A practical example of a collaborative learning experience for engineering students: How to build accesible indoor maps

Conference Paper · December 2015

DOI: 10.1109/ICBL.2015.7387647

4 authors:

Tania Calle-Jimenez
Escuela Politécnica Nacional

Sandra Sanchez-Gordon
Escuela Politécnica Nacional

Cristina Rivera
Escuela Politécnica Nacional

Sergio Luján-Mora
University of Alicante

A practical example of a collaborative learning experience for engineering students: How to build accesible indoor maps.

Tania Calle-Jimenez; Sandra Sanchez-Gordon; Cristina Rivera-Pastrano
Department of Informatics and Computer Science
National Polytechnic School
Quito, Ecuador
tania.calle@epn.edu.ec
sandra.sanchez@epn.edu.ec
cristina.rivera@epn.edu.ec

Sergio Luján-Mora
Visiting teacher at National Polytechnic School
Department of Software and Computing Systems
University of Alicante
Alicante, España
sergio.lujan@ua.es

Abstract. This paper presents the description of a project to develop an accessible indoor plan to help blind people to localize landmarks within a building from a pedagogical perspective. This project has been employed to foster engineering students to acquire the necessary knowledge and skills to use technologies and tools to develop accessible engineering products within the context of a collaborative learning experience. This paper describe the step by step process followed to make an accessible indoor plan that complies with WCAG 2.0 accessibility guidelines. Also, conclusions and future work regarding the potentialities of the approach used in this project for generating collaborative learning experiences are presented.

Keywords— Collaborative Learning Experience; Engineering Education; Accessible Indoor Maps; Web Accessibility; Blindness; WCAG 2.0; ARIA; SVG; CSS; HTM; JavaScript

I. Introduction

Nowadays, the advancement of several technologies has steadily increased, making life easier and enabling people to perform daily tasks readily, such as to be located in an unknown building.

However, there are several barriers in the use of technologies for daily tasks for people with visual disabilities, who cannot perceive all visual information in the same way than a person without disabilities. Blindness is the extreme case of a visual disability.

For floor plans or maps, most information is visual. These visual elements cannot be perceived at all by a blind person. For this reason, in this study it has been proposed the development of a practical example that allows affordable options to use an indoor map for mobility and location purposes for blind people in an unknown building. This practical example shows an approach for developing a collaborative learning experience for engineering students.

The rest of this paper is organized as follows. Second section explains concepts, standards, and technologies that were used in the development of the practical example. Third section describes the tools used for construction of the practical example, including a brief description of codification, edition, and evaluation tools, as well as screen readers. Fourth section contains an explanation of the practical example including the case study, the architecture used, and a comparative chart of compatibility analysis of four browsers. Fifth section describes the development process step by step. Finally, conclusions and future work are presented, describing the potentialities of the approach used in this project for generating collaborative learning experiences for engineering students.

II. Fundamentals

A. Web Accesibillity

Web accessibility states that web access should be universal, without dependency on software or hardware, network services, culture, geography, language, or users skills. It focuses on universal access to the web without excluding people with or without disabilities. Web accessibility considers most of the disabilities that affect web access including visual, auditory, motor, and cognitive and also helps people with disabilities. Web accessibility allows flexible design of websites to fit any need.

Some examples of web accessibility that can be added to a web content are:

- Device independence. Web content should be developed in such a way that does not require a specific type of device, such as a mouse. This also helps trained users that prefer the use of the keyboard and people using mobile devices.
- Accessible navigation. People with cognitive disabilities have low response times and problems processing visual information, so they are not on full capacity to navigate through web content that is not easy understandable. Consistency at designing web content also helps all people as it increases the usability of the web content [1].

B. Web Content Accessibility Guidelines

The Web Content Accessibility Guidelines (WCAG) are a set of 12 design guidelines organized under 4 principles: perceivable, operable, understandable, and robust. The

978-1-5090-0239-9/15/$31.00 ©2015 IEEE 9-11 December 2015, Tecnológico de Monterrey, Mexico City Campus, Mexico City
2015 International Conference on Interactive Collaborative and Blended Learning (ICBL)

guidelines include measures based in daily examples that explain how the design can influence or limit access to information published in the web. The WCAG also include check points to verify testable errors. Checkpoints are assigned priority levels are as follows:

- Priority 1. These are guidelines that a web developer must comply in order that certain groups of users can access the contents of a website.
- Priority 2. These are guidelines that a web developer must comply in order that certain groups of users do not have many difficulties to access the contents of a website.
- Priority 3. These are guidelines that a web developer must comply in order that certain groups of users do not have serious difficulties to access the contents of a website.

