https://doi.org/10.2352/EI.2024.36.1.VDA-361


# Interactive media planning data visualization

*Marina LJUBOJEVIC, Mihai MITREA; SAMOVAR, Telecom SudParis, Institut Polytechnique de Paris, Palaiseau, France*


## Abstract

*This study delves into the domain of mobile media planning applications and investigates advanced visualization and interaction modes within such an application. Media planning data is a collection of heterogeneous content (image/video/graphics, text, data analytics, logical expressions,…) that are aggregated, managed and displayed together. Consequently, ensuring the quality of experience for user interaction is challenging mainly in resources constraint environments, such as mobile devices. The paper reconsiders and extends an architecture we designed for media plan visualization, as to enrich its visualization and interactivity functionalities. The experiments correspond to objective assessments carried out on three anonymized real-life media planning databases. They show that the following beneficial features are granted: mobile data visualization solution for media planning, presenting all relevant information cohesively; the potential utilization of multiple technologies concurrently; ensuring ergonomic user interaction, regardless of the combination of technologies used to obtain the visualization solution.*


## 1. Introduction

*Media planning* is the process of strategizing and purchasing advertisement placements by determining the best combination of media to achieve the marketing campaign goals [1]. This process occurs for any type of product, regardless of its market share; for instance, Figure 1 shows the top digital advertisement spending verticals in 2023, including *Retail*, *Consumer Packaged Goods* (CPG), *Financial services*, *Computing products and consumer electronics*.

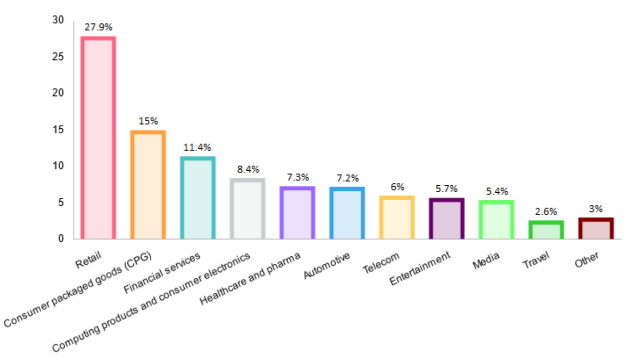


***Figure 1:*** *The top digital ad spending verticals in the U.S. in 2023* (source: Statista [2]).

The media plan deployment and usage refer to three levels: conceptual formulation, marketing provisioning, and software instantiation.

Firstly, conceptual formulation pertains to create a coherent and organized depiction of the diverse information essential for devising a media plan. This encompasses elements such as the digital advertising content, distribution channels (further referred to as *vehicles*, including TV, radio, newspapers, social networking, web ads), time behavior (whether regular, seasonal, or event-based), and the technical infrastructure necessary for content delivery (network, display devices, *etc*.).

Secondly, marketing provisioning focuses on optimizing the financial aspects of the media plan, particularly aiming to maximize the return on investment of the advertising campaign within the constraint of a predetermined total financial budget.

Finally, specific software tools are essential for end-to-end usage of a media plan. Data required for a media planning process is stored in a *media plan database*. Each database can contain information about multiple media plans, and even multiple plans for each of the brands in the database. Media planning software tools address the requirements for managing advertising content, providing interfaces for such management, and offering adaptive and interactive data visualization capabilities.

The utilization of data visualization within media planning context entails creation of visualization systems for integration, harmonization, and data representation [3], [4]. Thus, it firstly demands simultaneous management of heterogeneous content types (images, videos, graphics, text, data, logical connections, business analytics, *etc*.) as depicted in Figure 2. Secondly, this data compilation caters for the needs of various professional users with distinct habits and expectations, necessitating the dynamic adjustment of visual content based on user technical specifications (device, connectivity) and work practices (content initialization, preferred interactive modes, *etc*.). Lastly, the media planning community is accustomed to well-established operational practices, primarily relying on fixed computing environments (desktops) that have been in use for numerous years. Therefore, the introduction of emerging devices (such as convertible laptops, foldable smartphones, *etc*.) and alternative interaction methods (beyond mouse and touchscreen interactions) are expected to enhance the quality of user experience with visualized media plan content.

