Chapter V
Querying Web Accessibility Knowledge from Web Graphs

Rui Lopes
LaSIGE, University of Lisbon, Portugal

Luis Carriço
LaSIGE, University of Lisbon, Portugal

ABSTRACT

Web Accessibility is a hot topic today. Striving for social inclusion has resulted in the requirement of providing accessible content to all users. However, since each user is unique, and the Web evolves in a decentralized way, little or none is known about the shape of the Web’s accessibility on its own at a large scale, as well as from the point-of-view of each user. In this chapter the authors present the Web Accessibility Knowledge Framework as the foundation for specifying the relevant information about the accessibility of a Web page. This framework leverages Semantic Web technologies, side by side with audience modeling and accessibility metrics, as a way to study the Web as an entity with unique accessibility properties dependent from each user’s point of view. Through this framework, the authors envision a set of queries that can help harnessing and inferring this kind of knowledge from Web graphs.

INTRODUCTION

Since its inception, the Web has become more and more prolific in people’s lives. It is used as an information source, both one-way (e.g., newspapers) and two-way (e.g., blogging, forums, or even instant messaging). New Web sites and new content are produced and published each second by both professionals and amateurs, each one with different usability and accessibility quality marks. This fact, in conjunction with the Web’s decentralized, yet highly connected architecture, puts challenges on the user experience when interacting and navigating between Web sites.

At the same time, the attractiveness of the Web brings more users to use it on a regular ba-
sis. This means that user diversity will be closer to real life where both unimpaired and impaired users coexist. Since each user has its own specific requirements, (dis)abilities, and preferences, their experience is different for each one, resulting in different satisfaction levels. In the same line of user diversity, device prolificacy and Internet connection ubiquity also contribute to the range of possible user experiences on interacting with the Web and, consequently, also have a stake in accessibility issues.

For all these reasons, the shape of the Web itself deeply influences each user’s interactive experience in different ways. Users tend to navigate through the Web by avoiding Web sites that cannot be rendered correctly, which provide poor interactive capabilities for the specificities of the user or the device she/he is using to access the Web, reflecting negatively on users’ experience. Therefore, it is required to understand the Web’s graph of Web pages at a large scale from the point-of-view of each individual’s requirements, constraints and preferences, and grasp this information to devise future advancements on Web standards and accessibility-related best practices. The inability to adapt the Web, its standards, technologies, and best practices will pose severe problems on the society in general, by leaving untouched the barriers towards a proper e-inclusion level that can actually cope with everyone, independently of impairments and related needs.

The main contributions of this Chapter are: (1) the establishment of a Web accessibility framework that can be used to create complex knowledge bases of large scale accessibility assessments; and (2) a set of query patterns to infer critical aspects of the accessibility of Web graphs with a fine-grained control (based on users’ requirements and constraints). The proposed framework and the set of query patterns will form a core tool that helps analyzing the semantics of the accessibility of Web graphs. Next, we describe the relevant background work on Web accessibility and knowledge extraction from Web graphs.

**BACKGROUND**

Two main research topics have influence and contribute to the study of Web accessibility on large scale: the analysis of accessibility compliance of a Web page (or Web site), and the analysis of the Web’s graph structure.

The Web Accessibility Initiative (WAI, n.d.) of the World Wide Web Consortium (W3C, n.d.) has strived for setting up the pace of Web Accessibility guidelines and standards, as a way to increase accessibility awareness to Web developers, designers, and usability experts.

The main forces of WAI are the Web Content Accessibility Guidelines, WCAG (Chisholm et al., 1999). WCAG defines a set of checkpoints to verify Web pages for specific issues that have impact on accessibility of contents, such as finding if images have equivalent textual captions. These guidelines have been updated to their second version (Caldwell et al., 2008) to better handle the automation of accessibility assessment procedures, thus dismissing the requirement of manual verification of checkpoint compliance.

Until recently, the results of accessibility assessment were presented in a human-readable format (i.e., Web page). While this is useful for developers and designers in general, this is of limited use for comparison and exchange of assessment results. Therefore, WAI has defined EARL, Evaluation and Report Language (Abou-Zahra, 2007), a standardized way to express evaluation results, including Web accessibility evaluations, in an OWL-based format (Dean & Schreiber, 2004).

EARL affords the full description of Web accessibility assessment scenarios, including the specification of who (or what) is performing the evaluation, the resource that is being evaluated, the result, and the criteria used in the evaluation. However, EARL does not provide constructs to support the scenarios envisioned in macro scale Web accessibility assessments. It cannot cope with metrics (thus dismissing quantification
of Web accessibility) and with the Web’s graph structure. This way, EARL becomes limited to single Web page qualitative evaluations.

Lopes & Carriço (2008a) have shown that current Web accessibility practices are insufficient to cope with the whole spectrum of audiences (both disabled and unimpaired users), and that any user can influence everyone’s interactive experience on the Web (especially regarding accessibility issues). As Kelly et al. (2007) have predicted, to cope with every user, holistic approaches to Web accessibility have to be taken into account. This includes tailoring of accessibility assessment procedures to each individual’s characteristics, as thoroughly discussed by Vigo et al. (2007b).

Generalizing the concept of accessibility to all users (and not just to those that deeply depend on it – i.e., people with disabilities), the adequacy of user interfaces to each user’s requirements, limitations, and preferences is the ultimate goal of Universal Usability, as defined by Shneiderman (2000). As detailed by Obrenovic et al. (2007), one has to take into account users, devices, and environmental settings when studying accessibility in a universal way. However, to our knowledge, there is no work on how to measure the universal usability quality of a single Web page, from the perspective of a unique user (per definition of universal usability).

