User Interface adaptation for multi-device Web-based media applications

Mikel Zorrilla, Iñigo Tamayo, Angel Martin, Ana Dominguez

Abstract— **The quest to transform the television viewing experience into a digital media service is happening thanks to the addition of companion screens to the TV. Multi-device experiences become more intuitive and easier to use federating cooperative devices. They also bring new creative opportunities to schedule and distribute interactive content synchronised with the TV programme through any connected screen. The rise of HTML5 to develop responsive applications across multiple devices adds a significant amount of improvement enabling universal delivery. A key challenge to harness the power of navigation engaged with the story on the TV is the responsive design of a unique application spanning all the available screens. This paper presents user tests in order to explore the relevant parameters to create responsive User Interfaces for Web-based multi-device applications driven by media content.**

Index Terms— **Multimedia systems, Digital multimedia broadcasting, Interactivity, Future services of Broadcasting, Pervasive computing, Ubiquitous computing, Context-aware services, new human-device interaction.**

I. Introduction

COMPANION experiences engage consumers in relevant contents on second devices -smartphone, tablet or laptop- while watching something in the first screen -usually a TV-. The industry is adopting second screen viewing services to capture audiences and deploy new monetisation models.

Web-based applications enable universal delivery via HTML5 to reach those users who do not want to download an application. This trend is also driving the evolution of HTML5, making a wider range of devices capable of running applications that gain features previously available only through native SDKs [1]. Connected TVs are also following this trend towards Web technologies. HbbTV 2.0 [2] uses HTML5 and a set of related Web technologies including many modules from CSS level 3, DOM level 3, Web Sockets, etc. and provides mechanisms for the addition of companion screens to the TV.

Once the self-capacity and interoperability are being addressed by HTML5, the key challenge for next generation applications is to provide users coherent multi-device experiences with simple and intuitive interfaces that ease the navigation through the information provided with a right timing. This is a natural step in the adaption of the market and society to the growing behaviour of users accessing services from several devices simultaneously [3], aiming to have a single experience through multiple devices at the same time.

Despite the clear opportunities to catalyse new business models, the lack of standards hinders the creation of seamlessly connected, intuitively converged and conveniently continuous experiences across a heterogeneous ecosystem of devices. The priority for future research is shifting towards fully manageable, context-sensitive and controllable or self-regulating multi-device applications.

To overcome all the intrinsic features needed on a distributed web application, a complete capabilities stack is needed. It should comprise: a discovery service layer to federate other experience participants; a cross-platform user authentication layer for security; a communication layer to consolidate a synchronised multi-device context by means of autonomous information exchange and event triggering; and a cross-device application adaptation to distribute the experience across all the devices and suit the different visual components to each specific device conditions. The work presented in this paper is part of the research activities that are being performed in the MediaScape European project[1], which addresses the aforementioned challenges. In this case, this paper focuses on the User Interface adaptation challenge for media applications across multiple devices.

To really boost the interface creation and maintenance of multi-device Web applications, it is mandatory a universal mechanism that, based on Web standards, considers common possible situations that fire a set of orchestrated adaptation actions. Thus, it eases design and reusability while provides a default behaviour valid for a wide range of applications and contexts.

To this end, it is necessary to evolve application design

Manuscript received May 5, 2015. This work was supported in part by the European Commission under the FP7-ICT-2013-10 MediaScape project (Gran Agreement 610404, www.mediascapeproject.eu).

Mikel Zorrilla is with the Department of Digital TV & Multimedia Services, Vicomtech-IK4 GraphicsMedia.net. San Sebastian 2009, Spain (e-mail: mzorrilla@vicomtech.org; Phone: +34 943 309230; Fax: +34 943 309393).

Angel Martin is with the Department of Digital TV & Multimedia Services, Vicomtech-IK4 GraphicsMedia.net. San Sebastian 2009, Spain. (e-mail: amartin@vicomtech.org).

