DOCTOR OF SCIENCES

Methodology for automating web usability and accessibility evaluation by guideline

Beirekdar, Abdo

Award date:
2004

Awarding institution:
University of Namur

Facultés Universitaires
Notre-Dame de la Paix

A Methodology for Automating Guideline Review of Web Sites

Abdo Beirekdar

Thesis submitted in fulfillment of the requirements for the degree of Doctor of Sciences
(Computer Science Option)

- August 30th, 2004 -

Director:	Professor M. Noirhomme-Fraiture
Co-director:	Professor J. Vanderdonckt, Université Catholique de Louvain, Belgium
Jury:	Professor F. Bodart
	Professor J.-L. Hainaut (President)
	Professor Ch. Kolski, Université de Valenciennes, France
	Professor Ph. Palanque, Université Paul Sabatier - Toulouse III, France

Institut d'Informatique
NAMUR

Annex B
Application Examples

1. Introduction

In this chapter we will apply the GDL on some Web guidelines (usability, accessibility, design rules) in order to evaluate the ability of the language to formulate guidelines of different complexity, different automation level, etc. These guidelines are selected for the following reasons:

- We specify some guidelines from non-established sources to show that we can cover a variety of them.
- We apply the GDL on a heuristics-based rule to show the flexibility of the approach.
- We specify some guidelines from broadly used sets of guidelines because our tool must support these sets (requirement of the tool). We selected guidelines fully supported by well known tools to show that we are at least able to do the same as these tools.

The specifications of the following guidelines use the GDL DTD that we presented in chapter 5. This DTD is detailed in Annex1:

- GDL1: "*Do not use more than 2-3 fonts*" [Nogier 2002]. This example demonstrates a GDL specification is its most simple form.
- GDL2: "*For general public applications, use the pallet of Web-safe colors*" [Nogier 2002]. This example demonstrates the use of SEQ constructed data type.
- GDL3: "*The Web page must have less than Three italicized body words*" [Ivory 2001]. This example demonstrates the use of a metrics-based rule.
- GDL4 (the famous WAI guideline 1): "*Provide equivalent alternatives to auditory and visual content*" [W3C 1999].
- GDL5: "*Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup*" [Section508]. This example demonstrates the use ScopeOR elements scope.

2. Eample1 (simple guideline)

The first example is a simple guideline about the preferred maximum number of font in Web pages: "*Do not use more than 2-3 fonts*" [Nogier 2002].

Step1: interpretation

This guidelines is concrete, but we provide a default interpretation: "*The number of fonts in a Web page must be inferior or equal to 3*".

Setp2: HTML elements

We need one element: {Font.face}. We estimate that this covers all the aspects related to the original guideline, therefore, the automation of the guideline evaluation is **theoretically total**.

Step3: Evaluation sets

S1={Font.face[Page]}

Setp4: Evaluation conditions

We need one direct condition:

OP1: **NumberOfInstances**(S1) → V1 (Integer)

Result=ANY → Continue to Op2

OP2: V1<3 → Boolean

Result=TRUE → Success

Result=FALSE → Error, “Vous avez plus de 3 polices dans la page”

As we saw in chapter 5, the operation NumberOfInstances is a predefined operation on evaluation sets and it returns the number of captured instances of the evaluation set.

The implementation of this condition is possible, therefore, the automation of the guideline evaluation is **practically total**.

Following is the GDL specification of the guideline:

```
<GDL_Specification>
  <Guideline    ID_Guideline="G2" Ergo_Criteria="Usability" GLTitle="Font family"
                ID_References="Nogier02"
                GLStatement="Ne pas utiliser plus de 2-3 polices de caractères"/>
  <Interpretation>
    <Context  ID_Context="Default" User="All" Platform="All" Environment="All"
              Target_Objects="Text font"/>
    <Interpreted_Guideline ID_InterpretedGL="Inter1"
          IGLStatement="Le nombre de  polices de caractères dans une page doit être inférieur à 3"/>
  <Formal_Guideline>
    <Evaluation_Structure>
      <HTML_Elements>
        <HTML_Element E_ID="Font" E_Tag="Font" E_Attribute="face"/>
      </HTML_Elements>
      <Evaluation_Sets>
        <Evaluation_Set S_ID="S1" Priority="1">
          <Set_Element E_ID="Font" Scope="Page"/>
        </Evaluation_Set>
      </Evaluation_Sets>
    </Evaluation_Structure>
    <Evaluation_Logic>
      <User_Values>
        <Basic_Values>
          <Basic_Value V_ID="MaxNumber" Value="3" Type="Int"/>
        </Basic_Values>
      </User_Values>
      <Evaluation_Conditions>
        <Direct_Condition DC_ID="C1" Set_ID="S1">
```

```
            <Operation Op_ID="Op1" Op_Symbol="NumberOfInstances" Return_Type="Int">
              <Argument Arg_Type="SET" Arg_Value="S1" Pos="1"/>
              <Action Result="Any" What="Jump" Where="Op2"/>
            </Operation>
            <Operation Op_ID="Op2" Op_Symbol="LESS" Return_Type="Boolean">
              <Argument Arg_Type="Op" Arg_Value="Op1"/>
              <Argument Arg_Type="Val" Arg_Value="MaxNumber"/>
              <Action Result="false" What="Error"
                      Why="Guideline violated: Number of fonts is greater than 3"/>
              <Action Result="true" What="Success"/>
            </Operation>
          </Direct_Condition>
        </Evaluation_Conditions>
      </Evaluation_Logic>
    </Formal_Guideline>
    </Interpretation>
</GDL_Specification>
```

3. Example2 (simple guideline)

The second guideline is a simple one belonging also to [Nogier 2002]: "*For general public applications, use the pallet of Web-safe colors*".

