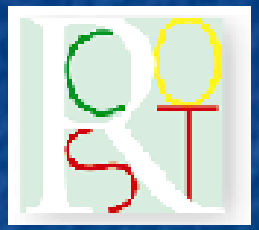
Web Site Accessibility: Identifying and Fixing Accessibility Problems in Client Page Code



Porfirio Tramontana

Anna Rita Fasolino

Dipartimento di Informatica e Sistemistica

University of Naples Federico II, Italy

Giuseppe A. Di Lucca

RCOST – Research Centre on Software Technology

University of Sannio, Benevento, Italy

Web Site Accessibility

- The importance of accessibility:
  - Tim Berners-Lee, the W3C Director and inventor of the World Wide Web: "*The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect.*"
  - Vanderheiden in 1990 states the importance of the design of Web Sites accessible to disable and elder peoples
  - Carter and Markel state that the development of an accessible Web Site opens vast potential markets and has spillover effects for all users.

WAI: Web Accessibility Initiative, http://www.w3.org/WAI/

G.C. Vanderheiden, "*Thirty-something million: should they be exceptions?*", Human Factors **32**(4), pp.383-396

J. Carter, M. Markel, "*Web Accessibility for people with disabilities: an introduction for Web developers*", IEEE Transactions on Professional Communication, **44**(4), 2001, pp.225-233

Web Site Accessibility

- W3C Definitions:
  - *Accessible*
    - Content is accessible when it may be used by someone with a disability
  - *Accessibility Problem (Web Content)*
    - Web content that fails to meet the requirements of the W3C We Content Accessibility Guidelines (WCAG)
  - *Accessible Web Content*
    - Web content with no Web content accessibility problems.

- ISO 9126:
  - *Usability:*
    - The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions.

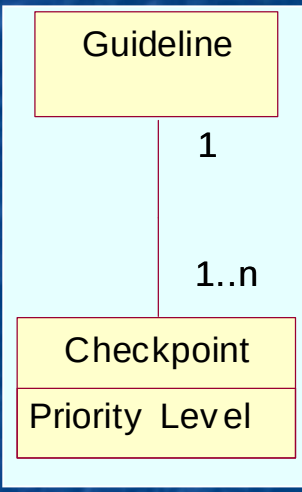
Accessibility References

- Governments and International Organizations
  - European Union
  - Single Governments (for the US: Section 508 Rehabilitation Act)
  - World Wide Web Consortium: Web Accessibility Initiative
    - ATAG: Authoring Tool Accessibility Guidelines
    - UAAG: User Agent Accessibility Guidelines
    - WCAG: Web Content Accessibility Guidelines
      - Explains how to make Web content accessible to people with disabilities.
      - Provides guidelines for Web content developers (page authors and site designers) and for developers of authoring tools.
    - …

European Commission, Information Society, "Accessibility and Information Society", http://europa.eu.int/information_society/policy/accessibility/index_en.htm
"Section 508 Rehabilitation Act: Web-based Intranet and Internet information and applications (1194,22)", http://www.access-board.gov/sec508/guide/1194.22.htm
Authoring Tool Accessibility Guidelines, http://www.w3.org/TR/WAI-AUTOOLS/
User Agent Accessibility Guidelines 1.0, W3C Recommendation, 17 December 2002, http://www.w3.org/TR/2002/REC-UAAG10-20021217/
Web Content Accessibility Guidelines 1.0, W3C Recommendation, 5 May 1999, http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/
Web Content Accessibility Guidelines 2.0, W3C Working Draft, 30 June 2005, http://www.w3.org/TR/WCAG20/

Web Content Accessibility Guidelines 1.0 (May 1999)

