IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 April 2024, accepted 21 May 2024, date of publication 28 May 2024, date of current version 6 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3406688

RESEARCH ARTICLE

UML Profile to Model Accessible Web Pages

KARLA ORDOÑEZ-BRICEÑO[1], JOSÉ R. HILERA[2], LUIS DE-MARCOS[2], SALVADOR OTÓN-TORTOSA[2], AND SAMANTA CUEVA-CARRIÓN[3]

[1]Instituto Superior Tecnológico Ismael Pérez Pazmiño, Machala 070211, Ecuador
[2]Department of Computer Science, Universidad de Alcalá, 28805 Alcalá, Spain
[3]Department of Computer Science and Electronics, Universidad Técnica Particular de Loja, Loja 110102, Ecuador

Corresponding author: Karla Ordoñez-Briceño (kfordonez@gmail.com)

ABSTRACT Creating accessible websites is essential to ensure the inclusion of users with disabilities, as defined in the Web Content Accessibility Guidelines (WCAG). In many countries, compliance with these guidelines is often legally mandated. Consequently, it is crucial to integrate accessibility considerations into Web development projects from the beginning. Unfortunately, accessibility requirements are often addressed only after the website is completed. To ensure compliance, it is proposed to incorporate accessibility requirements into system specifications and models. A Unified Modeling Language (UML) profile, called WebPageAcc, is introduced to streamline this process, allowing developers to incorporate accessibility requirements without extensive knowledge of accessibility standards. This profile provides stereotypes, data types, tagged values, and restrictions necessary to meet accessibility requirements. This study uses a four-stage method to create, apply, and validate the profile. Initially, an analysis of the WCAG 2.2 criteria is performed to define stereotypes, attributes and limitations aligned with the success criteria. Subsequently, the WebPageAcc profile is developed in Eclipse Papyrus, considering the identified elements. The third stage is to apply the profile to create a UML class diagram model of a real web page. Validation occurs in the final stage by executing Object Constraint Language (OCL) rules associated with attributes defined by the profile. Furthermore, an experiment is carried out to evaluate the usability difficulty of the profile among software developers. This multifaceted approach aims to improve the integration of accessibility into website development, ensuring inclusivity for a diverse user base while simplifying the implementation process for developers.

INDEX TERMS Accessibility web, MDA, MDD, model-driven engineering, profile UML-Web UML, OCL, WCAG.

I. INTRODUCTION

Web developers commonly lack extensive knowledge and experience basic web accessibility concepts [1]. Consequently, websites often encounter accessibility issues for disabled users. Accessibility testing is not frequently performed throughout the development process; instead, it typically occurs only after the website is completed using automated evaluation tools [2]. This delayed validation often results in late detection of accessibility errors, stemming from the failure to consider web accessibility features during development. Rectifying accessibility errors in the final stages of the software development process can require significant time and financial resources and may even necessitate website redesign.

According to a study by [3], one of the primary reasons for the persistence of inaccessible websites is the developers' lack of knowledge and experience regarding accessibility standards. This deficiency hampers software professionals' ability to design and develop appropriately accessible solutions.

Another prevalent issue related to web accessibility is the failure of many developers to consider it in the early stages of software construction. Presently, it is common practice to address accessibility issues only towards the end of software development, aided by automated evaluation tools [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Lukasz Wisniewski.

However, to ensure software quality, it is imperative to verify not only functional requirements but also nonfunctional requirements, such as accessibility.

For many developers, ensuring accessibility is a critical but time-consuming task, particularly since accessibility is not considered a fundamental quality requirement in many contemporary projects, especially within smaller teams, smaller companies, and less developed countries [3].

Additionally, developers frequently rely on automated evaluation tools to identify potential accessibility issues, typically at the end of the software construction process, leads to software redesign costs. Therefore, it is imperative that software functionality incorporation originates from requirements analysis [5], as accessibility is a quality feature that ensures not only access to information for individuals with disabilities, but also enhances overall usability for all users.

Accessibility issues may arise at various stages throughout a project's lifecycle. In fact, the costliest accessibility issues often arise after the project's completion [3]. Furthermore, organizations may face expensive legal actions in countries with robust accessibility legislation. Additionally, it is crucial to recognize that certain accessibility features may become functional requirements of the software for individuals with disabilities.

To address this challenge, the present proposal has been developed, allowing for the incorporation of accessibility features using a UML Profile in the later stages of software development, thereby mitigating the necessity for developers to possess advanced knowledge of accessibility standards.

In this context, to ensure that accessibility is considered from the outset of the web development process, we propose a model-driven development approach [6] to promote efficient utilization of system models in the software development process. Accordingly, a UML profile [7] was created to model accessible web pages.

The proposed profile is defined by a set of stereotypes, each extending the UML language to accessible websites. Each created stereotype represents different HTML tags of an accessible web page, such as images, videos, audios, menus, links, definition lists, headings, paragraphs, content sections, forms and tables. Additionally, constraints and tagged values are applied to represent the main properties of the model conceptually. This study utilized Object Constraint Language (OCL) [8] to specify constraints associated with the stereotypes defined in the profile, considering the Web Content Accessibility Guidelines (WCAG) 2.2 [9], thereby avoiding arbitrary and erroneous usage of each defined stereotype.

This proposal is compatible with the Object Management Group's (OMG) Model-Driven Architecture (MDA) [10], promoting efficient use of system models in the software development process, and relies on automation tools to facilitate gradual transformations between models. Consequently, the designer can define a Platform-Independent Model (PIM) for each web page based on the components and constraints established in the profile, and then automatically transform the model into HTML code [11].

This approach represents significant progress in overseeing the development of accessible websites and offers productivity gains, increased reliability, and better management of changing requirements. Designers can integrate accessibility requirements that website components and interactions must meet into the models. Subsequently, part of the HTML code is automatically generated from the templates, ensuring that it remains accessible and does not present issues for users with disabilities.

There are various proposals advocating for the integration of accessibility into model-driven software development (MDD), encompassing a wide range of applications such as web, mobile, and desktop applications, as well as various fields of knowledge such as content management systems (CMS), e-learning, and multimedia resources [12]. However, to date, no comprehensive proposals have been developed to address the challenges of web accessibility in a single model. This is due to the difficulty many authors encounter in constructing metamodels that cover all aspects of web accessibility in a generalized manner.

In this context, several works in the software development field stand out for using Model-Driven Development (MDD) to enhance web accessibility. For instance, the work of [1] assists developers in creating accessible web menus using a model-based approach.

Another significant study is that of [2], which employs MDD to generate graphical models of namespaces and produce structured wiki code. This approach also advocates for the inclusion of accessibility features in models based on official W3C guidelines such as WCAG and ATAG [2]. Additionally, the Accessed methodology has been proposed, enabling the generation of accessible mobile applications [3]. Likewise, there is an initiative to develop accessible graphical editors for multimedia players using MDD [7].

Despite the benefits offered by MDD in improving web accessibility, no proposal has yet been developed to generalize accessibility aspects. Most works focus on specific elements of a website, such as menus or multimedia elements, or solely target certain components of a mobile application. Moreover, none of these proposals provide users with clear rules on what can or cannot be added to the model.

In our proposal, we have employed the Object Constraint Language (OCL) to write constraints in the models. The use of UML profiles is advantageous, as they provide a generic extension mechanism for constructing UML models in specific domains. These profiles are based on stereotypes, constraints, and additional tagged values applied to elements or relationships in a diagram.

II. BACKGROUND

A. MODEL DRIVEN DEVELOPMENT (MDD)

Model-Driven Development (MDD) is a software development paradigm aimed at constructing applications from models that represent the system's functionality, structure, and/or behavior [13]. This paradigm proves highly beneficial as

it facilitates the segregation of models from code, thereby achieving platform independence and enabling the creation of applications for various programming languages from a unified model.

MDD offers several advantages, including increased productivity and reduced complexity in software development. Developers operate at a high level of abstraction, allowing them to focus on crucial aspects of the application domain without requiring extensive technical knowledge of the deployment platform. Consequently, applications can be developed swiftly and easily [14]. Additionally, MDD promotes reusability and portability by facilitating the creation of platform-independent models applicable to diverse application types—web, desktop, and mobile—without the need to rewrite the application's core logic.

Automatic transformations between models further simplify maintenance, as changes made to the models propagate seamlessly to the code, streamlining application evolution [15]. Moreover, MDD significantly aids in generating application documentation, as the models provide an accurate representation of the application they instantiate.

A fundamental standard employed in MDD is the Unified Modeling Language (UML) [16]. UML, a graphical modeling language, utilizes diagrams to specify software system artifacts. UML supports extension mechanisms such as stereotypes, tagged values, and constraints, facilitating customization through profiles to suit specific domains [17]. Constraints within models are described using Object Constraint Language (OCL) [8], which features a notation style akin to common object-oriented languages and offers ease of readability and writing. OCL expressions have no side effects, ensuring that system state remains unchanged upon evaluation.

Modeling tools are essential for creating profiles and models compliant with UML standards. In this endeavor, Papyrus—a robust, open-source, Eclipse-based modeling tool—has been employed. Papyrus enables the creation of UML profiles, empowering developers to craft domain-specific modeling languages effortlessly. Additionally, it offers validation capabilities to minimize modeling errors and ensure adherence to established domain rules. Papyrus was chosen for profile creation due to its cost-effectiveness, user-friendliness, and support for defining restrictions using the OCL language. Moreover, it proves invaluable for validating created models, thereby reducing errors and maintaining compliance with domain rules [18].

Despite the extensive benefits of model-driven development, certain limitations may arise, such as errors in the code generated through transformations [19]. Hence, it is imperative to rectify errors in the model and involve language programming experts in testing the resulting code's functionality.

This article introduces a novel UML profile named Webpage, crafted using Papyrus. Webpage comprises a set of stereotypes and design rules for developing accessible web pages. The profile categorizes various basic components of a web page and incorporates OCL constraints to fulfill web accessibility requirements outlined by the WCAG 2.2 standard [9], as detailed in the subsequent section.

B. WEB CONTENT ACCESSIBILITY GUIDELINES (WCAG)

Access to information on the web should adhere to specific accessibility requirements to ensure accessibility for all individuals, regardless of their abilities or usage conditions [20]. Web accessibility entails designing websites in a manner that enables individuals with disabilities—such as blindness, low vision, deafness, hearing loss, learning difficulties, cognitive limitations, limited mobility, speech disabilities, photosensitivity, or combinations thereof—to perceive, comprehend, navigate, and interact with web content. Therefore, implementing accessibility standards when designing a website is crucial, as it enhances the usefulness of content for all users, potentially increasing website traffic.

To enhance web accessibility, the World Wide Web Consortium (W3C) introduced the Web Accessibility Initiative (WAI) in 1997 [21]. Since its inception, WAI has developed guidelines aimed at making web pages more accessible to individuals with disabilities, with a prominent recommendation being the Web Content Accessibility Guidelines (WCAG) [9].

WCAG is a stable and widely recognized technical standard founded on four core principles: perceivability, operability, understandability, and robustness. Each principle is further elaborated into guidelines or fundamental accessibility objectives, each with defined testable success criteria. The latest version of WCAG (2.2) [9] comprises 13 guidelines and 86 success criteria. The accessibility level of a website is determined based on the specific criteria it satisfies. It is commonly agreed upon that websites should strive to achieve at least the AA level of conformance, which entails meeting 55 specific criteria in the case of WCAG 2.2.

The UML profile described in the following sections of this article, designed for modeling accessible web pages, will consider several essential criteria to achieve AA-level conformance. These criteria are associated with the 4 principles of the WCAG standard:

- Perceivable: Emphasizing the provision of text alternatives for non-text content, as well as providing equivalent alternatives for time-based media for prerecorded audio-only and video-only content.
- Operable: Considering guidelines related to adaptability and proper semantic markup, particularly when creating tables, forms, headers, and content sections.
- Understandable: Including criteria to make text content readable and comprehensible by defining the language of the page.
- Robustness: Maximizing compatibility with current and future user agents, including assistive technologies, to utilize HTML form controls along with their respective text labels.

C. BENEFITS OF MODEL-DRIVEN DEVELOPMENT FOR BUILDING ACCESSIBLE SOFTWARE

The application of an MDD approach enables the incorporation of quality requirements in the early stages of the software development life cycle, as demonstrated in the study conducted by [22]. Through this approach, software accessibility requirements can be specified in conceptual models, facilitating transformations between models [23]. This allows for the creation of platform-independent models that can then generate accessible software code in various programming languages.

In a model-driven development process, the productivity challenge is significantly addressed by enabling the generation of source code in multiple programming languages from platform-independent models [24]. This capability aids in the system's adaptation to technological changes or evolving requirements [25]. Additionally, it streamlines maintenance tasks by providing easily interpretable models, particularly in contrast to navigating through extensive lines of code in large-scale systems, especially when considerable time has passed since their construction or when there are changes in the software development team. Consequently, this model-driven development process facilitates the generation of system documentation from the initial stages of software development [26].

It is crucial to employ a model-based approach in developing accessible software as it significantly ensures the quality of the software product and meets user requirements, thereby fulfilling their expectations [27]. Software quality requirements, such as functionality, performance, compatibility, usability (including accessibility), reliability, security, maintainability, and portability, are precisely represented in the quality model of ISO/IEC 25010 [28].

Therefore, an MDD approach certifies compliance with several characteristics proposed by the ISO/IEC 25010 standard, enhancing productivity through automatic code generation and facilitating software reuse, portability, maintainability, and reimplementation due to the management of platform-independent models.

Among the benefits offered by employing an MDD approach for the development of accessible web portals is the elimination of the necessity to be an accessibility expert, thereby saving time and effort for the development team [14]. This is made possible by the ease of incorporating usability and accessibility requirements into models created in the early stages of the life cycle, tailored to the characteristics and preferences of users, including those with disabilities, thus ensuring easy, rapid, and secure access to the software for all users.

Model Driven Development (MDD) [12] has been effectively utilized to address complex scenarios, as software models, despite their intricacies, provide a comprehensive visualization of all software features. Furthermore, MDD has been widely acknowledged as a means to ensure quality, reduce time and effort, by enabling automatic transformation of models to generate source code in various programming languages [19].

D. RELATED WORK

There are various approaches to assess the accessibility of a website, including automatic evaluation, manual evaluation by an expert, and evaluation by the end user [29]. Often, technological tools are employed to automatically detect accessibility issues in software [4]. However, these tools may not always identify all problems and can generate false positives, highlighting the importance of expert oversight in the evaluation process. Additionally, relying on these tools for evaluation only in the final stages of software development can lead to increased redesign and maintenance costs [30].

In response to these challenges, several initiatives have been developed to integrate web accessibility features from the early stages of software development using a model-driven development (MDD) approach. For instance, a systematic literature review conducted by [12] analyzes various accessible software development proposals employing MDD. Notably, the proposal by [31], known as Accessorial - Moodle acc^{+}, aims to automate the development of web applications catering to a diverse user base, particularly users with disabilities, through a new CMS (Content Management System) leveraging MDD concepts.

Similarly, [32] identifies accessibility barriers in mobile learning environment chats, proposing a model-driven design strategy for creating accessible mobile chat applications. Additionally, the AWA methodological proposal [33] facilitates the design and development of accessible web applications, aligning with WCAG 2.0 navigation requirements throughout the application life cycle.

Other works, such as those by [34] and [35], advocate for the inclusion of accessibility considerations from the software design stage, leveraging abstraction models inspired by IMS Access for All specifications. In the field of e-learning, [36] suggests creating models considering accessibility standards using an MDD approach to adapt user interfaces based on device characteristics and user preferences or needs.

Moreover, [37] proposes a UML model for automatically generating user profiles with disabilities in XML format, enabling e-learning platforms to adapt their interfaces based on user accessibility needs. Lastly, [38] introduces Accessed, utilizing an MDD approach for developing accessible mobile applications for visually impaired individuals, employing a domain-specific language (DSL) within a model editor for defining models.

Comparing our approach with similar works, we offer a solution based on a UML Profile for modeling specific accessibility requirements outlined in WCAG 2.2, using OCL language to establish restrictions aligned with accessibility guidelines. Our approach allows for the incorporation of accessibility features from the software analysis and design stage, providing extensibility for additional features as needed. By raising the level of abstraction and enabling

rapid adaptation to technological changes through model-to-code transformation, our proposal significantly reduces development time and effort.

Furthermore, our approach facilitates early resolution of accessibility issues, unlike reliance on automatic validation tools in the final stages of development, which can lead to time-consuming and costly corrections. Additionally, our implementation utilizes freely accessible tools like Eclipse and the UML modeling language, minimizing implementation costs.

III. METHODS AND MATERIALS

Research in the realm of web accessibility, particularly following a model-driven development approach, holds significance given the challenges developers face in constructing accessible web applications and the critical role accessibility plays in fostering the inclusion of individuals with disabilities in technology usage. Thus, to provide direction to this research endeavor, the primary objective is as follows: Develop a UML profile for accessible web pages based on the WCAG 2.2 standard, enabling the integration of accessibility features from the early stages of the software development process. Aligned with this objective, our study is guided by the following research question: ¿How can web accessibility features be effectively addressed at the outset of the software development life cycle?

A. PROCEDURE

To accomplish the defined objective, a method comprising four stages has been devised, as delineated in Figure 1 and expounded upon in Section VI.

