How Inclusive is Conceptual Modeling? A Systematic Review of Literature and Tools for Disability-aware Conceptual Modeling

Aylin Sarioğlu, Haydar Metin, Dominik Bork

To appear in:

42nd International Conference on Conceptual Modeling (ER 2023)

How Inclusive is Conceptual Modeling? A Systematic Review of Literature and Tools for Disability-aware Conceptual Modeling

Aylin Sarioğlu, Haydar Metin, and Dominik Bork[0000−0001−8259−2297]

TU Wien, Business Informatics Group, Favoritenstrasse 9-11, 1040 Vienna, Austria
`dominik.bork@tuwien.ac.at`

Abstract. The reports on Disability by the World Health Organization show that the number of people with disabilities is increasing. Consequently, accessibility should play an essential role in information systems engineering research. While software and web engineering research acknowledge this need by providing, e.g., web accessibility guidelines and testing frameworks, we show in this paper, based on a systematic review of the literature and current modeling tools, that accessibility is, so far, a blind spot in conceptual modeling research. With the paper at hand, we aim to identify current research gaps and delineate a vision toward more inclusive, i.e., disability-aware conceptual modeling. One key finding relates to a gap in research and tool support concerning physical disabilities. Based on these results, we further present the first modeling tool that can be used keyboard-only, thereby including users with physical disabilities to engage in conceptual modeling.

Keywords: Conceptual Modeling · Accessibility · Disability · Modeling tools · Systematic Literature Review · Tool Review.

1 Disability in Information Systems Engineering

As the world's population continuously grows, the number of people with disabilities also increases. Over the years, the World Health Organization (WHO) has published different reports on disability, the latest article [44] states, that about 16% of the world's population live with some form of disability. Each disability is as unique as the person who is affected by it. Some are minor and temporary, while others are more severe and long-lasting. Disabilities can be grouped into five categories (see Table 1), while it is not always possible to assign disabilities to one of them, as multiple disabilities, changing abilities, and situational limitations also exist. People with disabilities often face obstacles in different aspects of their everyday lives - and using information systems is one of them.

Recently, information systems moved from heavy-weight desktop applications to lightweight Web-based applications that run in the browser or browser-like client applications. The Web is widely used for sharing and exchanging information. It is a constantly expanding and evolving system, so creating websites and web applications that everyone can access is essential. The Web is designed to work for all people, and that must include those people with disabilities, as stated by Tim Berners Lee in 1997 [36]: *"The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect."* Consequently, the term Accessibility is used in many diverse contexts and areas. In this paper, Web Accessibility is the main topic of discussion. The term Web Accessibility is defined by the Web Accessibility Initiative (WAI), a field of the World Wide Web Consortium (W3C), as follows [38]: *"Web accessibility means that websites, tools, and technologies are designed and developed so that people with disabilities can use them."* According to WebAIMs accessibility report of 2023 [42], about 96.3% of the top 1.000.000 home pages examined include Web Content Accessibility Guidelines (WCAG) 2 faults [40]. Further developments in web engineering are increasing the complexity of online pages, making the attainment of accessible websites all the more difficult [28,42].

The software and web engineering communities have already acknowledged the importance of accessible applications and started to use, e.g., the aforementioned web accessibility guidelines or test frameworks to make their applications accessible to a broader audience. One example is the WCAG standard [37] which aims to guide designers, software and web programmers to achieve digital accessibility for their applications. Additionally, it is possible to evaluate the state of digital accessibility for existing software products. The WCAG system classifies how well an application conforms to the standard. However, it would also be possible to integrate accessibility aspects during the software development process to pursue a disability-aware approach from the beginning [26]. Furthermore, literature about accessibility exists in software and web engineering (see Paiva et al. [28] for a recent systematic literature review). Moreover, several publications focus on the fundamentals of accessibility and disability (cf. [26,15,5,2]) and point out gaps and problems in the software & web engineering field. Others try to reach awareness for diversity by discussing or presenting ideas and solutions. There are publications that focus in particular on visual disabilities (cf. [23,21,18]), while others speak about disabilities in general (cf. [43,31,19]). Other disability categories were not well represented.

Contrary to the field of software & web accessibility, the accessibility situation is different in conceptual modeling, where accessibility—as we will show throughout this paper—is so far a blind spot. This not only hampers modeling to be an inclusive discipline that accounts for the diversity and heterogeneity of modelers, it even excludes many people having disabilities from engaging in conceptual modeling. State of the art in disability research in conceptual modeling literature and the accessibility of current web modeling tools are systematically surveyed and analyzed in Sections 2 and 3, respectively. Based on the identified research gaps, a research agenda toward disability-aware conceptual modeling is sketched in Section 4. In Section 5, we make a first contribution toward mitigating one of the identified research gaps by presenting the realization of the first keyboard-only web modeling tool that enables humans with physical disabilities to engage in conceptual modeling. Finally, we conclude this paper in Section 6.

Table 1. Classification of disability types

Disability	**Description**
Auditory	A person experiencing different extents of hearing loss.
Cognitive, Learning, & Neurological	A person experiencing neurodiversity, neurological disorders, behavioral, or mental changes. This may affect any part of the nervous system, such as speaking or hearing ability, or problems in comprehending information.
Physical	A person experiencing impeded movement, sensation, or control caused by muscular weakness, pain, limitation or lack of coordination, joint disorders such as arthritis, or missing limbs.
Speech	A person with a disability to speak clearly and be comprehended by others (e.g., difficulties in loudness or clarity of speech).
Visual	A person experiencing different extents of vision loss in one or both eyes (i.e., “low vision“), severe and uncorrectable vision loss in both eyes (i.e., “blindness“), or lack of sensitivity to brightness or (specific) color (i.e., “color blindness“).

2 Disability Research in Conceptual Modeling

This section reports the structure and the findings of a systematic literature review (SLR) [17,29] that explores the current state of research on (web) accessibility in conceptual modeling. A detailed version of the SLR steps and results can be observed in this submission's supplementary material[1]. The SLR shall respond to these research questions:

- **RQ-1: What is the state of research and its evolution regarding web accessibility in the field of conceptual modeling?**
- **RQ-2: Which disabilities are covered in existing literature?**
- **RQ-3: Which solutions are proposed to improve accessibility?**
- **RQ-4: What is the current state of web modeling tools in terms of their support for individuals with disabilities?**

To respond to these research questions, a search string was defined that combines keywords about disability and web accessibility (D) with conceptual modeling keywords (CM). The query was not constrained to specific years, was focused to find matches in any or all of the *Title*, *Abstract*, and *Keywords* of the publications, and was run on 15.05.2023 in the scientific databases Scopus, IEEE, and ACM. We further used two seminal works (cf. [19,45]) to conduct an additional search for relevant papers using ConnectedPapers [7].