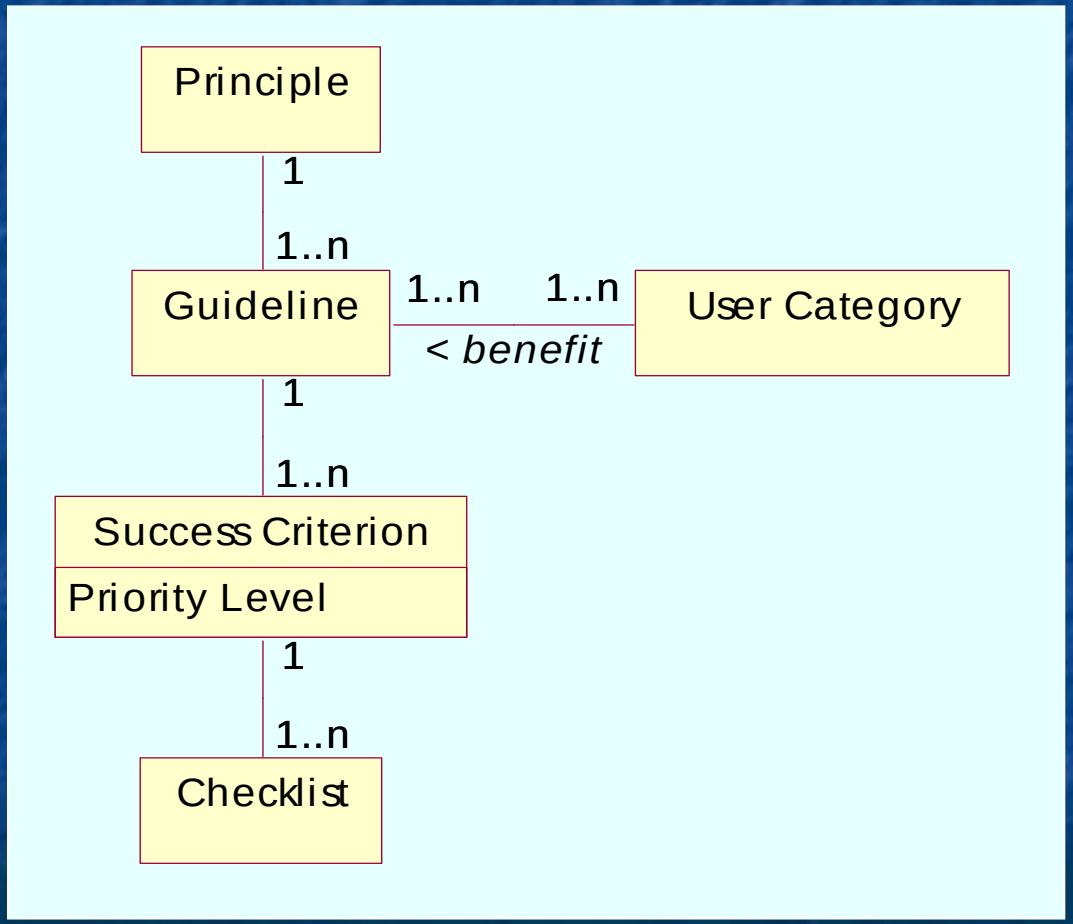
- 14 Accessibility Guidelines (i.e. Indications for Web Authors)
  - Example:
    - Guideline 1. Provide equivalent alternatives to auditory and visual content.
      - Provide content that, when presented to the user, conveys essentially the same function or purpose as auditory or visual content.



- 63 Checkpoints to evaluate the accessibility of a Web Page (with 3 priority levels)
  - Example:
    - 1.1 Provide a text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content). ... [Priority 1]
      - In HTML: Use "alt" for the IMG, INPUT, and APPLET elements, or provide a text equivalent in the content of the OBJECT and APPLET elements ...
- A 3 level measure of accessibility conformance (A – AA – AAA)

