanameuemura@gmail.com

あらまし 本報告では、スクリーンリーダーで Web 会議ツールを利用するために採用されている方法を記述し、スクリーンリーダー・ユーザーに生じている学習コストを質的に検討する。この検討を通じて、直感的なユーザインタフェースを採用するアプリをスクリーンリーダー・ユーザーが利用する際に求められる学習は、個人の努力によって到達可能な程度を超えていることを指摘する。そして、直感的なユーザインタフェースをキーボードショートカットによって、スクリーンリーダーでの操作を可能にするというアプローチが、アクセシビリティを実現するための方法として適切かについて問題提起する。

キーワード アクセシビリティ・キーボードショートカット・視覚障害

User learning costs when an intuitive user interface is used with a screen reader

– Focusing on web conference tools

Kaname UEMURA

Chuo University 742-1 Higashinakano, Hachioji-shi, Tokyo, 192-0393 Japan

E-mail: kanameuemura@gmail.com

Abstract In this report, I describe a method for using web conference tools with a screen reader and qualitatively examine the learning costs that arise for screen reader users. Through this examination, I show that the learning required of a screen reader user by applications that employ an intuitive user interface exceeds what is attainable through individual effort. I then raise the issue of whether the approach of making intuitive interfaces operatable with a screen reader through keyboard shortcuts is an appropriate method of realizing accessibility.

Keyword accessibility, keyboard shortcuts, people with visual disabilities

1. 目的

本報告では、直感的ユーザインタフェース（以下「UI」）をキーボードショートカットによってスクリーンリーダー（以下「SR」）での操作を可能にするというアプローチが適切かについての議論を喚起することを目標に、SR 環境下で Web 会議ツールを利用する

ために採用されている方法を記述し、SR ユーザーに生じている学習コストの内実を質的に検討する。

以下では、まず2において、本報告が事例とする環境を記す。そして、3、および4において、全盲の視覚障害者がPCを利用するための支援技術であるSRと、直感的UIを採用するアプリとしてWeb

会議ツールを事例に、そのアクセシビリティ機能、および、それぞれの学習コストを確認する。そのうえで、5において、SR環境下でのWeb会議ツールのユーザー体験（以下「UX」）を記述する。この事例を通して、直感的UIをSRで利用可能にするための取組について、その方向性が適切であるかについて問題提起する。

2. 事例とする環境

次節からは、Web会議ツールをSRで利用するために、ユーザーに求められている学習の内実を示す。そこで、OS、SR、Web会議ツールのいずれについても、網羅的に示すことはせず、代表的なサービスを選択することとした。具体的には、以下のとおりである。

OS

OSは、Windows10とした。一般的にも利用者が多く、日本の視覚障害者に最も利用されているOSであることから選択した。

SR

SRは、PC-Talker Neo(*1)と、NVDA日本語版(*2)の2者を用いることとした。PC-Talker Neoは、高知システム開発が開発、提供するSRであり、日本国内の視覚障害者に最も利用されている。NVDAは、オーストラリアの非営利法人 NV Access を中心とするコミュニティが開発するSRであり、英語以外の54ヶ国語に翻訳して提供されている。その日本語版はNVDA日本語チームが提供しており、日本国内の視覚障害者における利用者数は2番目に多い。この両者は性質や操作性に違いがあることから、両者を用いることとした。

Web会議ツール

Web会議ツールは、Zoom(*3)とMicrosoft Teams(*4)の2サービスについて、いずれもネイティブアプリを用いることとした。ネット情報を参考に、広く利用されていると推測されるサービスとして、この2サービスを選択した。

3. SRにおけるコマンド

SRは、画面に表示されている文字列をコンピュータの合成音声で読み上げる機能と、PCを操作する機能の大きく2系統の機能で構成されている。SRは、全盲の者がPCやスマートホンを利用するときに用いる支援技術として開発されることから、音声を耳で聞いてキーボードで操作するという形が採用されている。SRは、画面を読み上げ、PCを操作するために、様々なコマンドを有している。そうしたコマンドを実行するには、メニューを開いて目的のコマンドを選択肢で実行するということをしては、円滑な操作が困難である。そこで、大半のコマンドにはショートカットキーが設定されている。SRのショートカットキーは、二つから四つ程度のキーを同時に入力するという形で設定されている。ユーザーは、目的に

