Chapter III
A Service Oriented Ontological Framework for the Semantic Validation of Web Accessibility

Rui Lopes
LaSIGE, University of Lisbon, Portugal

Konstantinos Votis
Pattern Recognition Laboratory, University of Patras, Greece
Informatics and Telematics Institute, CERTH, Greece

Luís Carriço
LaSIGE, University of Lisbon, Portugal

Spiridon Likothanassis
Pattern Recognition Laboratory, University of Patras, Greece

Dimitrios Tzovaras
Informatics and Telematics Institute, CERTH, Greece

ABSTRACT

The Web serves as the principal mediator for information sharing and communication on a worldwide scale. Its highly decentralized nature affords a scale free growth, where each endpoint (i.e., Web site) is created and maintained independently. Web designers and developers have the onus of making sure that users can interact without accessibility problems. However, coping with users with disabilities poses challenges on how to ensure that a Web site is accessible for any kind of user. When (and if) this is done, designers and developers do it in a post-hoc way, (i.e., verify and tweak Web sites according to guidelines such as WCAG). In this Chapter the authors present SWAF, the Semantic Web Accessibility
A Service Oriented Ontological Framework for the Semantic Validation of Web Accessibility

INTRODUCTION

The increasing adoption of technologies from users puts the Internet in a central spotlight. The Web, as its major application, is accessed and interacted by users at constantly increasing pace, allowing them to quickly seek information, interact with their peers through social networks, or perform transactions from the comfort of their homes. For this reason, the way information is structured and presented is critical for the success of accessing it.

However, users have their own specific characteristics (e.g., abilities, impairments, preferences, knowledge). Consequently, the accessibility of each piece of information – such as a Web page – can differ significantly from user to user. While often dismissed in the Internet realm and, more specifically, on the Web, people with disabilities are not just a small population minority. If one takes into account people with mild disabilities, the slice of the population that requires some sort of software-based accessibility ramp is of the utmost importance.

The most important way to mitigate this problem is making sure that information providers (ranging from the individual to large corporations) do not overlook such accessibility issues. Internationally recognized organizations such as the World Wide Web Consortium (W3C, n.d.) play a critical role on helping information providers to cope with accessibility. Traditionally, this goes in the form of specifying accessibility-centric best practices, guidelines, and vocabularies to augment already existing Web languages.

Evangelization of accessibility practices, coped with the progressive intertwining of accessibility features in Web languages has brought Web accessibility more close to information providers. Consequently, each day, Web accessibility is gaining awareness. Guidelines such as WCAG, the Web Content Accessibility Guidelines (Chisholm, Vanderheiden, & Jacobs, 1999), are being followed more often, leading more users with disabilities to have access to information without barriers. In order to be so, these guidelines are presented as straightforward as possible, geared towards the largest set of Web designers and developers. However, mostly due to financial, human resources, and technological expertise problems, several companies (and individuals) totally dismiss the adequacy of Web sites to the different requirements of accessibility-dependent audiences, despite the fact that legislation is being pushed in several countries, in order to promote the rights of people with disabilities.

The dismissal of accessibility from information providers leverages the fact that such guidelines and standards for accessibility have inherent problems. Since they are specified in such a way that they require manual inspection of their conformance, developers have an increased effort on coping with accessibility issues. Furthermore, by being informally described (i.e., in natural language), they tend to lead to different interpretations from developers and accessibility experts and, for this reason, different and incoherent ways to ensure that a given Webpage is accessible.

When Webpages and Websites require an additional effort of supporting more specific and fine-grained audiences, the development of accessibility-centric solutions becomes cumbersome. Guidelines have an implicit assumption of which audiences they target to (and more often than not, in a very informal and loose way). Typically, these are often geared towards people with visually impairments (e.g., different kinds of blindness).
The way users from this audience interact with a webpage is also left implicit, but often assumed to include specific devices (e.g., screen readers). This leads to a lack of understanding what is the role of guideline checkpoints for each user and device characteristic they are tackling, thus posing more difficulties on developers on how to tackle fine-grained accessibility analysis and consequent development of accessible Websites and Webpages. Consequently, software developers need to have a conceptual framework in which to situate disabled-related guidelines, which they often do not have due to lack of experience with disabled people and their technologies.

This Chapter proposes SWAF (Semantic Web Accessibility Framework) as an ontological framework targeted to accessibility-aware Web design and development processes. This will enable large organizations, small enterprises, or even individuals (developers, designers, etc.) to produce Web sites of superior accessibility and usability, accompanied with appropriate measures, technologies and tools that improve their overall quality. This ontological framework can be used to answer questions about common accessibility standards, user abilities and disabilities, as well as about the technical capabilities and constraints of appropriate assistive devices, thus forming the context for semantic validation of Web accessibility. In This Chapter we present the general overview of the framework, and detail it in the context of existing Web accessibility standards, in order to facilitate accessibility assessment of Web sites across different audiences.

**BACKGROUND**

Designing for people with disabilities is becoming an increasingly important topic for a variety of reasons, but especially due to recent legislation in many countries that aims at promoting and enforcing the rights of people with disabilities. A number of philosophies and methodologies have been developed to support this process.

Firstly there has been the development of the *universal design*, *design for all*, and *universal usability* philosophies, as detailed by Shneiderman (2000). Many developers worry that they will be expected to produce a system that will be usable by every user, regardless of their abilities, and that they might have to seriously compromise their overall design to achieve this aim. Clearly this would be in no-one’s interest. With the increasing ability to personalize interfaces to meet the requirements of different users, this is not necessary.