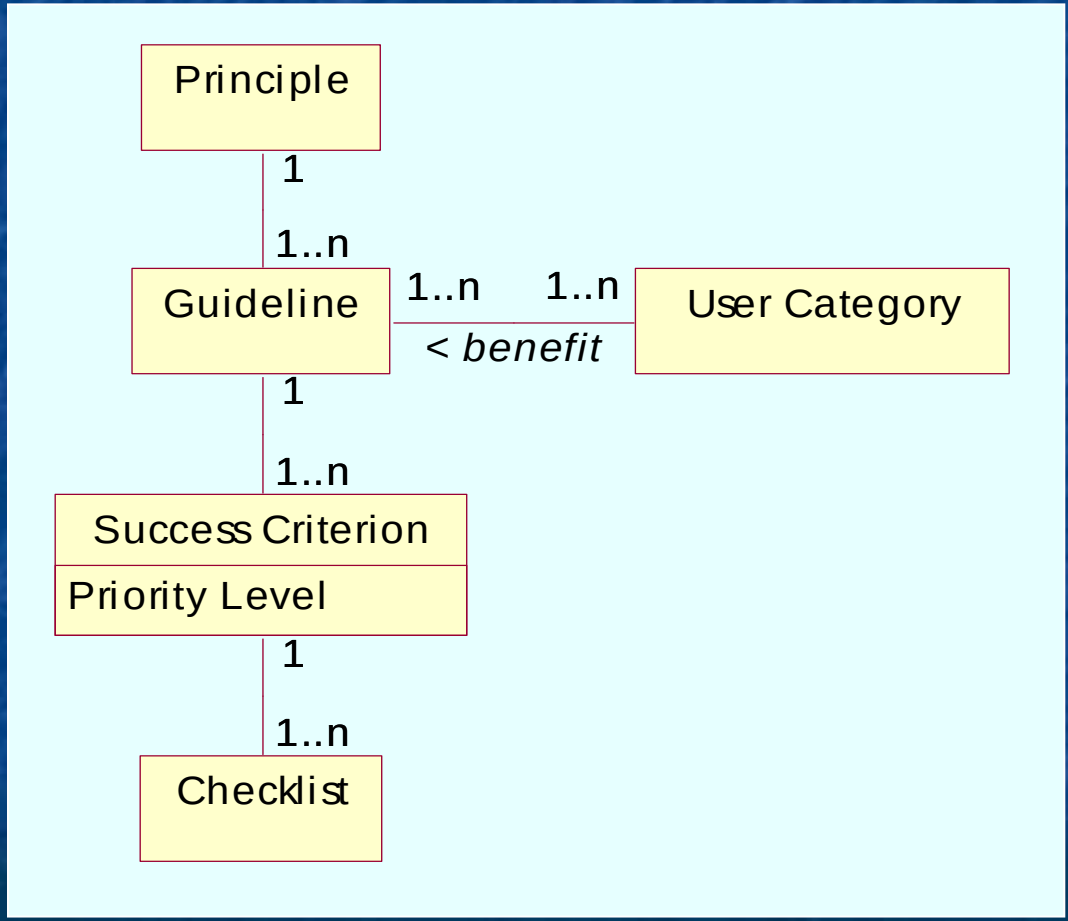
Web Content Accessibility Guidelines 2.0 Working Draft

- Currently under development (last release 30 June 2005)
- 4 Principles:
  - Content must be perceivable;
  - Interface elements in the content must be operable;
  - Content and controls must be understandable;
  - Content must be robust enough to work with current and future technologies.



Web Content Accessibility Guidelines 2.0 Working Draft

- 13 Guidelines
  - With indications of the categories of users that benefits by the conformance of the Web Site to the Guideline
- Success Criteria
  - Similar to Checkpoints, but non technologies-dependent
- Checklists of technology dependent rules
  - Contained in separate documents (to be published)



WCAG 2.0 Guidelines: An Example

- **Principle 2: Interface elements in the content must be operable.**
  - **Guideline 2.1 Make all functionality operable via a keyboard interface.**
    - **Level 1 Success Criteria for Guideline 2.1**
      - All of the functionality of the content, where the functionality or its outcome can be described in a sentence, is operable through a keyboard interface. [I]
      - **Note:** This includes author-provided accessibility features.
      - **Note:** Other interfaces (such as a mouse) can be provided in addition to keyboard operation.
      - **Note:** Refer to guideline 4.2 for information regarding user agent support.
    - **Level 3 Success Criteria for Guideline 2.1**
      1. All functionality of the content is designed to be operated through a keyboard interface.
    - **Who Benefits from Guideline 2.1 (Informative)**
      - Individuals who are **blind** (and cannot use pointing devices) can have access to the functionality of the Web content or site.
      - Individuals with **severe physical disabilities** can use speech input (which simulates keystrokes) to both enter data and operate the interface elements on the page.
  - **Examples of Guideline 2.1 (Informative)**
    - **Example 1: operation with multiple input devices.**
    - **Example 2: examples of Web content that would and would not be operable from a keyboard interface**