Web safe colors are those for which the hexadecimal value is a combination of 00, 33, 66, 99, CC or FF. For example, the red #FF0033 and the green #33FF66 are safe colors [Nogier 2002].

Step1: interpretation

Having the definition of web safe colors, and targeting all contexts of use but color of text only (no images, animations, etc.), we can re-formulate the guideline as: "*Pour des applications grand public, utiliser des couleurs dont la valeur de chaque composant R, G, B est une des valeurs 0, 51, 102, 153, 204 ou 255*". We target text only because, until now, we cannot capture colors of other objects with HTML only. We replaced hexadecimal values by integer values because the functions getRed, getGreen and getBlue of a color returns integer values [0..255]. This is an example of applying the first criteria of passing from the guideline to the interpretation: use HTML vocabularies.

Step2: HTML elements

As we deal with text only, we can use the following HTML elements: {Body.bgcolor, Body.text, Body.link, Body.alink, Body.vlink, Font.color, Table.bgcolor, TH.bgcolor, TR.bgcolor, TD.bgcolor}.

We estimate that this covers all the aspects related to the original guideline, therefore, the automation of the guideline evaluation is **theoretically total**.

Step3: Evaluation sets

The easiest way is to evaluate every color in the page without examining if it is really used or not (excluded by other color or not). In this case, we will examine colors separately, thus, every HTML object forms an evaluation set:

S1={Body.bgcolor[Page]}

S2={Body.text[Page]}

S3={Body.link[Page]}

S4={Body.vlink[Page]}
S5={Body.alink[Page]}
S6={Font.color[Page]}
S7={Table.bgcolor[Page]}
S8={TR.bgcolor[Page]}
S9={TH.bgcolor[Page]}
S10={TD.bgcolo[Page]}

Setp4: Evaluation conditions

To evaluate the guideline, we will need to apply the following evaluation logic on a color *C*:

RGBs=SET[00, 33, 66, 99, CC, FF]
Red component of C belongs to RGBs AND
Green component of C belongs to RGBs AND
Blue component of C belongs to RGBs

Clearly, we will define a Meta condition for all the cases:

Meta_Vars:

Color: HEX
SafeRGBs: SET[INTEGER]= SET [0, 51, 102, 153, 204, 255]

Meta_Model

OP1: **getRed**(Color)→ Integer
Result=ANY → Continue to Op2
OP2: **IN**(Color, SafeRGBs) → Boolean
Result=TRUE → Continue to Op3
Result=FALSE → Error, “Le composant ROUGE n’est pas Web safe.”

OP3: **getGreen**(Color) → Integer
Result=ANY → Continue to Op4
OP4: **IN**(Color, SafeRGBs) → Boolean
Result=TRUE → Continue to Op5
Result=FALSE → Error, “Le composant VERT n’est pas Web safe.”

OP5: **getBlue**(Color) → Integer
Result=ANY → Continue to Op6
OP6: **IN**(Color, SafeRGBs) → Boolean
Result=TRUE → Success
Result=FALSE → Error, “Le composant BLEU n’est pas Web safe.”

Notice in the Meta model the flexibility in generating customized error messages.

The implementation of this condition is possible; therefore, the automation of the guideline evaluation is **practically total**.