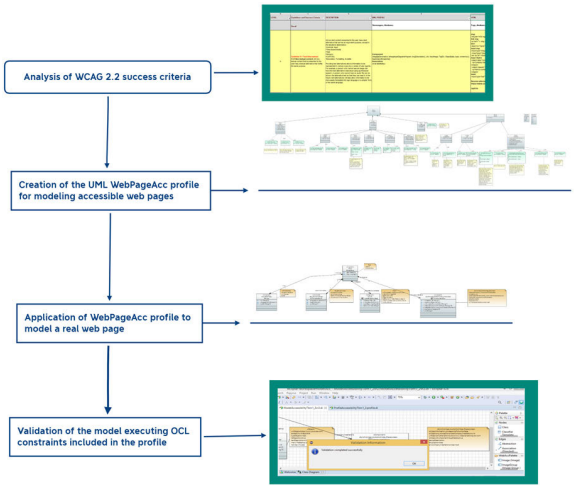


FIGURE 1. Webpage profile creation process is available at https://drive.google.com/file/d/10hWvFE6o0OeU_TDpXfyyK5HRWU_du5tE/view?usp=sharing.

During the initial stage, an analysis of the accessibility criteria outlined in the WCAG 2.2 standard is conducted. This analysis serves to define the requisite stereotypes, attributes, and constraints necessary to fulfill the success criteria.

Subsequently, in the second stage, the Webpage profile is constructed within the Eclipse Modeling tool using Papyrus. This stage entails meticulous consideration of each component, including stereotypes, enumerations, and constraints identified in the previous stage.

Moving forward, the third stage involves the application of the profile by creating a UML class diagram model of an authentic web page.

Lastly, the fourth stage centers on validating the model by executing the OCL rules linked to the attributes of the established stereotypes in the profile.

Furthermore, an experiment was conducted with a group of developers to assess the usability difficulty level of the profile.

For the initial iteration of the profile, the incorporated success criteria are derived from guidelines 1.1 Text Alternatives, 1.2 Time-Based Media, 1.3 Adaptable, 2.4 Navigable, 3.1 Readable, and 4.1 Compatible, in accordance with the WCAG 2.2 recommendation.

B. PROFILE MODEL

For the first prototype, a reduced set of requirements was selected, focusing on basic elements commonly found in an HTML page, such as images, audios, videos, paragraphs, links, forms, and tables. The selection of these requirements followed several principles:

- Prioritizing criteria from levels A and AA, as AAA level criteria are considered less important since they are not required by laws in many countries.
- Ensuring that the selected criteria could be easily incorporated into the profile to enable automated review, referring to WCAG 2.2 techniques.
- Including diverse criteria corresponding to key accessibility principles: Perceptible (P), Operable (O), Understandable (U), and Robust (R).

The specific criteria selected include:

- **Perceptible** (P): 1.1 Text Alternatives (1.1.1 Non-text Content-A), 1.2 Time-based Media (1.2.1 Audio-only and Video-only-A, 1.2.2 Captions (Prerecorded)- A, 1.2.3 Audio Description or Media Alternative (Prerecorded) - A, 1.2.4 Captions (Live) - AA, 1.2.5 Audio Description (Prerecorded) – AA), 1.3 Adaptable (1.3.1 Info and relationships-A).
- **Operable** (O): 2.4 Navigable (2.4.2 Page Titled - A, 2.4.4 Link Purpose - A, 2.4.6 Headings and Labels - A).
- **Understandable** (U): 3.1 Readable (3.1.1 Language of Page - A).
- **Robust** (R): 4.1 Compatible (4.1.2 Name, Role, Value - A)

C. TOOLS AND SUPPORT LANGUAGES FOR MDD

For the development of the proposal, the Eclipse Papyrus tool [39] has been utilized, along with languages such as UML for modeling the profile and its application in a specific system model, as well as OCL for creating constraints related to accessibility criteria in the model. These tools

were primarily selected because they are freely available and because Eclipse has extensive supporting documentation.

Papyrus, a tool for graphical modeling of UML2 applications, is an open-source project designed as a component of Eclipse and based on the EMF UML2 metamodel [40]. It can be used either as a standalone tool or as an Eclipse plugin [39]. Papyrus provides support for domain-specific languages and Sims. Moreover, it is designed to be easily extensible as it is based on the principle of UML profiles. UML Profiles serve as the mechanism provided by UML itself to extend its syntax and semantics to express specific concepts of a given application domain.

UML [16] is the most widely known and used language for software systems modeling today, endorsed by the Object Management Group. It is a graphical language that allows the visualization, specification, construction, and documentation of a system.

Object Constraint Language (OCL) [18] enables the formal specification of constraints within a UML model. It presents a formal syntax and semantics for OCL based on set theory, which includes expressions, invariants, and pre- and post-conditions. A formal base for OCL ensures precise constraint meanings, aiding in the elimination of ambiguities and inconsistencies.

IV. RESULTS

This section reports the results of applying the process for profile creation shown in Figure 1. A detailed summary of the activities carried out in each of the four stages for the development of the proposal is provided, including:

1. Analysis of the accessibility criteria established in the WCAG 2.2 standard.
2. Creation of the Webpage profile.
3. Application of the profile by creating a UML class diagram model representing a real web page.
4. Validation of the model by executing OCL rules.

Additionally, the results of the survey conducted with a group of developers are presented, showing each one's perspective on their experience with accessibility and model-driven development, as well as the level of difficulty encountered when using the proposed profile.

A. ANALYSIS OF WCAG 2.2 SUCCESS CRITERIA

Web Content Accessibility Guidelines (WCAG) 2.2 Success Criteria provide guidelines to ensure that web content is accessible to a wide range of people with disabilities. Designing a UML profile to model accessible web pages involves aligning with these success criteria and associated techniques. Table 1 describes the WCAG 2.2 criteria selected for the design of the proposed profile. These criteria pertain to fundamental elements of an HTML page, such as the proposed profile is defined by a set of stereotypes, each extending the UML language for accessible websites. Each created stereotype represents different HTML tags of an accessible web page, such as images, videos, audios, navigation menus, links, definition lists, headings, paragraphs, content sections, forms and tables. Additionally, constraints and tagged values are applied to represent the main properties of the model conceptually. This study utilized Object Constraint Language (OCL) [8] to specify constraints associated with the stereotypes defined in the profile, considering the Web Content Accessibility Guidelines (WCAG) 2.2 [9], thereby avoiding arbitrary and erroneous usage of each defined stereotype.

Success Criterion 1.1.1 ''Non-Text Content'' of WCAG 2.2 emphasizes the need to provide text alternatives to ensure equitable access to information. This criterion is crucial to ensure that non-text content, such as images and multimedia elements, is accessible to all, including those relying on assistive technologies like screen readers. Properly described text alternatives are essential for conveying visual information to users with visual impairments.

Success Criterion 1.2.1 emphasizes the inclusion of alternatives to ensure access for individuals unable to access primary multimedia content through traditional means.

Success Criterion 1.3.1 ''Information and Relationships'' ensures that web page information is structured and organized in a way that is understandable to all users, including those using assistive technologies.

Success Criterion 2.4.2 highlights the importance of page titles, facilitating user understanding and navigation.

Success Criterion 2.4.4 ''Link Purpose'' emphasizes providing clear information about link purposes, benefiting users with visual or cognitive disabilities.

Success Criterion 2.4.6 ensures that headings and labels are used effectively to provide a clear and meaningful page structure, benefiting navigation and content comprehension for all users.

Success Criterion 3.1.1 ''Page Language'' ensures accessibility and proper interpretation of content by assistive technologies and multilingual users.

Success Criterion 4.1.2 focuses on ensuring that the name, role, and value of each user interface component are programmatically determinable, benefiting users relying on assistive technologies when navigating or interacting with elements such as forms.

Designing a UML profile to model accessible web pages involves extending existing UML elements, introducing stereotypes, and incorporating additional notations to represent key aspects of WCAG 2.2 success criteria and associated techniques. The goal is to provide a modeling framework facilitating the integration of accessibility considerations in the early stages of web development.

B. CREATION OF THE WEBPAGEACC PROFILE

To address the research question of how to consider web accessibility features from the early stages of the software development life cycle, the UML profile called WebPageAcc is proposed, as shown in Figure 2. This profile advocates the inclusion of accessibility features from the beginning of the software development process, achieved by building a platform-independent model.

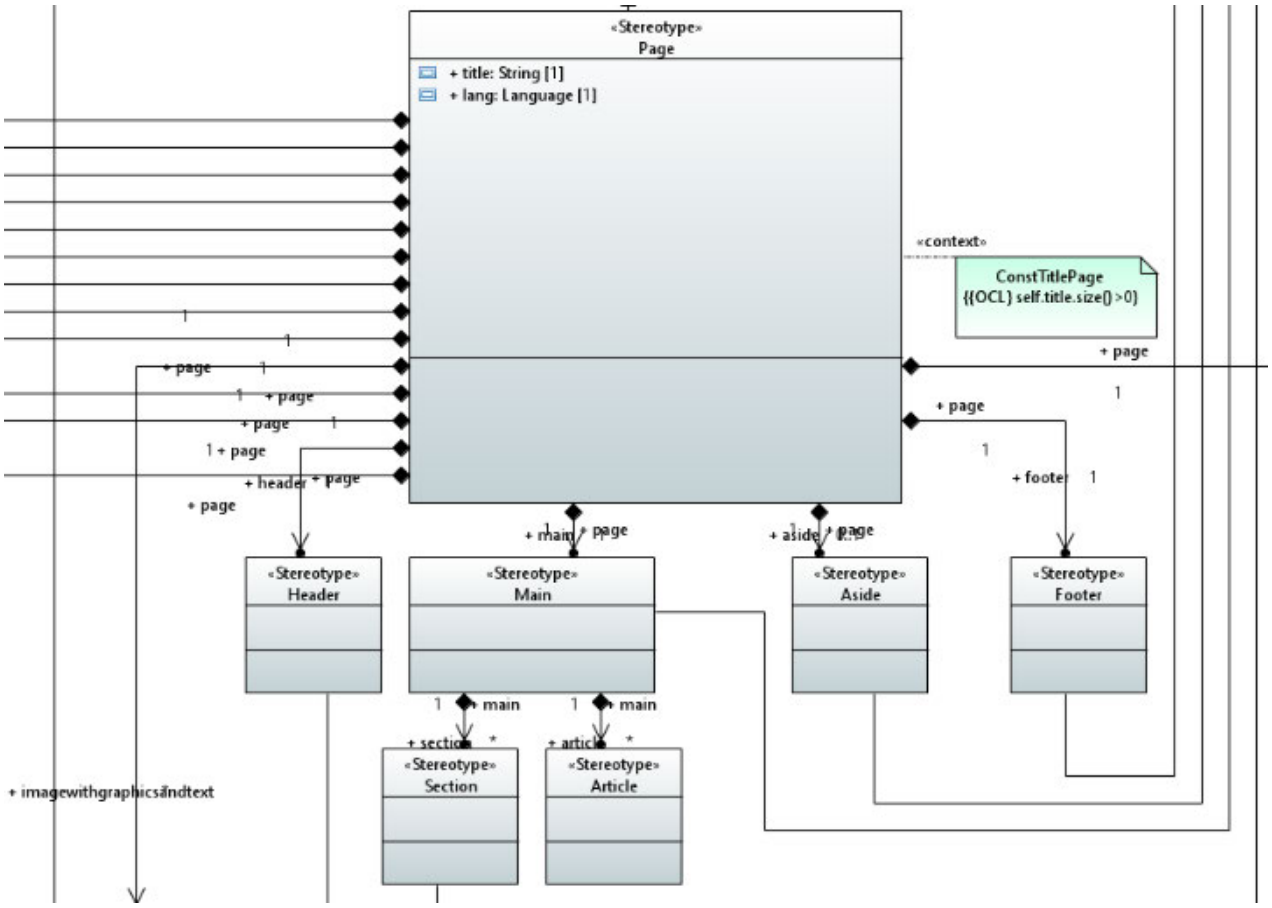


FIGURE 2. UML Profile WebAcc. The figure in high resolution is available at https://drive.google.com/file/d/1m7DbJLbUlJ-2MZWjgeOqK06XVAf00TZ3/view?usp=sharing.

The WebPageAcc profile makes it quick to model the main features of an accessible web page. Several success criteria outlined in the WCAG 2.2 guidelines have been incorporated into this profile, including 1.1 Text Alternatives, 1.2 Time-Based Media, 1.3 Responsive, 2.4 Navigable, 3.1 Readable, and 4.1 Compatible.

The profile creation process began with defining stereotypes that represent each element of the HTML web page, such as images, videos, audios, navigation menus, links, definition lists, headings, paragraphs, content sections, forms and tables. Figure 2 illustrates the WebPageAcc profile, composed of several stereotypes (≪…≫) linked to the tags used in an accessible web page. Detailed descriptions of each stereotype are provided in Table 2.

Additionally, elements of type ≪Enumeration≫ have been added to the profile, as described in Table 3. These enumeration literals, such as ≪Language≫, offer options such as Spanish, English, French, German, and Italian for their inclusion in the model.

In summary, the WebPageAcc profile provides a framework for integrating accessibility features from the early stages of the software development lifecycle by representing key elements of an accessible web page in a platform-independent model.

Below, we provide a description of each stereotype defined in the profile:

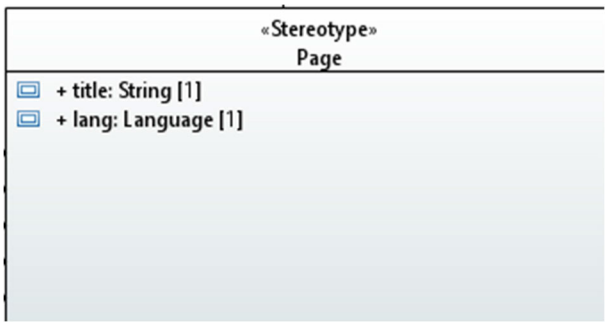


FIGURE 3. Stereotype ≪Page≫.

Among the various stereotypes defined in the profile, one notable example is the ≪Page≫ stereotype illustrates in Figure 3. This stereotype symbolizes the creation of a web page, which necessitates a title and an associated language, as mandated by success criteria 2.4.2 Page Title and 3.1.1 Language of the Page.

In Figure 4, we can observe the ≪ImageNoDecorative≫ stereotype. The OCL restrictions associated with this

TABLE 1. Success criteria of WCAG 2.2 and associated techniques, analyzed to design the profile.

SUCCESS CRITERIA (SC)	GENERAL (G) AND HTML (H) TECHNIQUES
1. Perceivable **Guideline 1.1 Text Alternatives** **SC 1.1.1 Non-text Content (A):** All non-text content that is presented to the user has a text alternative that serves the equivalent purpose.	G92: Providing long description for non-text content that serves the same purpose and presents the same information G95: Providing short text alternatives that provide a brief description of the non-text content G196: Using a text alternative on one item within a group of images that describes all items in the group H2: Combining adjacent image and text links for the same resource H24: Providing text alternatives for the area elements of image maps H35: Providing text alternatives on applet elements H37: Using alt attributes on img elements H53: Using the body of the object element H67: Using null alt text and no title attribute on img elements for images that AT should ignore
1. Perceivable **Guideline 1.2 Time-based Media** **SC 1.2.1 Audio-only and Video-only (Prerecorded) (A):** For prerecorded audio-only an alternative for time-based media is provided that presents equivalent information for prerecorded audio-only content, and for prerecorded video-only either an alternative for time-based media or an audio track is provided that presents equivalent information for prerecorded video-only content.	G158: Providing an alternative for time-based media for audio-only content G159: Providing an alternative for time-based media for video-only content G166: Providing audio that describes the important video content and describing it as such H96: Using the track element to provide audio descriptions
1. Perceivable **Guideline 1.2 Time-based Media** **SC 1.2.2 Captions (Prerecorded) (A):** Captions are provided for all prerecorded audio content in synchronized media, except when the media is a media alternative for text and is clearly labeled as such.	G87: Providing closed captions G93: Providing open (always visible) captions H95: Using the track element to provide captions
1. Perceivable **Guideline 1.2 Time-based Media** **SC 1.2.3 Audio Description or Media Alternative (Prerecorded) (A):** An alternative for time-based media or audio description of the prerecorded video content is provided for synchronized media, except when the media is a media alternative for text and is clearly labeled as such.	G8: Providing a movie with extended audio descriptions G58: Placing a link to the alternative for time-based media immediately next to the non-text content G69: Providing an alternative for time based media G78: Providing a second, user-selectable, audio track that includes audio descriptions G159: Providing an alternative for time-based media for video-only content G173: Providing a version of a movie with audio descriptions G203: Using a static text alternative to describe a talking head video H53: Using the body of the object element H96: Using the track element to provide audio descriptions
1. Perceivable **Guideline 1.2 Time-based Media** **SC 1.2.4 Captions (Live) (AA):** Captions are provided for all live audio content in synchronized media.	G9: Creating captions for live synchronized media G87: Providing closed captions G93: Providing open (always visible) captions
1. Perceivable **Guideline 1.2 Time-based Media** **SC 1.2.5 Audio Description (Prerecorded) (AA):** Audio description is provided for all prerecorded video content in synchronized media.	G78: Providing a second, user-selectable, audio track that includes audio descriptions G203: Using a static text alternative to describe a talking head video H96: Using the track element to provide audio descriptions
1. Perceivable **Guideline 1.3 Adaptable** **SC 1.3.1 Info and Relationships (A):** Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.	H101: Using semantic HTML elements to identify regions of a page G115: Using semantic elements to mark up structure AND H49: Using semantic markup to mark emphasized or special text G117: Using text to convey information that is conveyed by variations in presentation of text G140: Separating information and structure from presentation to enable different presentations G138: Using semantic markup whenever color cues are used H51: Using table markup to present tabular information H39: Using caption elements to associate data table captions with data tables H63: Using the scope attribute to associate header cells and data cells in data tables H43: Using id and headers attributes to associate data cells with header cells in data tables H44: Using label elements to associate text labels with form controls

TABLE 1. *(Continued.)* **Success criteria of WCAG 2.2 and associated techniques, analyzed to design the profile.**