When scaling up to the size of the Web, other aspects of analysis have to be taken into account. The characterization of the Web (e.g., its size, analysis metrics, statistics, etc.) is a hot topic today. Web Science is emerging as a discipline that studies the Web as a dynamic entity, as described by Berners-Lee et al. (2006). It is centered on how infrastructural requirements, application needs, and social interactions depend and feed each other in the Web ecology (Hendler et al., 2008).

At a more fundamental level, one of the core aspects of studying the Web concerns on how it is universally usable, as hypothesized and defended by Shneiderman (2007). However, since this discipline is fairly new, little is know about the Web from a universal usability point-of-view. It is known that the evolution of Web standards has influence on the way users navigate and interact with the Web (Weinreich et al., 2006), but not to what extent and what is the impact on each individual’s characteristics. By having a proper characterization of the Web’s graph from each individual’s point of view (i.e., requirements, needs, constraints, preferences), more complex studies can be preformed at higher abstraction levels, such as in-depth Social Network Analysis (cf. Berger-Wolf & Saia, 2006) and other types of social studies.

In Lopes & Carriço (2008b) the authors presented a mathematical model to study universal usability on the Web. It supports the analysis of the Web from the point-of-view of each user’s characteristics, and explains how the Web’s structure influences user experience. While the authors have hypothesized how this model can be used to observe the evolution of the Web, it just provides a theoretical framework for the analysis of accessibility. Nevertheless, this model provides interesting contributions on how the query patterns presented in this Chapter should be formulated.

WEB ACCESSIBILITY KNOWLEDGE FRAMEWORK

In order to open the way to querying different Web accessibility properties from Web graphs, we have defined a supportive knowledge framework. This framework groups four different components, as depicted in Figure 1: Web Graphs, Web Accessibility Assessment, Audiences, and Metrics. The framework has been design according to the following requirements:

- Universal. The framework should not be limited to “traditional” accessibility audiences (such as people with visual impairments), but cope with different kinds of
accessibility-prone issues, such as limited interaction devices (e.g., mobile phones), or adversary environment settings (e.g., poor lighting settings). The universality concept can (and, in fact, should) be also extended to all users and usage situations, thus allowing knowing the impact of Web accessibility and similar universal usability issues on any user.

- **Generalized.** The framework must not impose a priori any limitation or bias towards particular accessibility assessment concepts. It should define them at a meta-level, in order to be possible to define query patterns that are independent from particular instances (e.g., a query pattern depends on user characteristics, not on a user characteristic).

- **Extensible.** Since the accessibility assessment procedures change (mostly to enforce better analyses), the framework should support the application of different procedures.

- **Fine-grained.** As discussed earlier, current accessibility evaluation practices are black-boxed, leading to having just a general view of evaluation results. The framework should support fine-grained analyses, to support studying accessibility from the perspective of different audiences.

- **Scalable.** The framework should not impose limits to the size and complexity of encoded information (i.e., knowledge base).

Each component is defined through a specific OWL-based vocabulary, as the inclusion of already existing ontologies (mostly specified in OWL) lowers the burden of defining each component of the framework. Accordingly, we have developed this framework by extending the EARL ontology to support the elicited requirements. Next, each component of the framework is described in more detail. For details about the namespace prefixes used in the next Sections and their corresponding URI mappings, please consult the Appendix. Throughout this Section we will provide examples on how to describe accessibility knowledge based on the Notation 3 (N3) syntax (Berners-Lee, 2006).

**Web Graphs**

The first component in the framework relates to the specification of Web graphs. The goal of this component is to represent each Web page as a single resource, as well as its corresponding hyperlinking structure. Figure 2 presents the concepts that support the specification of Web graphs.

The main subject of constructing Web graphs is the Web page. Since the EARL specification
only supports the specification of subjects that are available on the Web (earl:Content class), we have further refined the concept to limit its scope just to Web pages (the core subject of accessibility assessment procedures), through the ev:Webpage class. Other types of content, such as images and CSS stylesheets (Bos, Çelik, Hickson, & Lie, 2007), were considered inherent of each Web page, from the perspective of evaluation procedures.

Two main properties (and their inverse) were defined to specify hyperlinks. The first, ev:linksTo (and its corresponding inverse property, ev:islinkedBy) establishes the direct relationship between two Web pages. The second property, ev:reaches (and its inverse, ev:isReachedBy), extends ev:linksTo with a transitive characteristic. This way, it becomes possible to query Web graphs from the perspective of reachability between two (or more) Web pages, not just on direct linking properties. This property will only afford knowing whether two Web pages are indirectly connected, leaving outside of the scope the number of links in between them. We have opted to explicitly define inverse properties, to afford the specification of queries that are more expressive and closer to natural language. To complement these constructs, we have specified the ev:Website class that, in conjunction with the ev:isComposedBy property (and its inverse, ev:composes), affords the direct specification of which Web pages belong to the same Website. To support out-of-the-box the specification of hyperlinking structure for Web sites, we have defined that ev:Website extends the ev:Webpage concept. However, the ontology cannot enforce the semantics that if two Web pages are linked, then their corresponding Web sites are also linked. Hence, we have devised two rules in SWRL (Horrocks et al., 2004) to afford linking scenarios, as presented next:

\[
\begin{align*}
\text{ev:linksTo}(?website1, ?website2) & \Rightarrow \\
& \text{ev:isComposedBy}(?website1, ?webpage1) & \& \\
& \text{ev:isComposedBy}(?website2, ?webpage2) & \& \\
& \text{ev:linksTo}(?webpage1, ?webpage2)
\end{align*}
\]

\[
\begin{align*}
\text{ev:reaches}(?website1, ?website2) & \Rightarrow \\
& \text{ev:isComposedBy}(?website1, ?webpage1) & \& \\
& \text{ev:isComposedBy}(?website2, ?webpage2) & \& \\
& \text{ev:reaches}(?webpage1, ?webpage2)
\end{align*}
\]
Next, we present a small example of how to define Web graphs, formally expressed (in the N3 format):

@base <http://example.com/>.