Iñigo Tamayo is with the Department of Digital TV & Multimedia Services, Vicomtech-IK4 GraphicsMedia.net. San Sebastian 2009, Spain. (e-mail: itamayo@vicomtech.org).

Ana Dominguez is with the Department of Digital TV & Multimedia Services, Vicomtech-IK4 GraphicsMedia.net. San Sebastian 2009, Spain. (e-mail: adominguez@vicomtech.org).

from developing different applications for concrete target devices and application roles, adding to each one ad-hoc mechanisms to control one by one their functionality.

The multi-device adaptation challenge addresses the seamless translation of a single application into a multi-device execution, providing a well fitted portion of the application on each device. And providing a responsive User Interface on each device is the goal of the research presented on this paper.

This approach differs from the current mechanisms to provide second screen experiences with standards such as HbbTV. [4] and [5] provide a solution where different applications need to be developed for the HbbTV device and for the mobile, while an event-driven server deals with the communication between them.

This paper presents user tests performed over 47 end users to analyse the impact of different parameters on how to arrange the User Interface for simultaneous Web-based multi-device media applications. These conclusions will help multi-device application developers to design a unique multi-device media application that will provide a coherent experience across multiple devices at the same time, with a responsive User Interface on each device depending on the parts of the application that is presenting regarding the multi-device dimension context.

II. State of the Art

This section provides a state-of-the-art of the existing technologies and frameworks to create responsive and device-adapted Web applications. Although there are available solutions and in-progress language recommendations in the field of adaptive interface of Web applications, these approaches are all designed for local adaptation, without considering that the application can be running in more than one device simultaneously.

W3C defined Web Components [6] with the aim of modularising Web applications. Web Components are made up of four valuable elements which empower recycling possibilities:

- **Custom elements:** personalized elements easily reusable in different Web applications including its behaviour.
- **HTML imports:** capability of HTML trees to include and reuse them in other Web applications.
- **Templates:** declaration of inert DOM (Document Object Model) subtrees ready to be instantiated in the final DOM.
- **Shadow DOM:** for maintaining functional boundaries between DOM trees and its interactions with each other within a document and enabling a better functional encapsulation.

However, native support of the browser is still limited. Therefore, there are some libraries, such as Polymer[2], that facilitate the development on top of Web Components specification and overcome the support lack with polyfills[3], providing universal browser compatibility.

Web Components provide a logic component definition of the Web application to make multi-device adaptation decisions on top of those. However, due to the enormous variety of existing devices, such as connected TVs, laptops, tablets and mobiles, Responsive Web Design (RWD) is one of the techniques that takes the lead in Web development. RWD returns a Web application able to adapt and change itself dynamically depending the target device, e.g. the capabilities of the device such as display size or resolution, providing a rich viewing experience.

To that end, HTML features a number of fall-back mechanisms. In order to obtain a responsive Web application, two different approaches could be followed. On the one hand, if a suitable media content is needed, e.g. a video changing his resolution and optimal bitrate mode for each device, a server-side detection could be used, where the server is responsible of selecting the right content, basing its decision on the embedded information in HTTP requests.

On the other hand, there is a client-side approach covered by the HTML mechanisms, where detection and adaptation happen at the client, allowing a wider range of possibilities. Here, HTML5 attributes (*srcset*) play an important role, giving the possibility of providing multiple resources varying resolutions for a single image [7]. A similar fact is also possible at audio and video tags, where the browser chooses which format to use on the current device after having defined different source types (*video/mp4*, *video/ogg*, etc.). Another powerful mechanism is based on some directives called Media Queries (*@media*) [8]. These directives can be included into CSS style sheets in order to change the properties based on characteristics of the end-device. In addition to the aforementioned, the trend with streaming protocols like MPEG-DASH moves the adaptation from the server to the client.

Thus, Web Components and Media Queries represent a perfect match for the RWD and the core technologies, among others, of the nearby future of Google apps [9]. In this field, W3C proposes some basic recommendations [10] to improve the development of rich and dynamic mobile Web applications.