$$\text{Query} = \left(\bigvee CM_i\right) \wedge \left(\bigvee D_j\right)\ \boldsymbol{where}$$

$$CM_i \in \{\textit{"Modeling Method"} \vee \textit{"Modelling Method"} \vee \textit{"Modelling Tool"} \vee \textit{"Modeling Tool"} \vee \textit{"Diagram Tool"} \vee \textit{"Modeling Editor"} \vee \textit{"Modelling Editor"} \vee \textit{"Diagram Editor"} \vee \textit{"Web Modeling"} \vee \textit{"Web Modelling"} \vee \textit{"Editor"}\}$$

$$\boldsymbol{and}$$

$$D_j \in \{\textit{"Accessibility"} \vee \textit{"Disabilit*"} \vee \textit{"Impairment*"} \vee \textit{"Accessible Internet"} \vee \textit{"WCAG"} \vee \textit{"Web Content Accessibility Guideline"}\}$$

---

[1] `https://drive.google.com/drive/folders/1ydHlKYoIYqc2QglPIBMSpDHeyS5skuln?usp=sharing`

The number of documents retrieved and filtered throughout this process is illustrated in Fig. 1. The performed search has led to **690 publications** in total and **495 publications** after removing duplicates. The following exclusion criteria (EC) were applied to eliminate irrelevant publications and to facilitate the subsequent steps: **EC-1:** Non-English publications; **EC-2:** Publications not related to the subject areas Computer Science or Engineering; **EC-3:** Publications with less than four or more than 60 pages; **EC-4:** Publications that are not accessible as full text or are non-scientific papers (e.g., posters, extended abstracts). After applying these ECs, we were left with **313** potentially relevant papers. Around 50% of them were published between the years 2015 and 2023.

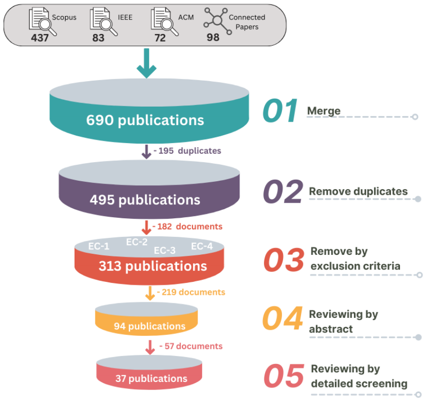


Fig. 1. Search and filtering steps

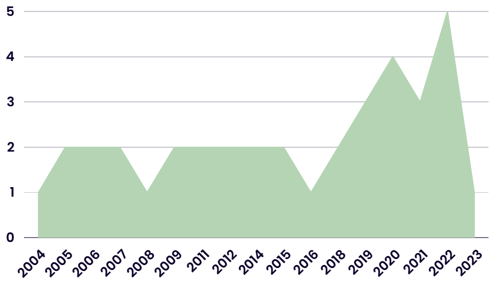


Fig. 2. Relevant documents per year

2.1 Screening of Publications

In this step, the goal was to identify publications that fit this research's purpose depending on their abstract. A first categorization of the possible relevant papers was conducted into the categories: **related to accessibility & conceptual modeling**; **related only to (web) accessibility**; and **not directly relevant to conceptual modeling**. The latter category was used if a publication or its subject area is not directly related to conceptual modeling but accessibility solutions are discussed which could also be useful for other domains.

After reading the abstracts, **94** out of the **313** potentially relevant publications remained potentially relevant. An initial grouping of these **94** publications indicates that exactly half of them (47) are related to topics about web accessibility. The other half can be split further: roughly half deal directly with accessibility in conceptual modeling (23 publications), and the remaining (24) are not directly related to conceptual modeling. Nevertheless, they were included because of their valuable insights that could be potentially useful in other areas if applied to this field. Eventually, we read the entire paper to select only the really relevant studies for our research scope. This led to a total of **37** eventually relevant publications. These papers have then been analyzed according to

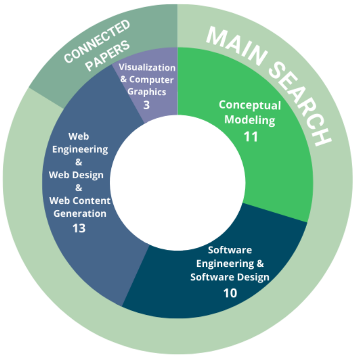


Fig. 3. Number of publications based on publication source and subject area.

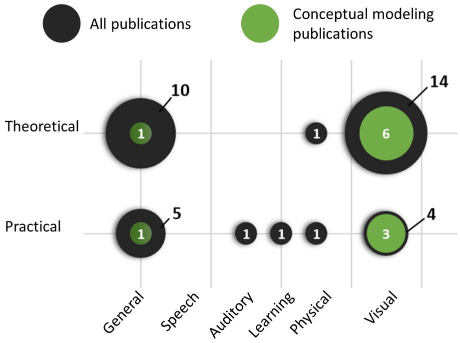


Fig. 4. Overview of the disability types combined with the contribution type.

different aspects to respond to the research questions stressed at the outset. In Fig. 3 it can be observed that most publications deal with (web) accessibility in general. Only a few works have been published specifically in the field of conceptual modeling, especially if we consider that there was no restriction in the year of publication. Interestingly, the use of ConnectedPapers proved very valuable as it contributed additional potentially relevant publications.

2.2 Findings

The findings are categorized by the **article metadata** (i.e., year and subject area) to respond to RQ-1, the covered **disability types** to respond to RQ-2, and the **type of proposed solutions** to respond to RQ-3. For the latter, we distinguish **(Theoretical)** used for all publications which introduce discussions, methods, prototypes, and possible solution approaches without technical artifacts or implementations; and **(Practical)** used for all publications which propose implementations, tools, and similar technical artifacts.

RQ-1: The publications in the area of web accessibility have increased significantly over the years (cf. Fig. 2). The topic gains in relevance with an increasing number and diversity of available publications. Fig. 3 show the distribution of the documents in specific subject areas and their source. It can be observed that around 84% of the eventually relevant papers originate from the search query while the other 16% originate from the search via ConnectedPapers. The majority of the documents were published in the subject areas of Web Engineering & Web Design & Web Content Generation (a total of 13), and Conceptual Modeling (a total of 10). It can be derived that our query was exhaustive with respect to the core focus on research at the intersection of disability and conceptual modeling, while several relevant works were found in adjacent domains like disability and web engineering through ConnectedPapers.

