

May SmartPhones Help to Maintain Audience Attention During Presentations?

Matteo Ciman¹, Yari Formaggio¹, Ombretta Gaggi¹ and Marco Regazzo²

¹*Department of Mathematics, University of Padua, via Trieste, 63, Padova, Italy*

²*Primo Round S.r.L., via Durando 39, Milano, Italy*

Keywords: Interactive Presentations, Slideshows, Mobile Applications.

Abstract: In this paper we describe a new tool for interactive presentations: the speaker shows a common slideshow on the screen, and the users can see the same slide in their smartphone or tablet, independently from its model and brand. The system can be used for both *on-line* and *on-site* presentations and encourages the user to perform social activities, e. g., comment or like a slide. The author can also submit a questionnaire or a poll to the audience and see in real-time the answers. Our tool has been used during 37 different events, followed by 3753 users. We recorded the actions of the users and, depending on the different kind of events, the average of actions ranges from 1 to 8 per user per presentation. This data shows that our tool helps to increase users engagement and to maintain user attention.

1 INTRODUCTION

One of the key problems for a speaker is to maintain user attention. This problem is particularly relevant for teachers, who has to transmit knowledge to their students, or for speakers who presents a new product, e. g., a medicine, to potential buyers. For this reason, computer-based multimedia presentations has rapidly replaced the use of chalkboards, overheads and hand-outs, both in academic lectures and in presentations in conference rooms. (Guadagno et al., 2011) even demonstrate that the use of animated PowerPoint® presentations may affect the persuasive power of the lesson/event on users' decisions. Users perceive technological sophistication of a presentation as a heuristic cue for high quality of the content, therefore it inflates positive evaluations from users. Moreover, (Snell, 1999) presented several reasons why interactive lecturing and presentations are better than a static presentation, e. g., active involvement can increase student learning, attention and motivation and can also provide feedback not only to the student but even to the teacher, with a double benefit.

Although the persuasive power of computer-based multimedia presentations, they do not completely solve the problem, since it often happens that many users do not follow a presentation from its beginning to the end, due to the limited attention threshold. The diffusion of smartphones and tablets increases this

problem, since a bored audience often begins to read email through smartphones or tablets or to perform social activities like using Facebook or Twitter. But can smartphones be used in a useful way?

In this paper we propose a system to involve users during talks with the aims at turning smartphones from a distracting tool, i. e., an enemy for the speaker, into a useful tool for interaction for the user itself or between the speaker and the user, thus encouraging the user to keep attention on the presentation.

Our system allows to show PowerPoint® presentations both through a projector and users devices. The speaker follows the natural trend of the presentation, can submit to the audience questionnaires or polls and receive in real-time the answers, showing the results on a dedicated slide. Moreover, it is possible to add comments on a slide that can be seen immediately or in a second moment by the presenter.

Due to the widespread diffusion of PowerPoint®, our system is based on an editor which translates a presentation from its format to web pages, and allows the insertion of questionnaires and polls. This clearly aims at removing the learning curve, that is typically associated with the adoption of a new technology, and at targeting the highest number of presenters, even those not skilled with HTML5, CSS and Javascript languages. We also support other formats, e. g., ODF, as discussed in Section 5.

2 RELATED WORK

PowerPoint® presentation conversion to web pages and audience engagement has been investigated in literature from several points of view.

Joubert et al. (Joubert et al., 2011) converts presentations into compressed scalable vector graphics. The conversion is performed to adapt presentations to phones which are not able to read PowerPoint® format, but they do not provide any kind of interaction more than stepping to the next slide.

Commercial tools like Adobe Captivate¹, iSpring² and SlideGo³, that let the user convert from PPTX format to HTML5 pages, besides being expensive, produce not valid pages or introduce SWF elements inside the pages. Another solution adopted by these products is the simple conversion of a slide into an image, which is placed inside an HTML page. All these solutions do not allow the portability of the system over all the mobile platforms. Our approach can be used both with a desktop, a tablet or a mobile device. Moreover it adapts the slides to user preferences, like the dimension of the font.

Interaction between the speaker and the audience is studied in (Triglianos and Pautasso, 2013). ASQ allows to deliver interactive HTML5 presentations which can contain questionnaires and polls, but the teacher must create a web presentation which is a skill not commonly available among teachers. The same problem is also present in (Edge et al., 2013), where the system, although it uses PowerPoint® application and slide format as a basis, requires to the authors the knowledge of a simple markup language.