In this context, there are several frameworks that support and facilitate the development of responsive applications. CSS3 Flexbox [11] suggests a CSS box model optimised for the user interface design, where the children of a flex container can be laid out in any direction –both horizontal and vertical – in a flexible way. CSS3 Regions [12] allows elements moving through different regions where CSS Regions library[4] expands the browser support with broader feature coverage. Grid-layout [13] defines a two-dimensional layout system, where the

[1] http://mediascapeproject.eu
[2] Polymer library's website: https://www.polymer-project.org/

[3] A polyfill is a downloadable code which provides facilities that are not built natively in a Web browser
[4] CSS Regions library, April 2014. https://github.com/FremyCompany/css-regions-polyfill

Fig. 1. a) A grid template layout example on a tablet in the context with ID number 6. b) A PiP template layout example on a TV in the context with ID number 1. c) A menu template layout example on a TV in the context with ID number 2. d) A horizontal template layout example on a smartphone in the context with ID number 9.

children of a grid container can be positioned into arbitrary slots in a flexible or fixed grid. Gridster.js[5] is a jQuery plugin to build dynamic drag-and-drop muti-column grid layouts. XUL[6] is Mozilla's XML language to design user interfaces. Kontx[7] is the proposal of Yahoo to develop TV apps. EnyoJS[8] is a framework to develop HTML5 apps focused on layouts and panels design. Bootstrap[9] is a framework for developing responsive mobile-first projects on the Web that includes a grid system to scale the layout as the device or viewport size changes.

All these languages, recommendations and frameworks for adaptive interfaces and responsive design are very useful, but they do not face the adaptation of an application in the multi-device domain, including only part of the application in each device.

III. Multi-device Model

Our research aims to explore which parameters have a relevant impact on the arrangement of the components to be shown on each device in a simultaneous multi-device application.

We will have a media Web application implemented with different logic parts developed with Web Components, and on top of this, we assume there will be an adaptation engine, out of the scope of this paper, taking the decisions of which components present on each device. For instance, if a user is consuming a media application through different devices at the same time (e.g. a TV and a smartphone), the adaptation engine will decide which components to show on the TV and which ones on the smartphone. But once the adaptation engine decides the components to be shown on each device, a responsive User Interface should be created, able to adapt to the context changes (new devices connected or disconnected by the user).

When application developers are facing a single-device user interface, they define CSS templates to organise the items in the layout, usually creating a different template for each target device. However, when developers are dealing with multi-device applications, this approach becomes unachievable. For instance, for an application with 6 different items, within a multi-device scenario, a single device could show all the components, only 5 of them, 4 of them, etc. Furthermore, when for example 4 items are shown on a device, the combinatory of selecting 4 components from a total of 6 rises 15 different options (see equation 1 where n is the total number of items of the application and k is the number of items shown in the device, in the example k=4 and n=6). As a result, an application of 6 items has 64 different combinations to create a layout template for each target device.

$$f(n,k)=\frac{n\cdot(n-1)\cdot(n-2)\ldots(n-k+1)}{k\cdot(k-1)\ldots2\cdot1} \quad (1)$$

A more versatile solution is desirable in order to compose automatically a responsive user interface following some patterns provided by the application developer and based in layout templates. This paper aims to explore which parameters are relevant to create a responsive User Interface under these circumstances.

Our hypothesis underlines these four parameters to affect to the User Interface:

- **The device**: As happens in a single-device application, the target device is very relevant to build a responsive Web application. In the same way, the devices involved in a multi-device application are expected also to be relevant.
- **The number of Web Components**: The quantity of pieces of information to be shown in that device can affect on how to present the content.
- **The nature of the application**: This parameter could be important to decide the arrangement of the User Interface. For instance, if there is a main video and related information on the device, or if the video is being displayed on another device and that device is being used only for extra information.
- **Other devices being used at the same time**: We want to know if having a second device being used

