



HAL
open science

Towards Accessible Complete Plain Text to General Public Reader

Frederic Serin, Katerine Romeo-Pakker Romeo

► **To cite this version:**

Frederic Serin, Katerine Romeo-Pakker Romeo. Towards Accessible Complete Plain Text to General Public Reader. A. Petz, K. Miesenberger. 17th International Conference on Computers Helping People with Special Needs (ICCHP2020), A. Petz, K. Miesenberger, pp.35-42, 2020, Future Perspectives of AT, eAccessibility and eInclusion, 978-3-9504630-2-6. hal-03380637

HAL Id: hal-03380637

<https://hal.science/hal-03380637v1>

Submitted on 15 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards Accessible Complete Plain Text to General Public Reader

Frederic Serin, Katerine Romeo
Normandy University, LITIS/CNRS FR 3638, France
Frederic.Serin@univ-lehavre.fr
Katerine.Romeo@univ-rouen.fr

Abstract

In this article, we analyze the difficulties to access spatial information and propose some solutions. Our objective is two-fold: to create any type of document easily, and to make it accessible and intelligible for all audiences, blind and sighted people. Digital document accessibility, which in addition to readability also includes understanding and appropriation, should be improved. We propose some solutions to enhance the efficiency of screen readers for difficult 2D objects such as graphics and tables. Linear reading can be facilitated by inserting additional tags and attributes proposed by HTML5 or WAI-ARIA. Thanks to a simple language with lightweight tags, we offer the possibility to automatically generate more accessible documents without specialized knowledge of these functionalities. From the plain text document, HTML and SVG codes are generated. To complete the formatted document, a description file is produced in parallel to accompany the understanding of complex documents - UML diagrams or tables - and also to present the structure of the complete document.

Introduction

This article presents some solutions of accessible document writing for people with visual impairments in order to communicate with sighted audiences [1,2]. It innovates by the fact that it does not place the visually impaired person (VIP) as a reader but as an author of documents. We also wanted to make this project attractive to other audiences: students learning to design schemes, diagrams and Web developers sometimes bewildered by the difficulty of making their work accessible.

We propose to enhance the level of accessibility for digital documents [3,4]. We suggest with our approach, not to increase the number of constraints proposed by the WCAG [5], but to decrease the quantity of rules that an editor must have in mind.

Our approach has the particularity to be developed for a blind teacher which expresses the wish to be able to communicate with well-sighted students. Our project, which is progressing according to an iterative method, differs by taking into account a proactive participation of the readers [6].

If you are a sighted person, you have about a 36% chance of not having a VIP among your 200 readers. But, if you are a VIP, the probability to have at least 199 sighted readers is close to 100%. This example uses the probabilities of a binomial law with a prevalence of 0.5% and it is based on statistics provided by the WHO [7]. Each generated document must be accessible but also must bring to students added value for their understanding.

In the first section, we present how a simple text could be written by blinds. This text requires additional information for a complete perception. Indeed, the visual rendering is not always identical to what is heard from the automatic reading by a screen reader.

The inclusive text editor for tabular structure and diagram description has been already implemented. Table representation will be presented in the second section.

Finally, we shortly speak about diagram conception, needed enhancement, and we conclude with perspectives of development.

Linear Composition

Digital documents are made accessible mainly through the constant improvement of screen readers [8]. As with listening, the vocal restitution of any document is linear [9]. So, it is important to understand any complex structure [10,11].

We create the documents using a generic basic tool: a text editor. We follow the principle KISS: keep it simple and intuitive. If a basic literary text is relatively simple to understand, it is quite different for a more complex structure. We have drawn inspiration from languages such as Markdown and PlantUML [12,13] to extend the functionalities of the basic text editor in order to facilitate the creation of documents and allow their universal use. We want to maintain, as much as possible, the principle of least surprise, therefore a lightweight markup language with simple, non-obtrusive tags is used.