<b.html> a ev:Webpage.
<c.html> a ev:Webpage.
<a.html> a ev:Webpage;
    ev:linksTo <b.html>;
    ev:linksTo <c.html>.

<> a ev:Website.
<> ev:isComposedBy <a.html>;
    ev:isComposedBy <b.html>;
    ev:isComposedBy <c.html>.

Web Accessibility Assessment

The essential aspects for accessibility assessment results concern the description of the tests and their resulting outcome of applying them to a Web page. Consequently, the EARL ontology affords an extensible way of describing Web accessibility assessment results, in the form of earl:Assertion predicates. This includes, amongst other predicates, the specification of which test is being applied (i.e., earl:TestCase) and what is the result of its application to the Web page that its being evaluated (i.e., earl:TestResult).

In the second component of our framework, we have extended the EARL predicates for accessibility assessment by refining test cases (i.e., earl:TestCase) with appropriate semantics about the nature of the tests, regarding the different technologies used in Web pages. This will afford the fine-grained analysis of Web pages according to technological criteria, as depicted by the concepts in Figure 3.

The main predicates for describing the nature of the tests are: ev:TestContent, for the specification of tests applied to the actual contents (in different media) of Web pages; ev:Structure, for tests applied directly on the HTML structure itself; ev:Style, when testing styling properties (such as analyzing CSS); and ev:Behavior, to represent tests over scripts (e.g., Javascript).

To better illustrate the usage of this ontology, we present next a classification of some WCAG 1.0 guidelines:

@prefix wcag10: <http://www.w3.org/TR/WCAG10/>.

wcag10:gl-color a ev:TestStyle.
wcag10:gl-structure-presentation a evTestStructure.

Figure 3. Web accessibility assessment ontology
Audiences

We have defined a third component in our ontological framework to support the specification of audiences. This will ensure that different queries can be performed to a knowledge base of Web accessibility assessment according to the necessities and characteristics of different audiences. We based this support on earlier works on audience modeling, such as those described in Lopes & Carriço (2008a). Figure 4 depicts the complete ontological vocabulary to describe audiences (for simplicity, inverse properties are omitted).

The atomic concept in this vocabulary is \textit{ev:AudienceCharacteristic}. Its purpose is to represent a single concept of an audience (e.g., a specific disability, a device characteristic, etc.). Since characteristics may represent concepts at different abstraction levels, they should be structured taxonomically. We introduce the \textit{ev:refines} property (and its inverse, \textit{ev:isRefinedBy}) to afford this expressivity.

The inherent nature of audiences raises the fact that they are often defined by several characteristics. Accordingly, this vocabulary introduces \textit{ev:AudienceClass} as a way to represent them, along the side of the \textit{ev:audienceClassContains} property (and its inverse, \textit{ev:audienceCharacteristicIsContainedBy}) to map characteristic inclusion by an audience. However, since this association is merely syntactic, incoherent audiences might be described. To mitigate such issues we have introduced two additional concepts in the vocabulary. The first, \textit{ev:dependsOn}, affords mapping dependencies between characteristics (such as total blindness depends on screen reader). The second, \textit{ev:incompatibleWith}, allows the specification of incompatibilities between characteristics (e.g., total blindness is incompatible with screen). With these two properties, the semantics of audiences can be verified automatically. These properties, in conjunction with \textit{ev:refines}, form the set of semantic relations that can be established between characteristics. Therefore, we introduce a generalization concept, \textit{ev:characteristicRelation}, as an abstraction for the three concepts. This term affords inferring, e.g., if two characteristics have any kind of dependency between them.

While analyzing Web graphs from the perspective of a single audience can provide interesting...
results, the scope of such results is limited. It is often to perform comparative analyses of the results for a set of audiences. To support such scenarios, we have defined an audience aggregation concept, \( ev:AudienceDomain \), to represent the domain of audiences that will be analyzed. The inclusion of an audience by a domain is represented through the \( ev:audienceDomainContains \) property (and corresponding inverse, \( ev:audienceClassIsContainedBy \)).

Lastly, we introduce in this vocabulary another concept to explore the synergies and differences between audiences, through the \( ev:audienceClassExtends \) property (and its \( ev:audienceClassIsExtendedBy \) counterpart). This extension mechanism is based on traditional object oriented modeling practices, i.e., an audience that extends another audience inherits its characteristics, thus creating parent-child relationships between audiences within a domain. Moreover, due to the fact that characteristics are taxonomically organized (through the \( ev:refines \) property), the characteristics of child audiences can be inferred and generalized to their common parent audience. A simple example follows, where a small taxonomy of characteristics is defined, and used in the definition of an audience domain.

```owl
@prefix tx: <http://taxonomy.com/>.
@prefix au: <http://audiences.com/>.

tx:characteristic a ev:AudienceCharacteristic.
tx:disability a ev:AudienceCharacteristic.
tx:blind a ev:AudienceCharacteristic.
tx:colorBlind a ev:AudienceCharacteristic.
tx:device a ev:AudienceCharacteristic.
tx:screen a ev:AudienceCharacteristic.


au:domain1 a ev:AudienceDomain.
au:blind a ev:AudienceClass.

au:totallyBlind a ev:AudienceClass.

au:colorBlind a ev:AudienceClass.
au:colorBlind a ev:audienceClassContains tx:colorBlind.
au:colorBlind a ev:audienceClassContains tx:screen.