Secondly, there have been numerous sets of guidelines to help developers produce systems that are accessible and usable by people with disabilities. These range from very general guidelines to the very specific guidelines for Web user agents (e.g., Web browser), authoring tools, and content creators. However, it is not clear whether providing guidelines is an effective method for ensuring usable designs, since these might be differently interpreted by developers and designers. Developers need to have a conceptual framework in which to situate disabled-related guidelines, which they often do not have due to lack of experience with people with disabilities and assistive technologies.

Consequently, even if people with disabilities want to be independent and do things for themselves, unfortunately, most Web sites and Web applications are not fully accessible today. There are a number of reasons for this as explained below:

- The Web has evolved over the latest years, and the importance of accessibility has only begun to be appreciated and encouraged in recent years. Older solutions are unlikely to be fully accessible (or accessible *at all*). Making an existing Web site accessible is often very difficult and expensive, in much the same way as making an existing building...
wheelchair-friendly can be very difficult, as well as un-aesthetic. Although efforts should be made to improve accessibility, it will be typically easier to do so during a major refurbishment.

- Many developers and, more surprisingly, designers, are not aware of the importance or need for accessibility. Consequently, new developments are being built in blissful ignorance, as many of them do not have the necessary knowledge or skills for building accessible Web sites.
- Some market stakeholders believe that creating accessible solutions will have prohibited costs and, at the same time, make them boring and less attractive to the majority of users (read: the non-impaired).
- Existing design and development tools give little out-of-the-box assistance in most cases or, at worst, make it impossible to develop accessible solutions.

### Accessibility Standardization

Up to now, there are several initiatives concerning guidelines, tools and technologies for Web accessibility. The major steering body for accessibility is the World Wide Web Consortium and its Web Accessibility Initiative (WAI, n.d.). WAI has three main tracks: the Web Content Accessibility Guidelines (Chisholm et al., 1999), the Authoring Tool Accessibility Guidelines (Treviranus, Richards, Jacobs, & McCarthy, 1999), and the User Agent Accessibility Guidelines (Jacobs, Gunderson, & Hansen, 2002). The activities of W3C and WAI are the result of collaboration of groups and organizations from different countries, like the TRACE Research and Development Centre (TRACE, n.d.), which is responsible for compiling and publishing the original set of Web accessibility guidelines that provided the backbone of WAI guidelines.

Apart from the guidelines, there are also legislative and standards initiatives for accessibility. Strong governmental support in the Unites States has led to initiatives such as the Americans with Disabilities Act (ADA, 1990). The UK equivalent is the Disability Discrimination Act (DDA, 1995) amended in 1999, and now extended to the Special Educational Needs and Disability Act (SEND, 2001). The rulings of ADA are also extended to Section 508 of the US Rehabilitation Act (Section 508, 1998). This legislation defines processes and the monitoring role of the US federal government in the procurement of electronic and information technology. Regarding accessibility, it states that regardless of medium, government must ensure that disabled federal employees and members of the public have the same accessibility as non-disabled members. Where accessibility is not present, government is directed to provide alternative means. Although, Section 508 is intended for the US federal government, many organizations and software houses worldwide are making efforts to address its mandate, which puts it as a central destination for Web accessibility verification practices.

Major standards bodies such as the US Human Factors and Ergonomics Society (HFES, n.d.) are engaged in furthering the accessibility drive. Their efforts extend to features and functions of the operating systems, drivers, application services, other software layers upon which the application depends and applications that increase accessibility with a general aim of reducing the need for add-on assistive technologies. The International Standards Organisation ISO/TS 16071:2003 (ISO 16071, 2003), Ergonomics of Human–system Interaction also provides guidance on accessibility for HCI interfaces. The guidelines were designed to complement general design for usability covered by related standards on usability.

As thoroughly discussed in this Section, there exist several initiatives and standardization bodies concerning guidelines, standards and methodologies for accessibility assessment that can be effectively applied in the context of Web technologies. It is also a fact that the existing standards
and best practices concerning accessibility are in most cases confusing and incomplete (Lopes & Carriço, 2008). Therefore, developers need to have a conceptual framework in which to situate Web accessibility-related guidelines, which they often do not have, due to lack of experience on technologies for the disabled. The fundamental aspect of pushing forward accessibility on Web site design and development practices is to provide concrete and objective rules and standardized guidelines that homogenize accessibility assessment and quality control procedures. Consequently, existing software that aims at assessing accessibility based on such guidelines (a thorough list of such software packages can be found at http://www.w3.org/WAI/ER/tools/complete) will provide incomplete and overly generalized answers to whether a given Web page or Web site is accessible.

While it is clear that determining what truly represents accessibility in the customer’s view can be elusive, it is equally clear that the number and frequency of problems and defects associated with a Web site are inversely proportional to its accessibility. Software problems and defects regarding accessibility are among the few direct measurements of software processes and products. Such measurements allow us to quantitatively describe trends in defect or problem discovery, repairs, process and product imperfections, and responsiveness to customers. Problem and defect measurements also are the basis for quantifying several significant software accessibility attributes, factors, user characteristics and criteria.