In early 1993, Nickerson (Nickerson, 1993) registered a patent for a real-time wireless audience response system to collect data from a plurality of participants. This idea could be identified as the beginning of the active participation in public talks, where users are engaged and requested to provide their opinion or data to particular questions.

To increase audience engagement, Maynes-Aminzade et al. (Maynes-Aminzade et al., 2002) developed a system that lets the audience control on-screen activities using three different interactions: leaning left or right on the seats, batting a beach ball while its shadow is used as a pointing device and pointing lasers at the screen. Although these interactions cannot be used with professional audience, it was clear that this sort of interaction could easily engage people and increase their attention.

¹<http://www.adobe.com/it/products/captivate.html>

²<http://www.ispringsolutions.com/ispring-converter>

³<http://www.slidego.com/>

Barkhuus et al. (Barkhuus and Jørgensen, 2008) investigated the usage of a voting system at rap competitions. They showed how augmented interaction can increase the sense of participation among the audience. One investigation on a more serious context was carried on by Jones et al. (Jones et al., 2006). In this case, they investigated the usage of text messaging as answer provider to polls submitted by the speaker. The system receives the SMS with the answer to the poll, records the data and let the speaker show the overall answers provided. From their tests, they showed how about 15% of the audience directly took part at the polls, that could be higher if the students had not to pay the costs of the SMS.

Differently from these approaches, our system does not require any programming skills or to change the tool to create presentations, but allows to use presentations already available for lessons or talks and to integrate more interaction through a simple web page. Moreover, it is portable over all the operating systems and mobile devices that integrate a browser that supports HTML5. Our system does not translate each slide into an image or a canvas, thus losing accessibility and portability of its content, but it translates the slide in pure and valid HTML code, i. e., paragraphs, tags <p>, for text and tags only for pictures and icons. In this way we preserve not only portability and accessibility, but each slide adapts itself also to user preferences as font size.

3 SYSTEM ARCHITECTURE

Our system must support both *on-site* presentations, i. e., audience and speaker meet together in a room, and *on-line* events, i. e., the presentation is distributed through the web, but there is not a speaker in the users' room. In the first case, the speaker presents the slides to the audience, following the natural trend of the presentation. The main objective is to increase the attention and the engagement (interaction) of the audience with the speaker, and viceversa. Therefore, the system supports the possibility for the user, i.e. a person of the audience, to add a "like" or a comment on a particular slide or, in some cases⁴, navigate through the slides, without waiting for speaker presentation. Moreover, the system supports the possibility, for the speaker, to add a slide with a questionnaire or a poll. The idea is to use these slides to interact with the audience, and receive in real-time the answers. The user initially sees only the question and the possible answers, then the system collects in real-time all the

⁴If the speaker enables this action during the configuration phase.

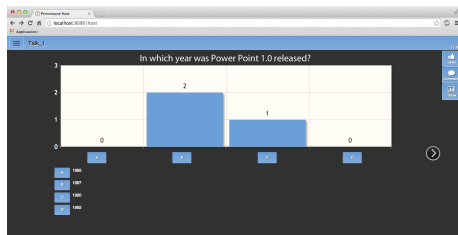


Figure 1: Slide report with the result of a questionnaire.

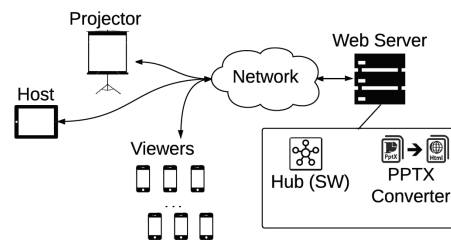


Figure 2: System architecture.

answers, elaborates them and shows the results in a dedicated slide (see Figure 1).

During *on-site* presentations, “like” and comments are notified in real-time to the speaker, hence giving him/her a feedback of the audience response during the presentation. Moreover, if comments from the audience are enabled, this immediate notification gives him/her the possibility to answer immediately to possible questions.

Our system supports even *on-line* presentations, where the presentation is recorded and spread through the Web. In this case, presentation timing is managed by the user, who can also submit “like”, comments or questions to which the presenter can answer in a second moment through email.