	H65: Using the title attribute to identify form controls when the label element cannot be used H71: Providing a description for groups of form controls using fieldset and legend elements H85: Using optgroup to group option elements inside a select H48: Using ol, ul and dl for lists or groups of links H42: Using h1-h6 to identify headings H97: Grouping related links using the nav element
2.Operable **Guideline 2.4: Navigable:** **SC 2.4.2 Page Titled (A):** Web pages have titles that describe topic or purpose.	G88: Provide descriptive titles for web pages AND associate a title with a web page using one of the following techniques: H25: Provide a title using the title element (HTML)
2.Operable **Guideline 2.4: Navigable:** **SC 2.4.4 Link Purpose (A) (In Context):** The purpose of each link can be determined from the link text alone or from the link text together with its programmatically determined link context, except where the purpose of the link would be ambiguous to users in general.	G91: Provide text on links that describes the purpose of the link H30: Provide text in links that describes the purpose of the link for anchor elements (HTML) H24: Provide textual alternatives for area elements in image maps (HTML) G189: Provide a control near the beginning of the web page that changes the text of links G53: Identify the purpose of a link using the text of the link along with the text of the phrase that contains it Provide a complementary description of the purpose of a link using one of the following techniques: H33: Complement the text of a link with the title attribute (HTML) H77: Identify the purpose of a link using its text combined with the containing list element (HTML) H78: Identify the purpose of a link using its text combined with the paragraph that contains it (HTML) H79: Identify the purpose of a link using its text combined with the containing cell and associated table headers (HTML) H80: Identify the purpose of a link using its text combined with the heading element preceding it (HTML) H81: Identify the purpose of a link using its text combined with the parent list element of the containing list (HTML)
2.Operable **Guideline 2.4 Navigable** **SC 2.4.6 Headings and Labels (Nivel AA):** Headings and labels describe topic or purpose.	G130: Providing descriptive headings G131: Providing descriptive labels
3.Understandable **Guideline 3.1 Readable** **SC 3.1.1 Language of Page (A):** The default human language of each Web page can be programmatically determined.	H57: Using the language attribute on the HTML element
4. Robust **Guideline 4.1 Compatible** **SC 4.1.2 Name, Role, Value (A):** For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.	G108: Using markup features to expose the name and role, allow user-settable properties to be directly set, and provide notification of changes H91: Using HTML form controls and links H44: Using label elements to associate text labels with form controls H64: Using the title attribute of the iframe element H65: Using the title attribute to identify form controls when the label element cannot be used H88: Using HTML according to spec

stereotype enable compliance with criterion 1.1.1 Non-text content., which mandates that non-decorative images have alternative text, described in the "alt" attribute of HTML "IMG" elements. The use of Alt Text in an image implies providing a concise textual translation related to the image, offering necessary information to describe it adequately.

It is crucial to note that alternative text serves several functions on a website: Firstly, it is read by screen readers in place of images, thereby ensuring that the content and function of the image are accessible to individuals with visual or cognitive impairments. Secondly, it furnishes semantic and descriptive meaning to the image, which can later be utilized by search engines to determine the image's content. Consequently, images on a web page not only enhance its presentation but also serve as a means of conveying information.

In Figure 5, we can observe the stereotype ≪Image Decorative≫, which enables us to illustrates decorative images that do not contribute any information to the content. Their sole purpose is to provide visual embellishment to the page; therefore, they should not be accompanied by text alternatives. Which helps with compliance with criterion 1.1.1 Non-text content.

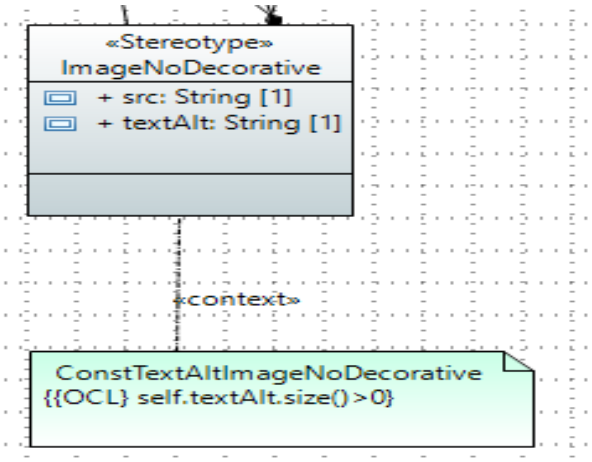


FIGURE 4. Stereotype ≪ImageNoDecorative≫.

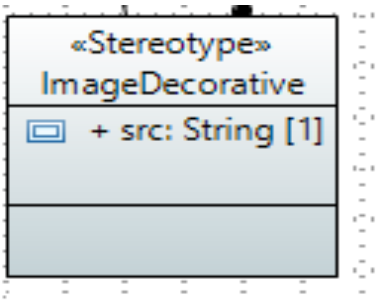


FIGURE 5. Stereotype ≪ImageDecorative≫.

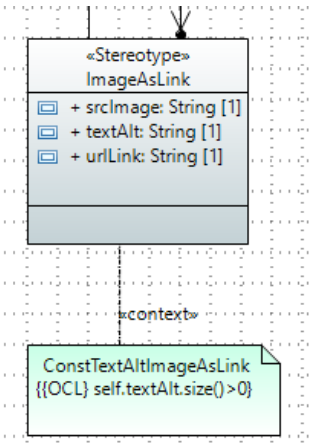


FIGURE 6. Stereotype ≪ImageAsLink≫.

Figure 6 illustrates an image functioning as a link. This can be achieved in HTML by employing the <a> (anchor) tag in conjunction with the <img> (image) tag. To ensure accessibility, it's crucial that the 'alt' attribute of the <img> tag provides descriptive alternative text conveying the purpose or content of the image. What helps to contribute with compliance with criterion 1.1.1 Non-text content.

The profile also encompasses the necessary stereotypes to represent image maps in HTML, such as ≪ImageMap≫ and

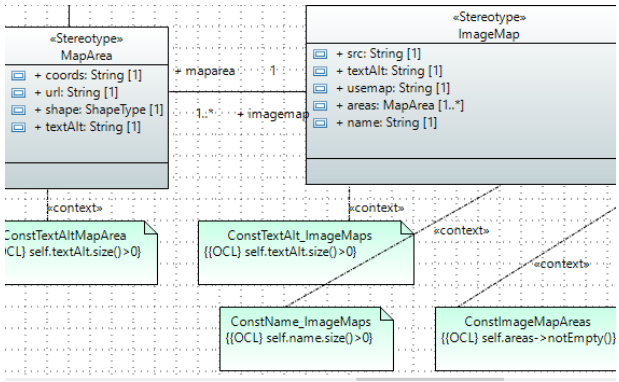


FIGURE 7. Sterotype ≪ MapArea ≫ ≪ ImageMap ≫.

≪MapArea≫, as illustrates in Figure 7. An HTML image map, consisting of its map and area tags, enables authors to delineate regions on an image or object and assign specific actions to each region (e.g., open a document, run a program, etc.). Upon activation of a region by a user, the assigned action is executed.

To ensure accessibility, it is crucial to provide alternative text for each <area> tag within the image map. Therefore, the OCL constraint of the ≪MapArea≫ stereotype mandates that the 'textAlt' attribute cannot be empty. As each hotspot functions akin to a link, the alt text should convey meaningful information even when presented out of context.

Moreover, the ≪ImageMap≫ stereotype incorporates the 'textAlt' attribute, facilitating the specification of the image map's description. Which helps with compliance with criterion 1.1.1 Non-text content.

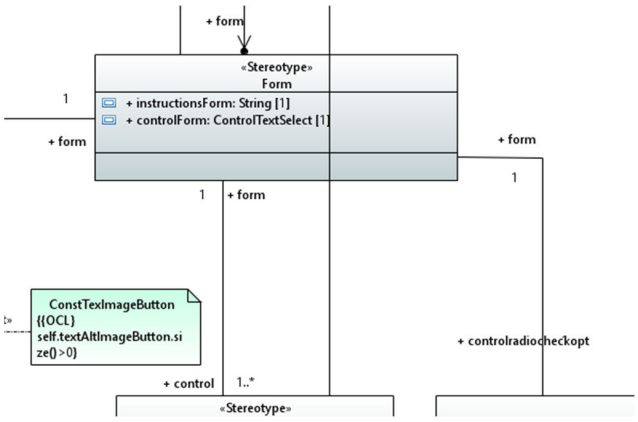


FIGURE 8. Stereotype ≪ Form ≫.

Figure 8 illustrates the ≪Form≫, ≪FormControl≫, and ≪Button≫ stereotypes, representing the accessible forms of an HTML web page. Accessibility in forms entails organization and usability, ensuring they are well-structured and easy to navigate.

To align with accessibility criteria concerning accessible forms (1.3.1 Info and Relationships and 4.1.2 Name, function, value), various OCL restrictions have been incorporated.

Effective form creation involves organization and labeling, where labels play a crucial role in identifying form controls. Hence, the association of the 'id' property of the form control and 'for' attribute of the label is imperative. Consequently, OCL rules have been defined within the profile to enforce non-emptiness of the 'id' and 'for' properties, alongside a restriction in code generation process ensuring that the 'id' properties of the control and 'for' attributes of the label are identical.

For instance, positioning the <label> tag adjacent to the form element (e.g., text box, checkbox, radio button, menu/list) is recommended. Although labels can sometimes be hidden from view, they remain accessible to screen readers. The ≪Button≫ stereotype within the form facilitates the submission and resetting of input elements. Clear instructions regarding the information to be submitted are also essential. If form elements are mandatory, users should be informed accordingly.

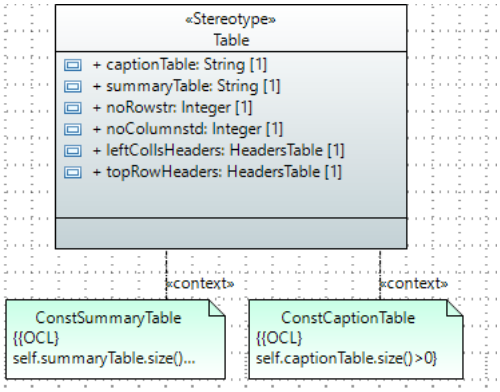


FIGURE 9. Stereotype ≪Table≫.

Figure 9 shows the ≪Table≫ stereotype, equipped with properties and constraints essential for meeting the WCAG 2.2 criteria concerning accessible tables (1.3.1 Info and Relationships). Consequently, associated restrictions mandate that the 'summary' and 'caption' attributes of the table must not be left empty.

Understanding a table's data on a webpage is straightforward when viewed in its entirety but challenging when examined piecemeal. Users relying on non-visual browsers face difficulties as they navigate tables linearly, losing the comprehensive view and reference points vital for interpreting cell content. Isolated cell content may lack context without knowledge of its row and column placement. To mitigate this issue, headers—defined using the <th> tag in HTML—can be employed, encompassing both vertical and horizontal headers. To facilitate this, the stereotype incorporates two Boolean properties, enabling selection of row and/or column headers, thereby ensuring their inclusion in the resulting webpage code.

Moreover, including a summary that succinctly outlines the table content is imperative. In HTML, the table summary is specified using the 'summary' attribute of the <table> tag. Non-visual browsers, such as screen readers or those utilizing braille displays, relay this attribute's content to users. Similarly, correctly defining the table title necessitates utilizing the <caption> tag. Hence, OCL restrictions have been established for these attributes to prevent their omission when adding a table to the model.

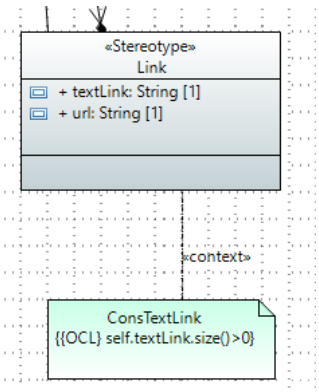


FIGURE 10. Stereotype ≪Link≫.

Figure 10 illustrates the ≪Link≫ stereotype, featuring corresponding attributes and constraints designed to ensure adherence to WCAG 2.2 accessibility criteria concerning links (2.4.4 Purpose of links (in context)).

In line with accessibility standards, links must possess descriptive attributes, clearly delineating their purpose within the link text. The text associated with a link should effectively communicate its intended function. Thus, as stipulated by OCL restrictions, it is imperative that the 'textLink' attribute, representing the text to be displayed in the link, is not left empty.

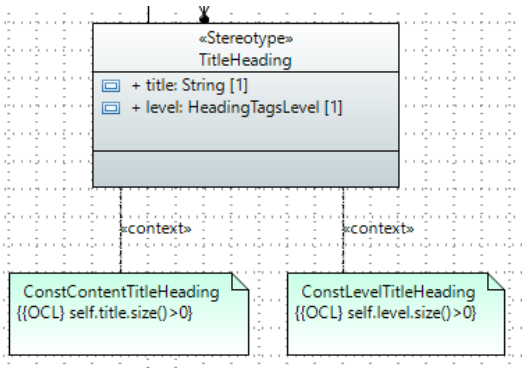


FIGURE 11. Stereotype <TitleHeading>.

Figure 11 shows the representation of Heading Tags on a web page through the ≪TitleHeading≫ stereotype, aimed at ensuring compliance with the corresponding WCAG 2.2 criteria (1.3.1 Info and Relationships).

Headings play a pivotal role in conveying the organizational structure of page content on the web. They serve as navigational aids for web browsers, plug-ins, and assistive technologies, facilitating page navigation by virtue of their

hierarchical arrangement. Heading tags are hierarchically nested, with <h1> denoting the highest level of importance and <h6> representing the lowest.

To adhere to these principles, an OCL restriction mandates the mandatory selection of the heading level and the title. This ensures that headings are appropriately structured and categorized, thereby enhancing the accessibility and navigability of the web page for all users.

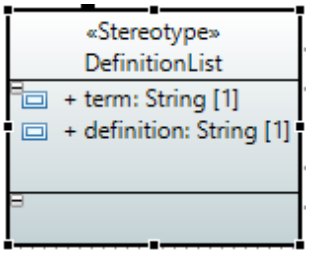


FIGURE 12. Stereotype ≪DefinitionList≫.

In Figure 12, the stereotype ≪DefinitionList≫ is represented, which allows the representation of a list of definitions. This stereotype includes properties for "term" and "definition," which reflect components typically found in a list of definitions.

The use of these attributes aligns with the success criteria described in 1.3.1 Information and Relationships, which belong to the Definition Lists. By ensuring that both the term and its corresponding definition are provided, the model adheres to accessibility guidelines intended to improve the clarity and organization of content for all users.

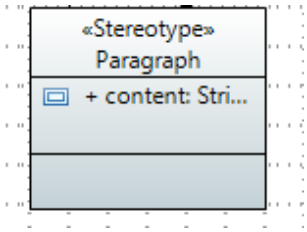


FIGURE 13. Stereotype ≪Paragraph≫.

In Figure 13, the ≪Paragraph≫ stereotype is illustrates, enabling the definition of content through paragraphs on a website, akin to the <p> tag in HTML. It's noteworthy that the Paragraph tag facilitates document navigation for users of screen readers and other assistive technologies, enabling them to navigate the content by jumping from one paragraph to another. This semantic representation of text in paragraph form aids in comprehending the structure and organization of the content.

By incorporating this stereotype, compliance with the WCAG 2.2 guidelines outlined in criterion 1.3.1 Info and Relationships, which pertain to paragraphs, can be achieved. This ensures that the model effectively represents the structural components of web content, enhancing accessibility and usability for all users.

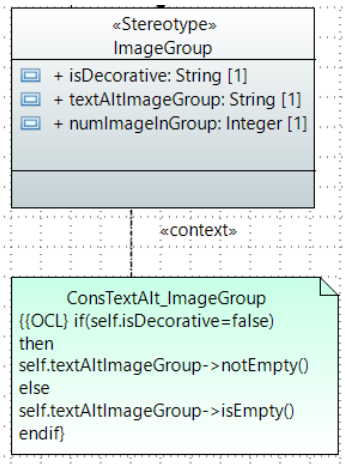


FIGURE 14. Stereotype ≪ImageGroup≫.

Figure 14 illustrates the ≪ImageGroup≫ stereotype, which symbolizes a cluster or set of correlated images. The associated OCL constraint mandates that the alternative text should not be left empty for groups comprising non-decorative images.

Sometimes, groups of images are employed collectively to convey a piece of information. For instance, a series of star icons might be utilized together to indicate a rating. In such instances, only one of the images necessitates a text alternative to describe the entire ensemble, while the remaining images possess a null (empty) alt attribute to be disregarded by assistive technology.

This adherence to the WCAG 2.2 guidelines, delineated in criterion 1.1.1 Non-text content, ensures the accessibility of image groups, enhancing the inclusivity of web content for all users.

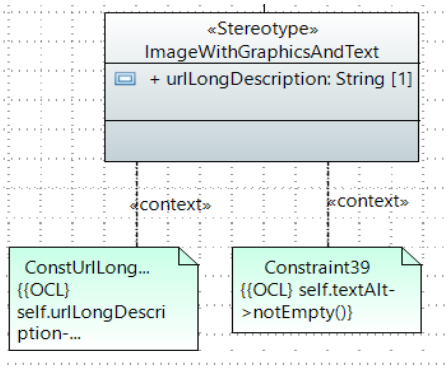


FIGURE 15. Stereotype ≪ImageWithGraphicsAndText≫.

Figure 15 illustrates the ≪ImageWithGraphicsAndText≫ stereotype, designed for images that integrate both text and graphics, such as diagrams, mind maps, infographics, tables, and statistical graphs. To adhere to the guidelines outlined in criterion 1.1.1 Non-text content concerning grouped images, two interconnected OCL restrictions have been established.