Although the advantages of measurement in the Web site design and development process are indisputable, the popularity of measurement methods, within accessibility terms, in practice is rather limited (McGarry, 2002; Varkoi, 1999; SEI, 2006). Very often difficulties arise when trying to focus the measurement. In many cases it is unclear what should be measured and also how the measurement data obtained should be interpreted (Habra, Abran, & Lopez, 2004; Kulik, 2000). Choosing the correct measurement entities and ranking the importance of measurement accessibility indicators is still a challenging task (Neely, 1998). Despite the difficulties, metrics such as those defined by Vigo et al. (2007) provide insightful cues on how to approach the problem of measuring the accessibility of Web sites, based on WCAG standards. This will help designers and developers to have a better (and measurable) understanding of accessibility on Web technologies.

**Ontologies for Disability and Accessibility**

There are several efforts towards the direction of the definition of ontological concepts and taxonomies for people with disabilities. These efforts try to cover adequately the personal requirements of the end users, including the person’s disabilities and individual preferences.

A central reference for classifying disabilities concerns the World Health Organization’s International Classification of Functioning, Disability and Health (ICF, n.d.), particularly tailored to impairment qualification on medical diagnosis tasks. Consequently, it stresses just on profound disabilities, leaving out several impairments such as color blindness. Obrenovic, Abascal, & Starcevic (2007) have leveraged ICF concepts into an accessibility description framework to help designers and developers discuss and describe multimodal interaction issues.

Gruber (1993) proposes an ontology architecture that tries to cover comprehensively the situation of persons with special needs for the purpose to utilize this information for customization of their home environment’s services is proposed. This approach tries to combine contextual information like personal aspects (e.g., disabilities, preferences), technical aspects (e.g., equipment, services, network) and natural aspects (e.g., location, time) in a way that the smart home environment’s services can adapt to the end user...
more or less automatically while keeping the user in control.

Several pre-existing ontologies for supporting context-aware smart environments, like CoOL (Strang, Linnhoff-Popien, & Fank, 2003), COBRA-ONT (Chen, Finin, & Joshi, 2003), CONON (Wang, Zhang, Gu, & Pung, 2004), SOUPA (Chen, Perich, Finin, & Joshi, 2004), and UbisWorld (Heckmann, 2005). All these ontologies share common concepts and structures. From these, SOUPA incorporates most concepts of previously defined ontologies and seems to be the most elaborated one of the listed ontologies. However, all of them still lack a specific support for persons with special needs towards a comprehensive specification.

A major contribution to the field of ontologies for disabilities was made from EU’s FP6 ASK-IT project (ASK-IT, n.d.). Within ASK-IT, ontology modeling and mapping produced a collection of shared sub-ontologies, which reflect mobility impaired people user needs, and relationally map available services to them. These needs were initially specified and afterwards, the ontology authoring procedure was based on content models derived from these specifications. It also defines the interrelationships that may rationally hold between user groups of people with disabilities and various user information needs of different content types, including multi-modal content.

The potential for applying ontologies in end user diverse environments and their potential for promoting a unified methodology is exemplified by the ontology devised by Uschold, King, Moralee, & Zorgios (1998). This ontology includes lexical and relational terms based on the idea of the activity (anything that involves doing) linked to the doer or operative unit which may be a person, organizational-unit or machine said to have capability and on occasion possessing roles in respect of an activity such as activity-owner.

Wooldridge, Jennings, & Kinny (2000) also adopted a role-oriented analysis as a natural step in the Gaia methodology. Another example, the Framework for Distributed organizational Memories (Abecker et al., 2001), describes the various actors in domain ontologies according to their goals, knowledge and competencies. Van Heijst, Schreiber, & Wielinga (2000) also capture the role of an ontology in the accessibility requirements specification process, where they illustrate how a methodology can extract semantics from an ontology at different levels of depth to produce conceptual models.

As stated by Masuwa-Morgana & Burrell (2004), an ontology for accessibility requirements could be centered in a similar fashion, on an activity such as a use case in which there are doers (people and access technologies). The only difference is that in that ontology for accessibility requirements there would be a need to reliably furnish (with clear identities and essence) descriptions of doers-people and patterns of doer-access technology and subsequent competencies and demands on interface design and interaction styles. Abecker et al. (2001) propose “AccessOnto” as an accessibility requirements theoretic ontological framework consisting of four components: a requirements elicitation subsystem, an inference engine, a requirements explanation subsystem and an accessibility knowledge base. It has the intention of extrapolating a requirements specification based on rules extracted from the accessibility knowledge database based on end user traits data elicited by the end user.

The aforementioned ontological frameworks emphasize the fact that there is little coupling between ontologies regarding accessibility and disabilities, and Web accessibility assessment practices (as they tend just to frame different accessibility scenarios). Our proposed ontological framework will be based on existing ontological models, as well as in best practices for ontology engineering, affording the design of a multi-layer knowledge base for accessibility and disability requirements mapping into Web accessibility verification procedures that can provide sup-
port for the requirements and needs of different accessibility-centric user groups.

**SEMANTIC WEB ACCESSIBILITY FRAMEWORK**

For many people, in particular for groups at risk of exclusion, the complexity and lack of accessibility and usability of Web sites is a major barrier to information access. We respond to this challenge by proposing a tailored accessibility assessment ontological framework, the Semantic Web Accessibility Framework (SWAF), which affords the specification of user characteristics and their requirements, and associate them to specific accessibility assessment procedures.