RQ-2: Fig. 4 shows a categorization of the state of the art of research on disability in conceptual modeling using the different disability categories (cf. Table 1) and the contribution type (i.e., theoretical or practical). The black bubbles represent the total number of publications selected as relevant, whereas the green bubbles present how many out of the total publications are related specifically to conceptual modeling. Notably, most relevant publications handle visual disabilities, with 14 theoretical publications and 4 with actual technical artifacts, including tools or implementations. About ten publications generally focus on disabilities without concentrating on a specific disability category. For the remaining four disability types almost no contributions exist. A majority of the publications provide a theoretical contribution. On the other hand, observing the publications related to conceptual modeling leads to the fact that only 11 of 37 selected papers are specifically handling modeling or modeling tools. Here, we can see that two contributions generally refer to disabilities without focusing on a particular type (marked in Fig. 4 as General, 1/10 & 1/5), and the rest explicitly targets visually disabled users (marked in Fig. 4 as Visual, 6/14 & 3/4). Furthermore, there was no existing research in the areas of *Physical*, *Auditory*, *Speech*, or *Learning* disabilities in the context of conceptual modeling.

RQ-3: The relevant papers have various foci. One central aspect of achieving accessibility is that this cannot be done additionally or at the end of a software development process. There is instead the need to address accessibility right from the beginning. Njangi et al. [26] present methods which should help to integrate the fulfillment of accessibility requirements into the software development phase. As Brophy et al. [5] presented in their publication, insufficient accessibility coverage can be detected using user-focused approaches to evaluate accessibility. This approach aims to ensure that disabled users are not overlooked. One reason for that, according to Kavicic et al. [15], is that traditional development processes assume that users do not possess any impairment.

Manual accessibility assessments are associated with increased effort. Weber et al. [43] therefore analyzed the potential for automation which yielded two significant limitations: *measuring readability* and *predictability of navigation* are subjective measures. This also leads to the question of whether there are existing features, especially in terms of Content Management Systems, that provide a more effortless and automatic way of achieving compliance with accessibility guidelines [31]. The authors state that this could be possible theoretically, however, they assess that the current technology is not prepared yet.

Some authors proposed the use of textual concrete syntaxes for visually impaired users (i.e., blended modeling). Luque et al. [19] discuss and evaluate how to make UML diagrams accessible for blind users and, therefore, evaluate the use of textual concrete syntaxes, as these textual models could be used in combination with screen readers and text-to-speech applications. An approach to using audio as a means to represent models was proposed by Metatla et al. [21].

The analysis shows, that the majority of the relevant publications (25 out of 37) mainly discuss different theories or propose potential methods. Only 12 out of 37 relevant publications present any existing tool or implementation (see Fig. 4). Furthermore, it is distinctly observable that the number of contributions specifically targeting conceptual modeling is relatively low, with only 11 out of 37 publications, with only four papers proposing practical input in this research field (see Fig. 4). Additionally, there is a clear gap regarding implementations and solution contributions specifically targeting physical, learning, auditory, and speech disabilities, as no relevant and suitable papers were found here.

RQ-4: This research question will be answered by examining ten well-known web modeling tools regarding the provided disability support. The analysis and findings of the assessment provide a comprehensive response to this research question. Section 3 elucidates these tools' strengths, limitations, and overall performance, offering insights into their effectiveness within the study's objectives.

2.3 Synopsis

The outcomes of this SLR show that accessibility is of increasing importance. While the web and software engineering communities made significant contributions with standards, methods, and tools, conceptual modeling research is currently scarce and focused on visual disabilities. Most existing papers on conceptual modeling analyze and present the detected issues that most disabled users have to deal with. This is achieved by providing evaluations and literature reviews, as done by e.g., Torres et al. [34] who reviewed the contributions in the field of accessibility in modeling for the visually impaired and found out that there is a research and solution gap in this field. Luque et al. [19] highlight the challenges visually impaired users may have while working with UML diagrams. Seifermann et al. [33] provide a survey that evaluates textual notation alternatives to replace existing graphical notations. On the other hand, some contributions provide systems that target specific disability types or solutions that only work for some modeling languages. As an example, several publications [16,32,18,8] target visually impaired and blind users and UML diagrams by presenting accessible systems or interfaces that make use of textual alternatives and editors. They aim to make the model content readable by screen readers or similar. Others provide auditory interfaces to allow operations via sound [35,21] and gestures [6], or want to avoid textual syntax as the only way and provide physically accessible prototypes [45].

In conclusion, the SLR has highlighted the research gaps in realizing more inclusive conceptual modeling. Based on the potential of web accessibility and the trend, that modeling tools move into the web, we review, in the following, current web modeling tools with respect to the extent to which they support modelers with disabilities.

3 Disability-Awareness in Web Modeling Tools

This section reports the findings of an in-depth evaluation of ten well-known current web-based UML modeling tools. The assessed tools serve as representatives of web modeling editors sourced from a curated list from the modeling community [22]. Naturally, the exhaustive examination of every existing modeling editor is unfeasible; thus, this compilation highlights ten web tools with a primary focus on UML, renowned for their widespread usage and established utility in modeling communities. Further details and references about the used web modeling tools can be observed in this submission's supplementary material[1]. When analyzing the tools, we used criteria proposed by the Web Accessibility Initiative (WAI) [39]. These criteria (see Table 2) provide a reasonable objective basis and summary of the most important aspects that should be provided by inclusive information systems like modeling tools. The criteria is structured along the disability dimensions *Visual Disabilities (V)*, *Cognitive, Learning, Neurological (CLN)*, and *Physical Disabilities (P)*. Finally, in Section 3.3, we describe the findings derived from assessing the tools with respect to the criteria.

3.1 Assessment Process

The evaluation has been conducted through *Observations* and *Experiments*. Each of the selected web modeling tools was evaluated according to whether it **satisfactorily**, **partially**, or **not** fulfills the evaluation criteria listed in Table 2. The tools were individually evaluated using the Google Chrome web browser. Consequently, the outcomes are based on the subjective judgment and expertise of the evaluator. Even if the evaluation would have been conducted with a group of disabled users, it would include subjective bias, as every person's disability is unique, and how they are affected by their limitations and barriers would influence the outcomes. A certain degree of bias is inevitable, but steps can be taken to mitigate its impact and ensure the analysis is as objective as possible. It is thus critical to establish well-defined evaluation criteria guiding the assessment in a transparent way and minimizing subjective bias through individual interpretation. The used evaluation criteria are based on objective evidence and standards, as their content is provided by the WAI, which is professionally dealing with accessibility and therefore has the necessary knowledge and experience in this particular area. Moreover, the evaluation was supported with various browser extensions for testing the accessibility of a given tool as described in Section 3.2. Furthermore, each step was noted in detail for better transparency. The following steps were followed for each of the assessed web modeling tools:

Check conditions of the evaluation criteria: The fulfillment for each criteria listed in Table 2 is assessed. Each assessment criteria is applied against *i*) the *Tool support & Graphical User Interface*, and *ii*) the *Canvas & Model* of the modeling tool. This step is supported by Browser extensions (cf. Section 3.2 for details on the tool and the assessment).