Our “tag text editor” is designed to be easy to write and, for the source text, also easy to understand through TTS (Text-To-Speech); indeed, a special care has been taken to ensure that the tags are easily identified by screen readers. Without knowledge of dedicated languages (such as HTML and SVG), the final output (generated file) meets the requirements of readability and visibility for all audiences. These languages are widespread for the diffusion of information and permit to include many possibilities to make accessible documents [2, 3].

We have developed our software, called Latitude, using design patterns from literature [14] and in particular: Factory, Builder, and Chain of Responsibility. Latitude (abbreviated to Lttd in the following diagram) is the short name for Light and Accessible Text Including Tags and Using Universal Design.

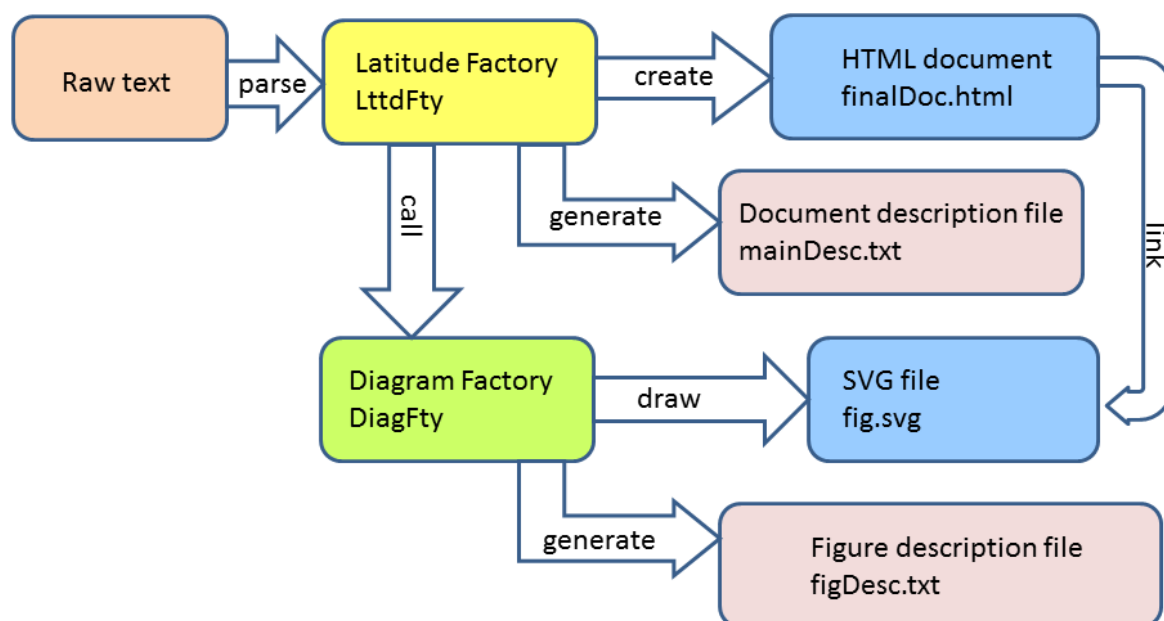


Fig. 1 Text document creation process

The user creates a simple text file, a plain text enriched with lightweight tags. This raw text is processed in the Latitude factory (Lttd Fty). From this point of entry, the structured and accessible HTML document is generated as well as a document description file. In addition, for diagrams, the definition elements are sent to a second factory (Diagram Fty). This second factory generates an SVG file for the schemas as well as a description document allowing a textual description.

According to the target audience, blinds and students, the description texts are very useful. This is why we would like to draw your attention to these complementary documents. They contain statistical

elements (size, complexity), warnings such as typographical elements (paragraphs or sentences that are too long) and suggestions for improvement.

For texts or tables, it will give indications such as size and reading complexity (Flesch score [15]), point out titles, items or paragraphs that are too long, make suggestions on structuring into sections and subsections. For a class diagram, it can give indications on the structure, its complexity, vocalize as a reader would the various component elements and indicate overlaps or problematic positioning.

The description file therefore helps blind people, who do not have a global perception of the document, but also students who want to benefit from an indicative feedback.