Based on these priorities, WCAG establishes the following levels of compliance:

- Conformance Level A. All the checkpoints of priority 1 are met.
- Conformance Level AA. All the checkpoints of priority 1 and 2 are met.
- Conformance Level AAA. All the checkpoints of priority 1, 2 and 3 are met [2].

C. *Accesible Maps*

Maps include visual content that consist of elements that are useful for positioning paths and spaces, to provide orientation information of the world around us.

However, these visual elements are not useful to a blind person, who do not have the ability to explore and navigate a map that does not have text alternatives that can be read by a screen reader. Therefore, it is necessary to integrate in maps alternatives that provide accessibility to content and information transmitted for blind users. An accessible map can be used for different types of applications:

- Movement and orientation. Refers to the type of information that is visually provided. These elements can be accessible, provided there are textual or sound alternatives. Additional elements can improve the navigability and structure of the map, for example, the useful content of the map can be perceived by short beeps, long beeps, and textual descriptions of map items.
- Location and exploration of interest points. This is a way to browse the map, point by point, by keyboard or alternative gestures. Thus, textual and sound tags are activated to perceive a description that provides new ways of navigability, increasing the accessibility of the map.
- Educational purposes. Geometrical figures, vibration alerts, and short videos about the most important components can be integrated to a map [3].

D. *Accessible Indoor Maps or Floor Plans*

Accessible indoor maps, also called floor plans, allow to plan ahead a visit to an unknown destination and identify the difficulties that a person might have to move around in an indoor place. Hence, they are useful to all people to improve their mobility and location enabling to identify points of interest within a building easily.

The most common way to make an accessible floor plan is to provide textual descriptions or touch options. Textual descriptions provide an overview of visual structure of the floor plan, while the tactile alternatives allow people to build mental space representations.

An accessible floor plan must provide descriptions of doors, windows, stairs, rooms, and all relevant information in the map that may be useful to help people to mobilize on the floor building. Information should be added about the activities undertaken in each room to guide people to know the functions of the different spaces within the floor building. These textual descriptions can be read by a screen reader through the floor plan code. Additionally, a position indicator helps identify elements that are around the person. From this, a guide route can be established to help people to arrive to a specific point in the floor building [4].

E. *ARIA*

Accessible Rich Internet Applications (ARIA) is a specification that defines a mechanism to add accessibility features to web content for disabled people. It specializes in advanced development with JavaScript, HTML, Ajax and others technologies, to improve accessibility of websites and applications.

One application of ARIA is the capability of identifying sections of websites that are most important to enable keyboard use, instead of only using the TAB key. The ARIA specification also includes roles that help to describe the type of element that are being used. The roles describe the structure of an HTML document. It also defines procedures for drag-and-drop features, to describe drag source and drop target. In addition, it provides keyboard navigation for the web items and procedures. [5].

III. TOOLS

This section presents four set of tools used in this study: codification tools, edition tools, evaluation tools, and screen readers.

A. *Codification Tools*

HTML

Hyper Text Markup Language (HTML) is a language used to display web pages. It works as a standard reference defined by the World Wide Web Consortium (W3C) in the development of web pages. HTML contains a structure based on code to display web content that can be text, images, data, audio, video, multimedia, data tables, among others. To build a website, it must be composed of HTML pages, Cascading Style Sheets (CSS) and scripts that define the behavior of the page [6].

SVG

Scalable Vector Graphics (SVG) is a two-dimensional vector language to create vector graphics, raster graphics, and text in XML format. SVG has the facility to be integrated directly to HTML code and use CSS for style. It is a scalable format that allows greater definition and can be adapted to be responsive according with the resolution or the specification of any device. [7].

CSS

Cascading Style Sheets (CSS) was created to manage a stylized format from web pages, separating style from HTML content. This separation between the structure and style of the page provides benefits in maintenance, performance, production and improve the quality of the styles [8].

JavaScript

JavaScript is a programming language. The code is developed through scripts that can be embedded in the HTML code or referenced from the HTML document. This language is used to give behavior to the elements of a web page, such as buttons, forms, and links. In web accessibility, JavaScript must be unobtrusive, this means separating the functionality of JavaScript or business layer from the content and presentation layers [9].

JQuery

JQuery is an open source library of JavaScript that was created to add extra features and allow the development of web pages on the client side. It serves to simplify navigation for HTML documents, create user interfaces, develop animations and effects, for programming Ajax applications, and manage events [10].

XAMPP

It is an open source platform that contains Apache HTTP Server, PHP, and MySQL. XAMPP has three components:

- Apache HTTP Server. It is an open-source HTTP server for modern operating systems. It provides a secure, efficient, and extensible server that give HTTP services in sync with the current HTTP standards [11].
- PHP. It is an interpreted an open source code language which is executed by a server. PHP code works with HTML code and can be combined with several web frameworks and templates.
- MySQL. It is a management system for relational databases with open source license that allows developing applications in multiple languages and platforms.