TABLE I
Combinations with the defined parameters

Id	The device	The number of Web Components	The nature of the application	Other devices being used at the same time
1	TV	3	At least one video	No
2	TV	6	At least one video	No
3	TV	3	No videos	No
4	TV	6	No videos	No
5	Tablet	3	At least one video	No
6	Tablet	6	At least one video	No
7	Tablet	3	No videos	No
8	Tablet	6	No videos	No
9	Smartphone	3	At least one video	No
10	Smartphone	6	At least one video	No
11	Smartphone	3	No videos	No
12	Smartphone	6	No videos	No
13	Tablet	3	At least one video	Yes, A TV
14	Tablet	6	At least one video	Yes, A TV
15	Tablet	3	No videos	Yes, A TV
16	Tablet	6	No videos	Yes, A TV

[5] Gridster.js website. http://gridster.net/
[6] Mozilla XUL. https://developer.mozilla.org/en-US/docs/Mozilla/Tech/XUL
[7] Kontx. https://developer.yahoo.com/connectedtv/kontxapiref/
[8] EnyoJS. http://enyojs.com/
[9] Bootstrap. http://getbootstrap.com/

Fig. 2. Images from user tests. In the left a user in front of the context situation with ID number 16 with the menu layout on the tablet. In the middle a user with the context situation with ID number 16 with the PiP layout in the tablet and in the right a user in the context situation with ID number 9 with the horizontal layout on the smartphone.

simultaneously has an impact on how the user wants to arrange the components in the first screen.

IV. User Tests

To find out the influence of each parameter, we enclosed them to these options:

- **The device:** 3 different devices. A Motorola Moto G Smartphone in portrait mode, A Nexus 10 tablet in landscape mode, and a Samsung UE40C8000 TV.
- **The number of Web Components:** Showing 3 or 6 components at the same time.
- **The nature of the application:** Two different scenarios. At least one of the components is a video, or there is not a video among the components.
- **Other devices being used at the same time:** Two possible situations. The evaluated device is the only one being used by the user or there is another device as a companion screen.

Testing images have been created simulating a broadcasted live F1 race scenario with all the possible combinations of the first three parameters (see Table 1 from Id 1 to 12).

Apart from these 12 context situations, we created 4 more to evaluate the "Other devices being used at the same time" parameter. We defined as a second screen a TV showing two fixed components and presented different contexts in a tablet, making the user think about how to present the content in the tablet, while they were also watching related content in the TV. As an outcome we have 4 new combinations in Table 1 from Id 13 to 16.

For each one of the 16 combinatorial contexts, we created always four different user interface arrangement patterns:

- **Grid Layout**: Based on the CSS Grid Layout Module Level 1[10] (see example in Fig. 1a).
- **Picture-in-picture Layout (PiP)** (see example in Fig. 1b).
- **Menu Layout** (see example in Fig. 1c).
- **Horizontal Layout** (see example in Fig. 1d).

The tests were done with 47 users, one by one, being always an expert presenting each one of the 16 context situations. The expert gave them a very brief description of the context of the testing and ask them to choose always the layout they would prefer on that moment to see the F1 race. All the tests were carried out in the Digital Home Lab of Vicomtech-IK4, where there is a similar environment on what we can find on a living room. From the 47 users, 40 of them were researchers in Vicomtech-IK4, with expertise on different fields, and 7 of them were administrative staff people. It took around 15 minutes to perform the test with each user, so around 12 hours in total, divided in three different days. Fig. 2 presents pictures took during the tests.