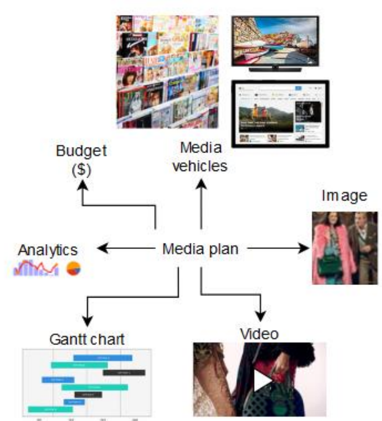


***Figure 2****: Heterogeneous data in media plan visualization* [5].

The present paper reconsiders and extends our previous study devoted to the software framework related for the interactive visualization of the media plans [5]. The main contribution consists in additional architectural modules designed for increasing the displaying functionalities featured by our solution, in the possibility of interacting with overlayed content, and in the possibility of managing complex interaction modes. The results obtained in this study correspond to three databases, namely *Retail*, *CPG*, and a composition of consumer electronics and home appliance; these databases are real, yet anonymized.

## 2. Problem Statement

Media planning is a two-folded problem, involving both the formulation of the media plan and its subsequent deployment strategy. Consequently, the media plan constitutes a non-linear function incorporating at least five main variables: advertising content, distribution channels, duration of distribution, pricing, and return on investment [6]. Complementing this, an expert system computes its optimum, translating marketing conventions into quantifiable entities.

The focus of this paper lies in the interactive visualization of media plans within mobile environments. The main challenges in handling and presenting such data arise from conceptual, usage, and technical aspects.

At the conceptual level, the vast amount of heterogeneous and interdependent data is required to be visually represented together. Such representations should combine basic data (text, numerical values), media data (image, video) throughout various types of statistical and time-series visualization graphics (plots, Gantt charts, sparklines, *etc*.). Visualization in context of media planning should also adhere to the general principles of data visualization, namely "*overview first, zoom and details on demand*" [5]. This principle assumes a general overview of the high-level information, with the possibility of zooming, and showing the details on demand. As media planning problem involves managing multiple variables, ensuring adequate information with details on demand poses challenges regarding the selection, placement, and presentation of detailed information.

The relevancy of different variables also varies according to the user profile (*e.g.* media planner, marketing expert or high-level manager). Therefore, according to the profile, different visualization methods need to be utilized.

The significance of various variables varies depending on the user profile, be it media planner, marketing expert, high-level manager, *etc*. Consequently, tailored visualization methods must be employed based on the specific profile to ensure optimal comprehension and utility.

In an applicative field dominated by strong legacy business practices about displaying and interaction with media planning applications, our study (whose preliminary results have already been presented in [5]) brings as novelty:

1. at applicative level: the shift towards a mobile environment of a traditional, business-oriented application; note that this shift is not trivial and development from scratch is not an alternative!
2. at methodological level: the conception, specification and implementation of comprehensive architecture modules for end-to-end data management from raw format in the database, across interactive visual content, and so forth, in iterative cycles;
3. at experimental level: the evaluations on the optimal technology blend, and gaining valuable insights into usability and user experience.

## 3. State of the Art

The present study reconsiders and extends the basis presented in [5], and specifically concerns constraint environments data visualization, interaction, and user profiles in media planning visualization.

The main concerns regarding visual data representation on the display environments are related to the spatial aspects and information density [8]. Spatial aspects include effective usage of a visualization space under limitations according to the device type. Information density represents the quantity of extracted and represented data across visualization space. As a solution for both aspects, dividing information and then allowing hierarchical access to different parts of that information is brought forth. That way, the main principles of data visualization are met, and issues in handling the data are lessened, as extracted information from the database is firstly hierarchically grouped and sorted, prior to the visualization process.

In [9], the interaction design is considered as the starting point for the study of information visualization. Several visualization techniques are presented, alongside with discussions on essential interaction design aspects to visual representation. Combining insights from various works on human-computer interaction and interaction in data visualization, [10] stated that interaction operates as a dialogue between the user and the visualization system, with data serving as the central focus. Success in this dialogue depends on the flexibility of software systems to accommodate a diverse range of user needs, as highlighted in [11], where stated that the interaction must be tailored to the user's profile to ensure effectiveness.