The first OCL restriction mandates that the group of images must include a comprehensive description of the image content. Meanwhile, the second restriction necessitates the inclusion of concise alternative text associated with each image.

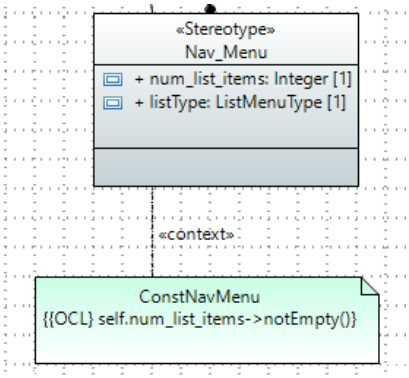


FIGURE 16. Stereotype ≪Nav Menu≫.

Figure 16 shows the representation of the navigation element of a web page using the stereotype ≪Nav_Menu≫. The HTML5 navigation element, introduced with the <nav> tag, plays a crucial role in improving accessibility. This tag allows you to clearly delimit the navigation elements within a web page, such as the main menu or the secondary menu.

Using the HTML5 <nav> tag significantly improves accessibility by providing a means to clearly identify the navigation elements present on a web page. This allows assistive technologies, such as screen readers, to identify and present navigation options to users when necessary. It is important to note that the menu is implemented using an HTML list (with <li> elements), a feature considered in the listType attribute of type ListMenuType. This is aligned with the requirements described in success criteria 1.3.1 Information and relationships.

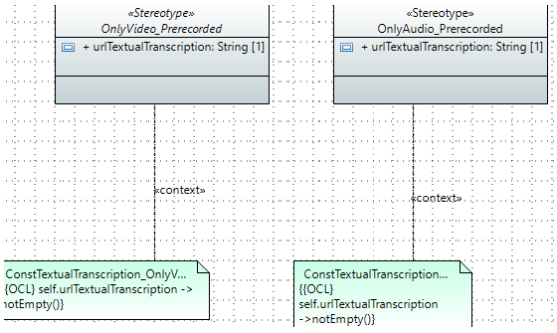


FIGURE 17. Stereotype ≪Only Video_Prerecorded≫ and ≪OnlyAudio_Precored≫.

To adhere to criterion 1.2.1 Audio-only and Video-only, the stereotypes ≪OnlyVideo_Prerecorded≫ and ≪OnyAudio_Prerecorded≫ have been established, as shows in Figure 17. These stereotypes address recorded audio-only and video-only content, respectively. In compliance with the criterion, except where the audio or video serves as a media alternative to text and is clearly identified as such, the following conditions apply:

For recorded audio-only content: The page must provide equivalent information, typically through a text transcript of the recorded audio. This transcript should include a verbatim record of all spoken content, identify speakers, and note significant sounds like applause, laughter, and audience questions. Hence, the associated OCL restriction mandates that the 'urlTextualTranscription' property must not be empty.

For recorded video-only content: Equivalent information must be provided, typically through a textual transcript. Similar to audio content, this transcript should offer a detailed account of the video's content. Therefore, the OCL restriction associated with the stereotype stipulates that the 'urlTextualTranscription' property must not be empty.

It's crucial to recognize that the goal of digital accessibility extends beyond catering to specific disabilities; it aims to provide an optimal experience for all users. Disabilities affect individuals differently, and various accessibility features cater to different needs. For instance, while some deaf a hearing impaired users may prefer subtitles, individuals with cognitive disabilities might find transcripts beneficial for reading at their own pace. Likewise, visually impaired users can utilize screen readers for video captions and refer to transcripts for descriptions of visual content.

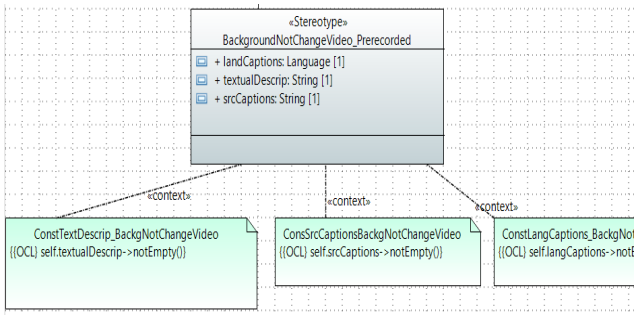


FIGURE 18. Stereotype ≪BackgroundNotChangeVideo_Prerecorded≫.

Figure 18 illustrates the ≪BackgroudNotChangeVideo_Prerecorded≫ stereotype, representing videos with a static background, such as speeches, to comply with the WCAG 2.2 criteria (1.2.3 Audio Description or Media Alternative (Prerecorded)). To adhere to this criterion:

- The 'textualDescrip' attribute must not be empty, as it requires a general description of the environment illustrates in the video.
- The 'srcCaptions' attribute should also not be empty, as it contains supplementary video Captions. In HTML5, these Captions are denoted by the <track> tag. It's worth noting that captions not only transcribe the audio but also describe other sounds such as music, screams, and impacts.

These provisions ensure that individuals with visual o hearing impairments can access and understand the video content effectively.

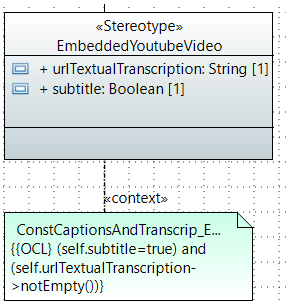


FIGURE 19. Stereotype ≪EmbeddedYoutubeVideo≫.

Figure 19 illustrates the representation of embedded YouTube videos using the ≪EmbeddedYoutubeVideo≫ stereotype. Embedded YouTube videos are generally accessible to most individuals with disabilities, including those who use screen readers and other assistive technologies. When using the standard embed code provided by YouTube, the video player already meets several accessibility criteria.

However, to fully comply with WCAG 2.2 requirements for subtitles and audio descriptions, additional steps are necessary. To meet the guidelines in criterion 1.2.3 Audio Description or Media Alternative (Prerecorded):

- The 'urlTextualTranscription' property must not be empty, as it requires a transcript that describes non-speech sounds and all visual material, with the sequence matching that of the video.
- Subtitles must be provided, as indicated by the 'subtitle' property, which must be set to true.

It's important to differentiate between transcripts and captions, as they serve distinct purposes. While captions provide audio translation for users of other languages, transcripts offer a comprehensive record of all audio content, including descriptions of visual elements.

Moreover, incorporating subtitles and transcripts in videos can enhance website SEO, as search engines index not only the media content but also the text in transcripts and subtitles. This boosts opportunities for better search engine ranking with relevant keywords and content.

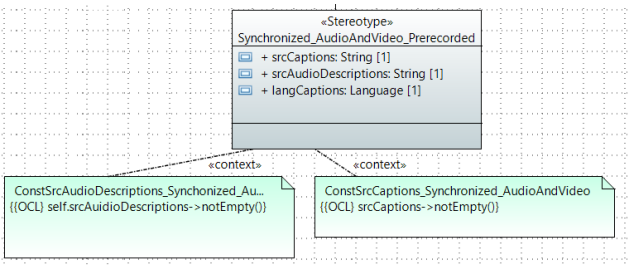


FIGURE 20. Stereotype ≪Synchronized_AudioAndVideo_Prerecorded≫.

Figure 20 shows the ≪Synchronized_AudioAndVideo_Prerecorded≫ stereotype, representing pre-recorded videos with synchronized audio. This enables compliance with several criteria from WCAG 2.2, including 1.2.1 Audio-only and Video-only (Prerecorded), 1.2.2 Captions (Prerecorded), 1.2.3 Audio Description or Media Alternative (Prerecorded), and 1.2.5 Audio Description (Prerecorded).

To adhere to these guidelines:

- The 'srcAudioDescriptions' property, containing audio descriptions, must not be empty.
- The 'srcCaptions' property, encompassing captions, must also not be empty.

At this point it is important to remember that captions are similar to dialogue-only subtitles, except that captions convey not only the spoken dialogue content, but also non-dialogue audio equivalents of information necessary to understand the content of the program, including sound effects, music, laughter, speaker identification, and location.

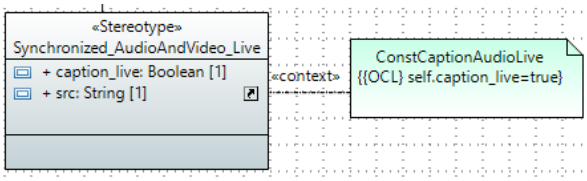


FIGURE 21. Stereotype ≪Synchronized_AudioAndVideo_Live≫.

Figure 21 illustrates the ≪Synchronized_AudioAndVideo_Live≫ stereotype, symbolizing synchronized live video with audio. To adhere to the WCAG 2.2 criterion 1.2.4 Captions (Live), an OCL restriction mandates setting the 'caption_live' property as true. This ensures that captions are attached to live broadcast videos, synchronizing with the media file to provide crucial audio information, including spoken words and significant sound effects.

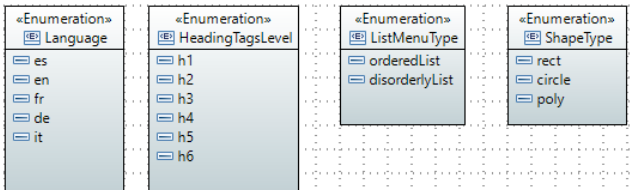


FIGURE 22. Stereotype Enumeration ≪Languaje≫ ≪HeadingTagsLevel≫≪ListMenuType≫≪ShapeType≫.

In Figure 22 Enumeration type elements have been introduced to define the required options for attributes in different stereotypes, such as language labels, header levels, menu types, shape types in image map areas. This ensures that users can only select from predefined options within the enumeration.

Figure 23 illustrates the semantic elements that constitute the structure of an HTML5 webpage, all of which are associated with the ≪Page≫ stereotype. These stereotypes have been defined to adhere to technique H101: Using semantic HTML elements to identify regions of a page, as outlined in success criteria 1.3.1 Info and Relationships.

HTML offers several semantic elements to delineate different sections of a webpage:

TABLE 2. Profile stereotypes.

Stereotype	Description	Success criteria (WCAG)	Attributes	Attributes type	Attributes description
<<Page>>	Represents a web page	**2.4.2 Page titled**	Title	String	Page title. Corresponds to <title> tag in HTML.
		3.1.1 Language of the page	Lang	Language	Default language of the page. Corresponds to "lang" attribute of the <html> tag in HTML.
<<Header>>	Represents an introductory content container or a set of navigation links	**1.3.1 Info and Relationships.**	-	-	-
<<Main>>	Defines the primary content of a page.	**1.3.1 Info and Relationships.**	-	-	-
<<Section>>	Defines a section on a page	**1.3.1 Info and Relationships.**	-	-	-
<<Article>>	Represents a section of content that forms an independent part of a document or site; for example, a magazine or newspaper article.	**1.3.1 Info and Relationships.**	-	-	-
<<Aside>>	Specifies the content of a page's sidebar.	**1.3.1 Info and Relationships.**	-	-	-
<<Footer>>	Defines a footer for a document or section.	**1.3.1 Info and Relationships.**	-	-	-
<<ImageNoDecorative>>	Stereotype that represents a non-decorative image.	**1.1.1 Non-text content**	src	String	Image file address. Corresponds to "src" attribute in HTML.
			textAlt	String	Textual description of the image. Corresponds to "alt" attribute in HTML.
<<ImageDecorative>>	Stereotype that represents a decorative image.	**1.1.1 Non-text content**	src	String	Image file address. Corresponds to "src" attribute in HTML.
<<ImageAsLink>>	Link to a web resource, clicking on an image.	**1.1.1 Non-text content**	srcImage	String	Image file address. Corresponds to "src" attribute in HTML.
			textAlt	String	Textual description of the image. Corresponds to "alt" attribute in HTML.
			urlLink	String	Link that redirects to a page.
<<ImageMap>>	Represents an image map. This is an image with clickable areas.	**1.1.1 Non-text content**	src	String	Image file address. Corresponds to "src" attribute of <img> tag in HTML.
			textAlt	String	Textual description of the image. Corresponds to "alt" attribute in HTML.
			usemap	String	Address of a file that contents a long description of the image. Images that have graphics and text such as diagrams and infograms need a longer description. If "alt" attribute value is longer than 150 characters, W3C suggests that a description file should be created than 150

TABLE 2. *(Continued.)* **Profile stereotypes.**

Stereotype	Description	Success criteria (WCAG)	Attributes	Attributes type	Attributes description
					characters. Correspond to "longdesc" attribute of <img> tag in HTML.
			name	String	Name of the image map. Correspond to "name" attribute of <map> tag in HTML.
			areas	<<MapArea []>>	Areas included in the image map
<<MapArea>>	Represents a clickable area within an image map.	**1.1.1 Non-text content**	url	String	Address of the link associated to the area. Corresponds to "href" attribute of <area> tag in HTML.
			coords	String	Values that specify the shape and location of the active area in an image, "coords" tag in HTML.
			shape	ShapeType	Defines the shape of the active region, of type rect, circle and poly.
			textAlt	String	Textual description of the image. Corresponds to "alt" attribute in HTML.
<<ImageGroup>>	Represents a group or collection of related images.	**1.1.1 Non-text content**	isDecorative	Boolean	Indicates if the group is decorative (true) o not (false).
			textAltImageGroup	String	Textual description of the group.
			numImagesIngroup	Integer	Number of images in the group
<<ImageWithGraphicsAndText>>	Images containing Diagrams, Mind Maps, Infograms, Tables, Statistical Graphics or images that combine text with graphics.	**1.1.1 Non-text content**	Src	String	Image file address.
			textAlt	String	Textual description of the image.
			urlLongDescription	String	Address of a file that contents a long description of the image. Images that have graphics and text such as diagrams and infograms need a longer description. If "alt" attribute value is longer than 150 characters, W3C suggests that a description resource should be created [W3C, 2000]. Correspond to "longdesc" attribute of <img> tag in HTML.
			ImageLink	Boolean	Link Image - Provide both text and iconic representations of links without making the web page more confusing or difficult for keyboard users or assistive technology users.
			imageLink	String	imageLink.- Provides the image link
<<Form>>	Web form is composed of one or more form controls, which can include single- or multi-line text fields, drop-down menus, buttons, check boxes, and radio buttons.	**1.3.1 Info and Relationships:** **2.4.6. Headers and labels** **4.1.2 Name, function, value**	instructionsForm	String	Provides general instructions to help users fill out the forms. Include instruction text at the beginning of the form.
			controlForm	<<ControlTextSelect l>>	Provides a checkbox, password, radio, text or select type control
<<ControlTextSelect >>	Form control checkbox, password, radio, text or Select.	**1.3.1 Info and Relationships** **4.1.2 Name, function, value**	isrequired	Boolean	Identifies required fields using the REQUIRED property
			id	String	Identifier of the form control, to relate it to the for label
			name	String	Name of the form control
			labelfor	<<Label>>	Association between control and a label
<<Label>>	It is an element used to associate descriptive text to a form element.	**1.3.1 Info and Relationships** **4.1.2 Name, function, value**	for	String	The "for" attribute specifies which form element a tag is bound to. This association is particularly useful, as clicking the label will focus or activate the associated form element.
			text	String	It is the text that is displayed as a label for a form element

TABLE 2. *(Continued.)* **Profile stereotypes.**

Stereotype	Description	Success criteria (WCAG)	Attributes	Attributes type	Attributes description
<<TextField>>	Text input element in an HTML form.	**1.3.1 Info and Relationships** **4.1.2 Name, function, value**	-	-	-
<<TexArea>>	Multi-line text area in a web form	**1.3.1 Info and Relationships** **4.1.2 Name, function, value**	-	-	-
<<Select>>	Options dropdown menu on a web form	**1.3.1 Info and Relationships** **4.1.2 Name, function, value**	num_options	Integer	number of options of the select type control
<<ControlRadioCheckOpt>>	Radio Button and Check Box	**1.3.1 Info and Relationships** **4.1.2 Name, function, value**	lengend	String	Shows text as a title
			name	String	Form Control name
			isrequired	Boolean	Identifies required fields using the REQUIRED property
			id	String	Identifier of the form control, to relate it to the for label
			num_options	Integer	Number of form control options
<<Button>>	Form button, which allows Send the form once the user has filled out all its fields.	**1.3.1 Info and Relationships.** **4.1.2 Name, function, value.**	name	String	Button name
			textButton	String	Assigns an initial value for the button.
<<ButtonImage>>	Button in a user interface that uses an image as its main visual content instead of text	**1.3.1 Info and Relationships.** **4.1.2 Name, function, value.**	textAltImageButton	String	Alternative text for image type buttons.
			srcimagebutton	String	Button image direction
<<Table>>	Table to organize data in rows or columns	**1.3.1 Info and Relationships:**	captionTable	String	Table title
			summaryTable	String	Table summary
			noRowstr	Integer	Number of rows, “tr” in HTML.