応じて適宜ショートカットキーを入力することで、音声読み上げをし、PCを操作する。

ここに、ユーザーには、予めこれらの各コマンドのショートカットキーのキー割り当てを覚えておかなければならないという学習コストが発生する。ただ、これらのコマンドは、利用するアプリやWebサイトによって変わるものではなく、PCを利用する際には汎用的に用いられるコマンドであることから、一度覚えてしまえば済むものではある。しかし、SRの性能が向上するにつれて、コマンドが増加し、現状では大変な数になっている。また、めったには行わない特殊な操作をするためのコマンドは、予め学習しておいたとしても、そのコマンドを用いたいときには忘れてしまっているということも生じる。

さらに視覚障害があるユーザーの学習コストを高くしていることは、各コマンドのショートカットキーのキー割り当てが、SRによって異なることである。一つのSRだけでも覚えきれないほどのコマンドを有するようになってきているにも関わらず、二つのSRを使用しようとする、2倍の学習をしなければならないのである。

例として、PC-Talker Neo(*5)とNVDA日本語版(*6)のごく限られたコマンドのみだが、ショートカットキーのキー割り当ての具体例を、表1に示す。まず、SRのフォーカスが当たっているアプリ（現在アクティブなウィンドウのタイトル）を確認するコマンドのショートカットキーについて、PC-Talker NeoではCTRL + ALT + 1に、また、NVDAではNVDAキー + tに割り当てられている。また、ワープロソフトなどにおいてカーソルがある行の1行全体を読み上げるコマンドのショートカットキーについて、PC-Talker NeoではCTRL + ALT + f、NVDAではNVDA + 1に割り当てられている。また、カーソルが当たっている文字の詳細読みを行なうコマンドについて、PC-Talker NeoではCTRL + ALT + m、NVDAではNVDA + ピリオド2回に割り当てられている。このように、SRは様々なコマンドを有し、ショートカットを設定しているが、SRごとにキー割り当てが異なるのである。

表1

	PC-Talker Neo	NVDA
SRのフォーカスが当たっているアプリの確認	CTRL + ALT + 1	NVDAキー + t
カーソルがある行の1行読み上げ	CTRL + ALT + f	NVDA + l
カーソルが当たっている文字の詳細読み	CTRL + ALT + m	NVDA + ピリオド2回

4. Web会議ツールに設定されるキーボードショートカット

近年、UIデザインは、直感的操作を重視する傾向にある。ここでいう「直感的操作」が、明言することなく前提としていることは、画面を目で見てマウスで操作するという操作方法である。従って、直感的操作を重視するUIを、それ以外のインターフェイスで操作する場合は、直感的に操作可能とは限らない。それが先鋭化して表

れるのが、SRで操作する場合である。SRには、音声を耳で聞いてキーボードで操作するという方式が採用されており、画面を目で見てマウスで操作するという前提が除外されているからである。

そこで、直感的操作を重視するUIを採用するアプリには、キーボード操作を可能にするために、ショートカットが設定されている。キーボードショートカットは、SRがアプリ内のどのオブジェクトやコントロールにフォーカスしている常体であっても、二つか三つのキーを同時に入力することによって、そのキーの組み合わせに割り当てられた機能が実行されるというものである。

ここに、ユーザーには、予めこれらのキーボードショートカットのキー割り当てを覚えておかなければならないという学習コストが発生する。このキーボードショートカットは、当該アプリに実装されている機能にのみ設定されるものであるから、数はそれほど多くはない。ただ、これらのキーボードショートカットは、利用するアプリに個別に設定されているものであるから、一つのアプリについて覚えたとしても汎用性はない。利用するアプリの数だけ学習しなければならないのである。

例として、Zoom(*7)とMicrosoft Teams(*8)のみだが、Web会議ツールにおけるキーボードショートカットのキー割り当ての具体例を、表2に示す。まず、オーディオのミュートとミュート解除は、zoomではalt + a、teamsではctrl + shift + mが設定されている。ビデオの開始/停止は、zoomではalt + v、teamsではctrl + shift + oに設定されている。画面共有の開始/停止は、zoomではalt + s、teamsではctrl + shift + eに設定されている。チャットを表示/非表示は、zoomではalt + h、teamsではctrl + nに設定されている。このようにWeb会議ツールにはキーボードショートカットが設定されているが、アプリごとにキー割り当てが異なるのである。