Such approach is disruptive with classic approaches to the accessibility: it takes it the other way round. Our extended editor enables VIP writers to produce a document that meets the expectations of both, VIP and (well-) sighted readers; it is an inclusive text editor.

The proposed approach may be attractive for other audiences: students learning to design inclusive text with graphics, web designers who must design accessible sites, etc. [2] The proposed approach gives insight in ways to make data presentations more ergonomic and easy to understand; moreover, it gives the opportunity to discover new ways to data and knowledge acquisition, and new ways for learning (inclusive) strategies.

Text Composition

We have drawn inspiration from languages such as Markdown to extend the functionalities of the basic text editor in order to facilitate the creation of documents and allow their universal use. We want to maintain, as much as possible the principle of least surprise, therefore a lightweight markup language with simple, non-obtrusive tags is used.

A document of a relatively large size is made up of sections that are themselves divided into subsections. Each section is identified by a title. In many languages, these titles are indicated by specific tags. These are not well identified by a screen reader. For example, we encounter brackets, exclamations points, dashes, etc. This is why we are introducing a symbol that stands out from the usual. We propose the equal sign (=), and we reverse the usage that the number of symbols increases with the depth of the nested sections. As we respect the typographical advice, we only offer four levels of nesting (division). The first level, equivalent to the most important title “h1” of HTML will be represented by five equal signs. The last level (the least important) will be composed with only two equals.

The advantage of this choice is double: the equal sign is well identified by vocal synthesis and the user can easily navigate with respect to the levels with the search function. The text created is enriched by light markup. The latter makes it possible to provide informative elements on the formatting such as the presence of a title, a word placed in italics, etc. The presence of the tags must guarantee that the legibility of the original text is maintained.

Each section is composed of a title, paragraphs and possibly, subsections. We automatically study if these sections are well balanced inside the document and indicate to the author what are the shortest or the longest ones. We suggest, if necessary, to add sections and subdivisions [6]. The paragraphs are decomposed in sentences, each of them in words and syllables. This process permits to apply the Flesch score and to measure the readability of the document [15,16].

The balance between sections and the determination of readability are very useful guidelines for blind authors. They are also useful for novice writers [17].

The perception of a good distribution of sections is not arithmetic but geometric. Therefore, we calculate the number of sections and subsections desired according to the overall size of the document. We indicate the imbalances found between sections.

To calculate the Flesch score, we need to be able to extract not only the words but also the syllables from a sentence [18]. We can thus indicate the readability of the text, but we can also point out the complexity of a paragraph, the inappropriate length of a title or an item in a list.

Creation of Tables

We present how to create a table and its visual and accessible restitution. A table consists of cells, each cell being situated at the junction of a column and a row [19].

In HTML, a table consists of four important notions: cells and among them, headings, rows, and caption.

With the progressive evolution of HTML, the WAI initiative [20], and the efficiency of screen readers, the tables are relatively accessible. But this accessibility is relative. It is limited, at the base, to the vocalization by the screen reader of the contents and the indication of the positioning of the cell: the number of its line and that of its column. The headings of the columns are indicated if they are correctly filled in [19]. If the table is a bit complex, i.e. if there are: several header lines with column groupings, cell merges, row headers, then the voice description is impossible without the introduction of specific tags and attributes proposed by the WAI-ARIA [20]. An example of a complex table with merged headings is given in table 1.

Table 1 An example of complex table

Country	Main Cities		Residents (millions)
	Capital	Metropole	
Algeria	Alger		34
Australia	Canberra	Sydney	42
Belgium	Bruxelles		12
Brazil	Brasilia	Sao Paulo	110

Current languages such as Wikitext for example, only allow these indications by writing these attributes in the tables created. However, they are very difficult to implement and, moreover, they are not known to the general public [21]. This public is precisely the one that proposes the tables. It is for this reason that we have developed a code analyzer that automatically inserts all the tags necessary to make any type of table accessible.