Check satisfaction of disability needs: This step is required to determine if a certain tool satisfies most of the disabled user needs for a disability type – a meta-assessment based on the fulfillment degree of each relevant criterion.

Determine Results: The end results were determined using a scoring system. *Satisfactorily fulfilled* is worth 1 point, *partially fulfilled* 0.5, and *not fulfilled* -1. The assessed web modeling tool *i*) **satisfies** the needs of a given disability type if the end score is positive, *ii*) **partially satisfies** the needs of a given disability type if the end score is zero, *iii*) **does not satisfy** the needs of a given disability type if the end score is negative. The assessment end result can be observed in Table 3. The detailed version of the table is available in this paper's submission supplementary material[1].

Table 2. Classification of disability types

V1-Customizing text and images size. The possibility to enlarge or reduce text or image sizes according to the user's needs. This criterion is partially fulfilled if only specific texts or images are resizable. It is not fulfilled if at least one visible element or area is not resizable.
V2-Customizing fonts, colors, and spacing. The color, spacing, and font impact the perception of specific visual impairments. This criterion is fulfilled or partially fulfilled if all or some of the mentioned features are customizable, respectively, or not fulfilled otherwise.
V3-Text-to-speech content synthesis. Visible elements should be recognizable by text-to-speech applications. The criterion is assessed as not fulfilled if at least the text-to-speech synthesizes of one visible element is not recognizable.
CLN1-Clearly structured content. This criterion describes the need for a structure that facilitates overview and orientation. This criterion is partially fulfilled if some element positions could be misleading. It is not fulfilled if the overall impression seems confusing and orientation is inefficient.
CLN2-Consistent labeling. The forms, buttons, and other content parts should have corresponding labels. This is important to make the content understandable, avoid misinterpretations, and make it accessible to text-to-speech readers. This criterion is not fulfilled if labels are missing. It is partially fulfilled if there are up to two elements without a label.
CLN3-Predictable interaction. This criterion describes that the outcomes of user interactions should be predictable, i.e., should do what it has indicated. This criterion is partially fulfilled if up to two interactions resulted in unexpected behavior and not fulfilled if this is more than three.
CLN4-Different navigating means. Using different navigational structures, e.g., hierarchical menu and search, allows users to use the most appropriate option for them. This criterion is fulfilled if at least two options, partially fulfilled if only one option is provided, and not fulfilled if e.g., only scrolling long options is available.
CLN5-Options to suppress distracting content. Animations or visual indicators, e.g., blinking and flashing, can be distracting. This criterion is fulfilled if there is an option to suppress distractions.
CLN6-Text supplemented by illustrations. Textual parts should have images, graphs, and similar supplements to improve comprehension. This criterion is only fulfilled if textual elements have a supplementary illustration.
P1-Full Keyboard Support. All possible interactions should be doable with the keyboard only (i.e., without a pointing device). The assessment of this criterion relies on the walkthrough option for manual accessibility checks, based on WCAG 2.1.
P2-Sufficient time limits to react. A person's reaction time should not lead to errors, interruption of the current task, or similar. This criterion was assessed by carrying out the same modeling actions at different speeds. This criterion is not fulfilled if at least one main modeling feature depends on the user's reaction time. It is partially fulfilled if this feature has corresponding alternatives for the same action or is not one of the main features.
P3-Controls, images, etc. with text alternatives. This criterion depends on alternative texts and ARIA labels to enable voice recognition. This criterion was not fulfilled if there were more than five (partially fulfilled if less than five) text alternatives or ARIA labels missing.
P4-Visual & non-visual orientation or navigational cues. This criterion is essential to navigate and make the current location/selection visible. It is not fulfilled if the cursor location or the marking of the elements is not visible, or there is no/only poor visual feedback or insufficient navigation support.
P5-Logical navigational mechanisms and page functions. This criterion describes that the page structure should not be misleading or that navigating does not show unexpected behavior. It is fulfilled if navigating is logical, i.e., in an expected and natural order.
P6-Large clickable areas. This criterion describes that the clickable area of action is large enough, so no fine motor skills are required. Additionally, the spaces between multiple elements are sufficient, so the probability of choosing the wrong one is low. This criterion is not fulfilled if there are multiple clickable traps.
P7-Error Correction Options. This criterion highlights the need of undo/redo or other correction options like deleting and renaming. This criterion is fulfilled if at least updating possibilities and redoing own actions is possible.

3.2 Assessment Criteria and Evaluation Software

We now first describe the assessment criteria for evaluating the tool support and the canvas & model before we introduce the software supporting the evaluation.

First, we evaluated each web modeling tool with respect to the provided support for each criterion by the **tool and its graphical user interface**. Generally, these kinds of tools' GUI contain a menu header, footer, and different side or panel menus, which include different interaction types. The outcomes for each tool can be observed in column *T - Tool Support & GUI* of Table 3.

Next, the evaluation was focused on the interaction between the **canvas and the diagram or model**. As the tools can deal with different diagram types and modeling languages, a default workflow was set beforehand to ensure the same process for each tool, thereby ensuring comparable results. The outcomes for each tool can be observed in column *CM - Canvas & Model* of Table 3. The default workflow was about creating a simple test UML class diagram if applicable, i.e., two test classes with properties and a relationship between them with multiplicities as a starting point. An exception was made for two web modeling tools, namely Miro, which only supports UML class diagrams in the premium version and BPMN.io not supporting UML diagrams, so a diagram was created using similar shapes and relations. Secondly, Diagramo only supports UML state diagrams, so this was used instead. The basic workflow was to check the CRUD functionalities of the diagram.

As not every evaluation criterion can be manually checked, **additional software**, primarily browser extensions, were used to automate the assessment. References to the used software can be found online[1]. The overview below describes how each extension was applied and for which evaluation criteria it was used.

Magnifying Glass (Hover Zoom): This extension can be used on any page as an additional aid to increase the size of elements or text, especially if the web pages do not provide a (satisfying) resizing functionality. It was used for the evaluation of **V1** to check if additional zooming aids can be applied to the tool without any loss of information or unexpected behavior.

OneLine: This tool is a reading aid extension that highlights the first row of the corresponding web pages to help disabled or impaired users by increasing their focus and reading efficiency. It was used for the evaluation of **V1** and **V2** to check if the tool allows its usage in a reasonable and efficient way, without the loss of information or making the web content unusable.

Read Aloud: This browser extension is a text-to-speech tool and is applied to check whether the evaluation criterion **V3** is fulfilled. In that case, this extension can check if the provided model's content is suitably prepared and if the content can be read aloud to the user so that it makes sense and also if it is possible to use the tool sufficiently.