TABLE 2. *(Continued.)* **Profile stereotypes.**

Stereotype	Description	Success criteria (WCAG)	Attributes	Attributes type	Attributes description
			noColumns td	Integer	Number of columns, "td" in HTML
			leftCollsHeaders	Boolean	Headings on the left (true or false)
			topRowHeaders	Boolean	Top row headers (true or false)
<<Link>>	Element of an electronic document that allows automatic access to another document or another part of it.	**2.4.4 Propósito de los enlaces (en contexto) Nivel A** **1.3.1 Info and Relationships:**	url	String	URL address of the website.
			textLink	String	Link Text
<<DefinitionList>>	Definition lists are used to make a set of elements with term-description pairs.	**1.3.1 Info and Relationships**	term	String	List term, dt tag in HTML
			definition	String	Definition of the list term, dd in HTML.
<<TitleHeading>>	Header Tags	**1.3.1 Info and Relationships:** **2.4.6 Headers and labels**	title	String	Heading title
			level	HeadingTagsLevel	Tag levels from <h1> to <h6>, to define HTML headers.
<<Paragraph>>	Paragraph of text starting on a new line Paragraph of text starting on a new line	**1.3.1 Info and Relationships:**	content	String	Paragraph content
<<Nav_Menu>>	Navigation menu, which represents a section that links to other pages or other parts of the same page.	**1.3.1 Info and Relationships:**	num_list_items	Integer	Number of item lists. tag for unordered list and for ordered list.
			listType	ListMenuType	Sorted or unordered list type
<<Time_based_Media>>	Time-Based Media. (videos, slides, films, audios or computers)	**1.2.1 Audio-only and Video-only (Prerecorded)** **1.2.2 Captions (Prerecorded)** **1.2.3 Audio Description or Media Alternative (Prerecorded)** **1.2.4 Captions (Live)** **1.2.5**	src	String	Address of the media resource

TABLE 2. *(Continued.)* **Profile stereotypes.**

Stereotype	Description	Success criteria (WCAG)	Attributes	Attributes type	Attributes description
		Audio Description (Prerecorded)			
<<OnlyVideo_Prerecorded>>	Information transmitted using recorded video-only content. An example of video recorded without audio information or user interaction is a silent movie.	**1.2.1 Audio-only and video-only (Prerecorded)**	Src Inherited from <<Time_based_Media>>	String	Address of the video resource
			urlTextualTranscription	String	Web address of textual transcription that presents equivalent information for prerecorded video-only content
<<OnlyAudio_Prerecorded>>	Information transmitted through recorded audio-only content. An example would be the recording of a speech.	**1.2.1 Audio-only and video-only (Prerecorded)**	Src Inherited from <<Time_based_Media>>	String	Address of the audio resource
			urlTextualTranscription	String	Web address of textual transcription that presents equivalent information for prerecorded audio-only content.
<<BackgroundNotChangeVideo_Prerecorded>>	Prerecorded video with an invariable background, such as speeches in which only one person speaks.	**1.2.3 Audio Description or Media Alternative (Prerecorded)**	Src Inherited from <<Time_based_Media>>	String	Address of the video resource
			srcCaptions	String	Address of captions file (subtitles for the deaf). Corresponds to "src" attribute of <track> tag in HTML.
			langCaptions	<<Language>>	Captions language. Corresponds to "scrlang" attribute of <track> tag in HTML.
			textualDescrip	String	Alternative text containing a general descriptionof the environment presented in the video.
<<EmbeddedYoutubeVideo>>	YouTube video embedded in a web page.	**1.2.3 Audio Description or Media Alternative (Prerecorded)**	Src Inherited from <<Time_based_Media>>	String	Address of the video resource
			urlTextualTranscription	String	Web address of textual transcription
			subtitle	Boolean	Indicates if the video includes subtitles (True) or not (False).
			title	String	Video title from YouTube link
<<Synchronized_AudioAndVideo_Prerecorded>>	Prerecorded video with synchronized audio	**1.2.1 Audio-only and Video-only (Prerecorded)** **1.2.2 Captions (Prerecorded)** **1.2.3 Audio Description or Media**	Src Inherited from <<Time_based_Media>>	String	Address of the video resource
			srcCaptions	String	Address of captions file (subtitles for the deaf). Corresponds to "src" attribute of <track> tag in HTML.

TABLE 2. *(Continued.)* **Profile stereotypes.**

Stereotype	Description	Success criteria (WCAG)	Attributes	Attributes type	Attributes description
		Alternative(Prerecorded) **1.2.5 Audio Description (Prerecorded)**	srcAudioDescription	String	Address of file with audio descriptive narration added to the soundtrack to describe visual details that cannot be understood from the soundtrack alone. For example in a movie where different graphic scenarios are presented.
			langCaptions	String	Caption language.
<<Synchronized_AudioAndVideo_Live>>	Video with synchronized audio played in real time.	**1.2.4 Captions (Live)**	Src Inherited from <<Time_based_Media>>	String	Address of the video resource
			caption_live	Boolean	Indicates if the video includes captions (True) or not (False).

TABLE 3. **Profile enumerations.**

Enumeración	Estereotipo	Descripción	Atributos	Descripción de los atributos
<<Language>>	<<Page>>	Language of the website.	es (Spanish) en (English) fr (French) de (German) it (Italian)	Languages on a website
<<HeadingTagsLevel>>	<<Heading Tags>>	Header elements.	H1 (Header level level 1, the highest) H2 (Header Level 2) H3 (Header Level 3) H4 (Header Level 4) H5 (Header Level 5) H6 (Header level 6, the least important level)	Header elements implement six levels of document heading, <h1> being the most important, and <h6> being the least important.
<<ListMenuType>>	<<Nav_Menu>>	Menu list type	orderedList disorderlyList	The menu list can be ordered or unordered
<<ShapeType>>	<<MapArea>>	Defines the shape of the active region (i.e., the clickable area) in an image map.	rest (rectangular area) circle (circular area) poly (polygonal area.)	Types of shapes to set the active region, rectangular area, circular area and polygonal area of a <<MapArea>>.

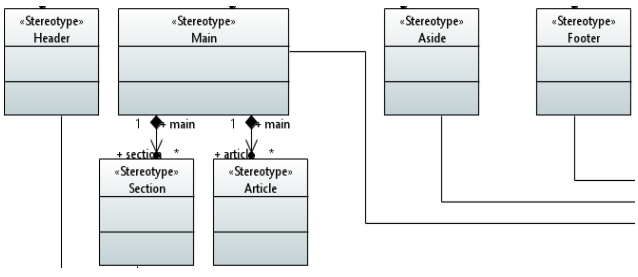


FIGURE 23. **Stereotype ≪Semantic HTML elements Stereotypes≫.**

- The <header> element represents introductory content or navigation links and typically includes header elements (<h1> - <h6>), logos or icons, and authoring information.
- The <main> element defines the primary content of a page.
- The <section> element defines a section on a page, such as a sponsor's promotion.
- The <article> element specifies independent, self-contained content.
- The <aside> element defines content for a page's sidebar.
- The <footer> element defines a footer for a document or section, which may include site contact details and copyright information.

TABLE 4. WebPageAcc profile OCL restrictions.

<<Stereotype>> [Attribute]	OCL Constraint	Description
<<Page>> [title]	ConstTitlePage self.title.size()>0	Required use of the title of the web page.
<< ButtonImage>> [textAltImageButton]	ConstTexImageButton self.textAltImageButton.size()>0	Required use of alt text on buttons containing images.
<<ControlTextSelect>> [id]	ConstControlIdLabelFor self.id=self.labelfor.for	It is mandatory that the 'id' attribute of the <<ControlTextSelect>> stereotype be equal to the 'for' attribute of the <<label>> stereotype.
<<Select>> [num_options]	ConstSelectOptions self.num_options>1	Number of control options must be greater than 1
<<ControlRadioCheckOpt>> [legend]	ConstLegendControlRadio self.legend.size()>1	Required use of the legend on the web page.
<<ControlRadioCheckOpt>> [num_options]	ConstNumOptionsControlRadio self.num_options>1	Number of control options must be greater than 1
<<Table>> [summaryTable]	ConstSummaryTable self.summaryTable.size()>0	Required use of summary in tables
<<Table>> [captionTable]	ConstCaptionTable self.captionTable.size()>0	Required use of a descriptive title for tables
<<Link>> [textLink]	ConsTextLink self.textLink.size()>0	Required use of a descriptive title for tables
<<TitleHeading>> [title]	ConstContentTitleHeading self.title.size()>0	Required use of a descriptive title of headers
<<TitleHeading>> [level]	ConstLevelTitleHeading self.level.size()>0	Required use of a level of headers
<<Nav_Menu>> [num_list_items]	ConstNavMenu self.num_list_items->notEmpty()	Required use of the num_list items attribute, to insert the number of items in the list.

<< ImageGroup >> [isDecorative, textAltImageGroup]	ConsTextAlt_ImageGroup
	if(self.isDecorative=false)
	then
	self.textAltImageGroup.size()>0
	else
	self.textAltImageGroup.size()<1 or
	self.textAltImageGroup->isEmpty()
	endif
		Required use of text alt attribute (textAltImageGroup) on non-decorative image groups.

<<ImageMap >> [textAlt]	ConstTextAlt_ImageMaps self.textAlt.size() > 0	Required use of long alt text in image maps.
<< ImageMap >> [name]	ConstNameImageMap self.name.size()>0	Required use of the 'name' attribute in image maps.
<<ImageMap>> [areas]	ConstImageMapArea self.areas -> notEmpty()	Required use of the arias attribute of type <<MapArea>>
<<MapArea>> [textAlt]	ConsTextAlt_MapArea self.textAlt.size()>0	Required use of the text Alt attribute, to insert alternative text to image map areas.
<<ImageAsLink>> [textAlt]	ConstTextAltImageAsLink self.textAlt.size()>0 self.textAlt.size()>0	Required non-empty textAlt if the image is decorative..
<<ImageNoDecorative>> [textAlt]	ConstImageIsDecorative self.textAlt.size()>0	Required non-empty textAlt if the image is decorative.
<<ImageWithGraphicsAndText >> [urlLongDescription]	ConstUrlLongDescripImagewithText self.urlLongDescription.size()>0	Images that require a description of more than 150 characters for better understanding, such as diagrams, infograms, maps, statistical tables, tables and images that combine text with graphics, must be associated with text with a long description.
<<ImageWithGraphicsAndText >> [textAlt]	ConsTextAltImageWithGraphicsAndText self.textAlt.size()>0	Required use of alt text on images that combine text and graphics.

TABLE 4. *(Continued.)* **WebPageAcc profile OCL restrictions.**

<<Stereotype>> [Attribute]	OCL Constraint	Description
<<OnlyVideo_ Prerecorded >> [urlTextualTranscription]	ConstTextualTranscription_OnlyVideo_Pre self.urlTextualTranscription.size()>0	Required textual description on video-only resources.
<<OnlyAudio_ Prerecorded >> [urlTextualTranscription]	ConstUrlTextualTranscription_OnlyAudio self.urlTextualTranscription.size()>0	Required on audio-only resources.
<<BackgroundNotChangeVideo_Prerecorded>> [textualDescrip]	ConstTextDescrip_BackgNotChangeVideo self.textualDescrip.size()>0	Required brief textual description in video assets with an unchanged background, such as speeches in which only one person speaks.
<<BackgroundNotChangeVideo_Prerecorded>> [srcCaptions]	ConstSrcCaptions_BackgroundNotChangeVideo self.srcCaptions.size()>0	Required use of subtitles on video resources with unchanged background. Unlike subtitles, captions, in addition to transcribing the audio, present information about sound effects or other relevant audio information.
<<Synchronized_AudioAndVideo_Prerecorded >> [srcCaptions]	ConstSrcCaptions_Synchronized_AudioAndVideo srcCaptions.size()>0	Required use of subtitles on pre-recorded video with synchronized audio.
<<Synchronized_AudioAndVideo_Prerecorded >> [srcAudioDescriptions]	ConstSrcAudioDescriptions_Synchonized_Audio self.srcAudioDescriptions.size()>0	Required use of audio description that must be added to the sound track to describe visual details that cannot be understood from the sound track alone.
<<EmbeddedYoutubeVideo>> [subtitle] [urlTextualTranscription]	ConstCaptionsAndTranscrip_EmbYoutubeVideo (self.subtitle = true) and (self.urlTextualTranscription.size()>0)	Required use of subtitles and URL of the textual transcript in YouTube videos.
<<Synchronized_AudioAndVideo_Live>> [captions_live]	ConstCaptionAudioLive self.captions_live=true	Required to indicate the language of the subtitles in the live audio.

FIGURE 24. UN web page for case study (https://www.un.org/en/).

1) OCL RESTRICTIONS DEFINED IN THE PROFILE

OCL is a formal language used to specify constraints. This language is based on first-order logic, which allows eliminating ambiguities and thus describing the rules that must accompany a model.

In the present work, a set of OCL rules or restrictions described in Table 4 has been proposed to create an accessible web page model. All rules or restrictions are linked to the compliance criteria of the WCAG guidelines "1.1 Text Alternatives, 1.2 Time-based Media, 2.4 Navigable, 3.1 Readable and 4.1 Compatible", which are directly related to the images, videos, audios, navigation menu, links, definition lists, headings, paragraphs, content sections, forms and tables. This in order to provide the end user with the ease of seeing, listening and interacting with the resources uploaded to the website, without having major complications, through the use of textual alternatives for non-textual content, or the use of captions (subtitles), verbatim transcripts or audio descriptions for time-dependent media such as audio or video. Also to provide a suitable structure for links, menus, forms and tables.

The description of the OCL rules for the profile stereotypes can be seen in Table 4, where the first column indicates the stereotype for which the rule is defined. The second column shows the OCL constraint defined using the attributes of the stereotype. Finally, the third column contains the description of each restriction.

Most of the restrictions exposed in Table 4 focus on certain properties that cannot be empty, such as the value of the textAlt property, when dealing with non-decoration images. This is so that visually impaired people can understand the meaning of the non-decorative images included in the web page, since the screen reader will interpret the content set in textAlt, the same one that refers to the 'alt' attribute of the < img > tag in HTML.

C. APPLICATION OF WEBPAGEACC PROFILE TO MODEL A REAL WEB PAGE

It is crucial to emphasize that one of the main objectives of UML profiles is to integrate new functionalities into UML models effectively. To understand how a profile works in practice, it is imperative to illustrate its characteristics through its application in a case study.

In this regard, specific sections of various web pages of the United Nations (UN), including its main page, will be used as illustrative examples. The selected components for modeling the main page include the logo, the main banner, and the multimedia content section containing videos and podcasts (audio), as shows in Figure 24. Figure 25 displays a sample web page including a form, while in Figure 26, a page featuring a table as a component is presented.

In the main page of the first illustrated example in Figure 24, which serves an informative purpose, two key images stand out: the logo and the main banner. In the multimedia section, there is a podcast analyzing the UN's response to the Beirut port explosions. Additionally, there are video elements with synchronized audio, one illustrating a world without plants, another showcasing ten inspiring activities, and a third conveying a message against racism. In the second example shown in Figure 25, a contact form is considered where users can input their data and corresponding message. Finally, in the third example of Figure 26, a table with headers at the top is presented.

Figure 27 presents the modeling of the UN's main page through a class diagram illustrating various components. In particular, the main component is IndexPage, categorized as ≪Page≫. Associated with IndexPage are images represented by LogoUN_img and Elephants_img, both categorized as ≪ImageNoDecorative≫. Additionally, a video without audio, RalphBuncheJustice_onlyvideo, is modeled as ≪OnlyVideo_Prerecorded≫. Moreover, two videos with synchronized audio and video, ResponsibleAgricultural_video and PreventDrownig_video, are categorized as ≪Synchronized_AudioAndVideo_Prerecorded≫. Finally, there is SpaceDiplomat_podcast, an audio-only component categorized as ≪OnlyAudio_Prerecorded≫.

In Figure 28, the model includes classes related to form elements. This comprises FormContactPage, categorized as ≪Form≫, and ButtonFormContact, of type ≪Button≫. Each control element, such as YourNameControl, YourEmailControl, SubjectControl, TextCaptcha, and MessageControl, categorized respectively as ≪TextField≫ and ≪TextArea≫, is accompanied by its corresponding labels, LabelName, LabelYourEmail, LabelSubject, LabelCaptcha, LabelMessage, categorized as ≪Label≫.

Regarding the table modeling presented in Figure 29, a class called TableEditorialUpdates has been created. In this class, all required fields have been included, such as title, summary, number of rows and columns, as well as the definition of headers at the top.

D. VALIDATION OF THE MODEL EXECUTING OCL CONSTRAINTS INCLUDED IN THE PROFILE

It is important to validate the successful compliance with the OCL constraints established in the profile, as described in Table 4. For this reason, the United Nations website model has been validated with various OCL rules. Figure 30 displays some of the validation issues related to the use of accessible images and videos, which were modeled from sections of the UN page, described in Figure 27. If violations of the OCL constraints are found during the model validation, they must be corrected to ensure the integrity and consistency of the model. The model validation process may require multiple iterations to identify and correct all violations of the OCL constraints. It is necessary to continue iterating and refining the model until all constraints defined in the profile are met, in order to comply with the WCAG 2.2 accessibility criteria defined in the profile.

Firstly, the validation errors found are related to the lack of a required value for the 'textAlt' field in the ≪ImageNoDecorative≫ classes. In this case, non-decorative images must have a brief alternative text associated so that people with vision impairments can access these resources using a screen reader. This applies to the LogoUN_img and Elephants_img classes. Additionally, if the image has an associated link that directs to another page, the value of the imageLink field cannot be empty. Figure 31 illustrates how to correct validation errors, such as inserting the text 'Index Page UN' in the 'textAlt' field of the LogoUN_img class. Similarly, since the value of the imageLink field cannot be empty, it was filled with the text 'https://www.un.org/en/elephant'.

Another issue detected in the model is the lack of required information for the urlTextualTranscription field of the RalphBuncheJustice_onlyvideo component ≪OnlyVideo_Precorded≫. This field refers to the mandatory use of textual transcription for video-only files, where it is necessary to describe the content transmitted through the video in textual format. This allows people with vision impairments to access such resources without difficulty. In this case, since the value of the urlTextualTranscription field cannot be empty, it was filled with the text 'https://www.un.org/en/transcription/ralphbunchejustice_video.vtt'.