The system can be divided into a *front-end* and a *back-end* component. The *front-end* component is responsible to manage all the possible interactions between the speaker, the users and the system. First of all, it provides the configuration pages used to upload and convert PowerPoint® presentations and to add interactivity, i.e. questionnaire slides. Secondly, the *front-end* component provides the proper interfaces during the presentation. We identified three different users’ roles that correspond to three different interfaces: the *Viewer*, the *Projector* and the *Host*. Figure 2 depicts all these components and their interactions. We call *Viewer* the interface dedicated to the audience and is responsible to manage users interactions. Through this interface, an attendant is able to interact during the presentation submitting comments, answering to polls or sharing slides.

An example of the interface is provided in Figure 3. Comments and “like” are notified to the speaker with two icons, a thumb up and a cartoon bubble with the number of received “like” and comments.

The *Host* is the interface dedicated to the speaker. During an *on-site* event, the speaker has the control of the slideshow. He/she can move to the next/previous slide and controls slides’ animations and interactive slides. Moreover he/she can read, in real-time, audience’s comments. The system allows to enable or not the possibility to show comments to the speaker during the initial configuration phase. If enabled, the



Figure 3: Example of interface provided for the *Viewer*.

speaker, or the user, can click on the cartoon bubble icon to display the comments. A visualization in the *Host* interface is propagated to all the *Viewer*. A visualization in a *Viewer* interface is interpreted as a temporary de-synchronization from the presentation. The speaker can decide to verbally answer to the comment in the case of *on-site* presentation, otherwise he/she can answer to the user through e-mail.

The *Host* is a reference point for viewers synchronization and it rules the progress of the slideshow. As soon as the speaker moves to a new slide, viewers who are following the speaker, i. e., who are not moving autonomously among the slides⁵, are also automatically synchronized with the current slide. Even if very different in terms of role inside the system, from a graphic point of view, the *Host* interface is very similar to the *Viewer*: it shows the number of comments and “like”, it allows to see a particular comment and to answer (this action is not possible in the *Viewer*). In case of questionnaire, the *Host* shows the collected data as depicted in Figure 1 and the *Viewer* shows the question and the list of possible answers.

Finally, the *Projector* is the interface dedicated to the room’s projector, if available. This is the most simple interface because it only displays the

⁵The speaker can decide to give or not this possibility.

slideshow and it does not allow any input or interaction. It also displays the number of comments and likes of the slide currently shown.

The *back-end* component, the rectangle in Figure 2, contains a *PPTX Converter* and the *Hub*. The former converts a PowerPoint® presentation into a set of HTML5 slides, the latter manages all the signals coming from the host and from all the viewers, to maintain the synchronization between the speaker and the audience and to collect users' feedback.

We must note here that a user from the audience may temporarily deviate from the state of the presentation imposed by the speaker, navigating through the slides. The *Hub* does not re-synchronize this kind of users, since a deviation from the current presentation state is interpreted as the wish to navigate autonomously. The user can decide, in every moment, to re-sync with the presentation displayed by the *Host*, clicking on the dedicated button. This is possible thanks to the *Hub* which knows, in every moment, the current state of the slideshow. Moreover, the *Hub* is also used to send the correct slide to new viewers that connect when the slideshow is already begun. In this way, the *Hub* always guarantees synchronization between speaker and the audience.

4 SYSTEM DEVELOPMENT

4.1 Managing Synchronization

The main task of the *back-end* component of the system, and in particular of the *Hub*, is to maintain the synchronization between all the devices connected to the server, e.g. the presenter and audience devices, and to manage all the exchanged messages. The server is implemented using *NodeJS*, in particular with the *Socket IO* library (*Socket IO*, 2015), that lets manage several parallel websocket connections reducing transferred data and increasing synchronization speed. As previously mentioned, the *Hub* component has to be aware about the current slide of the presentation, in order to be able to keep synchronization between all the connected devices, and to sync a new user that connects to the system.