Following is the GDL specification of the guideline. We will notice that the specification is relatively long because there are a lot of repeated parts:

```
<GDL_Specification>
  <Guideline   ID_Guideline="G1"
               GLStatement="Pour des applications grand public, utiliser la palette de
                            couleur web-safe"
               ID_References="Nogier02"/>
  <Interpretation>
    <Context   ID_Context="Con1"
               User="All" Platform="All" Environment="All" Target_Objects="text color"/>
    <Interpreted_Guideline
               ID_InterpretedGL="Inter1"
               IGLTitle="safe colors"
               IGLStatement="Pour des applications grand public, utiliser des couleurs dont la valeur
                            de chaque composant R, G, B est une des valeurs 0, 51, 102, 153, 204
                            ou 255"/>
    <Formal_Guideline>
      <Evaluation_Structure>
        <HTML_Elements>
          <HTML_Element E_ID="Bdbg" E_Tag="Body" E_Attribute="bgcolor"/>
          <HTML_Element E_ID="Bdtxt" E_Tag="Body" E_Attribute="text"/>
          <HTML_Element E_ID="link" E_Tag="Body" E_Attribute="link"/>
          <HTML_Element E_ID="alink" E_Tag="Body" E_Attribute="alink"/>
          <HTML_Element E_ID="vlink" E_Tag="Body" E_Attribute="vlink"/>
          <HTML_Element E_ID="Font" E_Tag="Font" E_Attribute="color"/>
          <HTML_Element E_ID="Table" E_Tag="Table" E_Attribute="bgcolor"/>
          <HTML_Element E_ID="Tr" E_Tag="Tr" E_Attribute="bgcolor"/>
          <HTML_Element E_ID="Th" E_Tag="Th" E_Attribute="bgcolor"/>
          <HTML_Element E_ID="Td" E_Tag="Td" E_Attribute="bgcolor"/>
        </HTML_Elements>
        <Evaluation_Sets>
          <Evaluation_Set S_ID="S1" Priority="1">
            <Set_Element E_ID="Bdbg" Scope="Page"/>
          </Evaluation_Set>
          <Evaluation_Set S_ID="S2" Priority="1">
            <Set_Element E_ID="Bdtxt" Scope="Page"/>
          </Evaluation_Set>
          <Evaluation_Set S_ID="S3" Priority="1">
            <Set_Element E_ID="link" Scope="Page"/>
          </Evaluation_Set>
          <Evaluation_Set S_ID="S4" Priority="1">
            <Set_Element E_ID="alink" Scope="Page"/>
          </Evaluation_Set>
          <Evaluation_Set S_ID="S5" Priority="1">
            <Set_Element E_ID="vlink" Scope="Page"/>
          </Evaluation_Set>
          <Evaluation_Set S_ID="S6" Priority="1">
            <Set_Element E_ID="Font" Scope="Page"/>
          </Evaluation_Set>
          <Evaluation_Set S_ID="S7" Priority="1">
            <Set_Element E_ID="Table" Scope="Page"/>
          </Evaluation_Set>
          <Evaluation_Set S_ID="S8" Priority="1">
            <Set_Element E_ID="Tr" Scope="Page"/>
          </Evaluation_Set>
          <Evaluation_Set S_ID="S9" Priority="1">
            <Set_Element E_ID="Th" Scope="Page"/>
          </Evaluation_Set>
          <Evaluation_Set S_ID="S10" Priority="1">
            <Set_Element E_ID="Td" Scope="Page"/>
          </Evaluation_Set>
        </Evaluation_Sets>
      </Evaluation_Structure>
```

```
<Evaluation_Logic>
  <User_Values>
    <Basic_Values>
      <Basic_Value V_ID="v1" Value="0" Type="Int"/>
      <Basic_Value V_ID="v2" Value="51" Type="Int"/>
      <Basic_Value V_ID="v3" Value="102" Type="Int"/>
      <Basic_Value V_ID="v4" Value="153" Type="Int"/>
      <Basic_Value V_ID="v5" Value="204" Type="Int"/>
      <Basic_Value V_ID="v6" Value="255" Type="Int"/>
    </Basic_Values>
    <Constructed_Values>
      <Seq_Value V_ID="RGBs">
        <Id_Value ID_Ref="v1"/>
        <Id_Value ID_Ref="v2"/>
        <Id_Value ID_Ref="v3"/>
        <Id_Value ID_Ref="v4"/>
        <Id_Value ID_Ref="v5"/>
        <Id_Value ID_Ref="v6"/>
      </Seq_Value>
    </Constructed_Values>
  </User_Values>
  <Evaluation_Conditions>
    <Meta_Condition MC_ID="mc1">
      <Meta_Vars>
        <Meta_Var Name="color" Type="Hex"/>
      </Meta_Vars>
      <Model>
        <Operation Op_ID="op1" Op_Symbol="getRed" Return_Type="Int">
          <Argument Arg_Type="Var" Arg_Value="color"/>
          <Action Result="ANY" What="Jump" Where="op2"/>
        </Operation>
        <Operation Op_ID="op2" Op_Symbol="IN" Return_Type="Boolean">
          <Argument Arg_Type="Op" Arg_Value="op1"/>
          <Argument Arg_Type="Val" Arg_Value="RGBs"/>
          <Action Result="TRUE" What="Jump" Where="op3"/>
          <Action Result="False" What="Error" Why="Red component is not Web safe"/>
        </Operation>
        <Operation Op_ID="op3" Op_Symbol="getGreen" Return_Type="Int">
          <Argument Arg_Type="Var" Arg_Value="color"/>
          <Action Result="ANY" What="Jump" Where="op4"/>
        </Operation>
        <Operation Op_ID="op4" Op_Symbol="IN" Return_Type="Boolean">
          <Argument Arg_Type="Op" Arg_Value="op3"/>
          <Argument Arg_Type="Val" Arg_Value="RGBs"/>
          <Action Result="TRUE" What="Jump" Where="op5"/>
          <Action Result="False" What="Error" Why="Green component is not Web safe"/>
        </Operation>
        <Operation Op_ID="op5" Op_Symbol="getBlue" Return_Type="Int">
          <Argument Arg_Type="Var" Arg_Value="color"/>
          <Action Result="ANY" What="Jump" Where="op6"/>
        </Operation>
        <Operation Op_ID="op6" Op_Symbol="IN" Return_Type="Boolean">
          <Argument Arg_Type="Op" Arg_Value="op5"/>
          <Argument Arg_Type="Val" Arg_Value="RGBs"/>
          <Action Result="TRUE" What="Success"/>
          <Action Result="False" What="Error" Why="Blue component is not Web safe"/>
        </Operation>
      </Model>
    </Meta_Condition>
    <Mapped_Condition Set_ID="S1" Meta_ID="mc1">
      <Meta_Mapping Meta="color" Instance="Bdbg"/>
    </Mapped_Condition>
    <Mapped_Condition Set_ID="S2" Meta_ID="mc1">
      <Meta_Mapping Meta="color" Instance="Bdtxt"/>
    </Mapped_Condition>
      ……
      ……
    <Mapped_Condition Set_ID="S10" Meta_ID="mc1">
```

```
                <Meta_Mapping Meta="color" Instance="Td"/>
              </Mapped_Condition>
            </Evaluation_Conditions>
          </Evaluation_Logic>
        </Formal_Guideline>
    </Interpretation>
</GDL_Specification>
```

4. Example3 (metrics-based rule)

We said that transform some ergonomic knowledge like the metrics of Ivory [2001] into guideline-like form and add it to the list of guidelines. Following is the specification of the example introduced in chapter 4.