Our parser scans the lines defining a table and automatically detects the separator. This frees the user from a limited choice. We allow the use of three commonly used cell separators: pipe, tabulation, and semicolon. The pipe has our preference for its perfect vocalization in the plain text by the screen

reader. Nevertheless, we allow tabulation for its common use by certain languages and writers. The semicolon is useful whenever interaction is possible with a CSV-formatted document.

With our proposed solution, it is no longer useful for the writer to provide tags to allow a complete description of a cell with its row or column headings. If cells are merged, if a header gathers several columns or if headers are placed at the beginning of a row, a table not using the ARIA recommendations will not allow voice output. However, the voice output is part of the information of the contents and structure of the table. We automatically process the data so that this information can be entered. When browsing the table, we create an object model to detect the headers that inform each cell. We then place the ARIA tags correctly so that the screen reader reads all the information that will help to navigate through the table.

Latitude creates an object-oriented image of a table, composed of typed cells: header and body. The groups of cells (fusion) define automatically references (id) in attributes. This id is added to the reference, headers, encapsulated into the td html tag. Different other attributes like role and scope are systematically included permitting accessibility [22].

For the user, it is no longer necessary to explicitly specify the headers: just leave a blank line to avoid placing a header. In the same way, the first cell of each row is the row header; a blank cell allows not specifying it. Finally, you must specify several separators in a row to merge cells. To create complex headers on several lines, a separator, following many dashes, visually and audibly separates head and body. According to our syntax, the table title is situated just after the dedicated `__table` tag. This possibility, indicated in the description file in case of omission, will be transformed as information elements of the html caption.

As to draw bar charts or pie charts the data is sometimes grouped in a table, we used the same format to make these diagrams. The first line of the table allows you to define the labels while we allow up to three lines of data that will allow this information to be presented on the same bar chart or with one pie chart per line, sized to be placed side by side in the same figure. In both cases, a calculation is automatically performed to correctly dimension the information. For the bars, it is necessary to estimate the height differences, in order to balance as well as possible and to allow the best visibility. A scale is determined automatically. For the choice of colors, a library is proposed by default, which allows distinguishing the contiguous zones by following the recommendations of accessibility in particular, those of contrast and color blindness.

Simple and complex tables are formatted to be accessible with the tags proposed by W3C. Their design remains, for the creator, as simple as possible. It is not required to know the attributes or behaviors of screen readers [4].

Accessible Diagrams for All

Our product allows easy integration of diagrams and graphics that we know can be made accessible with SVG [23], vector graphics structured by XML. We are inspired by PlantUML [13]. We add some complementary aims: readable code, accessible SVG use of ARIA attributes addition of a document given a textual description of the properties of the diagram.

Our first need was the easiness to ensure readability of the source code by a screen reader. The syntax must be understood by blinds throughout the vocalization [24]. Second, the description proposes a good rendering for visualization. Any sighted person must see what he does. Third, to facilitate the learning of the elements of the code, we have ensured consistency among the various proposed UML diagrams: class, object, sequence and activity. In Figure 2, an excerpt from the object model used by Latitude, is given: *There are three inheritance trees, designing the structure of the builder pattern used by Latitude. Each Block is either the main document or a section. The Principal and Article classes inherit from Block. Each Block can itself be composed of Articles. Within each Block we find individuals and, optionally, other*

Blocks. The instances of *Individual* correspond to different kinds of HTML elements: *Paragraphe*, *Table*, *Figure*... These subclasses inherit from *Individual* and structure the code. The class *Scheme* is linked to another external class called *Graphics*. The instances of a subclass of *Graphics*, translate different UML diagrams and generate the SVG accessible code.