[10] http://dev.w3.org/csswg/css-grid/

TABLE 2
RESULTS OF THE CHOSEN LAYOUTS ON EACH SITUATION

Id	Results: Layout: Number of Answers | Percentage			
	Grid	PiP	Menu	Horizontal
1	19 | 40%	13 | 28%	14 | 30%	01 | 02%
2	32 | 68%	08 | 17%	06 | 13%	01 | 02%
3	28 | 60%	16 | 16%	03 | 03%	00 | 00%
4	33 | 70%	07 | 15%	05 | 11%	02 | 04%
5	09 | 19%	26 | 55%	11 | 23%	00 | 00%
6	34 | 72%	08 | 17%	05 | 11%	00 | 00%
7	23 | 49%	14 | 30%	07 | 15%	03 | 06%
8	31 | 66%	09 | 19%	07 | 15%	00 | 00%
9	05 | 11%	04 | 09%	20 | 43%	18 | 38%
10	22 | 47%	03 | 06%	19 | 40%	03 | 06%
11	11 | 23%	08 | 17%	09 | 19%	19 | 40%
12	16 | 34%	08 | 17%	15 | 32%	08 | 17%
13	17 | 36%	17 | 36%	13 | 28%	00 | 00%
14	28 | 60%	07 | 15%	10 | 21%	02 | 04%
15	30 | 64%	06 | 13%	10 | 21%	01 | 02%
16	33 | 70%	03 | 06%	11 | 23%	00 | 00%

V. Results of the User Tests

Table 2 presents the answer that the users gave for each one of the 16 contexts. Moreover, the following sub-sections present the behaviour of each one of the users when one of the parameters changed, in order to evaluate which parameter changed their mind. It measures if the user selected a different layout when the context changed.

A. Changing the device

Number of users that changed the preferred layout when the context was changed from smartphone to tablet:

- From id 5 to 9: 38 | 80.85%
- From id 7 to 11: 34 | 72.34%
- From id 6 to 10: 26 | 55.32%
- From id 8 to 12: 26 | 55.32%
- **AVERAGE: 65.96%**

Number of users that changed the preferred layout when the context was changed from tablet to TV:

- From id 1 to 5: 24 | 51.06%
- From id 3 to 7: 14 | 27.78%
- From id 2 to 6: 21 | 44.68%
- From id 4 to 8: 20 | 42.55%
- **AVERAGE: 41.52%**

Number of users that changed the preferred layout when the context was changed from smartphone to TV:

- From id 1 to 9: 36 | 76.60%
- From id 3 to 11: 36 | 76.60%

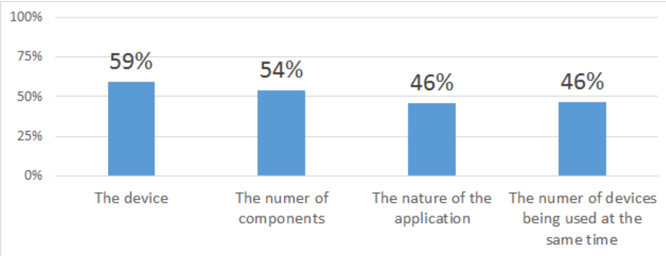

Fig. 3. Impact of the four defined parameters

- From id 2 to 10: 29 | 61.70%
- From id 4 to 12: 30 | 63.83%
- **AVERAGE: 68.68%**

The overall average impact of changing the device is **59.05%**.

B. *Changing the number of components*

Number of users that changed the preferred layout when the context was changed from an application with 3 components to 6 components when at least one of the components was a video:

- From id 9 to 10: 31 | 65.96%
- From id 5 to 6: 33 | 70.21%
- From id 1 to 2: 23 | 48.93%
- From id 13 to 14: 25 | 53.19%
- **AVERAGE: 59.57%**

Number of users that changed the preferred layout when the context was changed from an application with 3 components to 6 components when there was not a video component:

- From id 11 to 12: 34 | 72.34%
- From id 7 to 8: 23 | 48.95%
- From id 3 to 4: 19 | 40.42%
- From id 15 to 16: 14 | 29.78%
- **AVERAGE: 47.87%**

The overall average impact of changing the number of components is **53.72%**.