According to [12], users can be categorized into four distinct roles, commonly referred to as RACI – *Responsible*, *Accountable*, *Consulted*, *Informed*. These roles can be translated to the context of media planning within the business culture.

As our research focuses on integrating proof-of-concept solutions, the supporting software technologies are also explored. Thus, six visualization software libraries are selected based on their popularity at the time of the study: Snap.svg [13], D3.js [14], AnyChart [15], Highcharts [16], Chart.js [17], and Chartist [18].

It is worth noting that graphical content produced by libraries like Snap.svg, D3.js, AnyChart, and Highcharts is SVG-based, yet they offer built-in sets of visualization outputs in the form of charts with predefined options. Conversely, the Snap.svg-based solution represents a more general case where SVG code imposes no restrictions, yet it should be created from scratch. Therefore, throughout the remainder of this study, we will refer to Snap.svg (as well as D3.js) as a *generic* library, while AnyChart, Highcharts, Chart.js, and Chartist will be referred to as *specialized* libraries.

Table 1 shows results of an *a priori* comparaison among the six mentioned libraries, and according to various criteria for Gantt charts and sparklines. These two visual elements are selected as ground for comparison, as they relate to the main media plan functionalities: Gantt charts are the most effective solution for representing the time behavior, while sparklines are selected for visualizing statistical information. Comparison criteria includes interactivity, adaptivity, native support, and diversity. In case of Gantt charts, interactivity and code complexity are evaluated with comparison upon initial visualization and hybrid visualizations resulting upon following interactions. Such test is only done upon Gantt charts, as it is selected as initial visualization, and all the add-ons, including sparklines, should be built on top of it.

Upon first criterion, elementary Gantt visualization, AnyChart and Highcharts provide a general level of interactivity. While they may be outperformed in this aspect by Snap.svg and D3.js, they demonstrate effective handling of Gantt charts, even with substantial data volumes. Additionally, both libraries offer a wide range of charting options specifically designed for Gantt charts. Hence, both generic and specialized libraries have significant interaction capabilities. Although generic libraries offer more room for extensions, they are *a priori* on par with specialized libraries. Chart js and Chartist do not offer Gantt chart options at the time of the study.

During testing of hybrid visualizations, one notable aspect is the level of interactivity achievable through the Snap.svg approach that makes it possible for the degree of interactivity to be expanded to the full capabilities of JavaScript. In case of specialized libraries, the level of interactivity is restricted to predefined capabilities. While testing proved the possibility of adding higher-level interactions, the range of choices is constrained, and in some cases too complex and/or incompatible with media plan visualization. The primary challenge lies in accurately pinpointing the position of user interaction events. Specialized libraries typically offer hooks for handling code, but not all variables are readily accessible, thus increasing the complexity of the interactivity functions.

Regarding code complexity, generic libraries - Snap.svg and D3.js naturally impose higher complexity levels, as solutions are built from scratch. Complexity is more pronounced with D3.js, while Snap.svg is proven as more suited for media plan visualization solution. Going from basic to hybrid visualization, a rise in complexity is less pronounced compared to the specialized libraries (Highcharts, Anychart). Main reason for higher complexity is previously discussed limitations in adding new interactions to the specialized libraries.

**Table 1: Exploration of software technologies in media planning visualization context**

| | | SVG | D3 js | AnyChart | Highcharts | Chart js | Chartist |
|---|---|---|---|---|---|---|---|
| Gantt charts | Interactivity | 🟢 -> 🟢 | 🟢 -> 🟢 | 🟢 -> 🔴 | 🟢 -> 🔴 | 🔴 | 🔴 |
| | Native support | 🔴 | 🔴 | 🟢 | 🟢 | | |
| | Code complexity | 🟠 -> 🟠 | 🔴 -> 🔴 | 🟢 -> 🔴 | 🟢 -> 🔴 | | |
| Sparklines | Interactivity | 🟢 | 🟢 | 🟢 | 🟢 | 🔴 | 🔴 |
| | Native support | 🔴 | 🟢 | 🟢 | 🟢 | 🟠 | 🟢 |
| | Code complexity | 🔴 | 🔴 | 🟠 | 🟠 | 🟠 | 🟢 |
| | Diversity | 🟢 | 🟢 | 🟠 | 🟢 | 🟠 | 🟢 |

This state-of-the-art study brings to light that the design of an interactive and adaptive media plan software tool is still an open issue. If specialized libraries would a priori result in a richer and more detailed set of graphical objects to be interacted with, the generic libraries are appealing thanks to their unbonded set of promises that need to be implemented.