IF ((Italicized Body Word Count is not missing AND
(Italicized Body Word Count > 2.5)))
Class = Poor

Step1: default Interpretation

We will interpret the guideline as: *"The Web page must have less than Three italicized body words"*. We consider that this interpretation is for the default context: all users, all plateforms, standard environment, text.

Step2:HTML Elements

HTML provides a single element to italicize text: {I}

As argued in chapter 4, the guideline speaks about words count, and the tag I is not sufficient to find this information in a Web page, therefore, we will add the special HTML element: BodyText.

We estimate that this covers all the aspects related to the original guideline, therefore, the automation of the guideline evaluation is **theoretically total**.

Step3: Evaluation sets

We will have a single (atomic) set:

S1={I[Page], BodyText[I]}.

Step4: Evaluation conditions

We have one direct condition composed of one operation:

OP1: (**Length**(BodyText)) → V1

Result=ANY → Continue to Op2

OP2: (V1 **<** 3)

Result=True → "Good page"

Result=False → "Poor page because it has more that 2.5 italicized words".

The implementation of this condition is possible, therefore, the automation of the guideline evaluation is **practically total**.

Following is the GDL specification of this rule:

<GDL_Specification>

```
<Guideline    ID_Guideline="G3"
              GLTitle="Heuristic Rule for good Web pages" ID_References="Ivory2001 "
              GLStatement="if the page has no italicized text or the count of italicized words is less that
                           2.5, then the page is good"/>
<Interpretation>
  <Context   ID_Context="Default" User="All" Platform="All" Environment="Standard"
             Target_Objects="Text"/>
  <Interpreted_Guideline
             ID_InterpretedGL="Inter1"
              IGLStatement="The Web page must have less than Three italicized body words"/>
  <Formal_Guideline>
    <Evaluation_Structure>
      <HTML_Elements>
        <HTML_Element E_ID="itallic" E_Tag="I"/>
        <HTML_Element E_ID="text" E_Tag="BodyText"/>
      </HTML_Elements>
      <Evaluation_Sets>
        <Evaluation_Set S_ID="S1" Name="Italic text" Priority="1"
                        Description="find the number of italic words">
          <Set_Element E_ID="itallic" Scope="Page"/>
          <Set_Element E_ID="text" Scope="italic"/>
        </Evaluation_Set>
      </Evaluation_Sets>
    </Evaluation_Structure>
    <Evaluation_Logic>
      <User_Values>
        <Basic_Values>
          <Basic_Value V_ID="MaxItalicWord" Value="3" Type="Int"/>
        </Basic_Values>
      </User_Values>
      <Evaluation_Conditions>
        <Direct_Condition DC_ID="Dc1" Set_ID="S1">
          <Operation Op_ID="Op1" Op_Symbol="NumberOfInstances" Return_Type="Int">
            <Argument Arg_Type="SET" Arg_Value="S1"/>
            <Action Result="ANY" What="Jump" Where="Op1"/>
          </Operation>
          <Operation Op_ID="Op2" Op_Symbol="LESS" Return_Type="Boolean">
            <Argument Arg_Type="Op" Arg_Value="Op1"/>
            <Argument Arg_Type="Val" Arg_Value="MaxItalicWord"/>
            <Action    Result="False" What="Error"
                       Why="Rule violation: the page has more that 3 italicized words"/>
            <Action    Result="True" What="Success"/>
          </Operation>
        </Direct_Condition>
      </Evaluation_Conditions>
    </Evaluation_Logic>
  </Formal_Guideline>
</Interpretation>
</GDL_Specification>
```

5.Example 4 (WAI guideline)

In this section we will specify one of the most supported WAI guidelines in existing automatic evaluation tools, the guideline 1: "*Provide equivalent alternatives to auditory and visual content*" [W3C 1999].

The WCAG1.0 [W3C 1999] explain how to evaluate this guideline by providing the list of the following checkpoints:

- Provide a text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content). This includes: images, graphical representations of text (including symbols), image map regions, animations (e.g., animated GIFs), applets and programmatic objects, ascii art, frames,

scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video. [Priority 1].

- Provide redundant text links for each active region of a server-side image map. [Priority 1].
- 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. [Priority 1].
- 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.
- Until user agents render text equivalents for client-side image map links, provide redundant text links for each active region of a client-side image map. [Priority 3].

Step1: Interpretation

Practically, only checkpoints 1, 2, and 5 can be automated, whereas checkpoints 3 and 4 need the intervention of an evaluator to check them because, to our knowledge, HTML does not provide any mean to deal with them.

Therefore, the interpretation context will be limited to the objects of checkpoints 1, 2, and 5. In this context, we give the interpretation: "*Provide text equivalent for every non-text element and provide redundant text links for each active region of a server-side or a client-side image map*".