表2

	Zoom	Teams
オーディオのミュート/ミュート解除	alt + a	ctrl + shift + m
ビデオの開始/停止	alt + v	ctrl + shift + o
画面共有の開始/停止	alt + s	ctrl + shift + e
チャットを表示/非表示	alt + h	ctrl + n

5. SR環境下でWeb会議ツールを利用するUX

SRは様々なコマンドを有し、ショートカットキーが設定されていることから、使用するにはユーザーに一定の学習が求められる。そのため、ユーザーには1種のSRで全てのアプリを利用したいというニーズを生じる。アプリにキーボードショートカットを設定することでSRによる操作を可能にするアプローチは汎用性がなく、ユーザーの学習コストがさらに高いとしても、以前から行なわれていた。そのため、アプリについてもまた、ユーザーには、ワープロやブラウザなど1種のアプリについては一つのアプリで利用した

いというニーズを生じる。

1種のSRで全てのアプリを利用できるという結果を得られるからこそ、SRユーザーはコスト高であるとしても学習に努めてきた。確かに、各種ワープロソフトやブラウザのように、使用するアプリを自分で決めることができる場合は、ユーザーの学習コストが高いとしても、学習という投資と、SR環境下での各種アプリの利用という結果は、危ういながらも均衡をたもってきた。しかし、コロナ禍以降、この均衡が崩壊するほどにユーザーの学習コストは著しく高くなった。

SRユーザーの学習コストの増加を顕著に示すものがWeb会議ツールである。SR環境下で、キーボードショートカットによってSRによる操作を可能にするアプローチが採用されたアプリを利用することは、高い学習コストが求められるため、ワープロソフトやブラウザのように、一種のSRとキーボードショートカットを学習済みの1種のWeb会議ツールとの組み合わせで、全てのWeb会議をできることが望ましい。しかし、Web会議ツールは個人で決めることはできず、会議の参加者との兼ね合いで選択されるため、場合によっては、SRユーザーがキーボードショートカットを学習していないWeb会議ツールが選択されることもある。

初めてのWeb会議ツールを利用するときの学習コストは、画面を目で見てマウスで操作する人の場合にも生じる。そうはいつても、直感的操作を重視するUIデザインが採用されていることで、画面を目で見てマウスで操作する者にとっては、初めて利用するWeb会議ツールであっても、提供されている機能をある程度利用することが可能だろう。しかし、SRの音声を耳で聞いてキーボードで操作する者にとって、予めキーボードショートカットを学習していないWeb会議ツールを利用することは、非常に難易度が高い。その結果、利用頻度が高いものも低いものも、複数のWeb会議ツールのキーボードショートカットを学習しなければ利用困難という事態を生じている。

キーボードショートカットを入力したとしても、そこにあるオブジェクトやコントロールが様々であり、目的とするものを選択する操作、実行する操作もまた様々である。さらには、Web会議ツールを初見で利用しなければならないときや、キーボードショートカットを学習していない場合、Tabキーやカーソル矢印キーなどのホットキーや、SRのコマンドを駆使して操作することになる。こうして操作できることもあるが、必ずしも意図するおりに操作できることばかりではない。こうした状況で頼りになるのは、過去にユーザーが、SRで利用することを想定しないで設計されたWebやアプリをどれだけ試行錯誤して利用してきたかという経験知の蓄積である。ショートカットの学習と共に、この経験知こそがSRユーザーのICTスキルの中核を構成する。この試行錯誤は、本来、Webやアプリの設計者がアクセシビリティに配慮して構築すれば発生しないものであり、SRユーザーが直感的UIを利用するときには、Webやアプリの設計者の配慮不足がユーザーに転嫁されることによって発生する学習コストまで担っているのである。

一方、SRについても、Web 会議ツールが変わることで、汎用性が高いはずのSRにまで1種では済まないという事態を生じている。Teamsのように世界的に利用されるアプリがアクセシビリティ対応を進めるとき、SRとしてサポートされる支援技術は、多言語で利用可能なNVDAとJAWSである日本語圏でしか利用されないPC-Talker Neoは、サポートされない。そのため、当該Web会議ツールを、NVDAやJAWSであれば操作できるが、PC-Talker Neoでは操作できないことがある。そこで、PC-Talker Neoユーザーであっても、NVDAやJAWSを使用せざるを得ない状況が発生する。