B. Edition Tools

GIMP

GIMP is a program to edit digital images, graphics and photos in bitmap. It has General Public License (GPL) and the distribution is free. Gimp has tools to retouch and edit images, free graphics, change size, cut, convert to different image formats, and other tasks. It also creates route images that can be converted in any kind of digital format as SVG [12].

NOTEPAD++

It is a source code editor for multiple languages. It is licensed under the General Public License (GPL). It enables editing several pages in the same interface and has features to recognize the actual language, tabbed editing, drag-and-drop, among other functionality [13].

C. Evaluation Tools

Examinator

It is an online tool that checks the code of a website and performs a series of tests related to the conformance with the WCAG 2.0 guidelines, awarding a grade in a scale between 1 and 10 as an accessibility indicator of pages and provides a detailed report of the tests. The tool gives automatic results and facilitate understanding of the problems to make necessary corrections [14].

TAW

TAW is a set of tools for accessibility analysis of websites. TAW tools review all items and pages that compose a website. The main TAW tool is the web pages analyzer. The common core of the TAW set of tools is the scanning engine [15].

D. Screen Reader Tools

NDVA

NonVisual Desktop Access (NVDA) is a screen reader that allows using computers to blind and vision impaired people. It is licensed under GPL. The tool reads the text information of the screen with a digital voice and can be controlled moving the mouse to a specific area of the screen or using keyboard options. Other function of NDVA is converting the screen text in braille, if the computer has a braille display device. This program enables blind and vision impaired people access to online education, shopping, banking, access to information, among others [16].

ChromeVox

ChromeVox is a screen reader extension of Google Chrome browser that reads web content. It allows blind people to navigate in websites to access to information, news, banking, and social websites. As all screen readers, ChromeVox reads tags in HTML that contains extra information about elements of the code [17].

IV. PRACTICAL EXAMPLE

The practical example uses as case study the System Engineering Building, Second Floor of the National Polytechnic School of Ecuador. It has four main areas. The first area is dean office and undergraduate administrative offices. The second area is the library. The third and fourth sections correspond to office space of faculty members. In addition to the sections described, the second floor contains rooms for research projects, bathrooms, and storage rooms.

One big problem in the System Engineering Building, especially the second floor, is the lack of information on the distribution of offices and location of accessible elements for disabled persons.

For this reason, this study propose as a solution to create an accessible website with information and an accessible indoor map that can be accessible for blind people, with alternative textual descriptions of map elements that can be read by a screen reader, providing accessibility features to both blind people and people without disabilities.

Figure 1 shows the architecture of the practical example. The information is consulted in a database containing the persistent data about routes. A JSON request returns this information as a DOM object that is loaded by an AJAX function. The resultant web page that is presented to the end user can be read by a screen reader.

