

The ALT-server ("An eye for an alt")



A accessibility/collaboration project proposal

Daniel Dardailler
dd@w3.org

Status

This memo was written between [July 1997](#) and [Jan 1998](#). Most of it right after Xmas 97 (on a personal touch, I didn't write any algorithm since - 10 years after).

It has no W3C official status, it's not even a W3C Note.

Introduction

For most people, visiting a web page is a one-to-one experience where one client program, the user's browser, gets and presents some resources coming from one information provider, usually the provider's web server. It's the client/server paradigm as we understand it. The client usually gets an "initial" HTML file from which it derives a complete presentation made of pieces found in the document itself (text, markup, style, alt text, etc) and additional pieces fetched by going back to the provider servers (images, audio, longdesc, etc).

It doesn't have to be always that way. The web addressing and transport architecture is flexible enough so that the set of resources that makes one's web session can seamlessly integrate from independent providers, or chains of providers.

This paper presents the application of this principle to Web Accessibility. The design of a system to retrieve and generate missing textual description of particular HTML elements (such as images) is examined, as well as the human collaboration foundation on which it is based.

Web Accessibility

Web Accessibility covers a very broad set of issues. There are of course different kind of disabilities to consider, such as a visual or hearing impairment, which all relate to different types of access denial (e.g. a missing caption for an audio stream, or the inability to linearize the content of a table for speech output). For the purpose of this paper, we will focus on one important aspect of accessibility for non-visual based user-agent: the textual description, or rather the lack thereof, attached to graphical images on the web, i.e. the well known missing ALT text in HTML. However, we believe the system presented can be generalized to other kind of resources.

The current situation is the following: when presented with a piece of HTML containing an image, a non-visual browser needs to "degrade gracefully" by presenting the user with a textual version of the image (that can be output as speech or braille). Most such systems currently only look for the textual information in the ALT attribute of the IMG element of the image. With progress happening in the browser area, other ways to find this textual/alternate text will soon be implemented, such as: look in the TITLE attribute, the HTTP stream, or the URL filename part.

What characterizes these approaches is that the textual description can only come from the origin document or the provider server.

Suppose...

As we alluded to in the introduction, another way of getting this information is to ask a different server altogether. Suppose there was a web server somewhere on the Internet whose primary job was to serve textual description of other servers' images. A non-visual browser (such as lynx) would then just have to query it when image ALT and TITLE attributes are missing and use the result in its presentation.

Let's consider an example expressed in pseudo HTTP sequences of queries and replies:

- A non-visual browser makes a request for an initial HTML document

```
GET www.merchand.com /order.html
```

- It gets back some HTML containing an image and no ALT text for it


```
Content-type: text/html
<HTML> ...
<IMG SRC="Images/card.png"> Order now!
```
- Since there is no ALT, nor TITLE, the browser is configured to make a query to an alt text server (wai.w3.org), giving it the absolute URL of the image as a parameter


```
GET wai.w3.org /ALT?url=www.merchand.com/Images/card.png
```
- The alt server returns a piece of text corresponding to the textual description of this image


```
Content-type text/ascii
A credit card logo
```

(if the alt text server hadn't returned anything, the browser could default to using "card.png" as the textual description of this image)
- The non-visual browser uses that in place of regular ALT text.

We'll look at the server issues later on, but for now, let's concentrate on the client side.

The important line in the example above is the query to the alt server.

```
GET wai.w3.org /ALT?url=www.merchand.com/Images/card.png
```

It's just a regular HTTP request that provides the server with the URL of an image and expects the ALT text for this image back.

The /SERVICE?name=value can of course be generalized to handle different type of resources and more information about one given resource.

Consider the following examples:

```
GET wai.w3.org /TABLE?url=www.foo.com/doc.html#n4
```

which asks the server for a linear/textual version of the fourth table found in www.foo.com/doc.html.

```
GET wai.w3.org /AUDIOCAPTION?url=www.merchand.com/Sounds/hello.midi
```

which asks for a caption of the audio track found at the given URL.

```
GET wai.w3.org /ALT?url=www.merchand.com/Images/card.png;isA;line=2
```

which mentions to the server that the image is used in the context of an A tag (a link anchor), and appears on line 2 of the document, making it more important to treat.

I think it's easily understandable what one can achieve there. Two things worth mentioning: the performance hit is nothing really to worry about: we merely add a web request in the overall building of a page, as graphical browsers do all the time. The minimal configuration on the browser side is also really small: declaring the alt server name to query from. So it's mostly transparent for the non visual browser user.

The implementation of this new GET functionality in a given browser (like lynx of amaya) is trivial.

Collaboration

At the beginning of the previous section, we assumed that "there was a web server somewhere on the Internet whose primary job was to serve textual description of other servers' images".

How do we go about implementing this alt server ?

Basically, I envision two ways of generating alt text for images.

The first is automatic extraction, the second human generation.

I will not expand on the first as this is an area of advanced research (shape and pattern recognition). I'll just mention that for a entire category of images, those representing text using some big fonts and colors, there exist algorithms out there (e.g. OCR) that could be used to extract the characters out of the graphics. A centralized server is well suited to integrate the latest and greatest solution in one location while readily serving the entire community.

The second way, human generation, is where I see the power of the web as a collaboration tool best applied.

This is how it could work.