Limitations of WCAG 2.0

- It lacks of practical methods and rules allowing the identification and fixing of accessibility problems.
- It has not a precise classification of user categories affected by accessibility problems

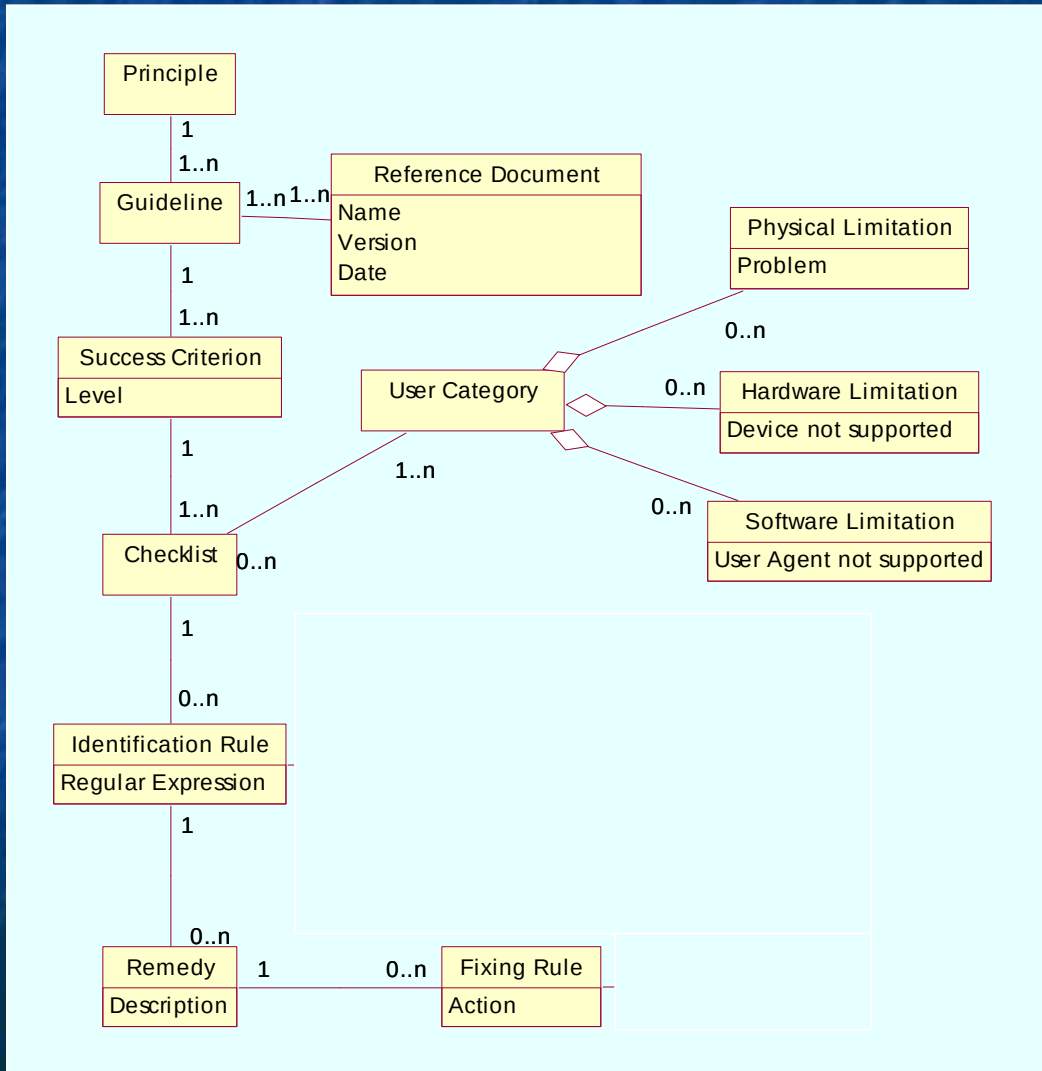
- To overcome these limitations, we propose:
  1. A conceptual model extending the one proposed in WCAG 2.0
  2. A process and a tool for identifying and fixing Web accessibility problems