```

However, affording the description of audience domains has limited applicability. To ensure that queries on Web graphs can be formulated based on the characteristics of audiences, there must be a mapping between audiences and accessibility assessment tests. Such vocabulary is synthesized in Figure 5.

In this vocabulary we have introduced a single property, \( ev:requiresCharacteristic \) (and its cor-
Querying Web Accessibility Knowledge from Web Graphs

responding \textit{ev:isRequiredByTest} counterpart), that maps a characteristic to a particular \textit{earl:TestCase} instance. With this property, any audience or even entire domain can be mapped to the battery of tests that must be performed to a Web page, in order to obtain results tailored to these audiences. Dually, if the entire set of tests is performed over each Web page, their results can be queried from the perspective of different audiences or entire domains. An example follows, where the two previous examples are bound together. More specifically, we map test cases to concrete audience characteristics that have been defined.

@prefix wcag10: <http://www.w3.org/TR/WCAG10/>.
@prefix tx: <http://taxonomy.com/>.


\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure5.png}
\caption{Audience/test mapping sub-ontology}
\end{figure}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure6.png}
\caption{Metrics ontology}
\end{figure}

**Metrics**

The last component in the framework concerns the specification of Web accessibility metrics, i.e., providing quantitative information about the accessibility of a Web page. Since different metrics can be applied to evaluation results, a supportive vocabulary for the specification of metrics must be extensible. This way, Web graphs can also be analyzed from the perspective of different metrics, thus allowing exploring which metric is better suited to different accessibility scenarios. Figure 6 depicts the vocabulary to support the specification of metrics.

The main concept in the metrics vocabulary is \textit{ev:Metric}. Its purpose is to afford the specification of metrics that are applied to each Web page, based on the results of corresponding tests. While some metrics might be independent from specific tests, more concrete metrics can depend on the application of them. Therefore, we introduce the \textit{ev:requiresTest} property to define dependency binds between metrics and tests (and its counterpart, \textit{ev:isRequiredByMetric}). This property can
Querying Web Accessibility Knowledge from Web Graphs

be used, e.g., to specify consistency verification rules on the application of metrics, based on their semantics. Furthermore, by crossing this property with the `ev:requiresCharacteristic`, metrics can be mapped indirectly to audience characteristics. However, metrics can also be directly related to audience characteristics. This affords tying up specific quantification procedures to characteristics. Hence, we introduce an extra property in the vocabulary, `ev:hasMetric`, in order to support this type of scenarios. Next, we present a simple example on how to bind metrics with tests and characteristics.

```
@prefix wcag10: <http://www.w3.org/TR/WCAG10/>.  
@prefix m: <http://example.com/metrics#>.  
@prefix tx: <http://taxonomy.com/>.  
m:simpleMetric a ev:Metric;  
   ev:requiresTest wcag10:gl-color;  
   ev:requiresTest wcag10:gl-structurePresentation.  
m:charMetric a ev:Metric.  

tx:colorBlind ev:hasMetric m:charMetric.  
```

We have introduced another concept on the vocabulary that is crucial to the specification of metrics. Each metric is supposed to have a concrete value, when applied to a Web page. Therefore, the vocabulary provides support to this feature through the `ev:hasMetricValue` datatype property, where metric values can be setup in the [0, 1] range (i.e., percentage). This way, since each metric does not yield an absolute value, Web graphs can be compared in the perspective of different metrics. With these constructs, the framework provides the support for specifying the resulting application of a given metric, in the context of an accessibility evaluation procedure. However, it is out of the scope of this Chapter to describe how these metrics are calculated (cf. Vigo et al., 2007a).

Consequently, since this property is abstract, concrete metrics properties must be derived from `ev:hasMetricValue` through subclassing. This extension to the metrics ontology is depicted in Figure 7.

As a simple example, we present how to use this extension to the metrics ontology, by specifying a new datatype property, as well as its application in a concrete set of Web pages.

```
@prefix m: <http://example.com/metrics#>.  
@base <http://example.com/>.  

m:hasSimpleMetricValue rdfs:subPropertyOf  
   ev:hasMetricValue.  
<a.html> m:hasSimpleMetricValue 0.2.  
<b.html> m:hasSimpleMetricValue 0.9.  
<c.html> m:hasSimpleMetricValue 0.45.  
```

However, with these constructs it is impossible to know what are the metric values associated with a specific characteristic or test case, for a given Web page. This happens due to `ev:Metric` instances are not automatically bound to datatype properties derived from `ev:hasMetricValue`. Consequently, query patterns cannot be created to explore complex mining scenarios, as each binding between metrics and datatype properties have to be artificially created on each query, which poses sever limitations on the generalization requirement for querying Web accessibility.

```
Figure 7. Metrics ontology extension
```

![Metrics ontology extension](image-url)
order to mitigate this situation, we defined another property, \textit{ev:relatesToMetric} (and its counterpart, \textit{ev:isRelatedToDatatypeProperty}), to draw both concepts together, as depicted in Figure 8.

Since we wanted to bind a datatype property directly to an \textit{ev:Metric} instance, we had to import the OWL schema into our own ontology. This is due to the fact that, per definition, object properties bind class instances. Because \textit{ev:hasMetricValue} (and subclassed datatype properties) are \textit{owl:DatatypeProperty} instances, we circumvented this to afford the specification of richer and more complex query patterns that can remain agnostic to particular concepts or instances.