Consequently, rather than making a choice on the use of a specific library, our study advances an architectural framework that ensure alternative usage of various libraries, as presented in the next section.

## 4. Advanced Solution: Interactive Multi-layer Visualization Architecture

To enable the creation of a given visualization element by using its most appropriated library, while maintaining overall spatio-temporal synchronization, we devised the architectural framework depicted in Figure 3 [5]. This framework is illustrated by considering a workflow of media plan information, from the raw data stored in the media plan to the interactive display of visual content on the user's device.

The workflow is executed in iterative cycles. First, an initialization is made based on some pre-established rules. According to these rules, relevant data is selected from the media planning database, and is fed to the *Data Aggregation and Preprocessing Manager*, which is the first block of the architecture. This block extracts the relevant information and aggregates the data to be visualized. Next, *Multi-layer Visualization Manager* processes and visualizes information according to the given data type. Finally, *Personalized Interaction Manager* is in charge with the management of the user interaction and of the user profile.

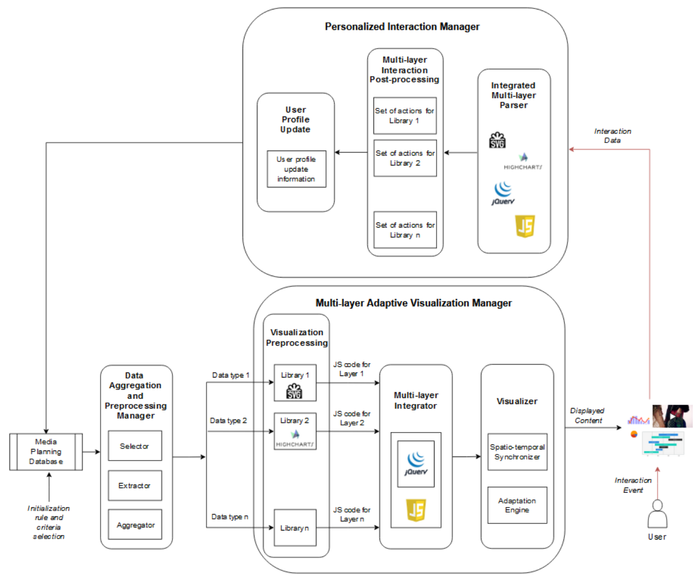


***Figure 3**: Interactive Multi-layer Visualization for media planning: architectural framework.*

### 4.1 Data Aggregation and Preprocessing Manager

*Data Aggregation and Preprocessing Manager* is the first block supporting the workflow and is composed of three sub-blocks: *Selector*, *Extractor* and *Aggregator*. It receives an input from the media planning database according to *Initialization rule and criteria selection* in the first iteration, or according to user requests based on user interaction with the visualization in subsequent iterations.

Media planning database is represented as a tree-structure (as previously shown in Figure 2), containing various details about the campaign such as budgeting plans, selected media vehicles or audio-visual data (videos, images). This database is stored in a JSON file accessible through a server.

*Data Aggregation and Preprocessing Manager* converts the selected data in a format fitted for the later visualization. Since JSON primarily contains textual data, *Selector* and *Extractor* modules are required. *Aggregator* is provided as a solution to one of the media plan visualization issues, as information in the database is vast and consist of combination of multiple time series data and statistical data.

Aggregation specifically focuses on selecting and combining interlinked time series data, as to provide general, entry view in the media plan visualization. This process merges the durations of planned campaigns according to the selected criteria. The final visualization of what is created in the aggregation module will later be presented in the form of a Gantt chart, where each line will represent combined durations of planned campaigns. As a result of aggregation, most relevant, high-level information about multiple campaigns can be shown in only one line of a Gantt chart, providing ergonomic solution for representation of vast data. This approach condenses a substantial amount of data into a single line on the chart.