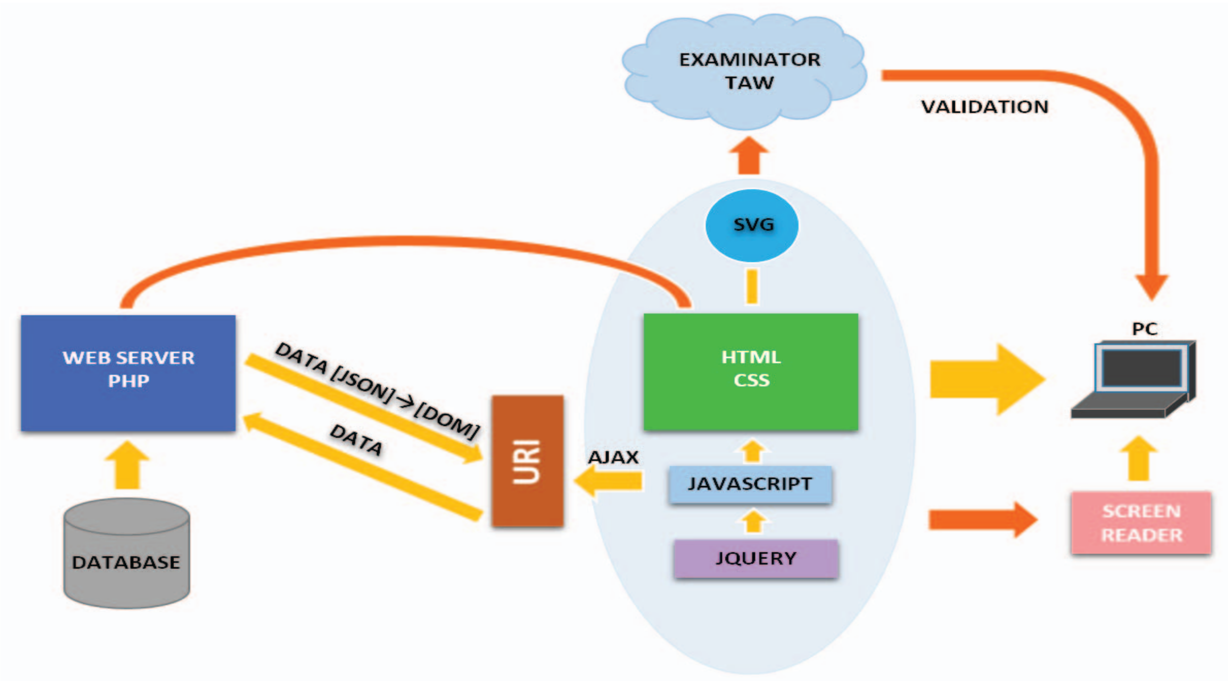

Fig. 1. Architecture.

The accessibility analysis of the practical example was performed with Examinator and TAW. Figure 2 shows the accessibility score of 8.3/10 obtained with Examinator. Figure 3 shows the accessibility analysis performed with TAW showing that conformance level WCAG 2.0 AA was acomplished.

Fig. 2. Accessibility evaluation score obtained with Examinator.

Fig. 3. Accessibility evaluation report obtained with TAW.

The website was tested in different browsers, having different behavior respect to SVG features, JQuery animations, CSS styles, and reading of speech database. Table 1 shows the compatibility analysis which shows that currently Mozilla Firefox is the most appropriate browser for displaying accessible indoor maps, because it has good compatibility with SVG elements and functions. Also, it fits properly to JavaScript and JQuery functions such as animations, uploading content, and graphic coordinate fetching from the database.

TABLE 1. COMPATIBILITY ANALYSIS OF BROWSERS

BROWSER	HTML COMPATIBILITY	SVG COMPATIBILITY	JQUERY COMPATIBILITY	CSS COMPATIBILITY
Google Chrome	Has total compatibility with HTML code, allows visualization of tabindex attribute and permit focusing in, on tabindex elements.	Has compatibility with graphics but is incompatible with other SVG functions like title positioning, and some SVG styles.	Dynamic elements as tool tips, are shown as a quickly and repetitively manner. Routes with coordinates are plotted immediately.	Has total compatibility with all web colors and styles.
Mozilla Firefox	Has total compatibility with HTML code, however, tabindex cannot be visualized clearly and doesn't work properly.	Displays SVG content with all format and styles, is compatible with most of SVG functions.	Content like tool tips are shown quickly and repetitively manner. The routes are plotted fast.	Has total compatibility with all web colors and styles.
Microsoft Edge	Has total compatibility with HTML code, and tabindex attributes are clearly shown.	Only SVG graphics as lines and colors are visualized, text and labels doesn't show.	Content like tool tips show quickly and repetitively manner. The routes are plotted with some delay.	Has total compatibility with all web colors and styles.
iOS Safari	Has slow charge of page content, size and distribution with content are showed different. tabindex are clearly recognized and visualized.	Has compatibility with graphics but is incompatible with other SVG functions like title positioning, and some SVG styles.	Content is charged slowly, tool tips appear lingering and always the focus is on close button.	Has total compatibility with most web colors and styles but some styles are not recognized.

V. A Step by Step Description of the Collaborative Learning Experience

In this section, we present a step by step description of the process used to design and build the accessible indoor map for the practical example.

We believe this approach enables a positive collaborative learning experience for engineering students.

First step. Ask students to team-up in odd numbers, i.e. 3 or 5 students per team. The reason to choose odd number of team members is that this facilitates team decision making.

Second Step. Each team takes physical measures for the indoor map. This allows team members to perform their first collaborative task.

Third Step. Each team plot their version of the indoor map. For this, it is necessary to use a tool like GIMP to export graphics into SVG format. Figure 4 shows the use of GIMP to plot the map. The exported code must be clean, i.e. it should not contain any type of tool dependencies, must be clearly structured, each map item may be graphed independently of the other elements. These elements are grouped into a single scheme maintaining dependence to add attributes to navigate. Map styles are worked independently of the indoor plan and descriptive labels of offices and spaces are added up. Figure 5 shows the indoor map made with HTML, SVG and CSS codes. It is advisable to use CSS and link design classes to SVG elements. After that, SVG code is added to an HTML document that contains all the main structure of an HTML page that will be displayed. Accessible attributes are added to SVG code integrated to HTML document, for example, tabindex to navigate the map with the Tab key, <text> to add visible graphic descriptions to map elements, <desc> to include a more detailed description of the plan components, <title> applied to each map to provide descriptions visible when the focus is on a specific element. ARIA attributes should be added to specify roles for each HTML attribute, as well as properties in regions of the web page, to include keyboard navigation.