To exemplify the usage of this property, we have bound a metric instance to a particular datatype based on the previous examples, as shown next.

\begin{verbatim}
@prefix m: <http://example.com/metrics#>.

m:hasSimpleMetricValue ev:relatesToMetric m:simpleMetric.
\end{verbatim}

However, by setting \textit{ev:relatesToMetric}’s domain to a generic OWL construct, one can bind metrics to any datatype property as there is no formal way to restrict the domain just to datatype properties derived from \textit{ev:hasMetricValue}. To mitigate this issue, there must be an appropriate semantic enforcement through rules. The following SWRL rule affords this scenario:

\begin{verbatim}
ev:relatesToMetric(?datatypeProperty, ?metric) =>
    owl:subPropertyOf(?datatypeProperty, 
    ev:hasMetricValue)
\end{verbatim}

**QUERY PATTERNS**

The extensions to the EARL ontology that we presented in the previous Section provide a comprehensive set of concepts that afford the full description of Web graphs from the perspective of Web accessibility and audience richness. This framework serves as the base ground for setting up Web graph knowledge bases that can be semantically queried in different forms. From the vast range of Semantic Web querying technologies, we opted to specify queries in the SPARQL language (Prud’hommeaux & Seaborne, 2008), as it is the \textit{de facto} querying standard in the Semantic Web stack.

All examples in this Section will be based on the following SPARQL prefixes mapping:

\begin{verbatim}
PREFIX earl: < http://www.w3.org/ns/earl#>
PREFIX ev: <http://hcim.di.fc.ul.pt/ontologies/evaluation#>
PREFIX m: <http://example.com/metrics#>
PREFIX tx: <http://taxonomy.com/>
PREFIX au: <http://audiences.com/>
PREFIX wcag10: <http://www.w3.org/TR/WCAG10/>$
\end{verbatim}

In this chapter, we have distinguished two different types of query patterns than can be applied to Web graphs: mining properties, and partitions extraction. Next, we describe each one of these pattern types.

**Mining Web Site Properties**

Web sites on their own can be analyzed from several perspectives. In this Section, we present some query patterns that can extract relevant
information about a single Web page, as well as a set of Web pages perceived as one single entity (i.e., Web site). For practical purposes, all SPARQL patterns are applied to a dummy Web page (http://example.com/a.html) or Web site (http://example.com), whose semantics are marked as ev:Webpage and ev:Website instances, correspondingly; other instances that appear on queries are based on the examples presented in the previous Sections.

**Metric thresholds.** One of the simplest ways of verifying the accessibility of a single Web page relates to setting up quality thresholds. We have devised several query patterns for this purpose. The simplest query concerns a strict threshold that yields whether a Web page has a minimum quality level for a specific metric:

```sparql
ASK {
  <http://example.com/a.html> m:hasSimpleMetricValue ?v.
  FILTER (?v >= 0.5)
}
```

Based on this pattern, one can generalize it for minimum and maximum boundaries, thus allowing checking if a Web page belongs to a particular quality cluster:

```sparql
ASK {
  <http://example.com/a.html> m:hasSimpleMetricValue ?v.
  FILTER(?v >= 0.5 && ?v <= 0.75)
}
```

On the other hand, thresholds can be used to understand what metrics are above a certain value (or between two boundaries). Along these lines, the previous query pattern can be rewritten as:

```sparql
SELECT ?metric
WHERE {
  ?metricValue ev:relatesToMetric ?metric.
  ?test rdfs:subClassOf ev:TestImageContent.
  FILTER (?v >= 0.5)
}
```

This query pattern can be extended to find out which types of test cases have an inherent quality above a given threshold (the DISTINCT query modifier has been used to remove duplicates):

```sparql
SELECT DISTINCT ?testType
WHERE {
  ?metricValue ev:relatesToMetric ?metric.
  ?test rdfs:subClassOf ?testType.
  FILTER (?v >= 0.5)
}
```

**Characteristic quality.** As explained earlier, characteristics can be bound to metrics. This feature of the framework allows the exploration of quality metrics similar to metric thresholds, but taking into account characteristics as the main feature to be analyzed:

```sparql
SELECT ?metric
WHERE {
  ?metricValue ev:relatesToMetric ?metric.
}
```

```sparql
ASK {
  FILTER (?v >= 0.5 && ?v <= 0.75)
}
```
This pattern can be extended in order to leverage which characteristics have a quality level above a certain threshold:

SELECT ?char
WHERE {
  ?char ev:hasMetric ?metric.
  FILTER (?v >= 0.5)
}

By having a quality mark associated to characteristics, these can also be compared, to verify which one is better supported in a Web page. This can be directly achieved with the following query pattern:

ASK {
  FILTER (?v1 > ?v2)
}

Furthermore, both patterns can be combined to extract which characteristics have a better quality than a predetermined one:

SELECT ?char
WHERE {
  FILTER (?v1 > ?v2)
}

**Audience quality.** One of the important aspects discussed earlier pertains to knowing if a Web page has a certain degree of quality in what respects to a particular audience. The previous query pattern can be adapted to support this feature:

ASK {
  ?char ev:hasMetric ?metric.
  FILTER (?v >= 0.5)
}

While this query pattern affords the explicit verification of the quality of a given audience, it is also relevant to explore and infer which audiences are supported in a Web page, with a given quality level. This pattern can be translated into SPARQL as:

SELECT ?audience
WHERE {
  ?char ev:hasMetric ?metric.
  FILTER (?v >= 0.5)
}
Domain quality. In the same fashion as the previous patterns, one can obtain information about whether a domain is supported by a Web page or not, according to a specific threshold:

ASK {
  ?char ev:hasMetric ?metric.
  FILTER (?v >= 0.5)
}

In the case where one wants to discover which domains are above a given threshold, the previous query pattern can be adapted in a simple way to cope with this requirement, as follows:

SELECT ?domain
WHERE {
  ?char ev:hasMetric ?metric.
  FILTER (?v >= 0.5)
}

Website quality. While all of the previous patterns are targeted just to a single Web page, it is relevant to find out information about the set of Web pages from a unique entity point of view (e.g., a Web site). By exploring the Web graph ontology provided in the framework, Web sites can be analyzed as a single entity:

SELECT ?page
WHERE {
}

More complex patterns can be devised for Web sites, based on this pattern and the set of patterns presented above for Web pages. For instance, combining this pattern with characteristic quality analysis, Web sites can be analyzed to find out which ones are entirely accessible for any audience characteristic above a certain threshold:

SELECT ?site
WHERE {
  ?site ev:isComposedBy ?page.
  ?char ev:hasMetric ?metric.
  FILTER (?v >= 0.5)
}

More complex patterns can be devised for Web sites, based on this pattern and the set of patterns presented above for Web pages. For instance, combining this pattern with characteristic quality analysis, Web sites can be analyzed to find out which ones are entirely accessible for any audience characteristic above a certain threshold:

SELECT ?page
WHERE {
}

More complex patterns can be devised for Web sites, based on this pattern and the set of patterns presented above for Web pages. For instance, combining this pattern with characteristic quality analysis, Web sites can be analyzed to find out which ones are entirely accessible for any audience characteristic above a certain threshold:

SELECT ?page
WHERE {
}

A variant on this query pattern can be defined as verifying if the average metric value is above the threshold, for a given characteristic. This would unify Web pages, thus analyzing a Web site as a single entity. However, SPARQL does not provide aggregation functions out of the box. Therefore, some implementations have circumvented this issue through, e.g., the AVG function. Without this function each metric value should be aggregated outside the query pattern and an average value calculation should be performed, which influences its scalability. Hence, this pattern uses the AVG function accordingly:

SELECT AVG(?v)
WHERE {
  tx:totallyBlind ev:hasMetric ?metric.
}
?metric:isRelatedToDatatypeProperty ?prop.
)

Semantically Extracting Web Graph Partitions

While capturing information about the accessibility of single Web pages or Web sites has value, it is more interesting to analyze Web graphs as a whole. The set of query patterns presented in the previous Section can be adapted to grasp new knowledge about entire Web graphs. In this Section we present query patterns that afford the extraction of Web graph partitions according to accessibility criteria. Along the lines of the previous Section, all SPARQL patterns are applied to a set of dummy Web pages (e.g., http://example.com/a.html) or Web site (http://example.com), with the semantics of ev:Webpage and ev:Website, correspondingly; other instances that appear on queries are based on the examples presented in the previous Sections.

Reachability. The simplest information that can be obtained about a Web graph concerns its edges, i.e., the link structure. Edges are described through ev:linksTo property instances. The transitiveness of the ev:reaches property, based on ev:linksTo, allows the exploration of connectivity between Web pages (and between Web sites, as well). This query pattern will be used as the base support for extracting Web graph portions according to different accessibility semantics. Reachability can be a property queried between Web pages, e.g.:

ASK {
}

This notion can be extended to explore which Web pages can be reached from a starting point:

SELECT ?page
WHERE {
}

The opposite pattern, knowing which Web pages reach a specific ending point, can also be explored similarly:

SELECT ?page
WHERE {
}

Lastly, based on these queries, Web graph portions can be extracted according to their linking structures. For these patterns, we use the CONSTRUCT query form provided in SPARQL. The simplest graph portion extraction concerns finding out the linking structure reached from a specific starting Web page:

CONSTRUCT {
  ?page ev:linksTo ?otherPage
}
WHERE {
}

By generalizing this query pattern, the entire information about a particular Web graph portion can be extracted. While we could use the DESCRIBE query form, we opted to use CONSTRUCT since it is required to be supported in every SPARQL implementation. The query pattern is as follows:

CONSTRUCT {
  ?page ?prop ?value
}
WHERE {
}

Lastly, all of these patterns can be further extended towards a macroscopic level, i.e., not centered on Web pages per se, but on Web sites. It is important to understand graph connectivity at this level, e.g. whether a Web site directly links to another one:

ASK {
}

Based on this pattern, it might be relevant to understand what are the linking sources in such cases:

SELECT ?page
WHERE {
}

Expanding further, one is able to find out which Web sites link directly to a given Web site:

SELECT ?site
WHERE {
  ?site ev:isComposedBy ?page.
}

This type of pattern can be applied to all of the subsequent query patterns accordingly. For simplicity purposes, each one of the next query patterns is applied to Web pages.

**Common characteristics.** Based on the characteristics quality pattern for Web pages and Web sites, the same type of information can be acquired from entire Web graphs. Here, a quality threshold dictates which characteristics are above it in the entire Web graph:

SELECT ?char
WHERE {
  ?char ev:hasMetric ?metric.
  FILTER (?v >= 0.5)
}

Based on this query pattern, Web graphs can be partitioned according to characteristic-oriented quality thresholds, following the same rules presented above:

CONSTRUCT {
  ?page ?prop ?value
}
WHERE {
  ?char ev:hasMetric ?metric.
  FILTER (?value >= 0.5)
}

While this last pattern extracts the entire RDF graph, there are cases where just the corresponding Web graph structure (i.e., just the Web pages and linking structure) is extracted. In these cases the pattern can be easily adjusted as follows:
CONSTRUCT {  
}
WHERE {  
  ?char ev:hasMetric ?metric.  
  FILTER (?value >= 0.5)
}