In particular, the most important messages exchanged with the *Hub* are:

- **change-slide:** from *Host* to the *Hub*, notifies that the presenter has changed slide. This message is then forwarded to all the followers currently synchronized with the presentation to move to the next slide;
- **update-like** and **update-comment:** from the *Hub*

to the *Host*, is used to update the number of “like” or the comments of a slide of the presentation;

- **user-resync:** from the *Viewer* to the *Hub*, used to sync the presentation of a viewer with the current slide of the presenter. The *Hub* will answer with the current slide of the presentation;
- **user-navigation:** from the *Viewer* to the *Hub*, informs the *Hub* that the current device is no more following the presenter, but is freely watching the slideshow.

Other signals between the *Hub*, the *Viewer* and the *Host* are exchanged to manage all the possible actions that can be performed with our system.

As mentioned before, the system supports interactivity between the presenter and the audience, in particular using *interactive slides*, that let the presenter ask questions to the audience, i. e., multiple choice questions or poll. When asked, people from the audience have a particular slide that is dedicated to let them submit their answer to the question. A specific message with the provided answer is then sent to the *Hub*, that collects all the answers from the audience and will send the final report to the *Host*. Then the *Host* itself shows the result in a proper slide. An example of the results of a poll is provided in Figure 1.

4.2 Module Converter

The module converter translates a PowerPoint® presentation into a set of HTML5 slides. We support only the .pptx format and do not support the old format .ppt, since the latter is not implemented using XML language. A description of the support of other formats is discussed in Section 5. We aim at, as accurately as possible, translation of the PowerPoint® presentation, considering images, text, animations and transition effects.

The core of this module is an XML parser that parses, in succession, the XML files that represent the slides and identifies the elements contained in every single slide. Analyzing these elements, the converter creates an HTML5 page adding the corresponding HTML5 element, if there is one, or adding a particular combination of HTML5 elements to reproduce the PowerPoint® object.

Consider, as an example, a slide containing an image: the converter creates a *div*, and sets the right image as source for its *background-image* property. The basic information that are needed to correctly translate a PowerPoint® image are: the *source image*, its *size*, its *position* and its *rotation*. The information about the *source image* is contained in the *r:embed* attribute of the `<a:blip>` XML element: *rId8* represents a unique ID that can be used to identify the

correct image (Fig. 4(a)). This ID must be searched into a PowerPoint®'s *relationships file*, (Fig. 4(b)), that is strictly connected to the slide XML file that contains the picture element. The `Target` attribute specifies the image file for the picture identified by the `rId8` ID.

Picture's *position*, *size* and *rotation* are contained into the `<a:xfrm>` XML element. The root attribute contains the image *rotation*. It is a simple type that represents an angle in 60.000ths of a degree. Positive angles are clockwise (i.e., towards the positive y axis); negative angles are counter-clockwise (i.e., towards the negative y axis). In this example the image is rotated 90 degrees clockwise.

The `x` and `y` attributes of the `<a:off>` element, describe the picture *position*. Values are measured in EMUs (English Metric Units)⁶ and they represent a distance from the top-left corner of the slide. In this example, the image has no offset.

The `cx` and `cy` attributes of the `<a:ext>` element, express the picture *size*, measured in EMUs. In our example, the image has a size of 720x540 points.

Moreover, if the slide contains a text area, the converter creates a `div` element, puts the text inside it, and sets all the necessary properties to maintain

```
<p:pic>
  <p:blipFill>
    <a:blip r:embed="rId8">...</a:blip>
    ...
  </p:blipFill>
  <p:spPr bwMode="auto">
    <a:xfrm rot="5400000">
      <a:off x="0" y="0"/>
      <a:ext cx="9144000" cy="6858000"/>
    </a:xfrm>
    <a:prstGeom prst="rect">...</a:prstGeom>
    ...
  </p:spPr>
</p:pic>
```

(a)

```
<Relationships xmlns="...">
  ...
  <Relationship Id="rId8" Target="../media/
    image5.jpg"/>
  ...
</Relationships>
```

(b)

Figure 4: XML fragment of a .pptx presentation containing an image (a) and its relationship file (b).

the correct font, font size, font color, alignment, line spacing and format, used in PowerPoint®.

⁶A EMU is defined as 1/360.000 of a centimeter and thus there are 914.400 EMUs per inch, and 12.700 EMUs per point.