A conceptual model for Web accessibility

1. It integrates guidelines belonging to different Reference Documents

2. It provides a set of technology-dependent checklists, describing situations causing accessibility problems affecting a specific User Category

3. For each Checklist, it provides a set of Identification Rules, based on known Programming Patterns responsible of the accessibility problem

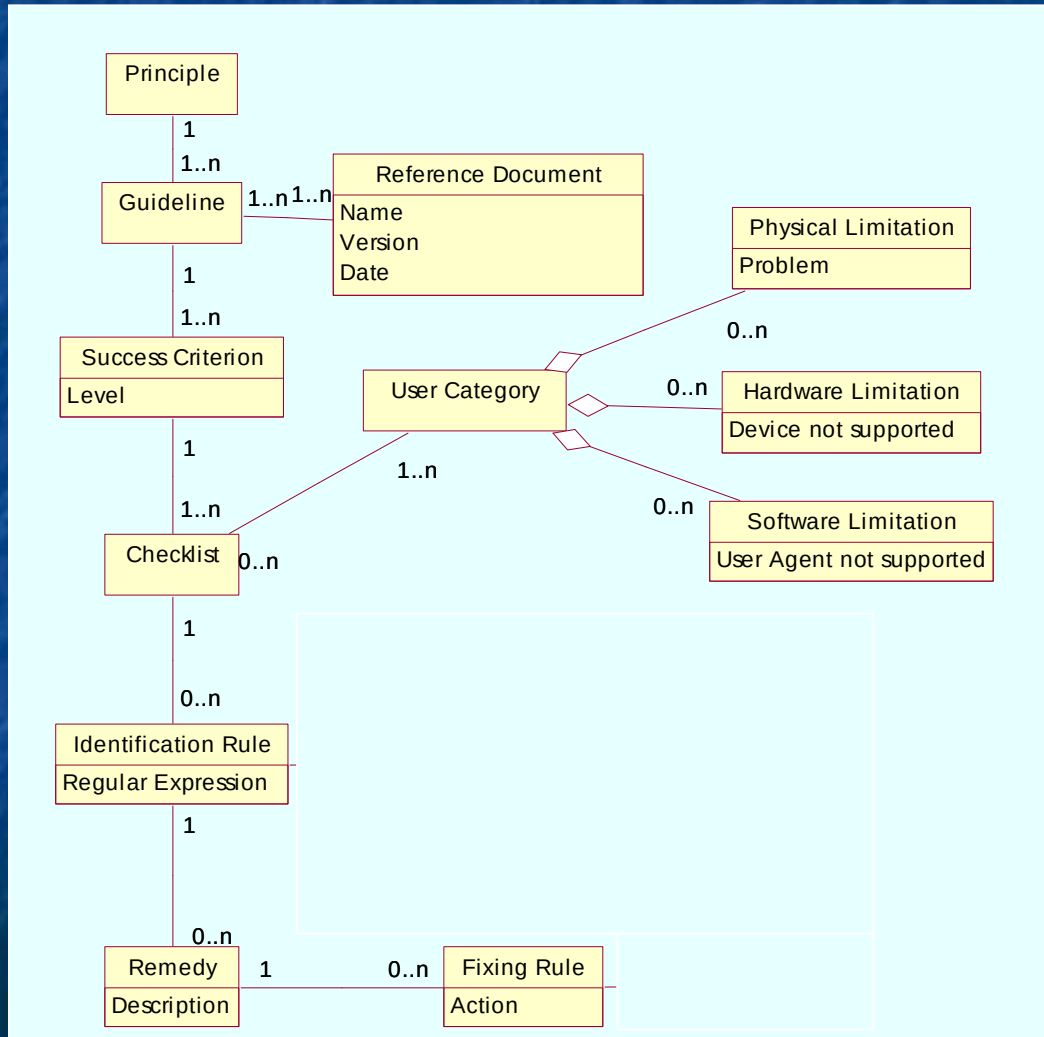


A conceptual model for Web accessibility

4. For each Identification Rule, it provides a list of Remedies (description of possible solutions to the identified problem).