Step2: HTML elements

If we look for available HTML support for the objects mentioned in the interpretation, we will find:

- IMG.alt: provides text alternative for images. This includes images, graphical representations of text, image map regions, images used as list bullets, spacers, and graphical buttons.
- Area.alt: provides text alternative for client-side images.
- INPUT.alt, INPUT.type: a graphical form submit button created with INPUT, type="image" creates a type of server-side image map. Therefore, we examine the element INPUT.alt only if the element INPUT.type has the value "image".
- Applet.alt: provides text alternative for applets.
- Object.type: Object element includes images or multimedia content if its type attribute has the values "image/…" or "video/…". There is no alt attribute for the tag Object, but we can add the text alternative as a content of the Object. In this situation, it is not possible to decide about the semantics of the content, therefore, we will use this object to generate warning messages if we encounter it with one of the above types.

Thus, the list of HTML elements is {IMG.alt, Area.alt, Applet.alt, INPUT.type, INPUT.alt, Object.type}. We can see that this list does not cover all the aspects

related to the original guideline; therefore, the automation of the guideline evaluation is **theoretically partial**.

Step3: Evaluation sets

According to the semantics of identified HTML elements, we group them as follows:

- S1={IMG[Page], IMG.alt[IMG]}: this set allows the checking of the presence of alt attribute for the images at different positions (simple image, client-side image, button image, etc.). We need the IMG tag alone to capture it when it has no alt attribute.
- S2={Applet[Page], Applet.alt[Applet]}: this set allows the checking of the presence of alt attribute for applets. We need the Applet tag alone to capture it when it has no alt attribute.
- S3={Area[Page], Area.alt[Applet]}: this set allows the checking of the presence of alt attribute for client-side image maps.
- S4={INPUT.type[Page], INPUT.alt[INPUT.type]}: if the type attribute indicates the presence of an image, we will check the presence of the alt attribute.
- S5={Object.type[Page]}: if the type attribute indicates the presence of an image, we will generate warning message asking the evaluator to check the page manually.

Step4: Evaluation conditions

By examining the evaluation sets we find that we can define a Meta condition for S1, S2, and S3, and two direct conditions for S4 and S5.

Meta condition for S1, S2, and S3

Meta_Vars:

Alt: setElem

SetElem indicates that the variable represents a set element.

Meta_Model:

Op1: **setInstanceHasElement**(alt) → Boolean

Result=True → Success

Result=False → Error, "Text equivalent is missing"

Notice that the error message is not very precise about what kind of element we are speaking about: image? Applet? Image map?. This is a shortcoming of using a Meta condition.

Mapped conditions:

S1: Meta=alt → Concrete=IMG.alt

S2: Meta=alt → Concrete=Applet.alt

S3: Meta=alt → Concrete=Area.alt

Notice that we did not use the elements IMG, Applet, and Area because there role is to enable the capturing of their instances in the evaluated Web page.

User values:

imageDataType="image"

videoDataType="video"

mutimediaTypes=SET[imageDataType, videoDataType]

Direct condition for S4:

Op1: INPUT.type **IN** multimediaTypes → Boolean

Result=True → Jump to Op2

Op2: **setInstanceHasElement**(INPUT.alt) → Boolean

Result=True → Success

Result=False → Error, "Text equivalent for button image is missing"

Direct condition for S5:

Op1: Object.type **IN** multimediaTypes → Boolean

Result=True → WarnStop, "Warning: verify that the object has text equivalent alternative in its content".