Common audiences. The same type of query pattern can be applied to find out if a given Web graph is tailored to a specific audience:

ASK {  
  ?char ev:hasMetric ?metric.  
  FILTER (?value >= 0.5)
}

Based on this query pattern, the Web graph itself can be partitioned according to this specific semantics:

CONSTRUCT {  
  ?page ?prop ?value
}
WHERE {  
  ?char ev:hasMetric ?metric.  
  FILTER (?value >= 0.5)
}

Characteristic reachability. This query pattern has been devised to find out which Web pages can be reached from a starting point, while maintaining a quality level above a specific threshold for a given characteristic:

SELECT ?page
WHERE {
  tx:totallyBlind ev:hasMetric ?metric.  
  FILTER (?value >= 0.5)
}

However, the way this query pattern has been devised misses the intermediate Web pages that might not have the desired quality level for the selected characteristic. To mitigate this issue, all intermediate Web pages have to be verified accordingly:

SELECT ?otherPage
WHERE {
  tx:totallyBlind ev:hasMetric ?metric.  
  FILTER (?value >= 0.5 && ?value2 >= 0.5)
}

Accordingly, this pattern can be adapted to extract the corresponding Web graph portion. This is done by creating an RDF graph consisting of `ev:linksTo` derived triples, where both
end-points have to be reached from the starting point, as follows:

CONSTRUCT {
  ?page ev:linksTo ?otherPage
}
WHERE {
  tx:totallyBlind ev:hasMetric ?metric.
  FILTER (?value >= 0.5 && ?value2 >= 0.5)
}

This last version of the query pattern can be further adapted to find out just whether there are any Web pages that cannot be reached according to the devised semantics:

ASK {
  tx:totallyBlind ev:hasMetric ?metric.
  FILTER (?value < 0.5 && ?value2 < 0.5)
}

If one wants to know what Web pages are not reached through this method, the previous version of the query pattern can be further adapted. Please notice that this version of the pattern simply inverts the filter, in comparison with the second version of this query pattern:

SELECT DISTINCT ?page
WHERE {
  tx:totallyBlind ev:hasMetric ?metric.
  FILTER (?value < 0.5 && ?value2 < 0.5)
}

**Audience reachability.** As audiences are more closely representative of users (by aggregating characteristics), it is also important to study the graph reachability from this point of view. The simplest query pattern for audience reachability concerns finding out what Web pages are appropriate for a specific audience:

SELECT ?page
WHERE {
  ?char ev:hasMetric ?metric.
  FILTER (?value >= 0.5)
}

Like in characteristics reachability, one has to take into account that all Web pages in between must also have a quality level above the threshold that has been set. Accordingly, this query pattern must cope with this issue:
SELECT ?otherPage 
WHERE {
  ?char ev:hasMetric ?metric.
  FILTER (?value >= 0.5 && ?value2 >= 0.5)
}

This pattern version can be easily adapted towards extracting the corresponding Web graph partition:

CONSTRUCT {
  ?page ev:linksTo ?otherPage 
}
WHERE {
  ?char ev:hasMetric ?metric.
  FILTER (?value >= 0.5 && ?value2 >= 0.5)
}

It is also possible to build on this query pattern version to find out if there is any Web page that cannot be reached with at least the same quality level:

ASK {
  ?char ev:hasMetric ?metric.
  FILTER (?value < 0.5 && ?value2 < 0.5)
}

Likewise, we can also extract from the Web graph the set of Web pages that cannot be reached according to this semantics:

SELECT DISTINCT ?page 
WHERE {
  ?char ev:hasMetric ?metric.
  FILTER (?value < 0.5 && ?value2 < 0.5)
}

**Domain reachability.** Along the lines of the previous two patterns, it is important to find out what partitions of a Web graph are reached from a starting point for all audiences within an audience
domain, according to a previously set quality level threshold. The patterns for domain reachability follow closely the ones for characteristic and audience reachability. Therefore, we present a query pattern representative of the specific details for domain reachability. The following pattern affords the extraction of a Web graph partition for all the Web pages that are reachable from a starting point, based on a quality threshold:

CONSTRUCT {
} WHERE {
  ?char ev:hasMetric ?metric.
  FILTER (?value >= 0.5 && ?value2 >= 0.5)
}

This query pattern can be further converted to find out what are these audiences. This way, researchers can ask what are the specific audiences that limit reachability. This pattern is as follows:

SELECT ?audience
WHERE {
  ?char ev:hasMetric ?metric.
  FILTER (?value < 0.5 && ?value2 < 0.5)
}

Another interesting pattern for domain reachability concerns finding out whether an audience domain has any audience that limits the reachability property:

ASK {
  ?char ev:hasMetric ?metric.
  FILTER (?value < 0.5 && ?value2 < 0.5)
}

**Inward linking quality.** As explained before, one of the great powers of the Web resides on how its linking structure is perceived and navigated by users. One important aspect of this property concerns the quality of the Web graph from the perspective of how Web sites are linked to each other. This query pattern explores linking to a specific ending point, i.e., all Web pages that
Querying Web Accessibility Knowledge from Web Graphs

link to a target Web page. First, it is important to extract the graph partition composed by the Web pages that point to it:

CONSTRUCT {
  ?page ev:linksTo <http://example.com/a.html>
}
WHERE {
}

Based on this simple query, quality thresholds can be set according to one of the query patterns presented in the previous Section (i.e., patterns for Web pages and Web sites), e.g., for characteristics:

CONSTRUCT {
  ?page ev:linksTo <http://example.com/a.html>
}
WHERE {
  tx:colorBlind ev:hasMetric ?metric.
  FILTER (?v >= ?v2)
}