WAVE & Accessibility Insights for Web: Both extensions can automatically evaluate a given web page/web modeling tool and create an accessibility report for known accessibility issues. The results are used as a combination, especially for the evaluation criteria **V3**, **CNL2**, and **CNL6**, e.g., to check if contrast errors, missing alternative texts, or descriptions exist.

Table 3. Tool Support & GUI (TG) and Canvas & Model (CM) for Cognitive, Learning, Neurological (CLN), Visual (V), and Physical (P) Disabilities.

Tool	Lucidchart		GenMyModel		Gliffy		diagrams.net		Creately		Cacoo		UMLetino		Diagramo		miro		BPMN.io	
	TG	CM	TG	CM	TG	CM	TG	CM	TG	CM	TG	CM	TG	CM	TG	CM	TG	CM	TG	CM
CLN1	●	●	●	●	●	●	●	●	◐	●	●	●	●	●	●	●	●	●	●	●
CLN2	◐	◐	●	◐	◐	◐	◐	●	◐	◐	◐	◐	●	●	○	●	○	◐	◐	◐
CLN3	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
CLN4	●	○	◐	◐	●	○	●	●	●	○	◐	○	○	○	◐	○	◐	○	●	○
CLN5	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
CLN6	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
V1	◐	●	●	●	◐	●	◐	●	◐	●	●	●	●	●	◐	●	◐	●	◐	●
V2	○	●	○	○	○	●	●	●	○	●	○	●	○	○	○	●	○	●	○	○
V3	○	○	●	◐	○	◐	○	○	○	○	○	○	●	○	○	○	○	○	○	○
P1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
P2	◐	●	●	●	◐	○	◐	◐	◐	◐	◐	●	◐	◐	◐	○	◐	●	●	●
P3	●	○	○	○	●	●	●	○	●	○	○	○	●	○	○	○	●	○	●	○
P4	◐	◐	◐	◐	◐	◐	◐	●	◐	◐	◐	◐	◐	◐	◐	◐	◐	◐	◐	◐
P5	◐	◐	◐	◐	◐	◐	◐	●	○	◐	◐	◐	●	◐	◐	◐	◐	◐	◐	◐
P6	●	◐	○	○	●	◐	●	◐	◐	◐	○	◐	●	●	●	◐	●	◐	●	●
P7	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

3.3 Findings

Table 3 shows the assessment results. It can be derived, that while there is sufficient support for cognitive, learning, neurological disabilities, support for physical and visual disabilities is insufficient if present at all in many tools.

Through this assessment, it is evident that especially the requirement *Full Keyboard Support (P1)* in the category of physical disabilities is not fulfilled at all by any of the evaluated web modeling tools. Furthermore, most tools could not return meaningful text-to-speech synthesizes if used with text-to-speech tools (*V3*). Moreover, the degree of customizability of tool settings, fonts, color, and contrast of the evaluated tools' GUI was unsatisfactory (*CNL5, V1, V2*).

It is also observable that some assessment criteria explicitly applied to the Canvas & Model perform differently than the GUI of the same tool, especially for the criteria *CNL2*, *CNL4*, and *V3*. The criterion *CNL2*, which assesses the sufficient labeling of the elements, shows that the labeling in the canvas or for the icons displayed together with the model elements is not sufficiently present. Furthermore, the criterion *CNL4*, which assesses the existence of different navigation types, is not fulfilled for the majority of the modeling tools in the area of canvas & model, as these tools do not offer a navigation option for created models. Furthermore, no tool offered a meaningful and helpful text-to-speech synthesis for the canvas & model (cf. *V3*). In conclusion, it can be said that especially for users with physical and visual disabilities dealing with the tools canvas and the created diagrams and models is not sufficiently possible.

4 Toward a More Inclusive Conceptual Modeling Future

The outcomes of the SLR (cf. Section 2) and the tool assessment (cf. Section 3) show that there is a blind spot in disability-aware conceptual modeling research aside from the apparent increasing importance and relevance. In the following, we present a research roadmap that aims to propose selected concrete avenues toward a more inclusive conceptual modeling future that would remove barriers for people with different disabilities and enable them to participate in modeling.

Visual Disabilities While there is research on e.g., improving visual notations of modeling languages [24,11,4] more research needs to be conducted and tools need to be improved to account for easy means to e.g., switch color schemes (color blindness), adjust font sizes and contrasts (visual impairment), or realize text-to-speech functionality such that alternative texts and ARIA labels for model elements can be read (blindness).

Physical Disabilities Research needs to address physical disabilities in conceptual modeling, especially because current modeling tools are heavily mouse-based. Virtual reality-based modeling (cf. [47,25]) tools might enable physically impaired people to engage in modeling. Additionally, advancing keyboard-only or audio-only interactions by smart support of complex modeling workflows (e.g., creation of edges) should be one area of future research.

Multiple and Complex Disabilities Disabilities are of course not binary. Different extents of e.g., visual impairment have different negative effects on the modelers. Moreover, problems increase when multiple impairments are given. Research needs to account for that by fine-tuning the techniques and tools to accommodate for the subjectivity adhering to disabilities.

Modeling Accessibility Assessment Research is necessary to prepare, similar to the web accessibility standards, frameworks, procedures, requirements, and tools for assessing the accessibility of conceptual modeling languages and tools. Thus, a revisiting of e.g., the seminal work by Moody [24] should be performed to assess a language's notation with respect to its accessibility.

Modeling Language & Tool Flexibility There has been a whole body of research on e.g., the flexibility in modeling tools, modeling notations, meta-modeling platforms, and the modeling process (cf. [9,13,14,27,30,46,3]). What is missing so far, and what our survey clearly shows, is flexibility with respect to e.g., alternative and customizable: *i*) representation of models (e.g., diagrammatic, audio, AR/VR, textual) and *ii*) interaction with models (e.g., keyboard-only, audio, AR/VR). The tool survey already showed a lack of basic customization features like changing font sizes, increasing contrast, etc.

Empirical Research Disability is per se human-centered. Consequently, a future research avenue needs to involve disabled persons. Languages and tools should be designed with disability in mind, and the resulting artifacts should, ideally, be tested for accessibility. This is of course not easy but necessary to truly involve disabled people in modeling.

The sketched research roadmap is by no means comprehensive. Instead, it should spark discussions by exploiting the research community in this very important but scarce researched area. Still, the few roadmap items already show the complexity of the many challenges toward disability-aware conceptual modeling. We believe convincing solutions to these challenges can only be achieved by collaborative measures, i.e., by building interdisciplinary teams composed of computer scientists (conceptual modeling, software engineering, human-computer interaction), social scientists, and maybe even medical scientists.