Default aggregation is done according to the vehicle type: for each brand, multiple durations of planned campaigns of each vehicle, or type of vehicle, are aggregated. Result of aggregation is combined and contains both original data, as well as aggregated timelines part.

Aggregation can be done by various criteria, as, for example, the brand. In this case, insight into planning for various vehicles over some time periods are provided: for each vehicle, multiple durations of planned campaigns of each brand, are aggregated.

### 4.2 Multi-layer Adaptive Visualization Manager

In its turn, the Multi-Layer Visualization module is composed of three sub-modules: *Visualization Preprocessing, Multi-layer Integrator* and *Visualizer*.

*Visualization Preprocessing* integrates a set of existing data visualization libraries that are required for media plan displaying. Each such a library can have a specific role, from time-series visualization to Gantt charts creation and video decoding.

For instance, Highcharts can be used for time-series data and Gantt charts, while jQuery is used for data related to analytics and sparklines (as illustrated in Section 5). JavaScript is used for formatting additional data types, such as images and videos. As mentioned, each library has a specific role, therefore a specific type of input as well. All the libraries are combined in the *Visualization Preprocessing* in such a way that the modularity is ensured. In case of need for changing the library, adding the new one, deleting or replacing the obsolete one, there will be no changing of the other modules. Therefore, it is possible to have other solutions, such as SVG visualization for Gantt charts, and again jQuery for data related to analytics and sparklines. In such a case, only codes for Highcharts and SVG solution are exchanged, while jQuery part, and all the other parts including other libraries, are kept the same.

*Multi-layer Integrator* ensures that each of the library outputs from *Visualisation Preprocessing* is appointed to its appropriate layer of visualization. Specifically, as pictured in Figure 3, the modules are conceived related to SVG and Highcharts as possible visualization solutions, as well as their joint usage over JS including jQuery. This integrator solution not only helps combining previous outputs, but also allows for JS or jQuery created layers, or parts of the layers in the final visualization solution.

*Visualizer* plays an important role by collecting the outputs generated by each of the data visualization libraries. It then proceeds to process these outputs with two primary objectives in mind. *Visualizer* is responsible for three key aspects: spatiotemporal synchronization, display adaptability and network adaptability. Firstly, it focuses on maintaining the synchronization of spatial and temporal aspects within the visual content. This ensures that all elements displayed are correctly coordinated in terms of their position and timing, contributing to a coherent and cohesive overall presentation. Secondly, display adaptability ensures that the visual content is appropriately adjusted to suit the specific characteristics of the user's device screen, ensuring an optimal viewing experience regardless of the device's size, resolution, or orientation.

### 4.3 Personalized Interaction Manager

*Personalized Interaction Manager* module consists of three blocks ensuring the detection of the user-generated event, the processing of the actions based on that event and the updating the information about user profile, based on their preferences.

Firstly, user-generated event is done by one of the three modes of interaction: interaction buttons designed to add a new layer of visualization, clicks on predefined parts of the visualization for expansion layer, or resizing/repositioning windows containing content managed by different libraries.

The resulting action based on that event is one of the expanded layers in the workflow iteration.

Updating the information about user profile results from the user generated actions. User profile is given as a JSON structure, which currently contains information about the RACI group user belongs to. According to a group which user belongs to, different information is represented by expansion of the layers of visualization. For example, if the user is part of the financial management group, upon expanding layers, additional data at sparklines level will give more statistical information regarding various budgeting options, such as planning and predictions. Specific action of the user is also saved.

## 5. Experimental Results

### 5.1 Testbed

In the previous work [5], we drew to light the PoC for multi-layer visualization, showing how this technology can be orchestrated for helping the media plan application. The workflow was tested on two databases (Retail and CPG) with Highcharts based solution.

That experimental testbed is expanded for the case of hybrid database, that includes a composition of consumer electronics and home appliances. Such database imposes additional variables regarding types of products, thus imposing new challenges to the semantics, aggregation options, grouping and analytics.

Semantics difficulty reflects in the size and composition of the parser function beneath the Selector sub-block in *Data Aggregation and Preprocessing Manager*, that should have a dynamic behavior.