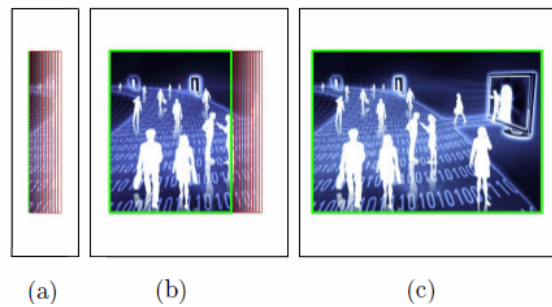


Figure 5: Wipe-in animation implemented in HTML5 and CSS3. (a), (b), (c) show, progressively, how the main `div` element, i. e. the green one, is expanded and how the red rectangles are moved according to it.

The converter is written using the *NodeJS* platform and Javascript language. First of all, it reads the picture's information, described above, from the XML file. The second step is to convert this information into valid CSS values. Images' position and size are converted in pixel, and images' rotation values in degrees; the size of the `div` element and of the font are scaled, according to the window size, to allow to adapt the slides to every device.

Moreover, since the slides are scaled to meet the screen size of different devices, also the images contained in the presentation must be scaled. This operation is performed on server side to obtain a second, important, goal: to save network bandwidth. In fact, it makes no sense to send a big image file, thus wasting time and bandwidth, to show it smaller, according to final user interface.

The converter is able to translate slides which contains objects like images, text-areas and shapes (only squares and rectangles). The system can reproduce user defined layouts with good results and can recreate a small set of PowerPoint® animations as fade-in, fade-out, wipe-in (from left, right, bottom, top), blink, entrance (from left, right, bottom, top) and exit (from left, right, bottom, top). The set of supported features, detailed in Table 1, is not complete but covers the most commonly used features, as discussed in following section.

4.3 Animations

Animations have great importance and they are translated in the slides using features offered by CSS3 standard and Javascript. The converter reads the animation information from the XML file: i. e., type, duration, involved elements and triggers. For each animation a specific CSS class is created and applied to the correct HTML5 element, at the right moment. Animations are played in real-time by a Javascript func-

Table 1: Summary of supported elements and features of a PowerPoint® presentation.

Elements	Features
Text	position, font, size, color, alignment
Images	position, source, size, rotation, cutting
Text Boxes	position, size, color background, text (font, size, color and alignment)
Bullet lists	position, color, type, text (font, size, color and alignment)
Shapes	
Square, rectangle	position, size, color background, text (font, size, color and alignment)
Animations	
Fade-in, Fade-out	duration, trigger (click or timer)
Slide-in, Slide-out, Wipe-in	duration, trigger (click or timer), direction (from left, from right, from on high, from on low)
Slide Layout	
Layout Grid	bullet list, title, etc. are displayed according to the chosen layout
Background	Slide background color
Placeholder	header, footnote, date, slide count, title, and text inherit their features from the layout of the slide

tion executed by user's internet browser.

It is important to note that the creation of animation inside HTML5 slide is not a simple translation between XML tags. This comes from the fact that it does not exist a correspondent CSS property for each animations provided by PowerPoint®, but only for the easiest one like *fade-in* and *fade-out*. Other type of animations are supported manually creating the animation effect.

Consider as an example the *wipe-in (left)* animation. It is an animation effect which reveals, progressively, an object placed on the slide, increasing its width, starting from a value equal to 0 pixels (the picture is not visible) and reaching the size of the object (the image is completely visible). To further emphasize the animation, a gradient effect is also applied along the vertical right side that reveals the image. This animation is implemented with the combination of two simpler effects. The first one changes the dimension of the `div` element which contains the object, colored in green in Figure 5. This animation is realized using a CSS3 transformation. The second animation adds the gradient effect and its implementation is more difficult. We create a number of `div` elements which contain only a slice of the original object. Then, different values of transparency are applied to these elements to simulate the gradient effect. Moreover, all these elements are moved according to the main `div` containing the object. The elements which realize the gradient are colored in red in Fig-

ure 5. To better support mobile devices, this second animation is *facultative* and it is not applied in case of old device or device with low CPU resources.

5 TEST

5.1 Coverage Test

We tested our system in different ways. First, we collected 16 sets of slides, created by 13 different authors, for a total of 449 slides, and we tested how many slides have been correctly translated and if any error occurs during conversion. Presentations has been selected from lectures' slideshows and presentation from commercial events; 15 out of 16 presentations contain images, 9 out of 16 presentations contains animations. The authors were not involved in the development of our system and did not have any notion about how it is built and its coverage of the features under evaluation.