Another aspect that can be explored based on this last version of the query pattern concerns knowing whether the target Web page has better quality than the Web pages that point to it. This allows us to understand if the target Web page can be perceived as an accessibility haven on navigation tasks:

ASK {
  tx:colorBlind ev:hasMetric ?metric.
  FILTER (?v < ?v2)
}

Outward linking quality. Dually to the previous query pattern, it is also important to understand the linking quality by setting up an initial starting Web page and explore the Web pages that it links to. The type of queries in this pattern follows closely the previous set of patterns with small changes. For instance, the following query leverages the Web graph partition of the Web pages that are safe to navigate:

CONSTRUCT {
}
WHERE {

Querying Web Accessibility Knowledge from Web Graphs

tx:colorBlind ev:hasMetric ?metric.
FILTER (v? >= ?v2)
}

**Verticality.** It is a fact that the Web is partially tailored to specific accessibility situations, e.g., “accessible versions” of a Web site. This property can be explored by studying the verticality of Web graphs. For example, given two different characteristics and a quality threshold, there might be an overlap between which Web pages are accessible to both. The amount of Web pages in this situation is directly related to the verticality of their corresponding partitions. This is done through the following query pattern:

```
SELECT ?page
WHERE {
FILTER (?value1 >= 0.5 && ?value2 >= 0.5)
}
```

**FUTURE TRENDS**

The framework presented in this chapter is just one of the initial steps that can help understanding the impact of Web accessibility and Web standards on users, in a large scale (i.e., the whole Web) and with a fine-grained control over what aspects of Web accessibility and users are to be studied. We envision that semantic technologies can disrupt the way Web developers and designers think of accessibility and its social impact in the way users feed and consume information of the Web.

To grasp this knowledge, the framework we presented must be supported by its implementation and use in the analysis of large portions of the Web. Hence, we foresee that the following trends will help in this complex task:

- **Scalable architectures.** Building large scale Web accessibility observatories require scale-free approaches to crawl, store, process, and query the Web. We expect that with ongoing and future developments of scalable architectures that can cope with these type of tasks will help providing further insights on the influence that the Web’s structure poses on Web accessibility issues.

- **Graph visualization algorithms.** There is a need for visualize large quantities of data (e.g., billions of metadata of Web pages), to grasp Web accessibility knowledge from semantic queries over Web graphs. Even when intelligent ways of extracting information from Web graph accessibility data, coping with billions of Web pages is not trivial. New graph visualization techniques can help lowering the burden of finding the needle in the haystack, i.e., the relevant information about the impact of Web accessibility at a large scale.

- **Automated verification.** Experts verify usability and accessibility problems in a manual/guided fashion. Since this approach is scale-bounded, there is the need for new automated verification procedures. With the advance in this research field (most probably with the aid of semantic technologies), more information can be obtained about usability and accessibility problems of the Web at a large scale. Significant advances to this challenge include understanding better
how humans interact with computers, new models and theories for human psychology, as well as more pragmatic approaches such as statistical content analysis.

- **Metrics.** Accurate metrics provide better answers for finding the impact of Web accessibility implementation for all users. Having a base framework such as the one we presented in this chapter will help comparing metrics (and their corresponding application to Web graphs) and improve their accuracy.

- **Predictive and evolutionary models.** By having available smart models, the Web can be studied from predictive and evolutionary perspectives, opening the way to improving Web standards and Web accessibility assessment tools.

With advancements on these fronts, we foresee that the work described in this chapter can be put together within existing Web crawling, indexing and searching facilities with minor tweaks, forming an architecture for large scale Web accessibility assessments, as presented in Figure 9.

In this architecture the central aspect resides on the **Web accessibility results** repository, which should follow the metadata structures defined in this chapter. This repository holds all information about the accessibility semantics of the Web graph, as grasped by** Accessibility Spiders** (similar to Web crawler’s spiders) and an aggregating **Web accessibility evaluator** module. Through the **Query Interface**, and the query patterns described in this chapter, we envision that this architecture will facilitate on visualizing Web accessibility at a large scale. We believe that this will provide clues on how Web standards and accessibility recommendations should evolve in the future towards a universally accessible and usable Web.

**CONCLUSION**

In this chapter we have presented a semantic knowledge framework for Web accessibility. This framework supports the definition of Web graphs and their accessibility properties. Through a set of query patterns, we have described a way to mine Web graphs in order to understand how the Web can cope with end users’ intrinsic and transient characteristics, such as disabilities, interactive devices, etc.

We are currently developing ongoing work to implement this framework within the context of the architecture proposed in the previous Section in cooperation with the Portuguese Web Archive (PWA, n.d.) and apply it to study the entire Por-

*Figure 9. Architecture for large scale Web accessibility assessments*
tuguese Web (around 40 million Web pages). We believe that the set of query patterns presented in this chapter will help us to understand the shape of the Web in what respects to its Web accessibility properties. More specifically, it will allow us discovering which Web sites are more accessible, and to verify if Web sites created by non-experts have significant accessibility problems, in comparison to those created by experts.

REFERENCES


KEY TERMS AND DEFINITIONS

Accessibility: The ability to access. Often tied to people with disabilities (e.g., total blindness), accessibility strives 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).

Metric: A quantification procedure based on several criteria. In the context of this Chapter, metrics quantify accessibility based on different accessibility checkpoints.

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).

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.

Web Graph: A formal representation of the Web’s structure. Web pages are represented as the graph’s nodes, whereas hyperlinks are represented as its arcs. By representing the Web as a graph, traditional graph analysis algorithms can be applied.

LIST OF NAMESPACE PREFIX/URI MAPPING

1. earl: http://www.w3.org/ns/earl#