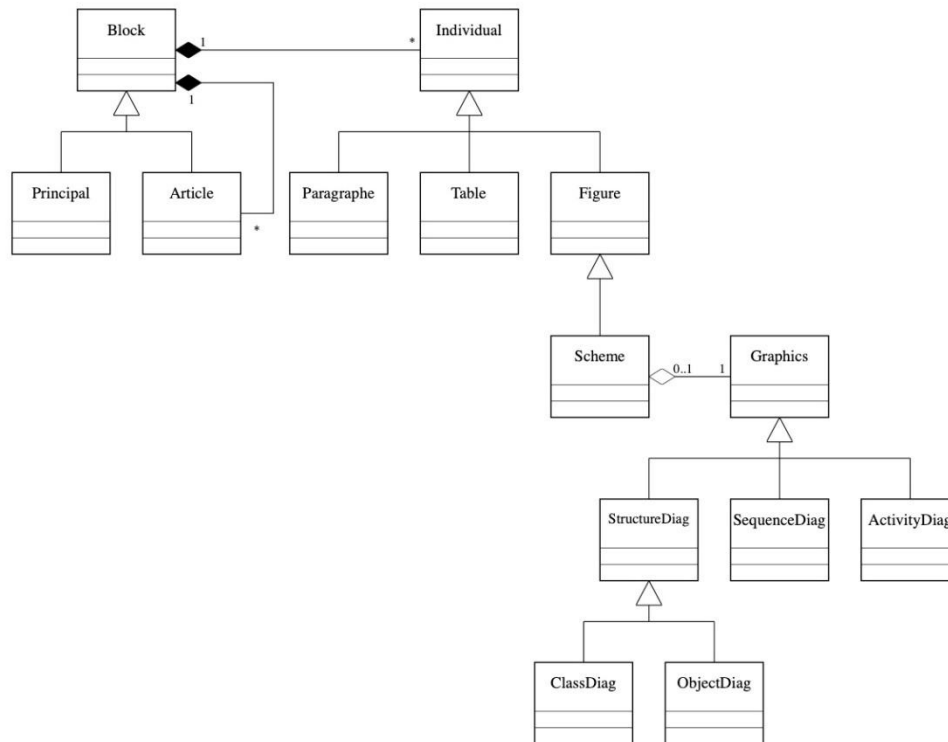


Fig. 2 An example of the object model coded in SVG, generated by Latitude (in depth description in text above).

Each individual is a part of a division. A scheme represents a diagram; it is a particular individual, coded in SVG. There are many other classes in our model, but this diagram gives a glimpse of our achievement. We can generate these diagrams either directly by calling a specific class or by inserting the code into a document containing, for example, text. Our original characteristics are the automatic generation of a SVG graphic including description tags for linear browsing, and including this diagram in a captioned figure in the global document. To enable this accessibility, we have used all the possibilities offered by the ARIA attributes.

Finally, in parallel with the generation of the diagram, our software creates a description document. This document analysis offers an alternative description, given information about the structure like: number of various elements, sentence explaining links (associations, messages), size and positions... All this information is given in the form of sentences that are clearly stated by the screen reader. If this annex is an invaluable aid for the blind, this document proves to be interesting for learning.

Conclusions and Perspectives

Our project is incremental. With each request, we add the expected functionalities. The primary target audience is the blind, who must be able to communicate with a normal audience, integrating appreciated and necessary visual elements [25]. On the other hand, we make a tool that is also useful to students because they have to formalize their diagrams and get the phrased description of their realization. Finally, we show that it is possible to enrich web page design tools by making them accessible without having to deal with an impressive number of constraints. This can be an alternative

solution to allow everyone to access websites. We need to improve automatic text analysis. To do this, we need to get closer to typographers, authors and educational experts.

For tables, we are in the process of adding the ability to merge cells vertically, and to allow content formatting: centered or aligned, automatically detecting whether columns or rows contain numbers for example. But the multiplicity of situations requires extreme caution in this area. Also, we aim to introduce a score, equivalent to the Flesch one, to indicate the complexity of a table. The proposed library of diagrams must be enhanced offering various functionalities. We add them as and when requested. Students are the best evaluators: they need to take ownership of the opportunities. At the frontier between tables and graphs, we plan to introduce curves drawn from data. Already, we can read a table and draw a bar chart or pie chart on demand. This is not trivial, it is necessary to be able to make a judicious choice of colors according to accessibility constraints and, also, to define an appropriate scale for the best possible visibility.