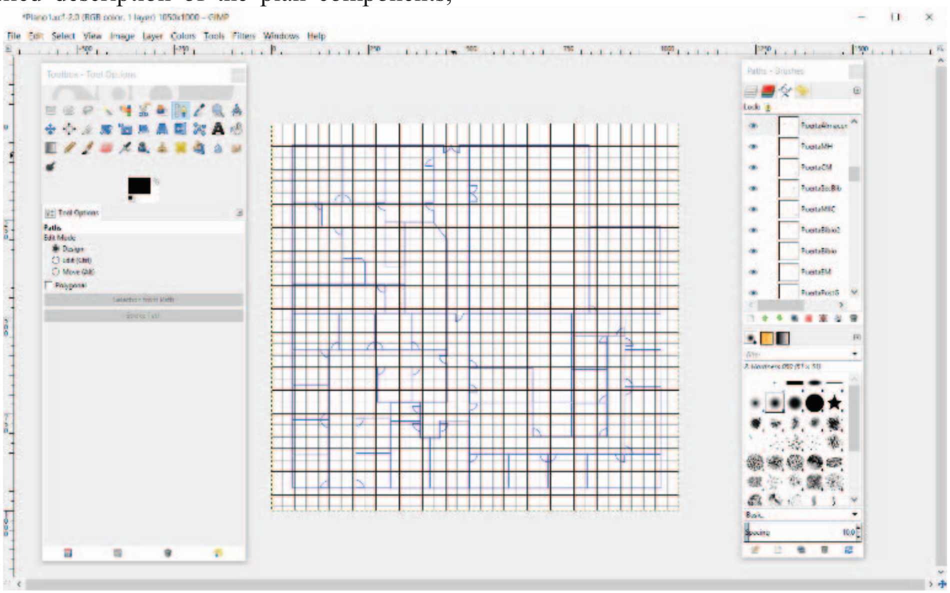

Fig. 4. Use of GIMP tool to plot the plan.

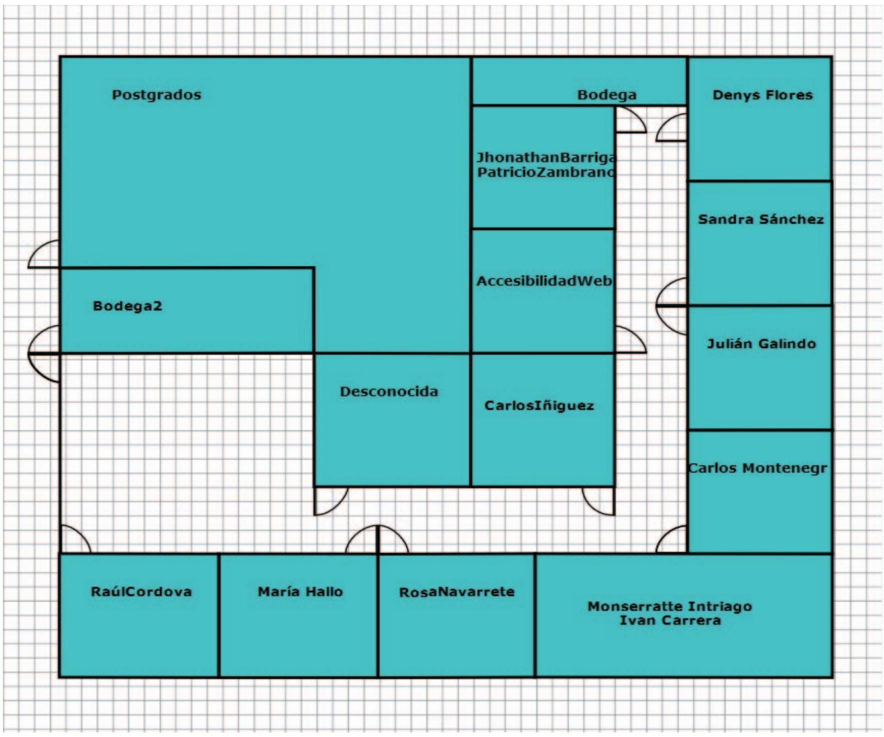

Fig. 5. Indoor map made with CSS, HTML, and SVG codes.

Fourth Step. Each team install and configures XAMPP as development environment to program logic in the accessible indoor map on a client-server scheme. For the server side, PHP is used as programming language. For the client side, JavaScript with JQuery are used. Figure 6 shows the final accessible indoor map for the practical example of the case study explained in the previous section.