The mandatory use of text in the 'srcAudioDescription' field is another problem detected, associated with the PreventDrownig_video and ResponsibleAgricultural_video classes of type ≪Synchronized_AudioAndVideo_Prerecorded≫, which refers to the use of an oral narration of what happens in the synchronized video, where characters and sounds cannot be identified through subtitles. It is essential to use oral narration when the information projected in the video does not completely match the audio. Undoubtedly, this helps people with visual impairments fully understand the content transmitted in the synchronized video.

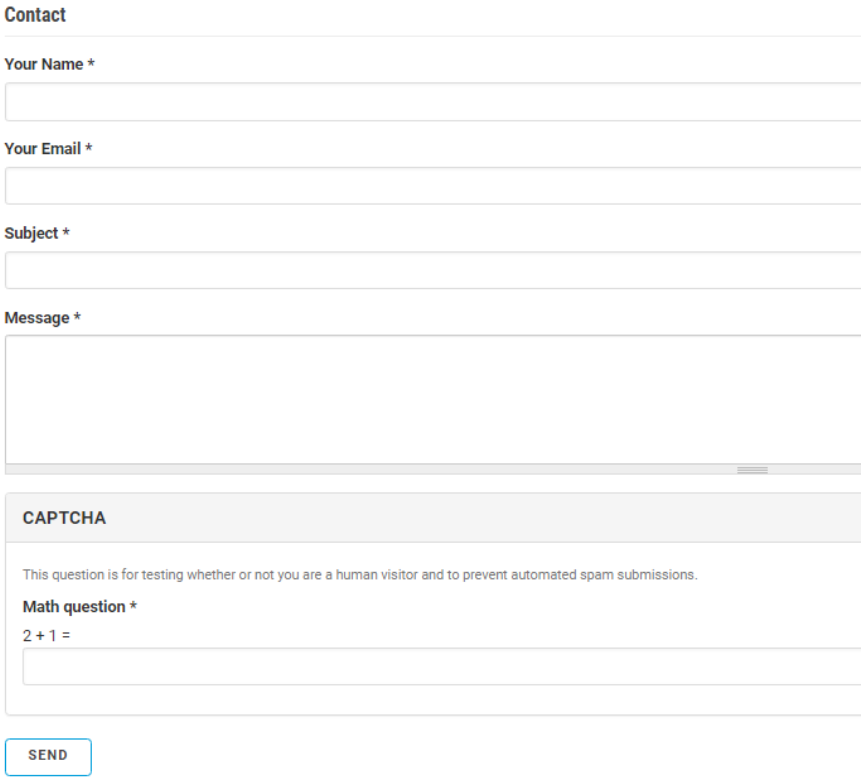


FIGURE 25. UN web form for case study (https://www.un.org/en/library/contact-us).

Date	Page	Change
8 May	Style/Spelling	The entries digitalize (adapt a system or process) and digitize (convert material into a digital format) were added
8 May	Numbers, dates and time	The example using an en dash to link to numbers was updated as follows: An en dash (–): A substantial increase in production (12–14 per cent) is expected. For more examples of the use of the en dash, see ***Punctuation***.
8 May	Punctuation	The examples under "En dash" were updated: pp. 17–19 (**but** pp. 17 and 18) paras. 19–21 (**but** paras. 19 and 20) para. 73 (b)–(e) (**but** para. 73 (b) and (c))
3 May	Style/Spelling	The entry "liveable" was added
20 April	Style/Spelling	The entry "cost-sharing" was added
23 March	Style/Abbreviations	The entry "KC (King's Counsel)" was added
16 March	Footnotes	The example for "Two consecutive page/paragraph numbers" was updated to read "pp./paras. 17 **and** 18"
16 March	Mastheads and cover pages	Under "Corner notation: agenda", the following was added: A footnote reference is provided for the preliminary list and the provisional agenda; **there is no footnote for the annotated preliminary list** or the adopted agenda
16 March	Punctuation	The entry for Apostrophe was edited to read: An apostrophe (**'s** or **s'**) is not used with an abbreviation or acronym, the name of a country **or the name of a body, for example, an intergovernmental organization, a court or a government ministry**.
16 March	Capitalization	"seat of government" was added as an example under Government
16 March	Style/Spelling	The entry "gender-responsive" was added
2 March	Capitalization	Under delegation, the following was added: *but* Delegation if used as equivalent of Permanent Mission: the Delegation of the European Union
17 February	Footnotes	An example was added for citing specific subparagraphs within a paragraph
17 February	Footnotes	An example of a shortened legal reference was added under repeated references
17 February	Style/Spelling	The entry "web map" was added
30 January	Punctuation	Examples were added under "Colon".
5 January	Capitalization	The section on proper names transliterated from Arabic was updated to read: The same rule applies to bin, **bint** and ibn; for example Osama bin Laden, Mr. Bin Hamdan, **Khawlah bint al-Azwar, Ms. Bint al-Azwar**, Abu al-Walid ibn Rushd (Averroës), Ibn Sina (Avicenna).
1 December	Style/Spelling	The entries "capacity development (noun)" and "capacity-development (adj.)" were added

FIGURE 26. UN web table for case study (https://www.un.org/dgacm/en/content/editorial-manual/updates).

In this case, since the value of the srcAudioDescription field cannot be empty, it was filled with the text 'https://www.un.org/en/audio/audioPreventDrownig.mp3' in the PreventDrownig_video class, and the text 'https://www.

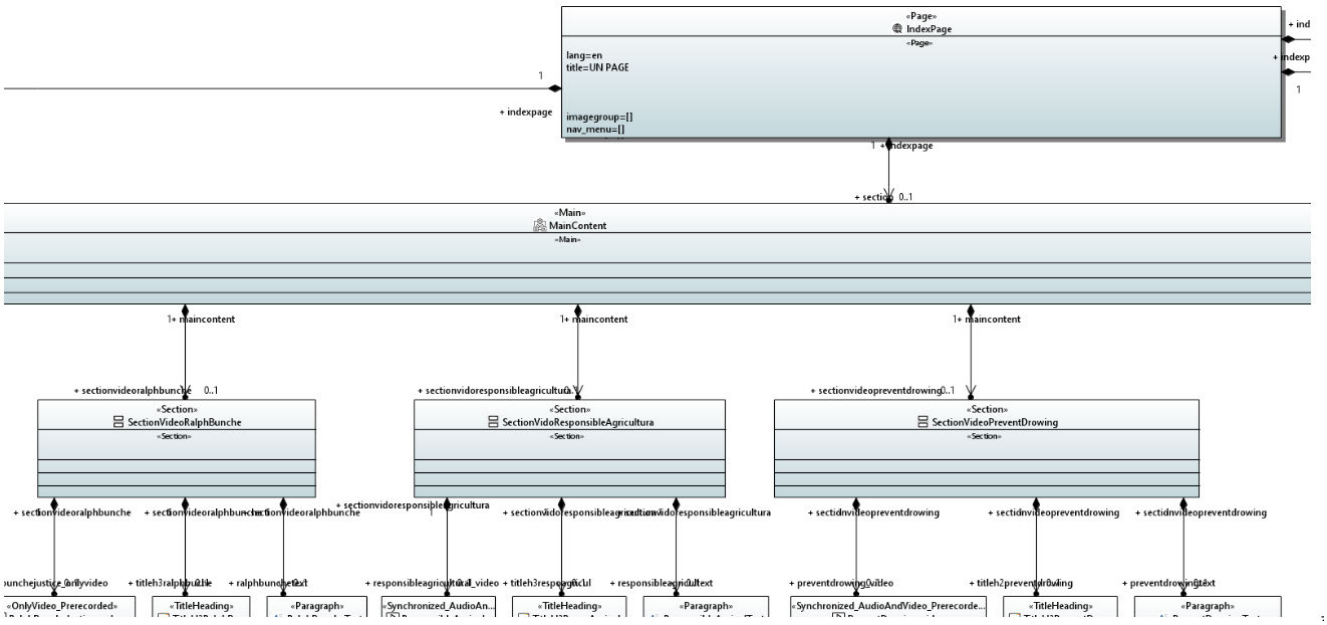


FIGURE 27. UN Website Class Diagram is available at https://drive.google.com/file/d/1-V0CLtrJiBfrmeQMBgyrXp29Y_UQ-rYR/view?usp=sharing.

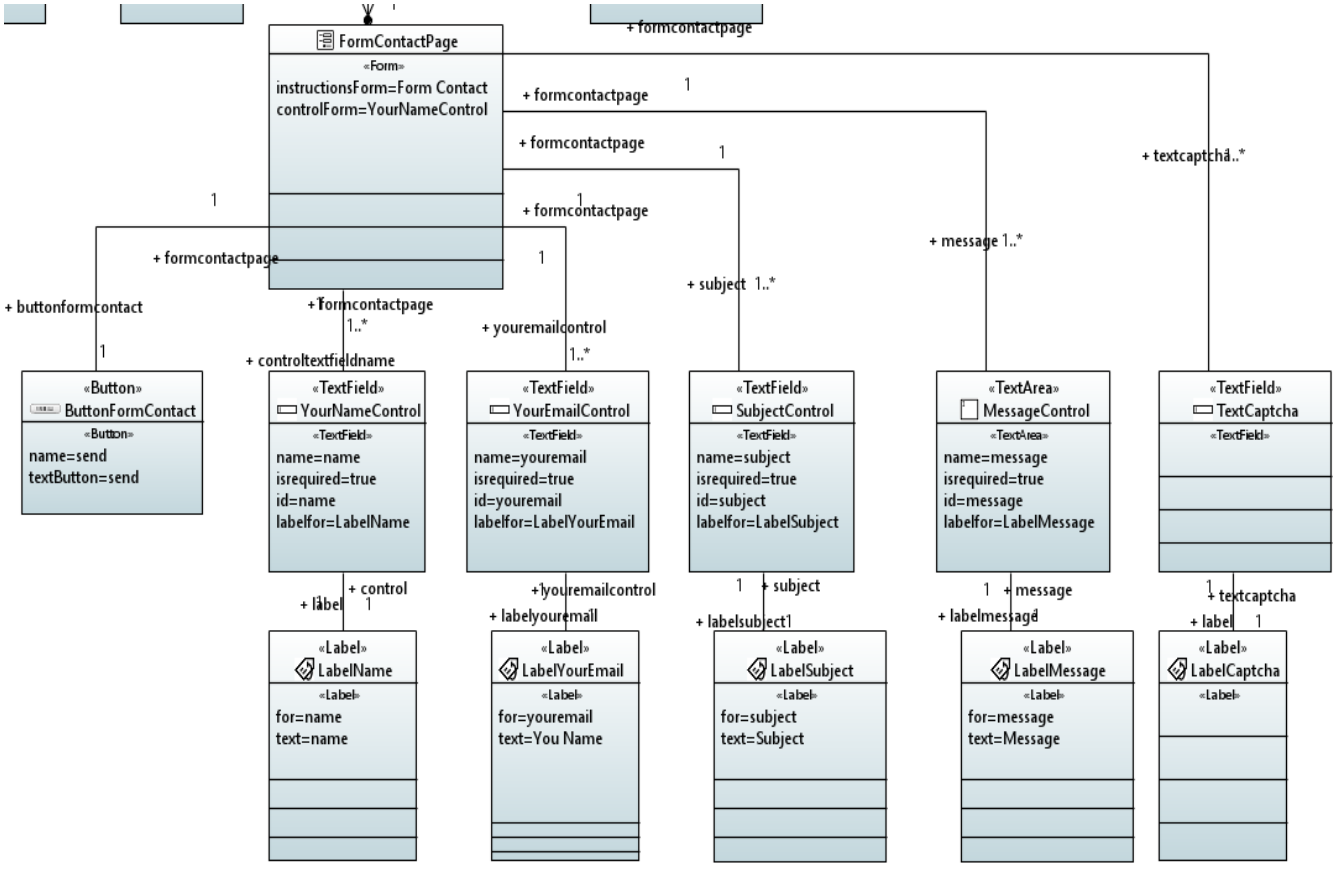


FIGURE 28. UN Web form class diagram.

un.org/en/audio/ResponsibleAgricultural_audio.mp3' in the ResponsibleAgricultural_video class.

Additionally, an error associated with the mandatory use of text in the srcCaptions (captions direction) field in

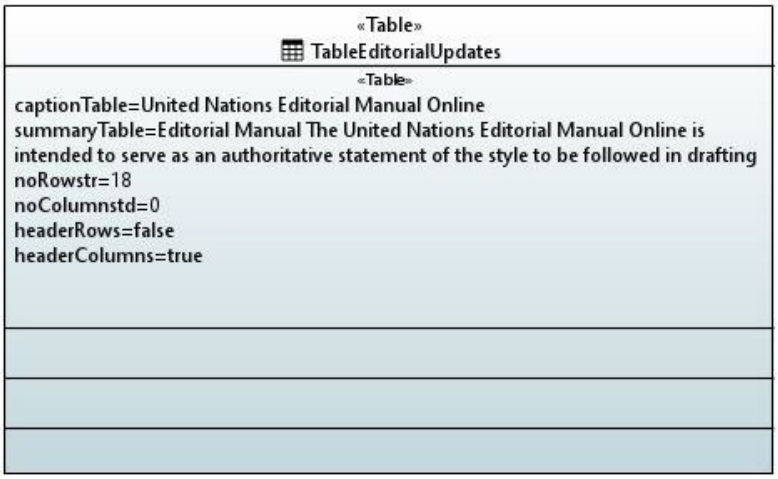


FIGURE 29. UN Web table class diagram.

synchronized audio videos was found. This applies to the PreventDrownig_video and ResponsibleAgricultural_video classes of type ≪Synchronized_AudioAndVideo_Prerecorded≫.

Captions, known as subtitles for the deaf a hearing impaired users, provide dialogue text by characters and include descriptions of important sounds. It is important to note that captions should not be confused with subtitles, as subtitles only provide text for dialogue and do not include descriptions of important sounds. In this case, since the value of the srcCaptions field cannot be empty, it was filled with the text 'captions_videopreventdrownig'.

Another problem detected is the mandatory use of the 'srcAudioDescription' attribute of the Responsible Agricultural_video component ≪Synchronized_AudioAndVideo_Prerecorded≫, which refers to the use of oral narration of what happens in the synchronized video, where characters and sounds cannot be identified through subtitles. It is important to use oral narration when the information projected in the video does not completely match the audio. This facilitates people with visual impairments to fully understand the content transmitted in the synchronized video. In this case, since the value of the srcAudioDescription field cannot be empty, it was filled with the text 'https://www.un.org/en/audio/ResponsibleAgricultural_audio.mp3'.

The lack of required information for the urlTextualTranscription attribute of the SpaceDiplomat_podcast component ≪OnlyAudio_Precorded≫ is another problem found in the model. This attribute refers to the mandatory use of textual transcription for audio-only files, where it is necessary to describe the content transmitted through audio in textual format. In this case, since the value of the urlTextualTranscription field cannot be empty, it was filled with the text 'https://www.un.org/en/transcription/SpaceDiplomat_podcast.vtt'.

Regarding the Form Model presented in Figure 28, an error related to the lack of association between labels and form elements was identified. Therefore, it is necessary to link each form control with its respective label, ensuring that the values of the "for" attribute of the label and the "id" of the control match. The use of labels allows the user to click on the label itself to set focus on the form element. Additionally, it was detected that the lack of instructions in the form is also a problem. Therefore, it is important to provide clear and concise instructions about the information to be transmitted by properly filling the 'instructionsForm' property.

In the table model presented in Figure 29, filling in the 'summaryTable' and 'captionTable' properties is mandatory. In this context, the summary is important because it provides a detailed description of the table that complements its title. The description should include an explanation of the content and structure of the table, such as the purpose of the table, the number of rows and columns it contains, as well as a description of the headers. Additionally, the table must have a title that provides a brief and clear description of its content. It is essential to have a correct table structure, as this facilitates people using non-visual browsers to understand the information quickly and easily. Therefore, it is necessary to properly label the tables to define their title, include a summary that briefly describes the table's content, define column and row headers, and use special tags and attributes to associate header cells and data cells for more complex headers.

Once the required information in each of the necessary attributes has been completed, as shown in the example in Figure 31, each model can be successfully validated, as shown in Figure 32. Therefore, it can be stated that the defined models comply with all OCL rules or constraints established in the profile.

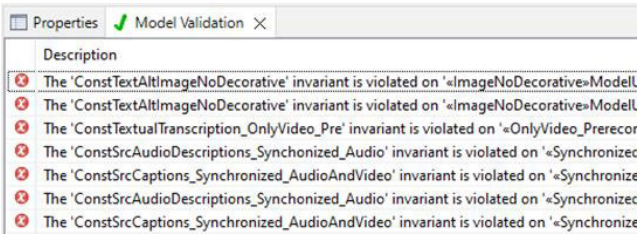


FIGURE 30. Problems detected in the validation is available at https://drive.google.com/file/d/182Mn5N5OqylmUw24XQAuWou0I2rZxOUp/view?usp=sharing.

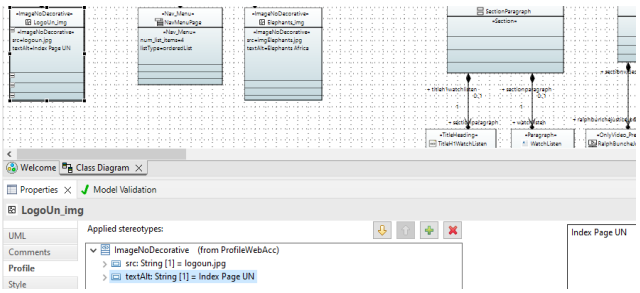


FIGURE 31. Correction of Alternative Text Problem (textAlt) in LogoUN_img ≪ImageNoDecorative≫ https://drive.google.com/file/d/1DWCHpXtmCEzH-IWXg7A02YikvCHYcQ1i/view?usp=sharing.