そうであれば、日常的に使用するSRとしてPC-Talker Neoを止めて、NVDAやJAWSにすればよいとも考えられる。しかし、日本語への対応は、NVDAやJAWSよりもPC-Talker Neoの方が充実しているため、日本語での利用が多い者には、PC-Talker Neoの方が利便性は高い。そこで、Web会議ツールを利用する機会が増えたことによって、日本語での利用が中心であり、日常的にはPC-Talker Neoを使用する者であっても、NVDAやJAWSを併用せざるを得ない状況が生じた。その結果、汎用性が高いはずのSRについても、複数のSRのショートカットを学習しなければならないようになったのである。

直感的操作を重視するUIをSRで利用する場合のUXは、様々な要素から構成されている。そのうちの一つの要素は、SRのショートカットの学習と、アプリのキーボードショートカットの学習との乗算によって形作られている。そして、ここまでの記述が示していることは、この積が、SRユーザー個人の努力によって到達可能な程度を超え始めているということだ。

6. まとめ

以上から、まず確実にいえることは、Webやアプリの設計者の配慮不足がSRユーザーに転嫁されることによって発生している学習コストは、本来担うべき者が担うようにすることが重要である。直感的UIとして、画面を目で見てマウスで操作する人に対して配慮するのと同程度に、アクセシビリティとして、音声を耳で聞いてキーボードで操作する者に対しても配慮されるべきである。問題は、配慮の方法である。

SEのようにコンピュータ関連の業務を行う人の場合は、これだけの学習コストを投資しても、得られる結果が業務に結びついていくという面もあるだろう。しかし、何らかの業務を推敲するための道具としてICTを利用する者にとって、ショートカットキーによってアクセシブルにするというアプローチは、SRユーザーの学習コストが高すぎるとはいえないか。ここまでの記述に基づいて、以下3点を指摘する。

1点目は、SRのコマンドに対するショートカットキーの学習についてである。SRは、場合によっては複数のSRを使い分けなければならない状況を生じてきており、SRごとにキー割り当てが異なることが、SRユーザーの学習コストを高くしている。そこで、同じ機能については全てのSRにおいて統一してはどうか。

2点目は、直感的UIを採用するアプリのキーボードショートカットの学習についてである。複数のアプリのキーボードショートカットを学習しなければならないことが、SRユーザーの学習コストを高くしている。そこで、同じ機能については、全てのアプリにおいて統一してはどうか。

3点目は、直感的操作を重視するUIをSRで利用可能にするためのアプローチについてである。2点目に記した案が実現できたとしても、Web会議ツールやワープロなど、アプリの種類ごとにキーボードショートカットを学習しなければならないことは変わらない。そこで、直感的UIをキーボードショートカットによってSRでの操作を可能にするというアプローチを見直してはどうか。

1点目と2点目については、開発者間の調整によって実現可能性がないわけではない。しかし、3点目については、新たなアプローチを考案するということであり、即座に解決できるものではない。代替案を持っているわけではないが、まずは問題提起する。

注

- (*1) 「PC-Talker Neo, Neo Plusのご案内」 <https://www.aok-net.com/news/pctkneonews.html>
- (*2) 「NVDA 日本語版 ダウンロードと説明」 <https://www.nvda.jp/>
- (*3) 「あらゆるデバイスからの対面ミーティング」 <https://explore.zoom.us/ja/products/meetings/>
- (*4) 「ビデオ会議、ミーティング、通話」 <https://www.microsoft.com/ja-jp/microsoft-teams/group-chat-software>
- (*5) PC-Talker Neoのショートカットキーは、同SRがインストールされたPCにおいてAOKメニュー内に掲載されている。
- (*6) 「NVDA 2023. 3. 3.jp コマンド一覧表」 <https://www.nvda.jp/nvda2023.3.3.jp/ja/keyCommands.html>
- (*7) 「ホットキーとキーボード ショートカット」 https://support.zoom.com/hc/ja/article?id=zm_kb&sysparm_article=KB0067056
- (*8) 「Microsoft Teams で使用するショートカット キー」 <https://support.microsoft.com/ja-jp/office/microsoft-teams-%E3%81%A7%E4%BD%BF%E7%94%A8%E3%81%99%E3%82%8B%E3%82%B7%E3%83%A7%E3%83%BC%E3%83%88%E3%82%AB%E3%83%83%E3%83%88-%E3%82%AD%E3%83%BC-2e8e2a70-e8d8-4a19-949b-4c36dd5292d2>