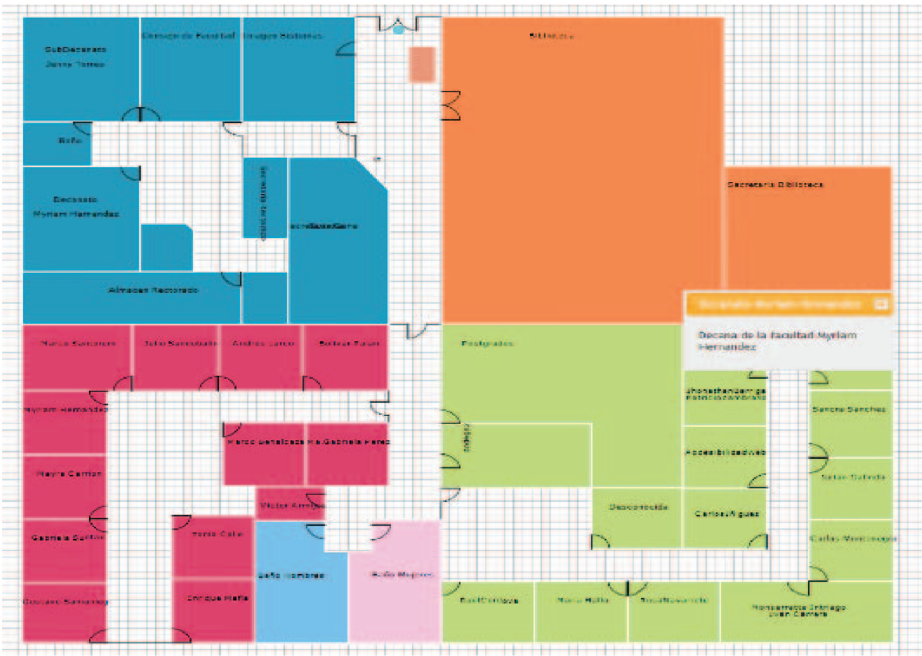

Fig.6. Accessible indoor map.

Fifth Step. Each team implement different routes within the indoor map for frequently needed points of interest. For this, JQuery is used to call a method that reads the database that contains the positions in (X,Y) for the route. Then returns that information in an array. This arrangement contains the positions in (X,Y) and sets a time interval between each coordinate graphing. There is a method that plots the route when called from a button on the indoor plan. The coordinates of the positions are dynamically draw resembling the steps of a person. Figure 7 and Figure 8 show the result of two routes plotted in the plan.

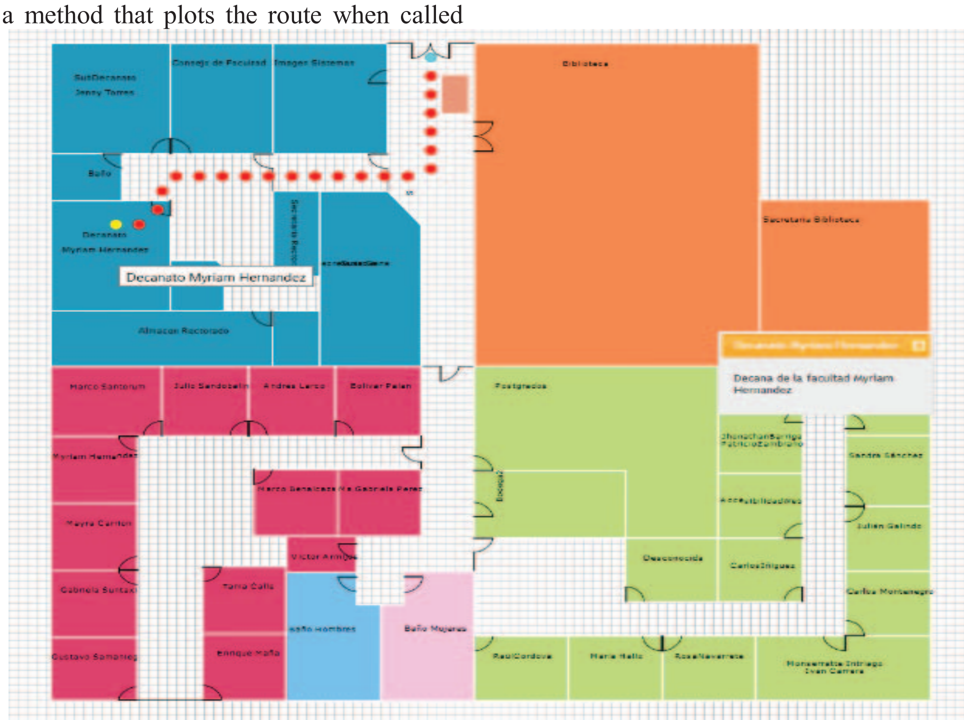

Fig. 7. Route to the Dean Office.