Aggregation needs to be tackled from three new perspectives: product as a main variable, product as an additional grouping variable, and product as a recursive grouping variable.

All databases processed in this study correspond to real-life applications and are anonymized in order to avoid any reference to any actual brand, product or distribution channel.

Beyond the database related capabilities, the previous testbed has also been extended to cater for the needs for generic SVG libraries, namely Snap.svg: all the related modules in *Multi-layer Visualization Manager* have been developed and integrated.

### 5.2 Illustrations

The experimental results, illustrated in Figures 4-11, correspond to an 8 steps walkthrough in the media plan visualization, unravelling new layers starting with aggregated Gantt chart. Such steps include various visual elements added on the top of aggregated Gantt charts: sparklines, detailed analytics, and media content.

Figure 4 and Figure 5 represent the first Gantt chart aggregated layer created by *Data Aggregation and Preprocessing Manager* and *Multi-layer Adaptive Visualization Manager* upon the workflow initialization, based on a specialized and generic library, respectively.

Figures 4 and 5 can be abstracted as visualization with two types of tables: on their left parts, marketing specific information (like brand, vehicle, start/end dates) are presented while the right parts are devoted to time-series visualization (Gantt charts in these examples). The architectural framework we designed supports the “overview first, zoom and details on demand” principle by allowing each of these two logical parts to be expanded/reduced. For instance, Figures 6 is obtained from Figures 4 by expanding (through user interaction request) the basic layer information.

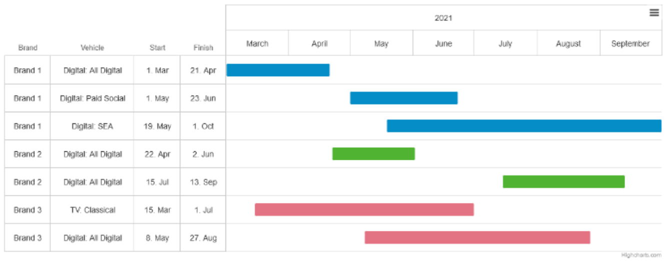


***Figure 4****: Media plan visualization: Gantt chart aggregated layer, based on Highcharts.*

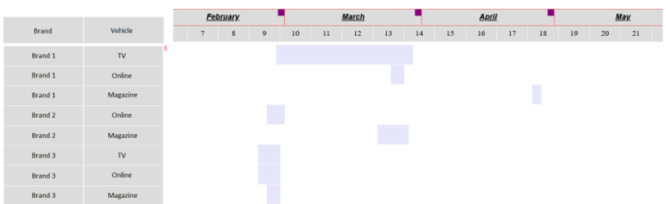

***Figure 5****: Media plan visualization: Gantt chart aggregated layer, based on Snap.svg.*

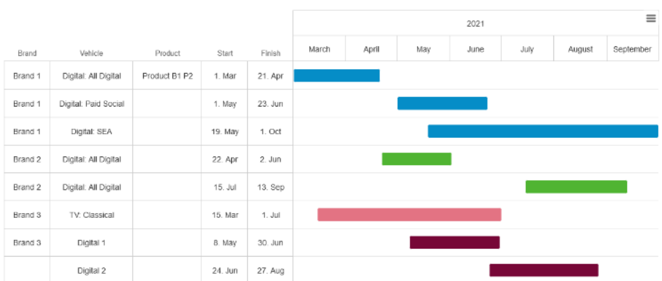


***Figure 6****: Media plan visualization: Gantt chart extended layer, based on Highcharts.*

As explained in Section 3, generic SVG libraries let the door open to richer interactivity options, as illustrated in Figure 7 where a layer with calendar options can be visualized and interacted with.

The same concept is further exploited for adding supplementary information over the basic aggregated Gantt chart. For instance, Figure 8 shows an additional column with totals, displayed as the result of a user interaction request. Such a functionality cannot be directly ensured for the Highcharts implantation. Figure 9 illustrates the possibility of adding fine-grain visualization objects, namely sparklines.