5. For such Remedies, it provides a Fixing Rule based on a known programming pattern.

6. It proposes a finer classification of the users ...



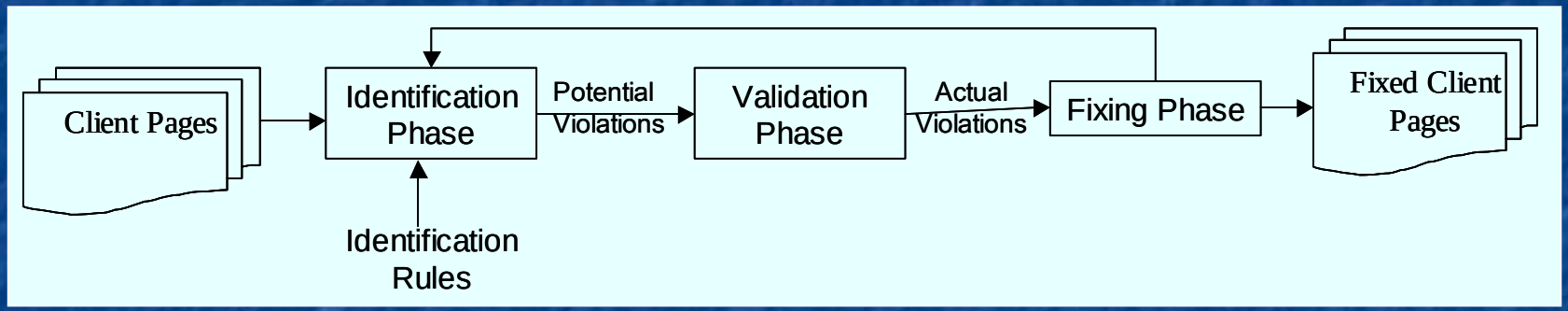
A finer classification of User Categories

- Based on three aspects:
  - The physical capabilities of the user
    - Blinds, with reduced sight capabilities, colour-blind, deaf, with cognitive problems, with hand problems, ...
  - The hardware devices used
    - Low resolutions screens, text only screens, Screen Readers, Braille devices, devices without mouse or keyboard, ...
  - The user agents used
    - Obsolete browsers, browsers with reduced capabilities, ...

An example: Functionalities activated exclusively by mouse events

- Description: A common technique used to improve Web pages usability consists in the use of scripts activated by mouse events.
- Principle: Interface must be operable
- Guideline: Make all functionality operable via a keyboard or a keyboard interface (WCAG 2.0 – Guideline 2.1)
- Reference: WCAG 1.0 – Checkpoint 9.3; WCAG 2.0 – Guideline 2.1; Section 508 – 1194.22 l
- Success Criteria: All of the functionality of the content is operable trough a keyboard or keyboard interface (WCAG 2.0 – Success Criteria 2.1.1 – Level 1)
- Checklist for Javascript: If mouse events are coded in the page, alternative keyboard events must be provided.
- Identification Rule: Search for scripts related to mouse events.
    - Regular Expression: `on(click|dblclick|mouse[a-z]+)[ ]*=.[^>< ]*`
- Affected User Categories:
    - Peoples using user agents without script support;
    - Peoples using Netscape Navigator releases older than 3.1;
    - Peoples using Internet Explorer releases older than 4.0;
    - Peoples that doesn't use a mouse (or equivalent device);
    - Peoples using a Screen Reader;
    - Peoples with sight problems.
- Remedies:
    - Use of alternative pages;
    - Insertion of keyboard events equivalent to mouse events.

A process for identifying Web accessibility problems



- Identification phase: The source code of existing Web pages is statically analysed in order to verify the occurrence of Identification Rules (revealing Potential violations of Accessibility Guidelines)
- Validation phase: The pages are executed in particular conditions in order to verify if the potential violation is an actual violation.
- Fixing phase: The pages containing actual violations are corrected (automatically, if fixing rules can be applied) and re-tested