The main goal of SWAF is to provide support for the formal and unambiguous definitions of accessibility domains, as well as the possible semantic interactions between them. We have specified SWAF to be integrated into accessibility verification environments (e.g., authoring tools, Web accessibility evaluators, integrated development environment - IDE). This will establish a common vocabulary for exchanging and describing the complex information that is related to accessibility assessment of Web sites. The framework aims to formalize conceptual information about:

- The *characteristics* of users with disabilities, devices, applications, and other aspects that should be taken into account when describing an audience with disabilities and developing tailored Web sites.
- Web accessibility *standards* and associated checkpoints.
- Semantic *verification rules* to help describing requirements and constraints of audiences, and associating them to accessibility checkpoints.

In order to cope with these goals, the framework must comply with the following requirements:

- To be as *formal* as possible, thus providing all the necessary definitions in a concise, unambiguous, and unified form;
- Provide information that can be *easily processed* by software applications and integrated into accessibility validation processes;
- *Easily implemented* by software developers and other users involved in the software development process of Web accessibility tools.

One of the main issues in designing and developing the proposed framework was to make it maintainable and extensible, while assuring model consistency within the framework. Therefore, we have separated SWAF into two distinct dimensions: *Web Accessibility Descriptions*, and *Web Accessibility Mapping*, as depicted in Figure 1. Each dimension is further explained in the following Sections.

*Figure 1. Semantic Web accessibility framework*
Web Accessibility Descriptions

The first dimension provides constructs to describe different Web accessibility concepts (WAD). To explore the differences and synergies between Web accessibility fields, and to support the inclusion of external concepts from other domains, the WAD dimension cuts the concept space into a Generic ontology and a set of Domain Specific ontologies, as detailed next.

Generic Ontology

The Generic ontology forms the core ontology and describes top-level entities and concepts that are critical for the semantic validation of Web accessibility. Thus this ontology provides more abstract and generic knowledge such as general characteristics and disabilities of users, devices, Web accessibility standards, and other main aspects that constitute the basis for applying accessibility-based approaches into the accessibility validation field.

Domains are specified in classes and subclasses providing a hierarchical model representing all the knowledge fields that are necessary for the accessibility validation. There are also a number of properties denoting the relationship between classes. A part of the Generic Ontology is depicted in Figure 2. This partial snapshot of the ontology consists of the main classes Disabilities, WAI_WCAG and Devices. The Disabilities class contains three subclasses: HearingImpairment, SpeechImpairment, and VisualImpairment. There is also a property of the type hasIncludedDomains denoting that the classes WAI_WCAG and Devices include disabilities.

Domain Specific Ontologies

To better illustrate how the Generic Ontology can cope with real scenarios within the Web accessibility domain, we defined several domain specific ontologies and integrated them into SWAF. These domain specific ontologies (DSO) cope with the key aspects that are part of the integration of Web accessibility into Web design and development processes. This way, ontologies are able to represent a more detailed description of their corresponding domain, fruitful for extensibility scenarios (e.g., using Web accessibility validation ontologies in Mobile Web tailoring scenarios). The purpose of distinguishing a generic ontology from the domain specific ones is to facilitate the extension of SWAF to different application domains (e.g., outside the scope of Web related accessibility guidelines and applications).

Each DSO uses the basic entities of the Generic Ontology to describe the specific concepts and structures that are needed for the semantic validation of Web accessibility. This ensures that all terms and their relationships utilized by each accessibility approaches separately are included in the generic ontology scheme. To cover every

Figure 2. Excerpt of the generic ontology
A Service Oriented Ontological Framework for the Semantic Validation of Web Accessibility

spectrum of applicability of accessibility assessment procedures, there should exist a corresponding domain specific ontology. Next, some of these ontologies are described.

General Characteristics and Disabilities Ontology

As discussed previously, validating accessibility is a process that must cope with user’s disabilities, as well as with each individual’s preferences. Thus it is of great importance to consider the users’ personal capabilities determined by her/his impairments. Consequently, different categories of disabilities (based on the ICF categorization) are incorporated within this ontology, such as:

- **Visual impairments.** Disorders in the functions of the eye ranging from reduced capability of sight, color-blindness to total disability to see (e.g., cataracts or retinal detachment).
- **Hearing impairments.** Disorders in perceiving audio, ranging from problems in understanding normal conversations to complete deafness (e.g., high or low tone hearing loss).
- **Specific learning impairments.** Disorders manifested by significant difficulties in the acquisition and use of listening, speaking, writing, reading, reasoning, or mathematical abilities.

The degree of the users’ disabilities determines the extent of Web accessibility concepts and guidelines that must be followed by Web sites (e.g., enlarging font sizes) to suit to users’ computing environment and usage context.

To afford the specification of such concepts, the General Characteristics and Disabilities Ontology provides a set of supportive constructs at a meta-level (e.g., generalizations). The main concepts are User and Characteristic. A hasCharacteristic property maps characteristics to users, thus affording the description of users. We have further detailed several meta-concepts under the Characteristic umbrella. Since this ontology is tailored to accessibility scenarios (in the broad sense of ability to access), we introduced a small taxonomy to afford the classification of Characteristic instances. This taxonomy distinguishes Ability, Disability, and Preferences, as well as more specific concepts (e.g., SensorialAbility or LearningDisability). Accordingly, the hasCharacteristic property has been refined to cope with the three main domains.