E. EXPERIMENT ON THE DIFFICULTY OF USING THE PROPOSED PROFILE

To test possible difficulty in using the profile by website developers, an experiment was conducted for use by several web developers. To obtain the participation of developers with knowledge of model-driven development, at least using UML, and knowledge of web accessibility, former students of the master's degree in web engineering at the University of Alcalá who were working as web developers were contacted, since In this master's degree, the indicated knowledge is taught. 12 developers were able to participate in the experiment.

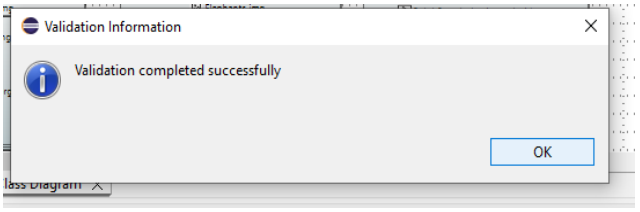


FIGURE 32. Validating OCL rules in the model.

To check their opinion on accessibility, some open questions were asked, such as the following, obtaining some interesting answers indicated below:

1. What is your opinion or perspective on accessible software design and development?
   a. The design and development of accessible software are essential to promoting inclusion, ensuring that all people, regardless of their abilities, can use and benefit from technology.
   b. It is necessary, but it is not always taken into account and is not applied in a general way, there is usually the idea that making accessible software implies making economic efforts to cover a small percentage of the population, because it is understood that accessibility is only for people who They have some type of disability, and that is not the case, it benefits all possible users.
   c. It is something that should be mandatory.
   d. Just as there is a TDD or a BDD, an ADD (Accessibility) should be imposed. Driven Development).
   e. If you want to make projects accessible after the fact, it will be much more costly in time, effort and money, and the result may not be as good.
2. What are the challenges you face in software projects when addressing accessibility issues?
   a. Accessibility is not considered from the beginning and the software must be adapted later.
   b. Lack of adequate training in accessibility, which requires constant up-to-date learning on best practices and regulations.
   c. Lack of time and resources to carry out accessibility tests.
3. What accessibility issues have you encountered when developing software?
   a. Limitations for keyboard navigation.
   b. Lack of descriptions of visual elements.
   c. Insufficient contrast in the interface.
   d. Videos without subtitles.
   e. Lack of clear labels on forms.
   f. Color perception problems.

Although all participants had knowledge about accessibility, not all of them applied it in the development of websites. After asking them, it was found that 58% had applied web accessibility standards, such as WCAG, while 42% had not applied them.

The experiment consisted of including in the model of a website based on the UML profile created, an element with the stereotype ≪ Image ≻ and another element with the stereotype ≪Page≫. Next, you had to give values to the properties created in the profile for ≪ ImageNoDecorative≫, at least to the 'textAlt' property, designed to guarantee that each image on a web page has alternative text from the design of the web page. Next, we had to give values to the properties of the ≪Page≫ element, at least the title property, to ensure in the model that the web page has ais title.

The third task was to validate the created model, to verify that the restrictions established in the profile with OCL are met. Th task was done simply by activating the Validate function offered by Eclipse Papyrus. Each participant was also asked to try making changes to the property values of the ≪ImageNoDecorative≫ and ≪Page≫ elements to force validation errors, to verify that if the property value is removed alternative text in the ≪ImageNoDecorative≫

element or that of the title property in the ≪Page≫ element, errors appear in the validation.

Once the experiment was carried out, each participant had to answer the three questions indicated in Table 5. 67% considered that it was not necessary to have high knowledge of accessibility standards such as WCAG to use the proposed profile, and the same percentage thought that the support of a technician was not necessary to use the profile. Furthermore, all participants believed that any developer could quickly learn to use the proposed profile.

TABLE 5. Relevant survey questions.

QUESTION	YES	NO
Do you think it is necessary to have high knowledge of accessibility standards, such as WCAG, to use the proposed profile?	33%	67%
Do you think technical support is necessary to use the profile?	33%	67%
Do you think developers could quickly learn to use the proposed profile?	100%	0%

To assess the difficulty of using the profile, four questions were also asked about the difficulty encountered when including in the model, a web page with a title and an image with alternative text. A scale of five possible difficulty levels was offered: 1 (Very easy), 2 (Easy), 3 (Neutral), 4 (Difficult), 5 (Very difficult). Table 6 and Figure 33 show the results.

TABLE 6. Questions about the difficulty of using the proposed profile.

QUESTION	Very easy (1)	Easy (2)	Neutral (3)	Difficult (4)	Very difficult (5)
Difficulty adding an image with alternative text to a UML model of a web page	33%	25%	33%	9%	0%
Difficulty adding a page title to a UML model of a web page	33%	17%	33%	17%	0%
Difficulty validating a UML model based on the profile created	50%	25%	8%	17%	0%
Difficulty in using the profile, from application to validation	17%	33%	42%	8%	0%

It can be seen that more than half of the participants consider it easy or very easy to use the proposed profile.

Descriptive statistics are presented in Table 7 and box plots are presented in Figure 34. We observe that the mean level of difficulty in all cases was Easy.

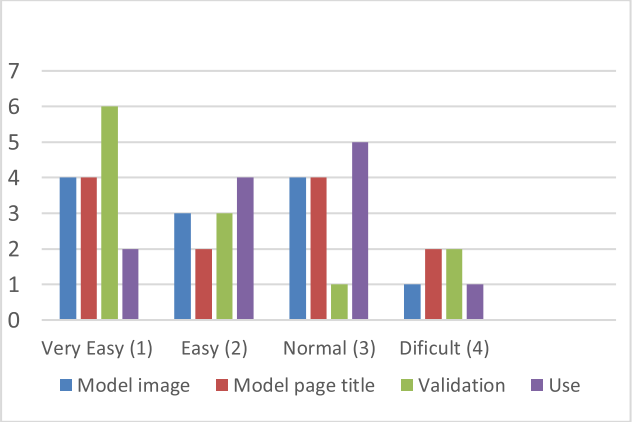


FIGURE 33. Bar chart on the difficulty of using the profile.

TABLE 7. Mean and standard deviation of the variables that measure the difficulty of using the profile.

DIFFICULTY	Mean (1-5)	S.D.
Difficulty adding an image with alternative text to a UML model of a web page	2.17 Easy	1.03
Difficulty adding a page title to a UML model of a web page	2.33 Easy	1.15
Difficulty validating a UML model based on the profile created	1.92 Easy	1.16
Difficulty in using the profile, from application to validation	2.42 Easy	0.9

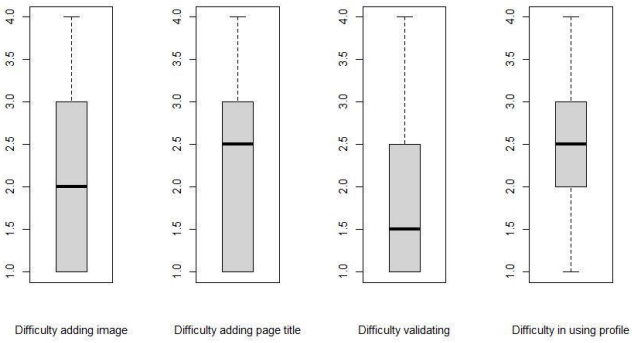


FIGURE 34. Box plots of the variables that measure the difficulty of using the profile.

We used Mann-Whitney non-parametric tests to check differences between the developers commonly using accessibility standards and those who doesn't use them. We chose non-parametric test because we cannot make any assumption on the distribution of answers given the number of answers. The only assumption is the homogeneity of variances which was checked using Levene's test. Results of the Mann-Whitney tests show that there is no statistical difference for any of the items evaluated (Table 8) suggesting that there is no difference of opinions on the difficulty for using the profile for participants.

TABLE 8. Results of Mann-Whitney tests checking differences between developers using standards and those who doesn't use them.

Parameter to contrast	p-value
Difficulty adding an image with alternative text to a UML model of a web page	0.75
Difficulty adding a page title to a UML model of a web page	0.94
Difficulty validating a UML model based on the profile created	0.57
Difficulty in using the profile, from application to validation.	0.57

Finally, we ran an item analysis to assess the internal reliability of the items that used a Likert scale (questions in Table 4). It returned a Cronbach's Alpha of 0.904 which is substantially higher than the commonly accepted threshold of 0.7 suggesting that the questionnaire has a high level of internal consistency. We also contrasted that removing any item would decrease the overall internal consistency (range 0.842-0.892 when items are omitted).

As a conclusion of the experiment carried out, it was shown that the difficulty of using the proposed profile is low and that any developer can use it, both those who apply accessibility standards and those who do not usually apply them.

V. DISCUSSION

The main contribution of this work is the formal definition of a UML profile that allows establishing constraints regarding the accessibility features that a website must comply with. In this initial version of the profile, several success criteria included in the guidelines 1.1 Text Alternatives, 1.2 Time-Based Media, 2.4 Navigable, 3.1 Readable and 4.1 Compatible from the WCAG 2.2 recommendations have been considered. In this first version, a total of 12 accessibility criteria have been considered. This profile was defined through its stereotypes, constraints, and tagged values, a mechanism provided by UML for extensions of its metamodel. Different OCL constraints were included to comply with the accessibility guidelines defined in the profile.

The profile presented in this article was developed with the aim of incorporating accessibility features from the early stages of the software development process, where developers do not need a high level of knowledge of accessibility standards, as evidenced by the satisfaction survey applied to a group of developers. Among the various advantages of this proposal is that it is easy to extend and adapt to changes in requirements because the business logic is captured in a model that is quicker to understand than thousands of lines of code. In this sense, one of the greatest benefits of our approach is the automatic code generation, as while the initial purpose of the profile is to generate accessible web pages, it could also be used to model the interface of the mobile application, as the profile includes components such as images and videos that could be added to a mobile application.

Our approach focuses on addressing the following accessibility issues:

Pages without title and without language. The HTML <title> tag allows you to define the title to identify the topic or main content of the web page. The ''lang'' attribute defines the primary language of the page allowing screen readers to recognize the declared language and make a correct pronunciation of the synthesized voice. When headings are properly defined, screen reader users can more easily perceive the hierarchy of content and use keyboard shortcuts to list and navigate through the different headings.

Page without sections. Defining the different sections of the page with semantic HTML elements allows accessible and orderly navigation for users. For example: header, nav, section, aside, footer, main.

Images without alternative text, as they may be inaccessible to people with some type of visual impairment. In this case, it is important to incorporate alternative text in the images.

Complex images without descriptive text, these types of images that contain text such as infographics or statistical graphics need an extensive description so that people who have some type of visual disability can interpret them. Additionally, a link (an anchor) to the description area must be included.

Videos without subtitles may be inaccessible to the hearing and visually impaired. Therefore, it is necessary to incorporate descriptions, subtitles and transcriptions in the audios and videos.

As image maps, the active area must have alt text describing the hyperlink it is linked to.

Lack of headers and subtitles on the page, since these allow the headers and subtitles to be separated from the rest of the content on the page. They give the page content an outline and structure, making it easier to consume and understand. HTML headers should maintain a logical descending numerical order from 1 to 6, with 1 being the most important and 6 being the least important. When HTML headers are not in order, the outline of the content is broken. The <H> tags allow you to define the headings and main topics of the page, assigning <H1> for the main topic and the tags <H2> to <H6>, for subtopics.

Links without texts. Each link must be accompanied by descriptive text that tells the user where it is going, providing content that makes sense out of context.

Lack of navigation element on the page. The navigation elements that a web page has must be clearly identified by the <nav> tag, so that supporting technologies such as screen readers can detect it and offer it to the user when they need it. Additionally, a screen reader can have a command (keyboard shortcut) to directly access the navigation elements of a page, just as it has a command to display the list of links or the list of headers on a page.

The forms are not clear and intuitive. The tab order between elements (or what is the same, access to them) should be easy to understand. Many people use a keyboard to navigate, so we must ensure that the form can be filled out using the keyboard. For example: Associate labels of the form with

inputs, with the use of the for attribute in the <label> and its corresponding id in the associated control. Additionally, it is important to provide a "submit" button to initiate a context switch. Likewise, indicate whether a field is mandatory in the LABEL associated with the field, for example through the text "(mandatory)" or through an asterisk explaining its meaning at the beginning of the form. Finally, when the form is short or has many fields of the same type, include instruction text at the beginning of the form instead of repeating the information in each field.

A table is very difficult to understand if you can only see one isolated piece of data at a time. This problem is suffered by users who use non-visual browsers, since they have to go through the tables in a linear manner but the nature of a table is two-dimensional, so they lose the global view of the table and lose the references that allow them to interpret the content of each cell in the table. To avoid this problem, the tables must be labeled correctly, first defining their title with the <caption> tag, and it is also important to include a summary that briefly describes the content of the table, by using the <summary> tag, Likewise, define the headings of the columns and rows.

There are different proposals suggesting the inclusion of accessibility in model-driven software development (MDD), for various types of applications such as web, mobile, and desktop, and for different fields of knowledge such as content management systems (CMS), e-learning, multimedia resources, and general applications. However, several of these approaches propose specific metamodels, which make it difficult to be implemented in different areas, for example, CMS, chat on mobile devices, multimedia players, and website menus. In contrast, our approach proposes a generic metamodel that provides ease of adaptation to different fields of study and different types of applications such as web and mobile.

Comparing our approach with similar works, we can affirm that we offer a solution based on the use of a UML Profile. This profile allows modeling specific accessibility requirements as established in WCAG 2.2, using the OCL language to establish constraints related to accessibility guidelines defined by the standard. To illustrate the application of the profile, we present a case study that demonstrates the simplicity and potential of its use. Additionally, with the survey applied to a group of developers, it was evident that the level of difficulty in using the profile is low, so it could be applied to a software project with ease.

In our proposal, accessibility features can be incorporated from the analysis and design stage of the software. This is achieved by building a model that can be easily extended if it is necessary to add other features, depending on the type of software, allowing us to quickly locate the different functionalities offered by the software. This flexibility allows us to adapt quickly to technological changes, as we can generate source code in different programming languages through model-to-code transformation. Additionally, it facilitates software maintenance in case of changes in requirements, thanks to the management of platform-independent models.

Our approach is promising, as from the early stages of the software development life cycle, it is recommended to incorporate accessibility features in the different multimedia elements of a website, such as inserting images with text, such as infographics or statistical graphs. Unlike automatic validation tools, which report possible accessibility errors at the end of software construction and require review by an expert to correct them. For example, with an automatic evaluation tool, it would not be possible to detect all accessibility problems of complex images, such as statistical graphs or infographics, as they not only need alternative text but also a more detailed description, so that people with visual or learning difficulties can understand their content. Additionally, the use of the profile could provide a documented structure for accessibility considerations in web design; therefore, development team members can have a clear view of the accessible elements by defining stereotypes and their relationships on the page.

Another highlight of our proposal is that for its implementation, we have used freely accessible tools such as Eclipse Papyrus and the UML modeling language, which implies low or no costs compared to other similar solutions. These tools, in addition to being free and having extensive support documentation, provide us with a complete working environment to develop projects with a model-driven approach.

To evaluate the effectiveness of the proposal, the UML model of several UN web pages has been developed. In this process, all OCL rules defined in the profile have been validated with the aim of ensuring compliance with accessibility requirements for various elements of the modeled website, such as decorative images, pre-recorded audio and video files, forms, and tables. Model validation has allowed identifying that it is much more convenient to integrate accessibility from the early stages of the project. This avoids the need to go back and redo the work later. By applying the OCL rules, we can detect possible accessibility errors present on the website during the site modeling stage, providing us with the opportunity to correct them in a timely manner. Subsequently, we can generate accessible HTML code for the different multimedia elements inserted on the website.

The experiment conducted by a group of developers demonstrated that the difficulty of using the proposed profile is low and that any developer can use it, both those who apply accessibility standards and those who do not usually apply them. With the tests carried out, it was evident that our proposal offers a clear structure to represent accessible elements, which facilitates understanding and communication within the work team, establishing a solid foundation for accessibility implementation.

Despite the numerous advantages that Model-Driven Development (MDD) offers, its initial implementation may pose significant challenges for software development companies. These challenges include the appropriate selection of a model-driven development method, the choice of appropriate

techniques and tools, as well as managing accessibility issues during the process. Therefore, it is crucial to train staff and adjust roles within the development team, which may lead to an increase in the organization's operating costs.

Among the most important potential challenges is that web accessibility often involves specific details that could be complex to accurately model using UML, especially in web applications that make use of advanced technologies such as AJAX and JavaScript. Additionally, determining the consistency of content provided in model fields can be complex. For example, ensure that titles correspond appropriately to the content of the page or that links are not generic or ambiguous, but rather offer clear information about their destination. Therefore, it is essential to provide training on the UML profile and its components, highlighting the importance of accessibility and its integration into the development process.

For software development companies, keeping the model up to date as the website evolves can present an additional challenge. Therefore, it is recommended to establish strong documentation practices and review and update the model regularly throughout the development cycle. Additionally, it is crucial to adapt the model to changes in accessibility standards, as these may evolve over time, requiring adjustments to the UML profile. Therefore, it is essential to be aware of updates to accessibility guidelines and update the UML profile accordingly.

Proactively addressing these challenges and adjusting the approach based on the specific needs of the team and project is essential. Experimentation and continuous feedback are essential to improve and optimize the application of the UML profile in the development of accessible web pages.