5 Toward Disability-Aware Web Modeling Tools - A Keyboard-Only Feasibility Study

When comparing the individual survey results, a research gap in the category of physical disabilities (see Fig. 4) and the missing fulfillment of the full keyboard support for all of the evaluated modeling tools (see *P1* in Table 3) inspired the implementation of a prototype in this area[2] This paper and the implemented prototype have an essential role as they form the building blocks for the first contribution in this direction, thus positively influencing further research and implementation in this area.

5.1 Motivation & Requirements

According to WAI [39], full-keyboard support for physically disabled persons is crucial, due to different types of mobility limitations, weaknesses, and limitations in muscular control and or pain that is involved in any kind of movement. Thus, using a keyboard over pointing devices, like a mouse, is often a better option as most of the needed movement, like dragging, moving, etc. is not applicable.

Our goal is not to improve efficiency, instead, we aim to realize a tool that enables persons with physical disabilities to fully engage in modeling. Modeling tools, traditionally heavily rely on a mouse for e.g., drag-and-drop interactions or the creation of edges, well-thought-through keyboard-only interaction possibilities are necessary. People with physical disabilities or impairments have difficulties or are even unable to use pointing devices (e.g., mouse), complex keyboard shortcuts, or to react fast in order to accomplish a task [15,39]. Thus, defining appropriate interactions require more awareness. Fortunately, keyboard interactions for HTML elements are already built-in [1] and developers can make use of it. This includes functionality like navigating through the page or interacting with controls by using the correct semantic HTML markup. However, not all developers pay attention to it, when developing custom functionality.

We developed our prototype together with the built-in functionalities and the keyboard accessibility developer guidelines by WebAIM [41] and MDN [20]. The following excerpt of the requirements was considered to fulfill the needs of keyboard-only web modeling.

- **Focus:** Only buttons, links, input fields, and custom interactive elements should be focusable to avoid leading users to elements, which cannot be interacted with and mislead or trap users in an unwanted state.
- **Navigation:** The tool should provide a mechanism to navigate through the model's content in a logical and intuitive way. Additionally, the focus from an element can be moved and is not trapped or locked there (cf. [37]).

---

[2] It should be noted that this prototype contribution does not necessarily mean that all people with physical disabilities can work entirely without limitations. Still, it should lower some essential barriers and therefore be more inclusive.

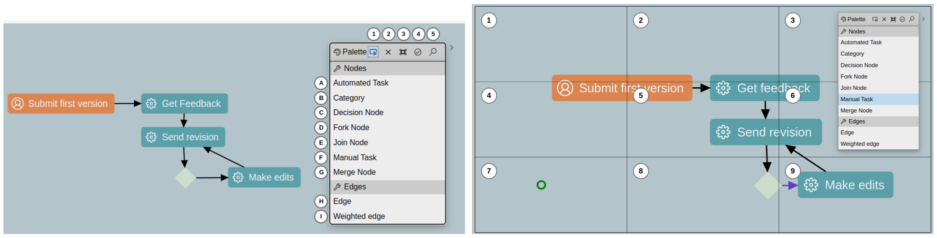
- **Shortcuts:** The selected shortcuts should allow easy and fast access to menus and functionalities. The shortcuts should be meaningfully designed, especially for frequently used actions and they should not conflict with standard keyboard shortcuts used by the operating system or assistive technologies (e.g., *CTRL+C* for Copying).
- **Visibility:** There should exist visual and non-visual orientation cues, page structure, and other navigational aids to help the user with better orientation and avoid misleading interactions. A clearly visible focus element should be ensured. This allows users to understand where they are and which element will receive their keyboard input next.
- **Consistency & Predictability:** Any interaction and functionality of the tool should provide the user with consistent and predictable behavior. Unexpected changes in behavior or focus can confuse or disorient users.
- **User Feedback:** In addition to providing keyboard shortcuts, there should be a mechanism that displays user feedback or information in real-time (e.g., short notifications about enabling/disabling a functionality), to keep the user informed about their interactions. These notifications should be clearly visible and not interfere with tool/model content or navigation.

Our prototype extends a Graphical Language Server Platform (GLSP) [12]-based workflow diagram editor with new keyboard interactions (see Section 5.2). GLSP is heavily used in industry and academia to realize web-based modeling tools with advanced visualization and interaction features (cf. [10]). The newly provided keyboard interactions aim to interact with the web modeling tool to accomplish a basic workflow of creating, editing, and observing a model.

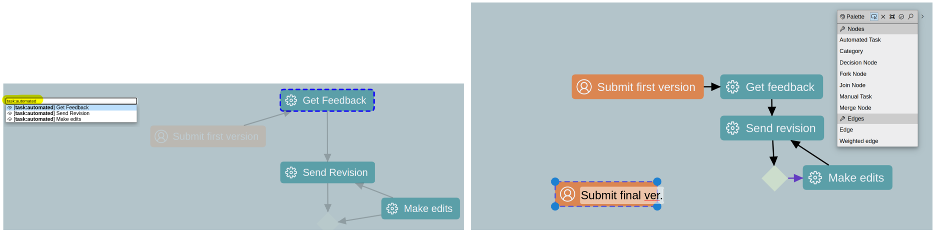
5.2 Modeling Operations

This section presents the most common interactions when working with modeling tools. The keyboard interactions assigned for these functionalities need to be intuitive, easy to understand and handle. We will introduce first basic *modeling CRUD operators*, followed by the functionality to *navigate* and to *explore* a model. Please note that a demo video showcasing how these functionalities work, can be found in this paper's submission supplementary material[1] .

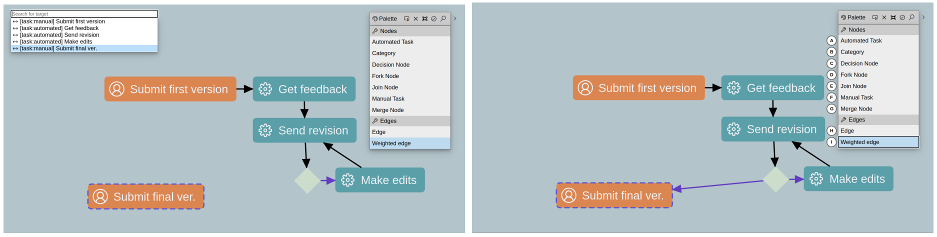
The following basic *modeling CRUD operators* have been conceptualized as keyboard-only interactions. The **tool palette** and its header menu are accessible using a shortcut (see Figure 5a). Afterward, the entries can be chosen by using the character keys or the header menu options via the numeric keys. The single-key shortcuts allow easy access to these frequently used actions. **Grid and pointer for node creation:** After selecting a node in the tool palette, a grid turns visible, where the modeler can choose the starting point of the pointer (i.e., a cursor) on the screen by using the numeric keys (see Figure 5b). Subsequently, moving the pointer using the arrow keys is possible, and finally, pressing the "enter" key finishes the node creation. The pointer also provides visual feedback on valid or invalid actions. **Edge auto-complete:** When creating an edge, an auto-complete palette appears that shows the valid source and target



(a) Tool Palette with Header Menu (b) Grid with Pointer



(c) Search with Highlight & Transparency (d) Rename



(e) Select node (f) Result

Fig. 5. Working with the keyboard-only modeling tool prototype

nodes of the selected edge and guides the user during the selection. The **search** functionality is one of the most essential interaction possibilities for a modeling tool, as models can get large quickly. Searching for specific labels, nodes, or edge types is a frequent task. In this particular prototype, one keyboard shortcut will reveal the search functionality and it is possible to intuitively search for nodes and edges. Afterward, the searched element will be focused and highlighted for further operations (e.g., renaming). The remaining elements of the model, which do not fulfill the search condition will become transparent (see Figure 5c).