Lastly, to afford a semantic extensibility and proper categorization of user characterization concepts, we defined an extendsCharacteristic property that maps between Characteristic instances in a taxonomical way.

All of these concepts provided by the ontology strive for a strong and expressive tool for Web designers and developers to describe and characterize their target users in a clean, thorough way, along the line of the descriptive ontologies devised by Obrenovic et al. (2007). Furthermore, by affording an extensible way of organizing user characterization concepts independently from users/audiences, Web designers and developers can build their own taxonomies with respect to their particular needs without being tied to a particular way of thinking and organizing information typical of stricter solutions.

Web Accessibility Standards and Guidelines Ontology

This domain ontology covers the main evaluation guidelines for Web accessibility assessment devised in the Web Accessibility Initiative, such as WCAG. These guidelines are divided into checkpoints and arranged based on their impact and priority. The combination of these factors is given in levels (none, A, AA, or AAA), depending on their evaluation outcomes. For instance, to claim conformance on level A, all the priority one checkpoints must be satisfied.
The table presented in the Appendix of this chapter reproduces the most fundamental Priority 1 checkpoints, which have been incorporated into this ontology. It is important to notice that some of these checkpoints may be irrelevant in different situations. Certain Web site instances might not have markup that can trigger accessibility problems. Furthermore, they might also be irrelevant based in the particular constraints and preferences of a user (from an accessibility point-of-view).

In particular, the checkpoints 1.2 and 9.1 apply (only) if image maps are used, the checkpoints 5.1 and 5.2 apply if tables are used, the checkpoint 12.1 applies if frames are used, the checkpoint 6.3 applies if applets or scripts are used, and finally the checkpoints 1.3 and 1.4 apply if multimedia is used. Most of these checkpoints are just relevant for those with visual impairments.

On the meta-level, this ontology introduces the Guideline and Checkpoint concepts, which can be mapped through an includesCheckpoint property. While Web designers and developers can leverage verification processes with the already supplied instances for WCAG, the extensibility provide by this meta-level affords the addition of new guidelines and checkpoints to their development processes in an effective way. Furthermore, this will allow them to leverage out-of-the-box all domain independent verification rules provided in SWAF’s Web Accessibility Mapping Dimension.

Devices Ontology

Owing to the rapid development of electronic technologies, it tends to be common to access Web sites outside the traditional field of a desktop PC and a computer screen (e.g., PDAs, mobile phones, assisting devices, etc.). This has brought more specific assistive technologies to improve interactivity for users with disabilities, as well as broad personal preferences. This includes the ability of coping with diverse input/output modalities combination within interactive scenarios. Since the diversity of these technologies varies along different axes (e.g., display resolution, images coloring, multimedia process, etc.), the way accessibility is assessed for Web pages must also cope with these differences.

This ontology provides a simple set of meta-level concepts to describe devices ecology: a Device hasFeature DeviceFeature. Like in the description of users and accessibility scenarios, this ontology affords out-of-the-box instances for common cases of devices and device characteristics without closing the door to extensibility and odd-case scenarios.

The description of devices can be used at two different fronts within development processes: it can help triggering semantic validation rules on user/device mismatches (e.g., user with total blindness and computer screen), and tying Web accessibility guidelines and checkpoints to particular devices and/or device features.

Web Applications Ontology

Web pages are not the only end for Web technologies. Nowadays, an increasingly number of applications is being ported from “traditional” desktop environments into Rich Internet Applications (RIA), by taking advantage of the richness of Web browsers, thus becoming easily available to any kind of users. However, since these technologies cannot cope with specific semantics of desktop applications widgets, accessibility issues may arise. The proposed domain ontology supports the development of accessible Web applications by affording the inspection of ARIA keywords (Accessible RIA) within Web pages, according to application requirements.

Figure 3 represents a partial snapshot of the ontology describing the knowledge domain of HearingImpairment. One of the main classes is the CompleteDeafness that consists of four subclasses Title, Parameters, Keywords and AssignedARIA. Each of these subclasses is characterized by a
set of properties that can be either a simple data property or an object property that denotes the relationship between two classes.

This ontology supports both HTML-specific concepts (such as key HTML terms that have influence on accessibility issues), besides Web application domain concepts. This way, Web pages are perceived as a subset of Web applications from the point-of-view of key concepts, thus affording reuse scenarios of the ontology.

For example, the Web Applications Ontology provides the GUICharacteristic and GUITechnology abstract concepts (coupled with more fine-grained concepts such as HTML), to support the description of the Web Applications domain.

**Web Accessibility Mapping**

Finally, SWAF is completed with the Web Accessibility Mapping (WAM) dimension. This dimension aims to cover the establishment of mapping relationships between the ontologies of the first dimension, and to validate the semantics of these relationships. These relationships can be used, e.g., for efficient navigation and searching inside the ontologies, as well as to afford the creation of semantic rules-based accessibility verification. Two ontology layers are provided in this dimension (as detailed next): Mapping Ontology, and Rules Ontology.

**Mapping Ontology**

The WAM dimension provides a mapping ontology comprising of a set of lexical and notational synonyms to express the semantics of the relationships between concepts within the General Ontology and Domain Specific Ontologies. This mapping is necessary, since each ontology domain represents the semantics of different knowledge domains. It is important to notice that these mapping concepts can be use to tie terms from the General Ontology to any Domain Specific Ontology, as well as between different Domain Specific Ontologies. This way, the Semantic Web Accessibility Framework can support different interdependent relationships between DSOs, thus affording richer Web accessibility validation scenarios.