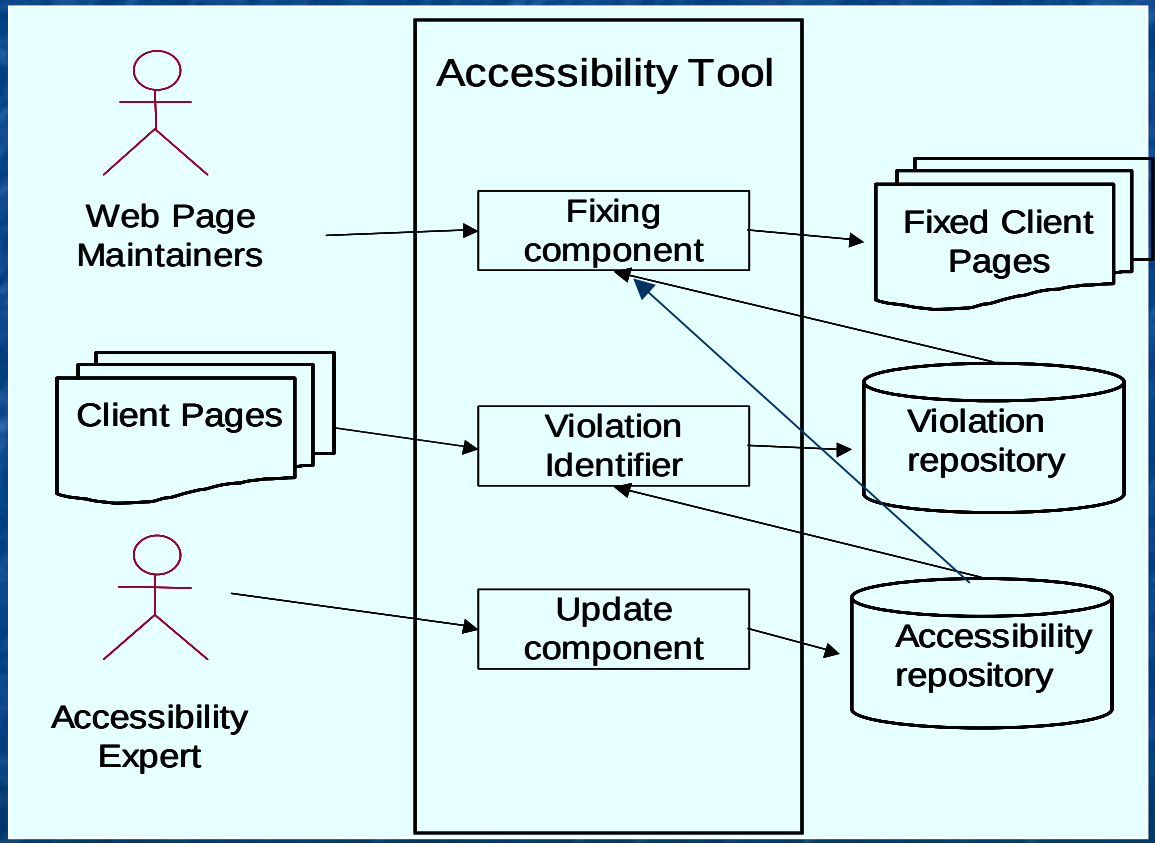
Violation

```
<script language=Javascript> location.replace("http://www.unina.it")
</script>
<a href=http://www.unina.it>Click here if you are not automatically
redirected </a>
```

Fix

The Accessibility Tool

- The Accessibility Repository is an instantiation of the Conceptual model for Web Accessibility. It is updated by Accessibility experts to take into account new hardware devices, new user agents, new standards, new violation patterns and so on
- Actual violations of accessibility guidelines are stored in the Violation Repository
- The Violation Identifier component performs the identification of potential violations and its validation
- Using the fixing component, the maintainer can correct the actual accessibility violations, (by using, sometimes,



Case Study

- A subset of accessibility problems, due to the use of client scripts has been considered
- 3 Reference documents

  (WCAG 1.0 – WCAG 2.0 Working Draft November 2005 – Section 508)
- 4 Principles
- 7 Guidelines
- 7 Success Criteria
- 16 Checklists
- 16 Identification Rules
- 9 Categories of users:

  Hardware without mouse;
  Blind users;
  Textual Browsers;
  Browsers without any script support;
  Release 3 of Internet Explorer;
  Releases of Internet Explorer older than 3;
  Releases 3 of Netscape Navigator;
  Releases 2 of Netscape Navigator;
  Releases of Netscape Navigator older than 2.
- 200 Client pages (20 pages of 10 Internet Web Sites of different business areas), randomly downloaded using a spider tool.