As already discussed in the previous section, our converter does not fully support all PowerPoint® features, but a reasonable set. We want to explore the effective coverage of features, since some PowerPoint® features are more widespread and other are seldom, if ever, used by the authors.

We must note here that some features, if not supported, can be easily replaced by other without loss of expressiveness of the presentation, e.g., a complex animation can be replaced by a fade in. We asked to one of the authors of the presentations and a developer of the system to grade the conversion of slides from PowerPoint® using this rating scale:

- 0, if the converted slide lacked of important elements (or they are empty) and the slide failed to convey its content;
- 1, if the converted slide contained some errors which can be considered as "minor" since they allowed to convey the slide's content but with some difficulty by the users, as depicted in Figure 6, and
- 2, if the converted slide was equal to the original or conveyed with the same efficacy its content.

We chose an author and a developer for the grading to combine a *black-box approach*, i. e., the author did not know anything about the development of the system, and judgment from a person that knew very well the system. When they did not agree on the grade, we reported the opinion of the author since she was not biased by grading something she has contributed to develop. Therefore, reported results can be considered a lower bound.

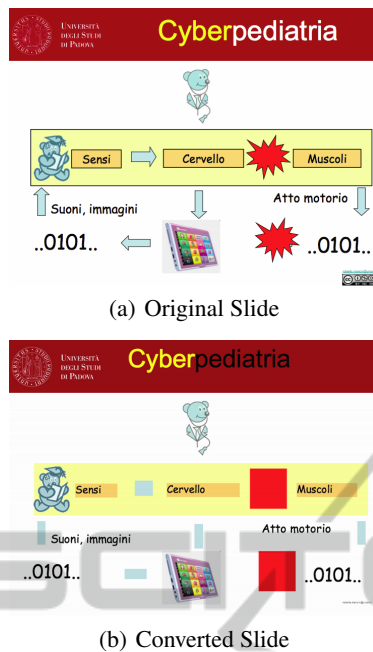


Figure 6: An example of slide whose conversion contains some errors: the font color of the title and the some shapes (arrows and explosions) are not supported.

For each set of slides, we calculated the number of total points available, i. e., the number of slides multiplied by two, and the number of total points collected. Our test showed that, on average, the presentations obtained 86% of available points, 78% of the slides were fully graded, 19% of converted slides contained some errors in the translation which can worsen the user experience and the efficacy of the presentation to convey information, and only 4% of the converted slides were evaluated to 0. Although 19% could seem a large percentage, we must note that a slide was graded to 1 if it was not equal to the original one, but it was still able to convey the whole content. So only the 4% of slides were not able to provide all the information to the audience. The complete results of this test are reported in Table 2, where we can see that two presentations were fully graded.

These results showed that, even if the coverage of PowerPoint® features is not complete, but we plan it would be in the future, it allowed to translate efficiently many presentations. As an example, text was always completely converted, but some problems might arise when bullet lists were included in the default layout and not explicitly defined. In this case the tool created a list of items, without the bullet, therefore the content was correctly conveyed. At the moment, the list of unsupported features contains, among others, geometric figures (we implement only squares and rectangles), gradients and tables. Only

Table 2: Percentage of slides evaluated with 0 (Class 0), 1 (Class 1) or 2 (Class 2) points for each set of slides analysed during the test phase.

Set #	Class 0	Class 1	Class 2
Set 1	0%	0%	100%
Set 2	0%	28,57%	71,43%
Set 3	3,54%	23,01%	73,45%
Set 4	10,53%	36,84%	52,63%
Set 5	9,68%	16,13%	74,19%
Set 6	0%	25%	75%
Set 7	0%	13,64%	86,36%
Set 8	26,67%	13,33%	60%
Set 9	7,14%	16,67%	76,19%
Set 10	0%	26,32%	73,68%
Set 11	0%	21,5%	87,5%
Set 12	0%	20%	80%
Set 13	0%	25%	75%
Set 14	0%	22,22%	77,78%
Set 15	0%	54,55%	45,45%
Set 16	0%	0%	100%
Average	3,56%	18,71%	77,72%

some types of animation are supported, in case of unsupported types, the converter introduces a fade in/out according if the animation puts or removes the object on/from the screen.