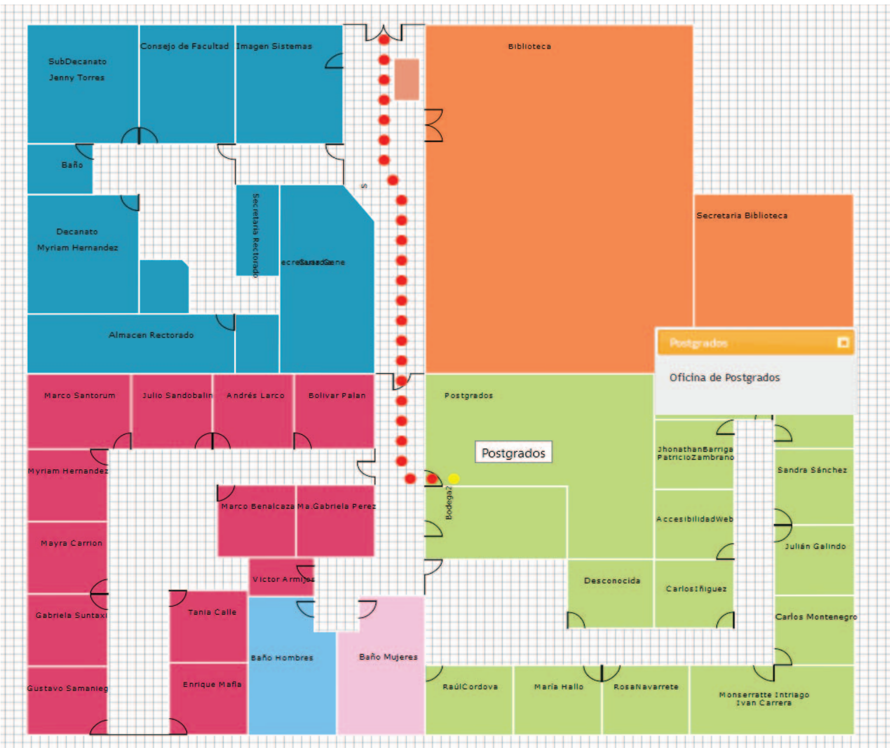

Fig. 8. Route to the Graduate Studies Office.

Sixth Step. Once the team has built the accessible indoor map, they have to perform evaluations with TAW and Examinator tools to determine the level of accessibility of their map. The evaluations results give information to improve or add new functionality to increase accessibility. A serious game is used in this step to motivate teams to improve the grade their map obtain and achieve the most accessible indoor map of the course. Extra grades are allocated to the winner team.

Seventh Step. Each team has to test their accessible indoor map with different screen readers and write a report with the results.

Eight Step. An accessible website is used to publish the indoor maps with the routes defined by all the teams. The website has the following structure. The first tab shows the overview of the project and the case study. The second tab contains information of the teams. The third tab shows the integrated accessible indoor map, as shown in Figure 9.

Fig.9. Accessible web site with accessible indoor map.

VI. Conclusions and Future Work

The conclusions of this study can be divided in two types: those regarding to the engineering solution built and those related to the pedagogical approach used to enable a collaborative learning experience for engineering students.

With the development of this project, engineering students achieved a better understanding of the difficulties that blind people has to face when they want to access an unknown or unlabeled building. Also, they learnt that the barriers for navigating buildings that face people with disabilities also affects to people without disabilities. Good signage, labeling and a correct representation of the spaces distribution of the building can collaborate to improve its accessibility.

Following the steps presented as a practical example for engineering students, it was possible to provide a collaborative learning experience to build an accessible indoor map of the case study. Both the website and the indoor map built meet level AA of the WCAG 2.0 accessibility guidelines, since they have features and alternative texts that can be read by a screen reader and options to allow keyboard navigation.

By using technologies like HTML, CSS, JavaScript, and SVG to locate the position and the elements around a person, engineering students managed to build an indoor map that improve the perception of the environment around a blind person.

As future work, we propose generating two future collaborative learning experiences. The first one to integrate to the solution the recognition of the actual position coordinates obtained by a triangulation method. The second one to improve the level of compliance to meet level AAA of the WCAG 2.0 both at the website and the indoor map.

References

[1] M. E. Holzschlag, "Web Accessibility," in *Web Standars and Regulatory Compliance*, United States of America, Friendsoft, 2012, pp. 2-20.

[2] W3C, "Web Content Accessibility Guidelines (WCAG) 2.0," 2008. [Online]. Available: http://www.w3.org/TR/WCAG20/ [Accessed 31 08 2015].

[3] M. C. Buzzi, M. Buzzi, B. Leporini and L. Martusciello, "Making Visual Maps Accessible to the Blind," in *Universal Access in Human-Computer Interaction,* Springer Berlin Heidelberg, pp. 271-280, 2011.