C. *Changing the nature of the application*

Number of users that changed the preferred layout when the context was changed from an application with 3 components and at least one of them a video component, to an application with 3 components but without a video component:

- From id 9 to 11: 30 | 63.83%
- From id 5 to 7: 30 | 63.83%
- From id 1 to 3: 27 | 57.44%
- From id 13 to 15: 17 | 36.17%
- **AVERAGE: 55.32%**

Number of users that changed the preferred layout when the context was changed from an application with 6 components and at least one of them a video component, to an application with 6 components but without a video component:

- From id 10 to 12: 20 | 42.55%
- From id 6 to 8: 18 | 38.29%
- From id 2 to 4: 14 | 29.78%
- From id 14 to 16: 16 | 34.04%
- **AVERAGE: 36.17%**

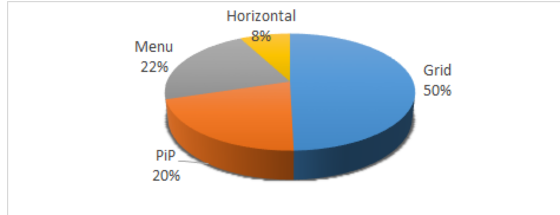

Fig. 4. Percentage of the selected layouts as an average of all the different contexts

The overall average impact of changing the nature of the application is **45.74%**.

D. *Changing the number of devices being used at the same time*

Number of users that changed the preferred layout when the context was changed from an application in the tablet to the same application in the tablet but having a TV with additional information:

- From id 5 to 13: 25 | 53.19%
- From id 7 to 15: 24 | 51.06%
- From id 6 to 14: 20 | 42.55%
- From id 8 to 16: 18 | 38.30%
- **AVERAGE: 46.28%**

The overall average impact of changing the number of devices being used at the same time is **46.28%**.

VI. Analysis of the Results

After the user tests performed we deduced the following from the results:

A. *The four parameters analysed as an hypothesis have a big impact on the user interface*

Analysing the impact of each one of the four parameters defined to be measured in this prototype, we conclude that the four ones are relevant and have a big impact above 45%. This means that at least 21 of the 47 users tested change the mind when one of the parameters changes in the context.

Fig. 3 shows that the most relevant parameter is the device, with an impact above 59%. As happens for “single device” application, it was also expected to be an important parameter for multi-device applications, where the user is consuming an application from more than one device at the same time. The number of components parameter is also very relevant being around 5 percentage points below. This means that it is important to change the arrangement of the user interface depending the outcome of the adaptation engine while the number of components to be shown changes.

The nature of the application and the number of devices being used at the same time are also relevant parameters, both of them around 46%, to take them into account to develop the final software of the UI Engine.

B. *There is a higher impact on the device parameter if we change the orientation of the device from landscape to portrait*

Being 59.05% the impact of the device parameter, it increases considerably when we compare how affects the user a change in the context when moving from a device in portrait (the smartphone in our tests) to a device in landscape (the tablet and the TV in our tests). In this situation the impact is 67.82%.

On the other hand, when there is a change in the device, but both are in the same orientation (changing from tablet to TV in our tests), the impact of the parameter is around 41%. Still relevant, but more than 25 percentage points below.

C. The grid layout is the preferred by users

The grid layout is the most selected by the users from the four possible layouts, with a percentage of 50%. Fig. 4 shows the average of the selected layouts by the users taking into account all the possible context situations. It is the preferred layout in 13 of the 16 contexts and it is never the worst layout.

D. The horizontal layout is the most rejected one by users

As Fig. 4 shows, the horizontal layout have been chosen in only 8% of the situations, 42 percentage points below the grid layout, and 12 percentage points below the second worst layout (the PiP layout). The horizontal layout is the least selected layout in 14 of the 16 different context situations. However, in those two circumstances where it is not the worst layout (context ID 9 and 11), it obtains a very good result, being the preferred one in context number 11. This analysis gives us the outcome for the next three conclusions.

E. The horizontal layout only has sense with a low number of components and in a portrait device

The results of the horizontal layout show that there could be very specific circumstances where a very specific layout has sense, but won't be useful out of that conditions. So our solutions needs to be open and easy to add a specific user interface for an application by the application developer.