Following is the GDL specification of this guideline:

```
<GDL_Specification>
  <Guideline   ID_Guideline="G4"
               GLStatement="Provide equivalent alternatives to auditory and visual content"
               ID_References="WCAG1.0" Ergo_Criteria="Accessibility"/>
  <Interpretation>
    <Context ID_Context="Default" User="All" Platform="All" Environment="All"
                                  Target_Objects="images"/>
    <Interpreted_Guideline ID_InterpretedGL="Inter1"
          IGLStatement="Provide text equivalent for every non-text element and provide redundant text
                        links for each active region of a server-side or a client-side image map"/>
    <Formal_Guideline>
      <Evaluation_Structure>
        <HTML_Elements>
          <HTML_Element E_ID="Img" E_Tag="IMG"/>
          <HTML_Element E_ID="imgAlt" E_Tag="IMG" E_Attribute="alt"/>
          <HTML_Element E_ID="Applet" E_Tag="APPLET"/>
          <HTML_Element E_ID="appAlt" E_Tag="Applet" E_Attribute="alt"/>
          <HTML_Element E_ID="Area" E_Tag="AREA"/>
          <HTML_Element E_ID="areaAlt" E_Tag="AREA" E_Attribute="alt"/>
          <HTML_Element E_ID="InputType" E_Tag="INPUT" E_Attribute="type"/>
          <HTML_Element E_ID="inputAlt" E_Tag="INPUT" E_Attribute="alt"/>
          <HTML_Element E_ID="ObjectType" E_Tag="OBJECT" E_Attribute="type"/>
        </HTML_Elements>
        <Evaluation_Sets>
          <Evaluation_Set S_ID="S1" Priority="1">
            <Set_Element E_ID="Img" Scope="Page"/>
            <Set_Element E_ID="imgAlt" Scope="IDREF"/>
          </Evaluation_Set>
          <Evaluation_Set S_ID="S2" Priority="1">
            <Set_Element E_ID="Applet" Scope="Page"/>
```

```
        <Set_Element E_ID="appAlt" Scope="IDREF"/>
      </Evaluation_Set>
      <Evaluation_Set S_ID="S3" Priority="1">
        <Set_Element E_ID="Area" Scope="Page"/>
        <Set_Element E_ID="areaAlt" Scope="IDREF"/>
      </Evaluation_Set>
      <Evaluation_Set S_ID="S4" Priority="1">
        <Set_Element E_ID="InputType" Scope="Page"/>
        <Set_Element E_ID="inputAlt" Scope="IDREF"/>
      </Evaluation_Set>
      <Evaluation_Set S_ID="S5" Priority="2">
        <Set_Element E_ID="ObjectType" Scope="Page"/>
      </Evaluation_Set>
    </Evaluation_Sets>
  </Evaluation_Structure>
  <Evaluation_Logic>
    <User_Values>
      <Basic_Values>
        <Basic_Value V_ID="imageDataType" Value="image" Type="Str"/>
        <Basic_Value V_ID="videoDataType" Value="video" Type="Str"/>
      </Basic_Values>
      <Constructed_Values>
        <Seq_Value V_ID="multimediaDataTypes">
          <Id_Value ID_Ref="imageDataType"/>
          <Id_Value ID_Ref="videoDataType"/>
        </Seq_Value>
      </Constructed_Values>
    </User_Values>
    <Evaluation_Conditions>
      <Meta_Condition MC_ID="mc1">
        <Meta_Vars>
          <Meta_Var Name="alt" Type="SetElem"/>
        </Meta_Vars>
        <Model>
          <Operation Op_ID="op1" Op_Symbol="setInstanceHasElement"
                                        Return_Type="Boolean">
            <Argument Arg_Type="Var" Arg_Value="alt"/>
            <Action Result="False" What="Error" Why="Text equivalent is missing"/>
          </Operation>
        </Model>
      </Meta_Condition>
      <Direct_Condition DC_ID="dc1" Set_ID="S4">
        <Operation Op_ID="op11" Op_Symbol="IN" Return_Type="Boolean">
          <Argument Arg_Type="SetE" Arg_Value="InputType"/>
          <Argument Arg_Type="Val" Arg_Value="multimediaDataTypes"/>
          <Action Result="True" What="Jump" Where="Op12"/>
          <Action Result="False" What="Success"/>
        </Operation>
        <Operation Op_ID="Op12" Op_Symbol="setInstanceHasElement"
                                        Return_Type="Boolean">
          <Argument Arg_Type="SetE" Arg_Value="inputAlt"/>
          <Action    Result="False" What="Error"
                     Why="Text equivalent for button image is missing"/>
        </Operation>
      </Direct_Condition>
      <Direct_Condition DC_ID="dc2" Set_ID="S5">
        <Operation Op_ID="op21" Op_Symbol="IN" Return_Type="Boolean">
          <Argument Arg_Type="SetE" Arg_Value="ObjectType"/>
          <Action    Result="True" What="WarnStop"
                     Why="Warning: verify that the object has text equivalent alternative in its
                          content"/>
        </Operation>
      </Direct_Condition>
      <Mapped_Condition Set_ID="S1" Meta_ID="mc1">
        <Meta_Mapping Meta="alt" Instance="imgAlt"/>
      </Mapped_Condition>
      <Mapped_Condition Set_ID="S2" Meta_ID="mc1">
        <Meta_Mapping Meta="alt" Instance="areaAlt"/>
```

```
            </Mapped_Condition>
            <Mapped_Condition Set_ID="S3" Meta_ID="mc1">
              <Meta_Mapping Meta="alt" Instance="appAlt"/>
            </Mapped_Condition>
          </Evaluation_Conditions>
        </Evaluation_Logic>
      </Formal_Guideline>
    </Interpretation>
</GDL_Specification>
```

6. Example 5 (Section508 guideline)

Here we re-present the guideline: "*Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup*".

Step1: Interpretation

It is very difficult to practically cover all possibilities to provide alternative means of conveyance for colored information. We will limit our interpretation to textual information and to some markup alternatives only.

HTML provides many possibilities to distinguish a piece of text that we want to highlight:

- **TT:** Renders as teletype or monospaced text.
- **I:** Renders as italic text style.
- **B:** Renders as bold text style.
- **BIG:** Renders text in a "large" font.
- **SMALL:** Renders text in a "small" font.
- **STRIKE** and **S:** Render strike-through style text.
- **U:** Renders underlined text.
- **EM:** Indicates emphasis.
- **STRONG:** Indicates stronger emphasis.
- **CITE:** Contains a citation or a reference to other sources.
- **DFN:** Indicates that this is the defining instance of the enclosed term.
- **CODE:** Designates a fragment of computer code.
- **SAMP:** Designates sample output from programs, scripts, etc.
- **KBD:** Indicates text to be entered by the user.
- **VAR:** Indicates an instance of a variable or program argument.
- **ABBR:** Indicates an abbreviated form (e.g., WWW, HTTP, URI, Mass., etc.).
- **ACRONYM:** Indicates an acronym (e.g., WAC, radar, etc.).
- Etc.