VI. CONCLUSION AND FUTURE WORK

A. CONCLUSION

The development of the model brings significant benefits to creating an accessible website. From enabling domain-level reuse for different types of software platforms to enhancing the model's quality through OCL constraint validation. Furthermore, it reduces costs by employing an automated code generation process and enhances the longevity of software solutions. In this manner, models can become assets rather than expenses, as they can be reusable across various software projects, including web, mobile, and desktop. In this regard, the proposed approach offers significant advancements for effective control over the development of accessible web pages, ranging from productivity gains, increased reliability, and better ways to handle changing requirements, to the ease of transforming models into code. Moreover, it's more efficient and effective to incorporate accessibility from the beginning of projects, as this avoids the need to backtrack and redo work later on.

The proposed UML profile, WebPageAcc, outlined in this document, is the result of a comprehensive study on the development process of accessible web pages using a model-driven development approach. This approach was applied in a practical case involving various UN web pages, where sections containing elements such as images, videos, audios, navigation menu, link, paragraph, headings, forms, and tables were considered. This allowed for the clear and straightforward modeling of the characteristics of an accessible web page through a class diagram, which is challenging to achieve with traditional UML elements. The aim was to analyze the feasibility of using UML profiles for different success criteria proposed by WCAG when designing an accessible web page.

The proposal's construction involved a four-stage process. Firstly, the analysis stage involved examining the accessibility criteria of WCAG 2.2, analyzing the techniques of each success criterion proposed by WCAG 2.2, and identifying the elements that would form part of the profile, such as stereotypes, attributes, enumerations, and constraints necessary to accurately represent an accessible web page in HTML language. Secondly, the development of the WebPageAcc profile utilized the Eclipse Papyrus tool to design the profile, considering all elements identified in the first stage. The third stage involved the application of the WebPageAcc profile in a UML model through a class diagram to apply the profile in a case study. Finally, the model validation stage involved OCL validation of the class diagram for the multimedia elements of the UN web page to determine if the model for accessible web pages complies with the accessibility characteristics defined in the profile.

To assess the ease of use of the profile, a practical experiment was conducted with a group of developers, demonstrating that the profile is easy to use, even for those without experience in accessibility topics. However, it is essential to provide prior training on the UML profile and its components, emphasizing the importance of accessibility and its integration into the development process. It is important to note that the level of coherence of the content proposed in the model cannot be validated in the profile.

It is very useful for web developers to have instructions, rules, elements, and specific notations for the domain to be modeled, in this case, for accessible web pages in HTML language. Therefore, within the proposed approach's development process, a prior analysis of the success criteria of WCAG Guidelines 1.1 Text Alternatives, 1.2 Time-based Media, 2.4 Navigable, 3.1 Readable and 4.1 Compatible was conducted to determine the components that the profile would have, whether these are stereotypes or enumerations. In this way, diagrams with greater definition and expressiveness are created.

Implementing accessibility principles in the early stages of development from a Web Engineering perspective using an MDD approach can lead to improvements on a website. If the analyst considers accessibility during their stage, and the designer focuses on accessibility issues related to the basic elements of their work (such as website components and their interaction), the programmer in the development stage only needs to apply accessibility to elements specific to programming, namely, the source code. This saves the effort of having to develop new behaviors and functionalities related

to accessibility requirements (on their own initiative). Therefore, the company could save time and money, especially in the maintenance and updating stages of the website, which are usually the most costly.

Although there are currently automatic tools available that allow for the evaluation of a website's accessibility features following WCAG recommendations, they are typically useful for a quick assessment and for finding common accessibility errors. Therefore, they are not sufficient to determine if a website complies with 100% of the accessibility features regarding compliance levels A, AA, or AAA. This is why manual evaluation is required, even though it is more costly because specialists in web accessibility standards are needed. In response to this, the proposed approach using the WebPageAcc profile would assist developers in creating an accessible website, starting from a conceptual model, without requiring extensive knowledge of accessibility standards, thus ensuring that the site complies with the required compliance level regarding the WCAG standard.

B. FUTURE WORKS

As future work aims to transform the resulting model into HTML code, this way you can obtain an accessible web page in an automatic format. Finally, we will work on incorporating other accessibility criteria in the profile, to consider a major number of lines suggested by the WCAG 2.2 standard.

REFERENCES

[1] B. Vollenwyder, G. H. Iten, F. Brühlmann, K. Opwis, and E. D. Mekler, "Salient beliefs influencing the intention to consider web accessibility," *Comput. Hum. Behav.*, vol. 92, pp. 352–360, Mar. 2019.

[2] G. Broccia, M. Manca, F. Paternò, and F. Pulina, "Flexible automatic support for web accessibility validation," *Proc. ACM Hum.-Comput. Interact.*, vol. 4, pp. 1–24, Jun. 2020.

[3] T. Bi, X. Xia, D. Lo, J. Grundy, T. Zimmermann, and D. Ford, "Accessibility in software practice: A Practitioner's perspective," *ACM Trans. Softw. Eng. Methodol.*, vol. 31, no. 4, pp. 1–26, Oct. 2022.

[4] J. Abascal, M. Arrue, and X. Valencia, "Tools for web accessibility evaluation," in *Web Accessibility: A Foundation for Research*. London, U.K.: Springer, 2019, pp. 479–503.

[5] W. T. Andrade, R. G. D. Branco, M. I. Cagnin, and D. M. B. Paiva, "Incorporating accessibility elements to the software engineering process," *Adv. Hum.-Comput. Interact.*, vol. 2018, pp. 1–17, Dec. 2018.

[6] G. Alpaslan and O. Kalıpsız, "Model driven web application development with agile practices," 2016, *arXiv:1610.03335*.

[7] D. Li, X. Li, Z. Liu, and V. Stolz, "Support formal component-based development with UML profile," in *Proc. 22nd Austral. Softw. Eng. Conf.*, Jun. 2013, pp. 191–200.

[8] MG. (2014). *Object Constraint Language (OCL)*. Object Mnagement Group. [Online]. Available: https://www.omg.org/spec/OCL/2.4/PDF

[9] W3C. (2023). *Web Content Accessibility Guidelines (WCAG)*. World Wide Web Consortium. [Online]. Available: https://www.w3.org/TR/WCAG22/

[10] O. Betari, S. Filali, A. Azzaoui, and M. A. Boubnad, "Applying a model driven architecture approach: Transforming CIM to PIM using UML," *Int. J. Online Eng.*, vol. 14, no. 9, p. 170, Sep. 2018.

[11] A. P. F. Magalhaes, A. M. S. Andrade, and R. S. P. Maciel, "Model driven transformation development (MDTD): An approach for developing model to model transformation," *Inf. Softw. Technol.*, vol. 114, pp. 55–76, Oct. 2019.

[12] K. Ordoñez, J. Hilera, and S. Cueva, "Model-driven development of accessible software: A systematic literature review," *Universal Access Inf. Soc.*, vol. 21, no. 1, pp. 295–324, Mar. 2022.

[13] F. Daniel and M. Matera, "Model-driven software development," in *Mashups*. Berlin, Germany: Springer, 2014, pp. 71–93.

[14] V. V. G. Neto, M. Guessi, L. B. R. Oliveira, F. Oquendo, and E. Y. Nakagawa, "Investigating the model-driven development for Systems-of-Systems," in *Proc. Eur. Conf. Softw. Archit. Workshops*, Aug. 2014, pp. 1–8.

[15] J.-P. Tolvanen and S. Kelly, "Model-driven development challenges and solutions: Experiences with domain-specific modelling in industry," in *Proc. 4th Int. Conf. Model-Driven Eng. Softw. Develop. (MODELSWARD)*, Feb. 2016, pp. 711–719.

[16] OMG. *Unified Modeling Language (UML)*. Object Manage. Group. Accessed: Sep. 19, 2023. [Online]. Available: https://www.omg.org/spec/UML/2.5.1/PDF

[17] I. Hsu, "Extending UML to model web 2.0-based context-aware applications," *Softw., Pract. Exper.*, vol. 42, no. 10, pp. 1211–1227, Oct. 2012.

[18] M. Richters and M. Gogolla, "OCL: Syntax, semantics, and tools," in *Object Modeling With the OCL*. Berlin, Germany: Springer, 2002, pp. 42–68.

[19] A. Shaikh and U. K. Wiil, "Overview of slicing and feedback techniques for efficient verification of UML/OCL class diagrams," *IEEE Access*, vol. 6, pp. 23864–23878, 2018.

[20] J. Carter and M. Markel, "Web accessibility for people with disabilities: An introduction for web developers," *IEEE Trans. Prof. Commun.*, vol. 44, no. 4, pp. 225–233, Feb. 2001.

[21] *WAI. Web Accessibility Initiative*. Accessed: Sep. 20, 2023. [Online]. Available: http://www.w3.org/WAI/

[22] L. Garćóa-Borgoñón, M. A. Barcelona, J. A. Garćia-Garćia, and M. J. Escalona, "Software process accessibility in practice: A case study," *Proc. Comput. Sci.*, vol. 27, pp. 292–301, Jan. 2014.

[23] A. H. Guimaraes, V. V. G. Neto, S. L. da Costa, and J. L. de Oliveira, "Web accessibility using model-driven development," *II Brazilian Workshop Model-Driven Develop.* São Paulo, Brazil, 2011. [Online]. Available: https://www.researchgate.net/profile/Valdemar_Graciano_Neto/publication/301526519_Web_Accessibility_Using_Model-Driven_Development/links/5717715008ae09ceb2649d9e/Web-Accessibility-Using-Model-Driven-Development.pdf

[24] E. Rios, T. Bozheva, A. Bediaga, and N. Guilloreau, "MDD maturity model: A roadmap for introducing model-driven development," in *Proc. Model Driven Archit.-Found. Appl., 2nd Eur. Conf. (ECMDA-FA)*, Bilbao, Spain. Berlin, Germany: Springer, Jul. 2006, pp. 78–89.

[25] A. U. Alrubaee, D. Cetinkaya, G. Liebchen, and H. Dogan, "A process model for component-based model-driven software development," *Information*, vol. 11, no. 6, p. 302, Jun. 2020.

[26] C. Ponsard and V. Ramon, "Survey of automation practices in model-driven development and operations," in *Proc. IEEE/ACM 4th Int. Workshop Bots Softw. Eng. (BotSE)*, May 2022, pp. 14–17.

[27] W. M. Watanabe, D. F. Neto, T. J. Bittar, and R. P. M. Fortes, "WCAG conformance approach based on model-driven development and WebML," presented at the *Proc. 28th ACM Int. Conf. Design Commun.*, Sep. 2010, pp. 167–174.

[28] J. Estdale and E. Georgiadou, "Applying the ISO/IEC 25010 quality models to software product," in *Proc. Eur. Conf. Softw. Process Improvement*, Sep. 2018, pp. 492–503.

[29] J. López-Zambrano, J. Moreira-Pico, and N. Alava-Cagua, "Methodology to valuate and classify web accessibility evaluation tools," *E-Ciencias de la Información*, vol. 8, no. 1, pp. 172–189, 2018.

[30] P. A. Duarte, S. I. Mari?o, P. L. Alfonzo, and M. V. Godoy, "Modelo de proceso software aplicado a la revisión de la accesibilidad web en desarrollos basados en IDE," *Revista Latinoamericana de Ingeniería de Softw.*, vol. 3, no. 4, pp. 177–182, 2015.

[31] M. Laabidi, M. Jemni, L. J. B. Ayed, H. B. Brahim, and A. B. Jemaa, "Learning technologies for people with disabilities," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 26, no. 1, pp. 29–45, Jan. 2014.

[32] R. Calvo, "Accessible chats for computer supported collaborative learning environments in mobile devices: Doctoral consortium paper," in *Proc. IEEE 7th Int. Conf. Res. Challenges Inf. Sci. (RCIS)*, May 2013, pp. 1–6.

[33] L. Moreno López, F. Valverde, P. Martínez Fernández, and y O. Pastor, "Supporting navigation accessibility requirements in web engineering methods," *J. Web Eng.*, vol. 12, nos. 3–4, pp. 181–202, 2013.

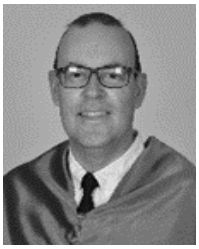
[34] M. Jemni, M. Laabidi, and L. J. B. Ayed, "Accessible E-learning for students with disabilities: From the design to the implementation," in *The New Development of Technology Enhanced Learning*. Berlin, Germany: Springer, 2014, pp. 53–74.

[35] H. Vieritz, D. Schilberg, and S. Jeschke, "Early accessibility evaluation in web application development," in *Automation, Communication and Cybernetics in Science and Engineering 2013/2014*. Cham, Switzerland: Springer, 2014, pp. 757–764.
[36] H. Hebiri, M. Laabidi, and M. Jemni, "User centered model to provide accessible e-Learning systems," in *Proc. 10th IEEE Int. Conf. Adv. Learn. Technol.*, Jul. 2010, pp. 720–721.
[37] S. Sanchez-Gordon, C. Aguilar-Mayanquer, and T. Calle-Jimenez, "Model for profiling users with disabilities on e-Learning platforms," *IEEE Access*, vol. 9, pp. 74258–74274, 2021.
[38] F. Dias, L. Duarte, and R. Fortes, "AccessMDD: An MDD approach for generating accessible mobile applications," in *Proc. 39th ACM Int. Conf. Design Commun.*, Oct. 2021, pp. 85–95.
[39] *Eclipse Papyrus*. Accessed: Oct. 10, 2023. [Online]. Available: https://www.eclipse.org/modeling/
[40] Eclipse.org. *Papyrus*. Accessed: Oct. 10, 2023. [Online]. Available: https://www.eclipse.org/papyrus/

KARLA ORDOÑEZ-BRICEÑO received the degree in information systems engineering computer from Universidad Técnica Particular de Loja, Ecuador, in 2013, and the master's degree in software engineering for the web from Universidad de Álcala, in 2016, where she is currently pursuing the Ph.D. degree in information engineering and communication technologies. She has been a Teacher with the Technical University of Machala and currently works with the Instituto Superior Tecnológico, Universitario Ismael Pérez Pazmiño. She was a part of the Technical Team, Latin American and Caribbean Institute for the Quality of Higher Distance Education (CALED). She has collaborated in the ESVIAL project as a virtual and face-to-face tutor for the Pedagogical Innovation in Quality Accessible Virtual Education Workshop, Peru edition. Her main research interests include educational technologies, web accessibility, and data mining.

JOSÉ R. HILERA received the B.S. degree in telecommunication engineering from the Polytechnic University of Madrid, Spain, in 1992, and the Ph.D. degree in science (mathematics) from the University of Alcalá, Spain, in 1997. He was a Lecturer with the Polytechnic University of Madrid. He is currently a Full Professor with the Polytechnic School, University of Alcalá. He was also the Director of the master's program in software engineering for the web and the Coordinator of the ESVI-AL cooperation network on accessible virtual education, funded by the European Union's ALFA III Program. He has authored or coauthored over 150 scientific works (books, articles, papers, and research projects), some related to learning technology and accessibility. Since their creation, he has been organizing several editions of two international conferences on accessibility, including the International Conference on Quality and Accessibility in eLearning (CAFVIR) and the International Conference on Application of Information and Communications Technologies to Improve Accessibility (ATICAcces).

LUIS DE-MARCOS received the B.Sc. and M.Sc. degrees in computer science and the Ph.D. degree in information, documentation, and knowledge, in 2001, 2005, and 2009, respectively. Since 2007, he has been with the University of Alcalá, Spain, where he has been a Full Professor, since 2022. He was an Invited Research Fellow with Lund University, Sweden, in 2007 and 2009, the University of Reading, U.K., in 2008, the Monterrey Institute of Technology, Mexico, in 2010, and the University of Zagreb, Croatia, in 2018. He was a Principal Investigator of the Research Team of the ProTego Project (H2020) on Cybersecurity (2019–2021) and two national research projects in Spain (2011–2013). He participated in the EU IAPP MC project Iceberg (2013–2017), in two Erasmus Intensive Programs (2009–2013), and two Erasmus+ Strategic Parnertship Projects (2014–2017). He is a also a member of the EU RISE uDevOps Project (2020–2024) on microservice architectures and leads the PARCHE project (2022–2025) on evidence recovery from darknets. He has over 160 refereed publications in conferences and journals. He has over 5000 citations in Google Scholar (H22), 2200 in Scopus (H14), and 1700 in Web of Science (H14). He leads the research group on Information Technologies for Training and Knowledge, University of Alcalá, with 22 researchers. He is the Director of the Ph.D. Program on Information and Knowledge Engineering. He mentored nine Ph.D. dissertations. His former students occupy academic positions with the University of Zagreb, the National University of Honduras, the University of Southampton, the University of Alcalá, and the National Institute of Aerospace Technology (INTA), Spain. His research interests include graph analysis, social network analysis (SNA), gamification, and learning technologies. He received the 2019 IEEE TRANSACTIONS ON EDUCATION Theodore E. Batchman Best Paper Award, awarded by the IEEE Education Society.



SALVADOR OTÓN-TORTOSA received the degree in computer sciences engineering from the University of Murcia, in 1996, and the Ph.D. degree from the University of Alcalá, in 2006. He is currently an Associate Professor with the Computer Science Department, University of Alcalá. He coordinates the master of agile software development for the web. He is the Coordinator of the European Project EduTech about accessible virtual education, granted by the Erasmus+ Program of the European Union. He is the author or the coauthor of more than 80 scientific works (books, articles, papers, and research projects), the majority of them directly related to learning technology, accessibility, and interoperability. His research interests include learning objects and e-learning standardization and mainly in accessibility, metadata, distributed learning object repositories, interoperability, and enterprise architectures.

SAMANTA CUEVA-CARRIÓN received the Ph.D. degree in advanced software development technologies, intelligent systems and distributed environments from the Polytechnic University of Madrid, Spain, in 2016. She is currently the Vice Dean of the Faculty of Engineering and Architecture, Universidad Técnica Particular de Loja. She has authored or coauthored scientific papers related to information technology and software engineering.