The *ALT* server logically maintains a list of tuples

(image-url, textual-description, state)

where state is one of to-be-described, being-described, and described.

Processing works as follows:

- queries for textual descriptions of images come in
- if the image, identified by its url, is in the list with a described state, the description is returned.
- if not, a failure code is returned and the image url is added with a to-be-described flag
- the server continuously prioritizes this list using generic information such as frequency of the query (how many clients asked for the same data) and specific information (the image appears in an anchor, at the beginning of the page, etc)
- the server provides a form interface for sighted web users to enter the textual description of the images
- once an image has gotten its textual description, it is moved in the described state

The part with the form filling needs to be detailed.

Each time a sighted user access the form (see [annex](#)), the next to-be-described image with the highest priority is presented while its state moves to being-described.

Sighted user can then enter the description in an input field aside the image and submit the form to the server, which validates the text and either move the entry in the described state of just unlock it my moving it back to to-be-described state (invalid might be empty text for instance).

The locking is necessary due to the asynchronous nature of web form filling: several users could access and fill the "same" form at the same time, and we only need one image description per image.

So this is basic principle of the alt text server: *use the eyes of sighted web volunteers to help those who cannot see.*

The reason why this system can work is based on two facts:

- there are much more sighted users than the opposite, and a lot of them are willing to help
- the same images are requested by a lot of different people

In addition, the number of images with no description should go down as the awareness of content providers to accessibility is raised and authoring tools are improved.

If the automatic extraction part is improved, this will also diminish the number of images actually needing some human collaboration.

Regarding implementation on the server side, see the annex for [pseudo code](#). I expect a first version handling the base service to take a week of programming. A more complete version (generating reports, ranking, and doing more automatization can take a couple to several months.

Misc

[to be developed]

- One issue is whether such a system can be detrimental to *native* alt as content providers start counting on the system to solve their alt problem...
- I think not because the system can be used as incentive for content provider to add alt text to their own images: anybody can query the alt server by site and see the "no alt worse of the week". In effect, the alt server can act a complaint repository.
- storing the describer id can be used to provide a ranked output as incentive to get more describers
- automatic recognition of simple decoration (one color line, repeated pattern, 1x1, 2xN images, etc) should be easy
- send email to webmaster@host for each 10 <http://host/image.png> with no alt.
- provide a way to check and do notification of mis-described image (empty, dirty words, etc)
- provide a way to check and do notification of broken link image
- provide n-at-a-time image/input-desc form
- provide site and url targeted form filling (e.g. a visual impaired friend asks me to help him with a particular site or image and the prioritization scheme needs to be shortcut - note that this could also be accomplished by running a private alt text server).
- provide a way to *seed* the database of the alt server using a robot that explores a given list of sites.
- sort images by base filename (www.foo.com/img.png and www.bar.com/dir/img.png) and do a image compare to see if they are really the same.
- do the same identical-image-check based on same size images pre-sort.
- hard to identify images used as pieces (jigsaw) in a bigger image laid out in a table

Annexes

1 - Form layout

Example of form used to query the sighter user.

Welcome to the ALT-server filling form

You are about to describe:

<http://www.merchand.com/Images/card.png>

(entered Dec 25 1997 and was queried 12 times)

Enter the description(*) of this image

 No Alt by definition

Select Language: ▼

Optional:

Email: Name:

(*)the description should be short and to the point: e.g. "a american express credit card", "a dog", "a map with a magnifier". No need to add "an image of" or "this is a".

[Link to Advanced query form](#) providing n-at-a-time, site or url targeted filling, and database dump ranked by images site name, describer id, base image file name, etc.

2 - Simple pseudo code for ALT-server script

This script handles both the queries for alt desc and the insertion of alt desc by sighted users for an hypothetical server hosted at <http://www.w3.org/WAI/altserv>

```
//
// for now ignore lang, date entered, number queries, id of describer,
// checking dup, validity, security, and additional services like
// ranking of bad site, good describers, etc)
//
// INPUT: 3 cases
//
// 1 (asking for textual description of url)
// http://www.w3.org/WAI/altserv?url=www.merchand.com/Images/card.png
//
// 2 (giving a textual description for url)
// http://www.w3.org/WAI/altserv?url=www.merchand.com/Images/card.png
// desc="A credit card logo"
//
// 3 (asking for form to fill in desc for a url)
// http://www.w3.org/WAI/altserv
//
// OUTPUT: see RETURN statement below
//
// maintains a persistent list of [url, desc, state]
// with state = d, bd, tbd (described, being described, to be described)
```

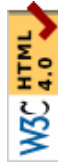
```

if url
if ldesc // case 1 : asking for textual description of url
if (url in list)
if (list[url].state = d)
RETURN list[url].desc
else
RETURN no desc
else
add url in list
list[url].state = tbd
RETURN no desc

else // case 2 : giving a textual description for url
// should check list[url].state = bd and desc valid
list[url].state = d
list[url].desc = desc
RETURN ok

else // case 3 : no param, asking for form to fill in desc for a url
get top url with list[url].state = tbd
list[url].state = bd // should check url valid with HEAD
RETURN form HTML with embedded image url

```



Copyright © 1997-2007 W3C® ([MIT](#), [ERCIM](#), [Keio](#)), All Rights Reserved. W3C liability, trademark, document use and software licensing rules apply. Your interactions with this site are in accordance with our [public](#) and [Member](#) privacy statements.