We must note here that, even if the set of presentations was relatively low, 16, they covered different authors, coming from different environment, ranging from public universities to companies, and even the presentations goals were very different (lessons, advertising, promulgation, etc).

5.2 Tests with Real Users

As a second test of the system, we tried to “measure”, in some way, how much the system helps to increase users engagement and interaction. Considering that the user probably has paid attention to the talk if he/she declare to like one or more slides and, in the same way, the user has to pay attention to the presenter to be able to answer to a questionnaire, the number of interactions can be considered a rough “measure” of the potential of the system to help to maintain users attention during a presentation.

We analyzed the behavior of students of a course of “Web Technologies” at the undergraduate program in “Computer Science” at the University of Padua in Italy during two lessons. Both the lessons proposed the same topic, exercises on XML technologies (i. e., XMLSchema and DTD) and were organized in the same way. The teacher proposed some exercises to the students and their solutions. A set of eight closed questions with a predefined set of possible answers

helped the discussion about the goodness of the proposed solutions.

Since the students were the same for both the lessons, exercises and questions were different to avoid annoying them. In the first lesson, we used the system described in this paper to allow the students to answer the questions and to record data about interaction. In the second lesson, the students were required to raise their hand to answer to the question when they agree with one of the possible answers and the teacher counted the number of raised hands to record data.

During the first lesson, 96 users registered to the system, 55 of them as “anonymous”. The system recorded 215 answers to the eight questions, on average, nearly 27 answers to each question. This number could seem low considering the number of users following the lesson, but we must note here that the questions were not easy for the students and they knew the system was recording their interactions so they want to avoid to give wrong answers. Nevertheless, considering only the authenticated students, the number of answers per users was, on average, 2,68.

The teacher counted 94 students attending the second lesson. In this case the number of answers to each question was, on average, 4,5, so the use of the system, during this test, increased nearly six times the number of answers. We must note here that this result can be due to many factors as, for example, the age and gender of the participants, but it is clear that the use of a device to answer, instead of a raised hand, can help to increase the number of received answers in a very significant way.

Finally, we collected data in 37 commercial presentations, where the system was used by different companies which aimed at increasing users engagement and attention during the presentation of their products. Five out of these 37 events were *on-line* presentations, the others were *on-site*. During these events, 3753 users accessed the slideshow using our system (2372 during the two on-line events, 1381 users were following the events on-site). Unfortunately, we were not always able to collect the number of people attending the events, but in the 11 events for which we know the number of attending people, on average, 75% of the audience followed the slideshows using their smartphones or tablets, with a peak of 100% for an event.

During the 37 slideshows, the users liked, on the whole, 4842 slides; the average of like operations per user is 1,66. In particular an event reaches 8,59 “like” per user.

The users filled out 2266 questionnaires, shared 38 slides on Facebook and sent 175 slides by email. Moreover, the system collected 920 comments.

All these data shows that the system is able to persuade users to interact with the presentation, and this is an interesting result. When we can collect the number of participants, we recorded that the system has involved, on average, 75% of the audience. Moreover, by interacting with the presentation, the users also gave some information about themselves to the server, e. g. the email, and this is really useful during commercial presentation since this data can be used to re-contact people after the event. In the same way, information like time spent viewing each slide can be used for commercial purpose, since it can help to understand what has most interested the user.

Since the number of interactions with the system is only a rough measure of its ability to engage users, at the end of the last three events we asked to the audience to answer a question about their satisfaction with the system: 1090 users answered the question (85% of the total audience), 63% of the users gave a very positive evaluation, 36% gave a positive evaluation and only 1% of the users gave a negative evaluation of the system, therefore we can state that the users liked the proposed system.

Finally, these tests have shown that the system is not simply a prototype but its coverage of PowerPoint® features is sufficient to make it usable. Moreover, during all these tests, both the users and the speakers never reported problems in the synchronization of the slides.

5.3 Tests with Other Formats

During the whole test phase we considered also other format, i. e., presentation created with Apple Keynote® or OpenOffice. In particular, we studied the ODF format(Organization for the Advancement of Structured Information Standards (OASIS), 2005), but it resulted completely different from the OOXML format(ECMA International, 2012) used by PowerPoint®, therefore our converter is currently not able to support this format. But our system, with some limitations, can be used also to distribute presentations created with Keynote® or OpenOffice, using a converter from the output from these software to the OOXML format.