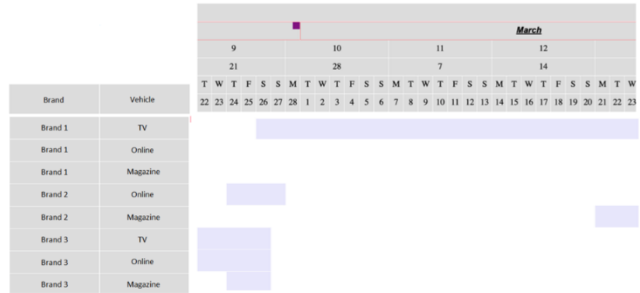

***Figure 7****: Additional visualization and interaction layers made possible for generic SVG libraries: calendar options over Snap.svg.*

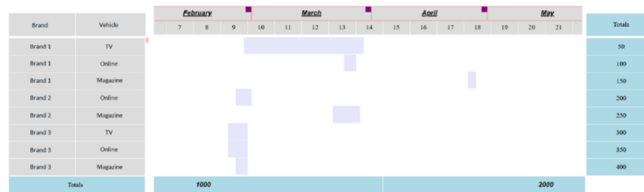

***Figure 8****: Additional visualization and interaction layers made possible for generic SVG libraries: total column options over Snap.svg.*

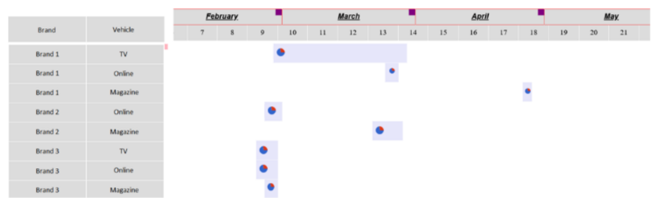

***Figure 9****: Additional visualization and interaction layers made possible for generic SVG libraries: sparklines over Snap.svg.*

As nowadays media plan databases also include multimedia content (audio, video), we also demonstrated the possibility of spatiotemporally aligning multimedia layers over Gantt charts, as illustrated in Figure 10 and 11 for the specialized (Highcharts) and generic (Snap.svg) libraries, respectively.

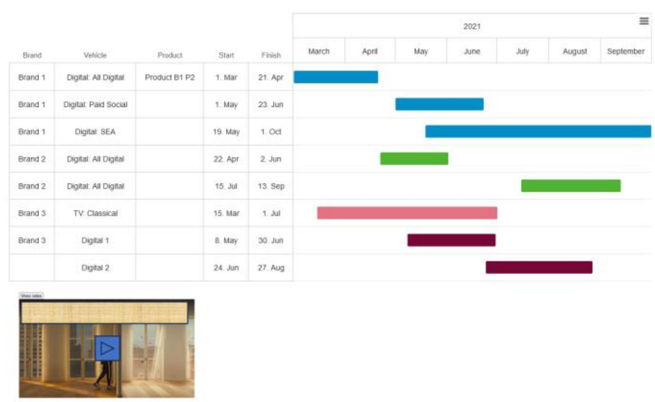

***Figure 10****: Additional visualization and interaction layers made possible for specialized libraries (Highcharts): media combined with any layer.*

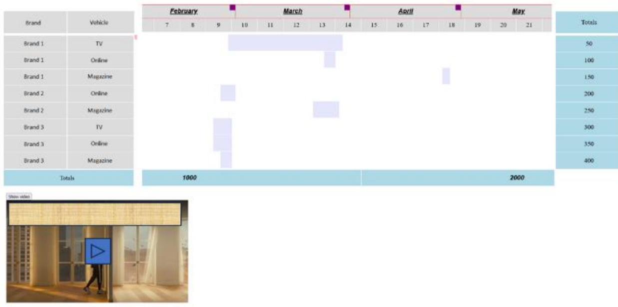


***Figure 11****: Additional visualization and interaction layers made possible for generic SVG libraries (Snap.svg): media combined with any layer.*

## 6. Conclusion

Media plan is a complex structure composed of heterogeneous information such as text, graphics, analytics, image, video, *etc*. Visualizing multiple plans over time requires interactive content visualization, according to an unpredictable sequence of user requests, on resource constrained devices.