[4] C. Goncu, A. Madugalla, S. Marinai and K. Marriott, "Accessible On-Line Floor Plans," in *ACM*, Australia, 2015.

[5] W3C, "WAI-ARIA Overview," [Online]. Available: http://www.w3.org/WAI/intro/aria [Accessed 07 09 2015].

[6] T. A. Powell, "Core Markup," in *HTML & CSS, The complete reference*, McGrawHill, 2010, pp. 55-60.

[7] W3C, "Scalable Vector Graphics," 2013. [Online]. Available: http://www.w3.org/Graphics/SVG/ [Accessed 02 09 2015].

[8] J. Ducket, "HTML & CSS : Design and Build Websites", Indianapolis, John Wiley & Sons Inc., pp. 226-242.

[9] M. Haverbeke, "A Modern Introduction to Programming," in *Eloquent JavaScript*, No Starch Press, 2014, pp. 6-8.

[10] M. A. Alvarez, "Manual de jQuery", 2012. [Online]. Available: http://dmaspv.com/files/page/07042011180222_manual%20de%20jquery%20en%20pdf%20desarrollowebcom.pdf 2012 [Accessed 04 09 2015].

[11] A. Project, "Apache HTTP Server Project," 2015 [Online]. Available: http://httpd.apache.org/ [Accessed 04 09 2015].

[12] W. Skaggs, Ć. Gémy , J. Hardelin , R. Ostertag , M. Boyce , D. Egger, R. Joost, O. Ellis and M. Quiñones, "GNU Image Manipulation Program," 2007.

[13] D. Ho, "About Notepad++," 2015. [Online]. Available: https://notepad-plus-plus.org/. [Accessed 07 09 2015].

[14] Examinator, "Evaluación de la accesibilidad web," 2015. [Online]. Available: http://examinator.ws/. [Accessed 04 09 2015].

[15] TAW, "TAW Analizador WCAG 2.0," 2010. [Online]. Available: http://www.tawdis.net/. [Accessed 04 09 2015].

[16] NVACCESS, "What is NVDA?," 2015. [Online]. Available: http://www.nvaccess.org/. [Accessed 07 09 2015].

[17] Google, "Introducing ChromeVox," 2015. [Online]. Available: http://www.chromevox.com/. [Accessed 07 09 2015].

[18] M. Stepp, J. Miller and V. Kirst, "HTML Basics," in *Web Programming Step by Step*, Washington, 2009, pp. 14-17.

[19] D. Duce, I. Herman and B. Hopgood , "SVG: An introduction," in *SVG Tutorial*, Oxford, 2013, pp. 7-10.

[20] P. Borowska, "The Next Big Thing: Responsive Icons," 2013. [Online]. Available: http://designmodo.com/responsive-icons/. [Accessed 02 09 2015].

[21] T. A. Powell, "Core Style," in *HTML and CSS: The Complete Reference"*, McGrawHill, 2010, pp. 430-438.

[22] T. Navarrete , "El lenguaje JavaScript", Argentina, 2006.

[23] W3SHOOLSs, "jQuery Introduction", 2012.

[24] PHP, "PHP: Hypertext Preprocessor," 2012. [Online]. Available: https://secure.php.net/. [Accessed 05 09 2015].

[25] OMS, "Ceguera y discapacidad visual," Centro de prensa, 08 2014. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs282/es/. [Accessed 16 08 2015].

[26] W3C, "Guía Breve de Accesibilidad Web," Web Accessibility Initiative, [Online]. Available: http://www.w3c.es/Divulgacion/GuiasBreves/Accesibilidad. [Accessed 16 08 2015].

[27] WBU, "Plan estratégico 2013-2016," 11 2013. [Online]. Available: http://www.worldblindunion.org/Spanish/Our-work/strategic-plan/Pages/default.aspx. [Accessed 16 08 2015].

[28] ONCE, "Discapacidad visual: Aspectos Generales," [Online]. Available: http://www.once.es/new/servicios-especializados-en-discapacidad-visual/discapacidad-visual-aspectos-generales. [Accessed 16 08 2015].

[29] A. Dix, "Encyclopedia of Database Systems," in *Human-Computer Interaction*, US, Springer, 2009, pp. 1327-1331.

[30] J. A. Voos , "Portal de Aplicaciones," Cordoba-Argentina, 2010, pp. 1-8.

[31] UNAD, "Sistema de Coordenadas Cartesianas," in *Coordinación de Matemáticas*, Monterrey, 2011, pp. 1-2.

[32] S. Duffey, "A More Accessible Map," 18 04 2006. [Online]. Available: http://alistapart.com/article/cssmaps. [Accessed 01 09 2015].