We tested three set of slide generated by other software and then converted to the OOXML format. The obtained results, converted with our system, were evaluated with the same method described above: each slide was evaluated to 0, 1 or 2 according to the quality of the conversion, by the same people discussed in Section 5.1.

The results shown that, on average, the presentations got 89% of the total available points. In partic-

ular, on average, 83% of the slides were fully graded, 12% received a score of 1 point, and only 5% of the slides were evaluated to 0. Therefore, we can assume that even in case of presentation generated with other software, the result of the converter is quite the same. This means that files which use ODP format, converted to the OOXML format, are usable as an input file for our converter, thus expanding the type of supported file, and so, the number of convertible presentations. However, in this case, in addition to the errors introduced by our system, there may be additional errors generated by the conversion of a presentation to the OOXML format.

6 CONCLUSION

In this paper we propose a system to distribute presentations to smartphones and tablet to make presentations more interactive, adding questionnaires and polls, and to improve users engagement.

The main component of our system is a converter which is able to translate PowerPoint® slides into web pages using HTML5. The user may access to the slides, comment on them, and make other social activities. Our tests showed that the system was able to involve, on average, 75% of the audience, and to make users more interactive (one test reported up to six time more interactive) with respect to presentation based only on slides. Therefore, the use of smartphone to distribute slides is promising.

Moreover, interaction can continue also after the talk, using the system in the *on-line* mode, which gives the possibility to navigate slides and leave comments. This last mechanism can be even more useful after the end of the talk to continue discussion with participants/clients. We must note here that our solution collect many data about the users, i. e., data about navigation and email, therefore some privacy issues arise. To partially solve this problem, the system allows the user to skip the authentication step and logs in as *anonymous*. In this case, the system does not record any email, so this user cannot be contacted in the future. However this is not a complete solution. Future works will be investigate how to clarify to the user that his/her activity is registered. Moreover, we want to increase the number of supported features and to explore the possibility to show users comments on a slide even on the projector screen. Actually, the projector shows the number of comments per slide, but not the comments themselves. We are currently implementing a new component, the *Moderator* interface, which can be used to select comment to display to all the users.

REFERENCES

- Barkhuus, L. and Jørgensen, T. (2008). Engaging the crowd: Studies of audience-performer interaction. In *Conf. on Human Factors in Computing Systems*, pages 2925–2930.
- ECMA International (2012). Standard ECMA-376, Office Open XML File Formats <http://www.ecma-international.org/publications/standards/ecma-376.htm>.
- Edge, D., Savage, J., and Yatani, K. (2013). Hyperslides: Dynamic presentation prototyping. In *Conf. on Human Factors in Computing Systems*, pages 671–680.
- Guadagno, R. E., Sundie, J. M., Hardison, T. A., and Cialdini, R. B. (2011). The persuasive power of powerpoint presentations. In *PERSUASIVE 2011*, pages 2:1–2:4.
- Jones, M., Marsden, G., and Gruijters, D. (2006). Using mobile phones & pdas in ad-hoc audience response systems. *ICI Global*.
- Joubert, J.-P., Greyling, J., and Cilliers, C. (2011). The conversion from powerpoint (ppt) to compressed scalable vector graphics (svgz). In *Conf. on Knowledge, Innovation and Leadership in a Diverse, Multidisciplinary Environment*, pages 123–132.
- Maynes-Aminzade, D., Pausch, R., and Seitz, S. (2002). Techniques for interactive audience participation. In *ICMI '02*, pages 15–20.
- Nickerson, R. (1993). Real-time wireless audience response system. US Patent 5,226,177.
- Organization for the Advancement of Structured Information Standards (OASIS) (2005). Open Document Format <http://www.opendocumentformat.org/>.
- Snell, Yvonne Steinert, L. S. (1999). Interactive lecturing: strategies for increasing participation in large group presentations. *Medical Teacher*, pages 37–42.
- Socket IO (2015). Socket IO Library <http://socket.io/>.
- Triglianos, V. and Pautasso, C. (2013). Asq: Interactive web presentations for hybrid moocs. In *WWW '13 Companion*, pages 209–210, Republic and Canton of Geneva, Switzerland.