The use of all these possibilities is the same: add the feature to the colored text. We need at least one of these possibilities to consider that the guideline is respected. For example, the following HTML code pieces respect our guideline:

```
<Body bgcolor=White>
   <Font color=Green><STRONG>
```

```
      Hello, I am STRONG
   </STRONG></Font>
</Body>
```

And

```
<Body bgcolor=White>
   <Font color=Green><STRONG><I>
      Hello, I am italic and STRONG
   <I></STRONG></Font>
</Body>
```

Therefore, we will use the following interpretation: "*all information conveyed with color must also be highlighted with at least one of these possibilities: Italic, Bold, and BIG*". The interpretation context is: all users, all platforms, all environments, textual colored information only, {italic, bold, BIG} alternatives.

Step2: HTML elements

According to the above interpretation, in addition to Italic (I), Bold (B), and BIG tags we will need HTML elements controlling the color of textual content: {Body.text, Body.bgcolor, Body.link, Body.vlink, Body.alink, Font.color, Table.bgcolor, TH.bgcolor, TD.bgcolor, TR.bgcolor}.

Let us see if we really need all these elements:

If we consider that text links (ex. this is a like to FUNDP) are usually automatically underlined, we can ignore them in our evaluation. Therefore, the considered HTML elements are: {Body.text, Body.bgcolor, Font.color, Table.bgcolor, TH.bgcolor, TD.bgcolor, TR.bgcolor, I, B, BIG}.

Step3: Evaluation sets

At first thought, we will need evaluation sets to cover the different positions of colored text as we explained in our example in chapters 4 and 5, and we will add the highlighting elements to each of them. We had the following sets:

- S1={Body.bgcolor[Page], Body.text[Page]}.
- S2={Body.bgcolor[Page], Font.color[Body.bgcolor]}.
- S3={Table.bgcolor[Page], Font.color[Table.bgcolor]}.
- S4={TR.bgcolor[Page], Font.color[TR.bgcolor]}.
- S5={TH.bgcolor[Page], Font.color[TH.bgcolor]}.
- S6={TD.bgcolor[Page], Font.color[TD.bgcolor]}.
- S7={Body.text[Page], Table.bgcolor[Body.text]}.
- S8={Body.text[Page], TR.bgcolor[Body.text]}.
- S9={Body.text[Page], TH.bgcolor[Body.text]}.
- S10={Body.text[Page], TD.bgcolor[Body.text]}.

However, S1 is not useful here because it controls the text of all over a Web page and not only a piece of it. Therefore, we estimate that we can ignore it.

In addition, we know that we can control the color of the text inside tables and table cells with the combinations {Body.text, Table.bgcolor}, {Body.text, TR.bgcolor}, {Body.text, TH.bgcolor}, and {Body.text, TD.bgcolor}. In these cases, we can consider that colored information is also conveyed by being enclosed inside a table or a table cell. Therefore, we estimate that we can ignore the sets S7..S10.

The remaining sets are:

- S2={Body.bgcolor[Page], Font.color[Body.bgcolor]}.
- S3={Table.bgcolor[Page], Font.color[Table.bgcolor]}.
- S4={TR.bgcolor[Page], Font.color[TR.bgcolor]}.
- S5={TH.bgcolor[Page], Font.color[TH.bgcolor]}.
- S6={TD.bgcolor[Page], Font.color[TD.bgcolor]}.

The common element among all these sets is Font.color. In fact, it is the element that indicates the presence of a colored piece of text whether it is inside a table, or a table cell, or not.

By conclusion of the above analysis, all what we need is the Font.color element!!! When we encounter it, we know that the text between <Font color=…> and </Font> is differentiated from the surrounding text by having a different color. Therefore, we will merge all the above sets in one set:

S1={Font.color[Page]}

By adding the selected highlighting possibilities to it:

S1={Font.color[Page OR I OR B OR BIG],

I [Page OR Font.color],

B [Page OR Font.color],

BIG [Page OR Font.color]}

Where the scopeOR is use because in HTML, <Font color=???><I>…..</I></Font> and <I><Font color=???>…..</Font></I> are equivalent.

Step4: Evaluation conditions

According to the specified evaluation set, the parser of Web pages will capture instances of the following forms:

- {Font.color, null, null, null}: instance of colored text without alternative way of conveyance. This happens when the parser encounters HTML code like <Font color=???>…..</Font>. This case is a violation of the guideline.
- {Font.color, I (and/or B and/or BIG)}: instance of colored text with at least one alternative way of conveyance. This happens when the parser encounters HTML code like <Font color=???><I>…..</I></Font>. This case is OK.

- {I (and/or B and/or BIG), Font.color}: second form of instance of colored text with at least one alternative way of conveyance. This happens when the parser encounters HTML code like <I><Font color=???>…..</Font></I>. This case is OK.
- {I (and/or B and/or BIG), null}: This happens when the parser encounters HTML code like <I>…..</I>. This case is irrelevant because there is no colored information.