F. Horizontal as a specific case of the grid layout

If we consider the horizontal layout as a specific case of the grid layout, where there is only one column, the grid layout is reinforced as the preferred layout increasing from the 50% to the 58% of the context situations. When the horizontal layout is preferred by the users, it can be considered as a transfer from grid to horizontal, since the sum of grid and horizontal remains stable for all the devices.

G. PiP and menu layouts are more neutral

Rarely are the most preferred layouts (PiP is the preferred one in 2 of the 16 situations and menu is the most selected one in 1 of the 16 circumstances), but almost never the most rejected ones (PiP layout is the most rejected one in 4 of 16 contexts while the menu layout is never the most rejected layout). For a considerable big screen (a tablet and a TV in our tests) the PiP layout is preferred to the menu layout, and for small screens (the smartphone in our tests) the menu layout is preferred to the the PiP.

In fact, for the smartphone the PiP was selected by the 11% of the users while the menu was preferred by the 34% of the users. However, for tablets and TVs, around 26% of the users prefer the PiP while around 16% of the users select the menu layout.

The results also make clear that in order to choose only one neutral layout from PiP and menu, the menu layout is more neutral since it is never the worst option for the user and Fig. 4 shows that it is more selected than the PiP.

VII. Conclusions

Existing standards, recommendations and frameworks address the creation of adaptive user interfaces and responsive design of an application. However, they do not consider the adaptation in the multi-device domain. The trend of consuming media content across multiple devices at the same time is transforming the television viewing experience, where users add companion screens to the TV.

This fact motivated the presented user tests to explore which are the relevant parameters to create responsive User Interfaces for Web-based multi-device applications, where it is not possible to define a specific layout for all the context circumstances and device combinations.

We proposed four parameters to be relevant as a hypothesis in the creating of multi-device adaptive user interfaces: target device, number of components, nature of the application and number of devices at the same time. From the user tests we concluded that the four parameters have an important impact on the user interface as the obtained rank for all of them is higher than 45%.

In the paper we also propose four different layout templates to be composed automatically based on the application definition: grid layout, menu layout, picture-in-picture layout and horizontal layout. In terms of layout templates, grid is the preferred one by the users. It is a more complex layout compared to the other ones to be automatically composed, so further analysis is needed to evaluate how to organise the components on a grid layout to provide a good experience. Menu (for small screens) and picture-in-picture (for bigger screens) could be the alternative in situations where grid layout does not satisfy the user, since they are almost never the most rejected ones. Finally, horizontal layout is the most rejected by the users, having only sense with a low number of components and in portrait orientation, which means that there could be very specific circumstances where a concrete layout has sense.

These results provide a valuable information in order to create responsive User Interfaces for Web-based multi-device applications driven by media content. However, further research and user tests are needed to collect information from a higher sample of users with different profiles.

References

[1] Anthes, G. (2012). HTML5 leads a web revolution. Communications of the ACM, 55(7),16-17.

[2] Merkel, K. (2011, March). Hybrid broadcast broadband TV, the new way to a comprehen-sive TV experience. In Electronic Media Technology (CEMT), 2011 14th ITG Conference on (pp. 1-4). IEEE.

[3] Courtois, C., & D'heer, E. (2012, July). Second screen applications and tablet users: constellation, awareness, experience, and interest. In Proceedings of the 10th European conference on Interactive tv and video (pp. 153-156). ACM.

[4] Zorrilla, M., Tamayo, I., Martin, A., & Olaizola, I. G. (2013, June). Cloud session mainte-nance to synchronise HbbTV applications and home network devices. In Broadband Mul-timedia Systems and Broadcasting (BMSB), 2013 IEEE International Symposium on (pp. 1-6). IEEE.

[5] Ziegler, Christoph. "Second screen for HbbTV—Automatic application launch and app-to-app communication enabling novel TV programme related second-screen scenarios." Con-sumer Electronics Berlin (ICCE-Berlin), 2013. ICCEBerlin 2013. IEEE Third International Conference on. IEEE, 2013.