The present study reconsiders and extends our previous results [5]. It provides an architectural framework that support end-to-end iterative displaying-interaction workflows for media plan visualization. The main architectural blocks are conceived so as to handle data fetching from the database, data visualization and user interaction. The main novelty concerning data fetching consists in the possibility of dynamically and automatically structuring row data according to their typology. The main novelty related to data visualization is the possibility of individually managing each type of content with its most suitable technology, thus obtaining different visualization layers, that are a posteriori spatiotemporally synchronized. User interaction management is also ensured in a novel way: by creating specific modules related to interaction capturing, interaction managing and user profile updating, these three functionalities can be seamlessly ensured. The experimental results are obtained out of processing three real-life (yet anonymized) media plan databases.

Future work will span around three main axes. First, subjective evaluation campaigns for the user quality of experience with the multi-layer visualization will be considered. In parallel, the work on user profile management will be extend so as to allow the user profile updating beyond the 4 basic categories in the RACI taxonomy. Finally, establishing a media plan ontology, covering the large variety of terms and relations among the media plan entities will be also part of our future work.

## References

[1] https://www.thebridgecorp.com/media-planning/ (last visit: February 20th, 2024)

[2] https://www.statista.com/statistics/1421001/industries-highest-share-digital-ad-spend-usa/ (last visit: February 20th, 2024)

[3] A. Mitchell, "The use of alternative knowledge-acquisition procedures in the development of a knowledge-based media planning system", International Journal of Man-Machine Studies, vol. 26, no. 4, pp. 399-411, 1987.

[4] P.C. Jha, R. Aggarwal, A. Gupta, "Optimal media planning for multi-products in segmented market", Applied Mathematics and Computation, vol. 217, pp. 6802-6818, 2011.

[5] M. Ljubojevic, M. Mitrea, "Multi-layer visualization for media planning", IS&T Electronic Imaging, Visualization and Data Analysis, conference proceedings, San Francisco, 2023.

[6] U. Kulbida, O. Kaneva, A. Zykina, "Media planning optimization treatment", Dynamics of Systems, Mechanisms and Machines, 2014.

[7] B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations", proceedings of the IEEE Symposium on Visual Languages, p.336-343, Washingtion, 1996.

[8] H. Shen, T. Bednarz, H. Nguyen, F. Feng, T. Wyeld, P.J. Hoek, E. Lo, "Information visualization methods and techniques: State-of-the-art and future directions", Journal of Industrial Information Integration, vol, 16, 2019.

[9] I. Avazpour, J. Grundy, L. Zhu, "Engineering complex data integration, harmonization and visualization systems", Journal of Industrial Information Integration, 2019

[10] E. Kandogan, H. Lee, "A Grounded Theory Study on the Language of Data Visualization Principles and Guidelines", Society for Imaging Science and Technology, 2016.

[11] L. Perkhofer, C. Walchshofer, P. Hofer, "Does design matter when visualizing Big Data? An empirical study to investigate the effect of visualization type and interaction use", Journal of Management Control, 2020

[12] https://www.forbes.com/advisor/business/raci-chart/ (last visit: February 20th, 2024).

[13] http://snapsvg.io/ (last visit: February 20th, 2024).

[14] https://d3js.org/ (last visit: February 20th, 2024).

[15] https://www.anychart.com/ (last visit: February 20th, 2024).

[16] https://www.highcharts.com/ (last visit: February 20th, 2024).

[17] https://www.chartjs.org/ (last visit: February 20th, 2024).

[18] https://gionkunz.github.io/chartist-js/ (last visit: February 20th, 2024).

## Author Biography

*Marina Ljubojevic received her BS in electrical engineering from the University of Belgrade (2018) and MsC in multimedia networking at Telecom Paris (2020). She is currently a PhD student with Telecom SudParis.*

*Mihai Mitrea holds an HDR degree Pierre and Marie Curie University in Paris (2010) and a PhD from Politehnica University in Bucharest (2003). He is currently Associate Professor at Telecom SudParis. He is vice-president of the Cap Digital's Technical Commission on Digital Content and serves as advisor for the French delegation at ISO/IEC JTC1 SC29 (a.k.a. MPEG)*

## Acknowledgment

Marina Ljubojevic’s participation to this study is founded by info3 (https://www.infocubed.com/), through a PhD grant. Yet, the present research study was conducted in the absence of any commercial or financial constraint that could be considered as a potential conflict of interest.