Since the proposed integration needs require that information be passed seamlessly among the different layers, generic and domain ontology mapping is absolutely necessary. This semantic information stems from the semantic metadata description of the content and has to be mapped to the corresponding classes and properties of the relevant domain description ontology. Therefore, each domain specific ontology will provide a set of concepts to the mapping ontology to support this type of mappings.
For instance, the Web applications ontology provides properties to map Checkpoint instances (inherent from the Web Accessibility Standards and Guidelines Ontology) to Application instances (described with concepts from the Web Applications Ontology). This mapping property allows the specification of which checkpoints are valid to the particular set of technologies available for the design and development of Web applications. Another set of mapping concepts provide support for bridging Checkpoint instances with Device instances and User instances, thus closing the loop between characterization of devices and audiences and tailored Web accessibility assessment processes.

Rules Ontology

The last piece in the Web Accessibility Mapping dimension of the Semantic Web Accessibility Framework concerns the specification of semantic validation rules for Web accessibility. This ontology will provide the required set of rules that go beyond the syntactic analysis of Web accessibility processes, such as the description of checkpoints, users, etc. The role of this ontology is, therefore, to bridge the semantic verification gap between the Web Accessibility Description domain and the Mapping Ontologies.

We have devised this ontology as a set of rules based on SWRL (Horrocks, Patel-Schneider, Boley, Tabet, Grosof, & Dean, 2004), a rules language that affords the specification Horn-like rules with OWL predicates. The Rules Ontology can be used to reason which concepts from other ontologies (both at the instance and meta levels) and which combinations of them are satisfied by accessibility validation procedures. By setting up these rules within an inference engine, relevant accessibility rules will be reasoned out according to the information residing within the ontologies of the Semantic Web Accessibility Framework’s Web Accessibility Domain dimension.

While some rules can be specified with General Ontology concepts, its use is fairly limited, as they are not tied to particular application domains. By using terms originated from Domain Specific Ontologies, and by combining them according to the semantics of existing validation processes, Web design and development processes can be augmented with more interesting verification rules that are triggered according to specific application/audience requirements. It is worth mentioning the fact that this ontology serves as an entry point for semantic validation processes. We devised it as a placeholder upon which the SWAF ontology can (and, in fact, should) be enriched with application-specific and technology-specific semantic Web accessibility validation rules.

As a simple example, we present the description of a set of rules for users that have been characterized as having some sort of visual disability, and how to cope with content presentation. This is one of the critical rule types that are to be supported within Web accessibility validation scenarios. User rules are defined as the set $UR = \{U1, U2, U3\}$, where each one of the rules represents a single semantic validation according to a specific user audience:

$U1$: if user is color blind then content of black and white images and black text are preferred.

$U2$: if user is partially sighted then content of audio and appropriate image is preferred.

$U3$: if user is totally blind then pure audio content is preferred.

The same approach can be used in other domains, e.g., for devices. When verifying if Web sites can cope with device capabilities, one can check if content can be appropriately fit into the constraints imposed by devices. Such rules domain, e.g., $DR = \{D1, D2, D3\}$ can be defined as:

$D1$: if device is a mobile phone then image depth must be black and white.
A Service Oriented Ontological Framework for the Semantic Validation of Web Accessibility

D2: if device is a PDA image color contrast must be high.
D3: if device is a PC then image depth can be of any size.

When defining such semantic rules, the constructs available on the different ontologies can be used to express the concrete situations that help Web designers and developers in the tailored Web accessibility verification processes in an effective, unambiguous way. As an example, we detail how to express the Checkpoint 2.1 from the Web Accessibility Content Guidelines 1.0 (as shown in the Appendix), which state: “Ensure that all information conveyed with color is also available without color, for example from context or mark-up”. When targeting to audiences composed by individuals with color blindness, this checkpoint can be expressed in SWRL as (compact syntax):

\[
\text{wao:hasGUITechnology}(?APP, ?x1) \& \\
\text{wao:HTML}(?x1) \& \\
\text{wao:hasGUICharacteristics}(?x1, ?x2) \& \\
\text{wao:characteristicName}(?x2, \text{“alternativeNonColoredInformation”}) \& \\
\text{wao:characteristicValue}(?x2, \text{“true”}) \& \\
\text{d o : h a s F e a t u r e } (\text{?DEV}, \text{Individual(“colorDepth1bit”)}) \\
\Rightarrow \\
\text{wasgo:isDefiningValidApplication(Individual(“WAI _ checkpoint2.1”), ?APP)}
\]

Here we are ensuring that conforming to Checkpoint 2.1, only for users (\text{?USER}) characterized by color blindness (\text{colorBlindness}), applications (\text{?APP}) encompassing HTML technologies (\text{wao:HTML}) must provide alternative non colored information. It is worth mentioning that the constructs provided by the Web Accessibility Ontology (\text{wao:hasGUITechnology}, \text{wao:HTML}, \text{wao:hasGUICharacteristics}, \text{wao:characteristicName}, and \text{wao:characteristicValue}) form the core validation rule, since they bind more general concepts (color blindness, WCAG checkpoint) to concrete concepts inherent of the application domain. All of these concepts would have to be substituted, if targeting the specific WCAG rule to other application technology (e.g., outside the scope of Web accessibility). Likewise, this type of rule can be easily adapted to device constraints such as the device domain rules described above, as follows:

\[
\text{wao:hasGUITechnology}(?APP, ?x1) \& \\
\text{wao:HTML}(?x1) \& \\
\text{wao:hasGUICharacteristics}(?x1, ?x2) \& \\
\text{wao:characteristicName}(?x2, \text{“alternativeNonColoredInformation”}) \& \\
\text{wao:characteristicValue}(?x2, \text{“true”}) \& \\
\text{d o : h a s F e a t u r e } (\text{?DEV}, \text{Individual(“colorDepth1bit”)}) \\
\Rightarrow \\
\text{wasgo:isDefiningValidApplication(Individual(“WAI _ checkpoint2.1”), ?APP)}
\]