Case Study

- Amazon (#10), Ibm (#4) and Chicago Tribune (#7) Web sites, makes a great use of scripts, pop-ups, mouse events, and other script features
- Google Web site (#9) presents potential violations due to a script setting the focus on a specific input field (e.g. the search text field in the home page).
- The tourism Web site (#10) presents problems making it not accessible to users with old releases of browsers
- University of Naples Federico II Web site (#3) is compliant to the 'A' level with respect to WCAG 1.0 recommendations, but it presents potential violations due to the use of scripts

Potential inaccessibilities identified by the tool, with respect to all the considered user caegories

#	Category	Web Site	Pages with potential violations	Total Number of potential violations
1	Accessibility	www.useit.com	2/20	7
2	E-commerce	www.amazon.com	19/20	308
3	Educational	www.unina.it	19/20	96
4	Enterprise	www.ibm.com	20/20	136
5	Government	www.governo.it	0/20	0
6	Government	www.comune.procida.na.it	3/20	3
7	On-line Newspaper	www.chicagotribune.com	13/20	469
8	Organization	www.w3c.org	0/20	0
9	Search Engine	www.google.com	16/20	422
10	Tourism	www.procida.net	18/20	65

To evaluate if the potential violations are actual violations:

1. Testing approach
   - Recreating the same limitation of users;
2. Source Code Analyisis approach
   - Searching for the use of solutions providing techniques providing equivalent behaviours for users with limitations

Examples

Browser with script support


```
<div id="searchTargetMenu" class="menu"> <div
class="searchmenuItem">
<a href="javascript:setSearchTarget('article')" id="searchTarget-
article" style="display:block" onmouseover="return true">
<img src="/images/search/search_trib.gif" border="0" width="150"
height="19" alt="chicagotribune" /></a> </div>
<div class="searchmenuItem"><a
href="javascript:setSearchTarget('google')" id="searchTarget-
google" style="display:block" onmouseover="return true"><img
src="/images/search/search_web1.gif" border="0" width="150"
height="19" alt="Google" /></a> </div> </div>
```

```
<script type="text/javascript" src="/javascript/popmenu.js"></script>
<script type="text/javascript">
<!-- var searchTargetMenu = new PopMenu("searchTargetMenuLabel",
"searchTargetMenu", 160, "sw", "nw", true); function setSearchTarget (target) {
document.Search.elements["target"].value = target; var menuLabel =
document.getElementById("searchTargetMenuLabel"); var menuItem =
document.getElementById("searchTarget-" + target); menuLabel.innerHTML =
menuItem.innerHTML; searchTargetMenu.close(); } // Ensure display set properly
when page loads. setSearchTarget(document.Search.elements["target"].value); //-->
</script>
```

Browser without script support

Examples

Browser with script support


```
Contact:
<script name="Contact" language="JavaScript"><!–
var e1 = "nielsen"
var e2 = "nn"
var e3 = "group"
var e4 = "com"
document.write("<a href=" + "mail" + "to:" + e1 + "@" + e2 + e3 +
"." + e4 + ">" + e1 + "@" + e2 + e3 + "." + e4 + "</a>;")
// //--> </script>
```

Browser without script support

Conclusions and Future Works (1/2)

- A conceptual model describing accessibility issues has been presented;

- A Process supporting identification, validation and correction of accessibility violations has been proposed;

- A Tool supporting the Process has been realized and tested in order to assess the effectiveness of the Process;

- Case studies show the usefulness of the tool in the identification of potential accessibility violations

Conclusions and Future Works (2/2)

•The problem of assessment of accessibility is very hard, due to the continuous introduction of:

- Hardware technologies
- Software technologies (user agents and languages)
- Reference documents
- Accessibility techniques

• Consequently, the effectiveness of the process depends on the freshness and richness of the accessibility repository

Future Work:

• Realization of an open XML accessibility repository, so as anyone can suggest new rules and remedies ... Collaboration is welcome!


Time is over ... Are there any questions?