Next, we describe the two different **navigation** algorithms we conceptualized to navigate within a model using the keyboard-only. The **default navigation** can be enabled via the shortcut *N* and allows to iterate through the models' nodes and edges via the *arrow keys* depending on the direction of the given relations. The **position-based navigation** can be activated with *ALT+N* and is used to iterate through the model via the *arrow keys* depending on their position in the canvas, i.e. their x and y coordinates without taking the relations and their directions into account.

Finally, we describe typical *model exploration* functionalities we conceptualized. To **move** a model's nodes or edges, the *arrow keys* can be used to move the selected node or the whole canvas in all directions. The selection can be done via the previously mentioned **search** functionality. To gradually adapt the **zoom** level of one element, a set of elements or the canvas, the shortcut *'+'* can be used to increase or *'-'* to decrease the zoom level. Additionally, with *CTRL+0* the default zoom level can be set and all other zoom activities will be reset. Furthermore, it is also possible to set the zooming level more refined, by displaying the grid and selecting a grid number in order to zoom in to the desired grid box (via *CTRL+'+'*). The **resize** functionality helps to set the size of the nodes. In most tools, the resizing action is accomplished by dragging the desired edge of the shape in another direction. To avoid this, we assigned a key shortcut to the resizing functionality. To activate the resizing mode *ALT+R* needs to be pressed. Afterward, *'+'* and *'-'* can be used to increase or decrease the size of the nodes' shape gradually. Via *CTRL+0* the default size of the node can be reset.

5.3 Workflow example

Fig. 5 shows a typical modeling workflow. The modeler wants to add a new node to the workflow diagram. First, she triggers the tool palette using the shortcut $ALT + P$ and a character (e.g., F) to select a specific node (Fig. 5a). This selection triggers the grid to become visible. Afterward, using a digit (e.g., 7), she places the pointer to the correct cell to finalize the new element creation by *enter* (Fig. 5b). Now, she can use the search to focus the element and press $F2$ to rename it (Fig. 5c-d). Lastly, she can connect the new element with the decision node by selecting the "weighted edge" in the tool palette and using the opened node selector to choose the source and target node for the new edge (Fig. 5e-f).

6 Conclusion

This article carries significant value by elaborating on the state of the art of accessibility research, and by sketching a research agenda for more inclusive, i.e., disability-aware conceptual modeling. Based on a systematic literature review of the literature and a selection of current web UML modeling tools, we establish a foundation for further research in this area. The observations showed that there are little to no contributions in research for the disability types physical, auditory, speech, and learning. Since this is the area where the most contribution is possible and the greatest need exists, a keyboard-only prototype was subsequently conceptualized and implemented, especially for users with physical disabilities. While the presented prototype is specific to a workflow modeling language, the generic implementation is currently under review to be integrated as a generic feature for the open-source Eclipse Graphical Language Server Platform. This enables other tool developers to easily plug-in our functionality to make their tool accessible for physically impaired modelers. In the future, we aim to invite impaired persons to empirically test our prototype. The current state of the prototype including a demo video is available online[1].

References

1. Accessibility Developer Guide: How to implement websites that are ready for keyboard only usage. `https://www.accessibility-developer-guide.com/knowledge/keyboard-only/how-to-implement/` (2018), accessed: 23.05.2023
2. Albusays, K., Bjorn, P., Dabbish, L., Ford, D., Murphy-Hill, E., Serebrenik, A., Storey, M.A.: The diversity crisis in software development. IEEE Software **38**(2), 19–25 (2021)
3. Bork, D., Alter, S.: Satisfying four requirements for more flexible modeling methods: Theory and test case. Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model. **15**, 3:1–3:25 (2020). https://doi.org/10.18417/emisa.15.3
4. Bork, D., Roelens, B.: A technique for evaluating and improving the semantic transparency of modeling language notations. Softw. Syst. Model. **20**(4), 939–963 (2021). https://doi.org/10.1007/s10270-021-00895-w
5. Brophy, P., Craven, J.: Web accessibility. Library trends **55**(4), 950–972 (2007)
6. de Carvalho, J.F., Amaral, V.: Towards a modelling workbench with flexible interaction models for model editors operating through voice and gestures. In: 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC). pp. 1026–1031. IEEE (2021)
7. ConnectedPapers: `https://www.connectedpapers.com/`, accessed: 15.05.2023
8. Cross, C., Cetinkaya, D., Dogan, H.: Transforming diagrams' semantics to text for visually impaired. In: Design, User Experience, and Usability. Interaction Design: 9th International Conference, DUXU 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings, Part I 22. pp. 339–350. Springer (2020)
9. David, I., Latifaj, M., Pietron, J., Zhang, W., Ciccozzi, F., Malavolta, I., Raschke, A., Steghöfer, J., Hebig, R.: Blended modeling in commercial and open-source model-driven software engineering tools: A systematic study. Softw. Syst. Model. **22**(1), 415–447 (2023). https://doi.org/10.1007/s10270-022-01010-3
10. De Carlo, G., Langer, P., Bork, D.: Advanced visualization and interaction in glsp-based web modeling: realizing semantic zoom and off-screen elements. In: Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems, MODELS 2022, Montreal, Quebec, Canada, October 23-28, 2022. pp. 221–231. ACM (2022)
11. De Carlo, G., Langer, P., Bork, D.: Rethinking model representation - A taxonomy of advanced information visualization in conceptual modeling. In: Conceptual Modeling - 41st International Conference. vol. 13607, pp. 35–51. Springer (2022)
12. Eclipse: GLSP. `https://www.eclipse.org/glsp/`, accessed: 23.05.2023
13. Gabrysiak, G., Giese, H., Lüders, A., Seibel, A.: How can metamodels be used flexibly. In: Proceedings of ICSE 2011 workshop on flexible modeling tools, Waikiki/Honolulu. vol. 22 (2011)
14. Guerra, E., de Lara, J.: On the quest for flexible modelling. In: Wasowski, A., Paige, R.F., Haugen, Ø. (eds.) Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS 2018, Copenhagen, Denmark, October 14-19, 2018. pp. 23–33. ACM (2018). https://doi.org/10.1145/3239372.3239376
15. Kavcic, A.: Software accessibility: Recommendations and guidelines. In: International Conference on "Computer as a Tool". pp. 1024–1027 (2005)
16. King, A., Blenkhorn, P., Crombie, D., Dijkstra, S., Evans, G., Wood, J.: Presenting uml software engineering diagrams to blind people. In: Computers Helping People

with Special Needs: 9th International Conference, ICCHP 2004, Paris, France, July 7-9, 2004. Proceedings 9. pp. 522–529. Springer (2004)

17. Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O.P., Turner, M., Niazi, M., Linkman, S.: Systematic literature reviews in software engineering–a tertiary study. Information and software technology **52**(8), 792–805 (2010)
18. Loitsch, C., Weber, G.: Viable haptic uml for blind people. In: International Conference on Computers for Handicapped Persons. pp. 509–516. Springer (2012)
19. Luque, L., Veriscimo, E.d.S., Pereira, G.d.C., Filgueiras, L.: Can we work together? on the inclusion of blind people in uml model-based tasks. In: Inclusive Designing, pp. 223–233. Springer (2014)
20. MDN: Keyboard. `https://developer.mozilla.org/en-US/docs/Web/Accessibility/Understanding_WCAG/Keyboard` (2023), accessed: 23.05.2023
21. Metatla, O., Bryan-Kinns, N., Stockman, T.: Constructing relational diagrams in audio: the multiple perspective hierarchical approach. In: Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility. pp. 97–104 (2008)
22. Modeling Languages: Top online UML modeling tools. `https://modeling-languages.com/web-based-modeling-tools-uml-er-bpmn/`, accessed: 01.05.2023
23. Monaco, F.: Color blind accessibility manifesto. Communications of the ACM **65**(8), 7–7 (2022)
24. Moody, D.L.: The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. IEEE Trans. Software Eng. **35**(6), 756–779 (2009)
25. Muff, F., Fill, H.G.: Initial concepts for augmented and virtual reality-based enterprise modeling. In: ER Demos/Posters. pp. 49–54 (2021)
26. Nganji, J.T., Nggada, S.H.: Disability-aware software engineering for improved system accessibility and usability. International Journal of Software Engineering and Its Applications **5**(3), 47–62 (2011)
27. Ossher, H., van der Hoek, A., Storey, M.D., Grundy, J., Bellamy, R.K.E.: Flexible modeling tools (flexitools2010). In: Kramer, J., Bishop, J., Devanbu, P.T., Uchitel, S. (eds.) Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2, ICSE 2010, Cape Town, South Africa, 1-8 May 2010. pp. 441–442. ACM (2010). https://doi.org/10.1145/1810295.1810419
28. Paiva, D.M.B., Freire, A.P., de Mattos Fortes, R.P.: Accessibility and software engineering processes: A systematic literature review. Journal of Systems and Software **171**, 110819 (2021)
29. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: An update. Information and software technology **64**, 1–18 (2015)
30. Sandkuhl, K., Fill, H., Hoppenbrouwers, S., Krogstie, J., Matthes, F., Opdahl, A.L., Schwabe, G., Uludag, Ö., Winter, R.: From expert discipline to common practice: A vision and research agenda for extending the reach of enterprise modeling. Bus. Inf. Syst. Eng. **60**(1), 69–80 (2018). https://doi.org/10.1007/s12599-017-0516-y
31. Schulz, M., Pieper, M.: Web compliance management: barrier-free websites just by simply pressing the button? accessibility and the use of content-management-systems. In: Universal Access in Ambient Intelligence Environments, pp. 419–426. Springer (2007)
32. Seifermann, S., Groenda, H.: Towards collaboration on accessible uml models. Mensch und Computer 2015–Workshopband (2015)

33. Seifermann, S., Groenda, H.: Survey on textual notations for the unified modeling language. In: 2016 4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD). pp. 28–39. IEEE (2016)
34. Torres, M.J.R., Barwaldt, R.: Approaches for diagrams accessibility for blind people: a systematic review. In: 2019 IEEE Frontiers in Education Conference (FIE). pp. 1–7. IEEE (2019)
35. Torres, M.J.R., Barwaldt, R., Pinho, P.C.R., de Topin, L.O.H., Otero, T.F.: An auditory interface to workspace awareness elements accessible for the blind in diagrams' collaborative modeling. In: 2020 IEEE Frontiers in Education Conference (FIE). pp. 1–7. IEEE (2020)
36. W3: IPO Announcement. `https://www.w3.org/Press/IPO-announce` (1997), accessed: 23.05.2023
37. W3C: Web Content Accessibility Guidelines (WCAG) 2.1. `https://www.w3.org/TR/WCAG21/` (2018), accessed: 18.09.2022
38. WAI: Accessibility Intro. `https://www.w3.org/WAI/fundamentals/accessibility-intro/` (2022), accessed: 22.05.2023
39. WebAIM: Diverse Abilities and Barriers. `https://www.w3.org/WAI/people-use-web/abilities-barriers/` (2017), accessed: 23.11.2022
40. WebAIM: WebAIM's WCAG 2 Checklist. `https://webaim.org/standards/wcag/checklist` (2021), accessed: 18.09.2022
41. WebAIM: Keyboard Accessibility. `https://webaim.org/techniques/keyboard/` (2022), accessed: 23.05.2023
42. WebAIM: The 2023 report on the accessibility of the top 1,000,000 home pages. `https://webaim.org/projects/million/` (2023), accessed: 23.05.2024
43. Weber, G., Weimann, K.: Editing a test suite for accessibility of interactive web sites. In: International Conference on Universal Access in Human-Computer Interaction. pp. 193–201. Springer (2007)
44. WHO: Disability. `https://www.who.int/news-room/fact-sheets/detail/disability-and-health` (2023), accessed: 22.05.2023
45. Wildhaber, F., Salloum, N., Gygli, M., Kennel, A.: Self-directed creation and editing of uml class diagrams on mobile devices for visually impaired people. In: Tenth International Model-Driven Requirements Engineering. pp. 49–57. IEEE (2020)
46. Wüest, D., Seyff, N., Glinz, M.: Flexisketch: a lightweight sketching and metamodeling approach for end-users. Softw. Syst. Model. **18**(2), 1513–1541 (2019). https://doi.org/10.1007/s10270-017-0623-8
47. Zhang, B., Chen, Y.: Enhancing UML conceptual modeling through the use of virtual reality. In: 38th Hawaii International Conference on System Sciences (2005)