Therefore, the direct evaluation condition will be the following:

IF (Font.color**<>**null) **THEN**
  IF (I=null **AND** B=null **AND** BIG=null) **THEN**
    Error
  ELSE:
    Success

The condition is decomposed into the following operations:

Op1: **setInstanceHasElement**(Font.color) → Boolean
Result=True → Continue to Op2
Op2: **setInstanceHasElement**(I) → Boolean
Result=True → Success, “colored text is also italic”
Result=False → Continue to Op3
Op3: **setInstanceHasElement**(B) → Boolean
Result=True → Success, “colored text is also bold”
Result=False → Continue to Op4
Op4: **setInstanceHasElement**(BIG) → Boolean
Result=True → Success, “colored text is also BIG”
Result=False → Error, “Colored text does not have alternative possibility of conveyance”

Following is the GDL specification of the guideline:

```
<GDL_Specification>
  <Guideline   ID_Guideline="G5" Ergo_Criteria="Accessibility"
               ID_References="section508"
               GLStatement="Web pages shall be designed so that all information conveyed with color
                            is also available without color, for example from context or markup"/>
  <Interpretation>
    <Context  ID_Context="Default"  User="All"  Platform="All"  Environment="Colored monitor"
              Target_Objects="text"/>
    <Interpreted_Guideline
              ID_InterpretedGL="Inter1" IGLTitle="Markup conveyance"
              IGLStatement="all information conveyed with color must also be highlighted with at least
                            one of these possibilities: Italic, Bold, and BIG"/>
    <Formal_Guideline>
      <Evaluation_Structure>
        <HTML_Elements>
          <HTML_Element E_ID="fColor" E_Tag="Font" E_Attribute="color"/>
          <HTML_Element E_ID="i" E_Tag="I"/>
          <HTML_Element E_ID="b" E_Tag="B"/>
          <HTML_Element E_ID="Big" E_Tag="BIG"/>
```

```
        </HTML_Elements>
        <Evaluation_Sets>
          <Evaluation_Set S_ID="S1" Name="ColoredAlternative" Priority="1">
            <Set_Element E_ID="fColor"  ScopeOR="Page i b Big"/>
            <Set_Element E_ID="i" ScopeOR="Page fColor"/>
            <Set_Element E_ID="b" ScopeOR="Page fColor"/>
            <Set_Element E_ID="Big" ScopeOR="Page fColor"/>
          </Evaluation_Set>
        </Evaluation_Sets>
      </Evaluation_Structure>
      <Evaluation_Logic>
        <Evaluation_Conditions>
          <Meta_Condition MC_ID="C1">
            <Meta_Vars>
              <Meta_Var Name="fg" Type="SetElem"/>
              <Meta_Var Name="italic" Type="SetElem"/>
              <Meta_Var Name="bold" Type="SetElem"/>
              <Meta_Var Name="big" Type="SetElem"/>
            </Meta_Vars>
            <Model>
              <Operation Op_ID="Op1" Op_Symbol="setInstanceHasElement"
                         Return_Type="Boolean">
                <Argument Arg_Type="Var" Arg_Value="fg"/>
                <Action Result="true" What="Jump" Where="Op2"/>
              </Operation>
              <Operation Op_ID="Op2" Op_Symbol="setInstanceHasElement"
                         Return_Type="Boolean">
                <Argument Arg_Type="Var" Arg_Value="bold"/>
                <Action Result="true" What="Success" Why="colored text is also bold"/>
                <Action Result="false" What="Jump" Why="Op3"/>
              </Operation>
              <Operation Op_ID="Op3" Op_Symbol="setInstanceHasElement"
                         Return_Type="Boolean">
                <Argument Arg_Type="Var" Arg_Value="italic"/>
                <Action Result="true" What="Success" Why="colored text is also italic"/>
                <Action Result="false" What="Jump" Why="Op4"/>
              </Operation>
              <Operation Op_ID="Op4" Op_Symbol="setInstanceHasElement"
                         Return_Type="Boolean">
                <Argument Arg_Type="Var" Arg_Value="big"/>
                <Action Result="true" What="Success" Why="colored text is also BIG"/>
                <Action Result="false" What="Error"
                        Why="Colored text does not have alternative possibility of  conveyance"/>
              </Operation>
            </Model>
          </Meta_Condition>
          <Mapped_Condition Set_ID="S1" Meta_ID="C1">
            <Meta_Mapping Meta="fg" Instance="fColor"/>
            <Meta_Mapping Meta="italic" Instance="i"/>
            <Meta_Mapping Meta="bold" Instance="b"/>
            <Meta_Mapping Meta="big" Instance="Big"/>
          </Mapped_Condition>
        </Evaluation_Conditions>
      </Evaluation_Logic>
    </Formal_Guideline>
  </Interpretation>
</GDL_Specification>
```

7. Summary

In this chapter we presented the GDL specifications of some selected guidelines with different complexity levels. We can now underline the following remarks about these guidelines:

- In general, the specification task is not difficult but it could be long.

- Example 1 (2-3 polices) shows demonstrated the use of the number of occurrences of a set instance, which to our knowledge is not possible in other tools.
- Example 2 (Web safe colors) demonstrated the use of constructed data types.
- Example 3 (metrics about number of italic words) demonstrated the flexibility of the approach in evaluating different forms of ergonomics-related expressions.

- Example 4 (text alternative) shows that, as other evaluation tools, we are unable to decide whether an Object of type image has equivalent text alternative or not, because the only way to do this is to examine the content of the object, which is not in the scope of our approach (it is not related to HTML). Of course, we can use the special element BodyText to capture the textual content of the object but we have no means to decide about its semantics and its correspondence to the semantics of the image included in the object.
- Example 5 (conveyance of colored information) shows that the specification is very dependent of the good comprehension of the guideline semantics and the HTML experience of the evaluator. As this is not usually the case, we suggest that specifications be accomplished by a human factor expert and a HTML expert (a Web designer).
- Examples 3 and 5 have Font.color in common but they are semantically different.