Here we attach Checkpoint 2.1 just to those devices (\text{?DEV}) that are severely constrained by color depth, e.g., just black and white displays (“colorDepth1bit”). All other atomic rules can be retained, thus showing that Web accessibility verification semantics are similar between different characterization and verification domains (e.g., color blind vs. color depth). In both rules, the test for alternativeNonColoredInformation provides the bridge towards an actual accessibility check. In this case, a software component attached to a SRWL rules processor is triggered and analyses an HTML document accordingly.

More complex rules can be built on simpler rules, thus affording sharing semantics between verification scenarios. This will afford a richer and more complete approach to the implementation and integration of SWAF into Web accessibility aware design and development tools. In the next example we rework the two rules presented
A Service Oriented Ontological Framework for the Semantic Validation of Web Accessibility

above by refactoring the common set of rules in a modular way:

\[ wao:\text{hasGUITechnology}(\text{APP}, \text{x1}) \land wao:\text{HTML}(\text{x1}) \land wao:\text{hasGUIDisabilities}(\text{x1}, \text{x2}) \land wao:\text{characteristicName}(\text{x2}, \text{"alternativeNonColoredInformation"}) \land wao:\text{characteristicValue}(\text{x2}, \text{"true"}) \Rightarrow ex:\text{verifyHTMLColor}(\text{APP}) \]

\[ ex:\text{verifyHTMLColor}(\text{APP}) \land gdo:hasDisability(\text{USER}, \text{Individual("colorBlindness")}) \Rightarrow wasgo:isDefiningValidApplication(\text{Individual("WAI\_checkpoint2.1")}, \text{APP}) \]

\[ ex:\text{verifyHTMLColor}(\text{APP}) \land dvo:hasFeature(\text{DEV}, \text{Individual("colorDepth1bit")}) \Rightarrow wasgo:isDefiningValidApplication(\text{Individual("WAI\_checkpoint2.1")}, \text{APP}) \]

FUTURE TRENDS

As described earlier, Web accessibility is gaining traction as time goes by, and as Web technologies mature. Web designers and developers are becoming more receptive to accessibility and inclusive design practices, to broaden the spectrum of users that can be targeted by a Web site or Web application. Still, they tend to follow blindly guidelines such as WCAG, thus lacking the perception that these do not cope with a high range of accessibility situations that are not taking into account.

Furthermore, the lack of integration of accessibility tools within the design and development process of Web sites and Web applications tend to leave accessibility assessment procedures to quality assurance tasks or, at best, usability testing tasks. This fact sets Web accessibility assessment as a patch to existing development processes, which has consequences on the adequacy of Web sites and applications.

We believe that the entry point for disrupting how Web accessibility is perceived nowadays must come from proper tool support, e.g., by means of open and free Web accessibility tools that can be plugged into existing Integrated Development Environments (e.g., NetBeans, Visual Studio, etc.) and design tools (e.g., Dreamweaver). This way, Web designers and developers will have an acute sensibility for Web accessibility issues during the design and development processes they are working in.

As an extra point, the Semantic Web Accessibility Framework detailed in this Chapter provides a fine-grained control of audiences and their characteristics, and how these can cope with existing Web accessibility guidelines. We expect that by providing this feature out-of-the-box, Web design and development teams will bring audience-modeling procedures to their development processes. This will give them more control on implementing Web sites and Web applications that are accessible and verifiable during development stages.

Lastly, while Web accessibility is an important issue to take into account, it is just the starting point for providing digital services to end-users that are totally accessible and universally usable. Other domains, such as mobile phones, desktop applications, or even embedded services must also be targeted by accessibility assessment procedures during early design and development stages. The SWAF framework described in this Chapter will be extended in the future to cope with these scenarios in a very effective way, since extensibility is one of the core concerns inherent to it. On Figure 4 we present our vision of the application of SWAF in the context of application development (not just to Web sites and Web applications), and how it can be tied to Integrated Development Environments.
This architecture builds on the core technologies and concepts defined in SWAF in different fronts. First, different application domains are supported through a plug-in fashion (e.g., Web accessibility would be one of the plug-ins), based on semantic technologies such as the Mapping and Rules ontologies, or the General Characteristics and Disabilities ontology. An inference engine would provide context-reasoning features tied to particular application and technology-dependent ontologies (e.g., Web sites and HTML). We envision three extra components in this architecture that complement SWAF: (1) an ontology-oriented search engine, where developers can search for information residing in SWAF-based knowledge bases, (2) an ontology integration tool, to afford the specification of new domain-specific ontologies, and (3) a set of reporting tools centered on providing concise information about accessibility assessment procedures.