[6] W3C Working Group Note (2014, July). Introduction to Web Components. http://w3c.github.io/webcomponents/explainer/

[7] W3C Editor's draft (2014, November). HTML 5.1 Nightly. A vocabulary and associated APIs for HTML and XHTML. http://www.w3.org/html/wg/drafts/html/master/embedded-content.html#the-source-element-when-used-with-the-picture-element

[8] W3C Recommendation (2012, June). Media Queries. http://www.w3.org/TR/css3-mediaqueries/

[9] Techrunch (2013, May). Google Believes Web Components Are The Future Of Web Development. http://techcrunch.com/2013/05/19/google-believes-web-components-are-the-future-of-web-development/

[10] W3C Recommendation (2010, December). Mobile Web Application Best Practices. http://www.w3.org/TR/mwabp/

[11] W3C Working Draft (2014, September). CSS Flexible Box Layout Module Level 1. http://www.w3.org/TR/css3-flexbox/

[12] W3C Working Draft (2014, October). CSS Regions Module Level 1. http://www.w3.org/TR/css3-regions/

[13] W3C Working Draft (2014, May). CSS Grid Layout Module Level 1. http://www.w3.org/TR/css3-grid-layout/

M. Zorrilla is with the Department of Digital Tv & Multimedia Services, Vicomtech-IK4 GraphicsMedia.net. He received his Telecommunication Engineering degree in 2007 from Mondragon Unibertsitatea (Spain) and an advanced degree in Computer Science from UPV-EHU University of the Basque Country (2011-2013, Spain).

He focuses on the research lines related to multimedia services and interactive media technologies. During his studies he worked in the area of communications of IK4 Ikerlan Research Centre (www.ikerlan.es) (2002-2006). Since 2007 he is working at Vicomtech-IK4 (www.vicomtech.org), where he designs, develops and leads projects of the Digital Television and Multimedia Services area. He is also a professor in the University of Deusto (www.deusto.es) in multimedia technologies in Computer Science.

A. Martin is with the Department of Digital Tv & Multimedia Services, Vicomtech-IK4 GraphicsMedia.net. He received his engineering degree in 2003 from University Carlos III, Spain.

He starts his Phd in the Communications and Signal Processing Department of the University Carlos III in the video coding research area in 2003. At the same time he collaborates with Prodys developing a standard MPEG-4 AVC/H.264 codec for DSP. In 2005 he starts to work on Telefonica I+D in projects related to multimodal interactive services for Home Networks. Also in Telefonica I+D, he goes deeper into image processing area in terms of 3D video and multiview coding. From 2008 to 2010 he worked in Innovalia as R&D Project consultant related with smart environments and ubiquitous and pervasive computing. Currently he is on Vicomtech-IK4 managing and developing R&D projects around multimedia content services.

I. Tamayo is with the Department of Digital Tv & Multimedia Services, Vicomtech-IK4 GraphicsMedia.net. He received his Computer Science Engineering degree in 2007 from Mondragon Unibertsitatea (Spain).

He focuses on the research lines related to multimedia services and interactive media technologies. He developed there the End of Degree Project developing a Quality Management System on Fundación Legarra Etxebeste. In 2007-2008 he collaborated with Geoaim in the development of a Geographical System Information. Since 2008 he is working at Vicomtech-IK4, where he designs and develops projects of the Digital Television and Multimedia Services area.

A. Dominguez is with the Department of Digital Tv & Multimedia Services, Vicomtech-IK4 GraphicsMedia.net. She received her Telecommunication Engineering degree in 2013 and her Telecommunication advanced degree in 2014, both from Tecnun, Universidad de Navarra (Spain).

She focuses on the research lines related to multimedia services and interactive media technologies. She developed at Vicomtech-IK4 her End of Degree Project developing web components based on Web Components standard for their subsequent integration on multi-device multimedia applications. Since 2015 she is working at Vicomtech-IK4, where she designs and develops projects of the Digital Television and Multimedia Services area.