References

1. Altinier, A. : « Accessibilité Web », Ed. Eyrolles, 2012
2. Romeo, K. ; Pissaloux, E. ; Serin, F. : « Accessibilité des sites web pour les personnes présentant une incapacité visuelle ». CNRIUT 2018, Aix-en-Provence, p163-165, 7-8 June 2018
3. El Andaloussi, M. ; Le Ludec, C. : « L'accessibilité numérique, entre nécessité et opportunité, Une obligation légale vis-à-vis des citoyens, un levier stratégique pour les acteurs », Ed. Conseil National du Numérique, 2020-02-05
4. Murphy, S.: "Accessibility of graphics in technical documentation for the cognitive and visually impaired", SIGDOC '05, Int. Conf. on Design of communication: documenting & designing for pervasive information, 12-17 Sept. 2005
5. Web Content Accessibility Guidelines (WCAG) 2.0 <https://www.w3.org/TR/2008/REC-WCAG20-20081211/> consulted on 2020-05-10
6. Van Der Perk, M.: "A guide for scientific writing", Utrecht University, April 2015
7. <https://www.who.int/publications/i/item/world-report-on-vision> consulted on 2020-06-28
8. <https://webaim.org/techniques/screenreader/> consulted on 2019-12-04
9. Romeo, K. ; Pissaloux, E. ; Serin, F. : « Accessibilité aux informations textuelles et visuelles sur les sites web pour les personnes avec une déficience visuelle ». HANDICAP 2018, 13-15 June 2018, Paris
10. Sutton, J. : "A Guide to Making Documents Accessible to People Who Are Blind or Visually Impaired". Pub. by the American Council of the Blind, Washington, DC. 2002
11. Leuthold, S.; Bargas-Avila, JA.; Opwis, K.: "Beyond web content accessibility guidelines: Design of enhanced text user interfaces for blind internet users". Int. J. Human-Computer Studies (66,257–270). 2008
12. Vanderheiden, G.: "Fundamental Principles and Priority Setting for Universal Usability", ACM Conference on Universal Usability, Arlington, Virginia, USA, p32-38, 2000
13. <https://plantuml.com/en/> consulted in february 2020
14. Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: "Design Patterns : Elements of Reusable Object-Oriented Software". ISBN 0-201-63361-2, Addison-Wesley, 1994
15. Rudolf Flesch : "The Art of Plain Talk", New York, Harper and Row, 1946
16. Rudolf Flesch : "How to test Readability", New York, Harper and Row, 1949
17. Ciampa, K. : "The effects of an online reading program on grade I students' engagement and comprehension strategy use", J. of Research on Technology in Education, 45(1), 27–59. 2012.
18. Georges, G. : « Lisibilité et compréhension », In: Communication et langages, n°45, 1er trimestre 1980, pp7-16,1980
19. Duval, R. : « Comment analyser le fonctionnement représentationnel des tableaux et leur diversité ? », Spirale. Revue de recherches en éducation, n°32, 2003.
20. <https://www.w3.org/TR/wai-aria-1.1/> consulted on 2019-03-18

21. <https://www.mediawiki.org/wiki/Help:Tables> consulted on 2020-02-7
22. Romeo, K. ; Pissaloux, E. ; Serin, F. : « Tableaux accessibles dans les documents numériques », CNRIUT 2019, Toulon, p138-140, 6-7 June 2019
23. Altmanninger, K.; Wöß, W.: “Dynamically Generated Scalable Vector Graphics (SVG) for Barrier-Free Web-Applications”, In: Miesenberger K., Klaus J., Zagler W.L., Karshmer A.I. (eds) Computers Helping People with Special Needs. ICCHP 2006. LNCS, vol 4061. Springer, Berlin. 2006.
24. Luque, L.; de Oliveira Brandão, L. ; Tori, R. ; Alves Franco Brandão, A. : “On the Inclusion of Blind People in UML e-Learning Activities”. Brazilian Journal of Computers in Education (Vol23, No 02). 2015.
25. Marty Beech. “Assisting Students with Disabilities”. Bureau of Exceptional Education and Student Services (BEES), Florida. 3rd Edition, 2010