A supportive Integrated Development Environment will tie these technologies to already existing features as a complementary facet of development (e.g., similar to a debugging/helper feature). We believe that enriching IDEs with such features, as well as supportive accessibility simulation and reporting facilities will bring Web accessibility and general accessibility assessment procedures to a wide range of designers and developers, thus lowering the burden of providing accessible applications to all users without any kind of barriers.

CONCLUSION

This chapter presented SWAF, the Semantic Web Accessibility Framework, as the foundation for the semantic description of Web accessibility audiences, concepts and verification rules. This framework provides the basic constructs for the creation of Web accessibility verification engines that are capable of performing assessments tailored to specific user audiences and interaction devices. We have divided SWAF into two dimensions, Web Accessibility Descriptions (which includes general and domain specific concepts) and Web Accessibility Mapping (which affords semantic mapping and rules between concepts from the first dimension), in order to afford the
extension of SWAF into different domains in the scope of Web accessibility.

Ongoing work is currently being done in several fronts in the SWAF realm, including: (1) building a comprehensive set of concept instances for user and device feature characterization, (2) providing support for guidelines and standards other than WCAG, (3) improving the Mapping Ontology to cover more situations, (4) implement a robust inference engine supporting SWAF concepts, (5) integrate this inference engine into existing Integrated Development Environments and other Web site development tools, and (6) extend the SWAF ontology to cover other application domains outside the scope of the Web.

REFERENCES


**KEY TERMS AND DEFINITIONS**

**Accessibility**: The *ability to access*. Often tied to people with disabilities (e.g., total blindness), accessibility thrives to break the barriers to information access. We follow the strict sense of accessibility by embracing any situation where the ability to access information can be disrupted by device or even surrounding environment constraints.

**Accessibility Guidelines**: A set of best practices that must be followed by designers and developers when implementing software solutions (e.g., Web site) that will help on providing accessible information. By being guidelines, it should not be assumed that content is accessible just by following them.

**Checkpoint**: A concrete verification task that materializes a (part of a) guideline. Checkpoints can be fully automated if application technology provides corresponding support (e.g., verifying if all images have associated textual captions).

**Integrated Development Environment**: A computer application used by developers that provides several features to ease the task of developing applications, such as text editor, compiler, automation features, etc.

**Universal Usability**: a research field that studies the adequacy of user interfaces and information to all users, regardless of their characteristics, knowledge, or mean of interaction (Shneiderman, 2000).

**Usability**: A research field that studies how adequate user interfaces are to users, how easily can they learn to perform tasks, and what are their levels of satisfaction when interacting with user interfaces.

**User Interface**: The “visible” side of an application, where users can acquire and interact with information.

**Web Accessibility**: The subfield of accessibility that is targeted to the specific technologies and architecture that compose the World Wide Web. This includes technologies such as HTML, CSS and JavaScript, as well as the HTTP protocol.
## APPENDIX: WAI WEB CONTENT ACCESSIBILITY PRIORITY 1 CHECKPOINTS

<table>
<thead>
<tr>
<th>Cp</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cp1.1</td>
<td>Provide a text equivalent for every non-text element (e.g., via “alt”, “longdesc”, or in element content).</td>
</tr>
<tr>
<td>Cp2.1</td>
<td>Ensure that all information conveyed with colour is also available without colour, for example from context or mark-up.</td>
</tr>
<tr>
<td>Cp4.1</td>
<td>Clearly identify changes in the natural language of a document’s text and any text equivalents (e.g., captions).</td>
</tr>
<tr>
<td>Cp6.1</td>
<td>Organize documents so they may be read without style sheets.</td>
</tr>
<tr>
<td>Cp6.2</td>
<td>Ensure that equivalents for dynamic content are updated when the dynamic content changes.</td>
</tr>
<tr>
<td>Cp7.1</td>
<td>Until user agents allow users to control flickering, avoid causing the screen to flicker.</td>
</tr>
<tr>
<td>Cp14.1</td>
<td>Use the clearest and simplest language appropriate for a site’s content.</td>
</tr>
<tr>
<td>Cp1.2</td>
<td>Provide redundant text links for each active region of a server-side image map.</td>
</tr>
<tr>
<td>Cp9.1</td>
<td>Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape.</td>
</tr>
<tr>
<td>Cp5.1</td>
<td>For data tables, identify row and column headers.</td>
</tr>
<tr>
<td>Cp5.2</td>
<td>For data tables that have two or more logical levels of row or column headers, use mark-up to associate data cells and header cells.</td>
</tr>
<tr>
<td>Cp12.1</td>
<td>Title each frame to facilitate frame identification and navigation.</td>
</tr>
<tr>
<td>Cp6.3</td>
<td>Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page.</td>
</tr>
<tr>
<td>Cp1.3</td>
<td>Until user agents can automatically read aloud the text equivalent of a visual track, provide an auditory description of the important information of the visual track of a multimedia presentation.</td>
</tr>
<tr>
<td>Cp1.4</td>
<td>For any time-based multimedia presentation (e.g., a movie or animation), synchronize equivalent alternatives (e.g., captions or auditory descriptions of the visual track) with the presentation.</td>
</tr>
<tr>
<td>Cp11.4</td>
<td>If, after best efforts, you cannot create an accessible page, provide a link to an alternative page that uses W3C technologies, is accessible, has equivalent information (or functionality), and is updated as often as the inaccessible (original) page.</td>
</tr>
</tbody>
</table>