Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Ciudad de México
School of Engineering and Sciences

Multi-Screen & Dynamic Visualization Metrics and Architecture for Decision-Making Centers

A dissertation presented by

Christelle Navarrete Corella

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Doctor

In

Engineering Science

Principal Advisor:
Dra. Juana Julieta Noguez Monroy
Co-advisor:
Dr. José Martín Molina Espinosa

Tlalpan, Ciudad de México, May 24th, 2021

Dedication

Thanks, Tata for the inspiration you have given us, for being such a wise man and teaching us to always learn and take every good opportunity given from life. Thanks Ave, for your unconditional confidence and the encouragement you have given me to be better always. Thanks, Saqib for believing in me and for your support in the hardest moments. Thanks, Ian, for your patience. You were my main motivation for pushing through this work.

Acknowledgements

Firstly, I would like to express my deepest gratitude to my advisors, Dr. Juana Julieta Noguez Monroy and Dr. José Martín Molina Espinosa, for their guidance and support throughout this work.

I'd also like to thank to my colleagues for their enthusiasm, companionship and fun moments during this research project.

Furthermore, I would like to thank my professors for their teachings and for sharing your knowledge.

Finally, I would like to express my sincere gratitude to Instituto Tecnológico y de Estudios Superiores de Monterrey for letting me be part of this program.

Multi-Screen & Dynamic Visualization Metrics and Architecture for Decision-Making Centers

by

Christelle Navarrete Corella

Abstract

Information is one of the most valuable resources, and its use is critical during the decision-making process within the organization. Technological evolution has given way to the development of tools that allow organizations to have more efficient processes. The information visualization tools are a resource that facilitates the analysis of critical information and that allows decision-makers to consider the different variables and their effect when taking a resolution. However, before the rise of the visualization tools as well as their diversity arise some questions. What characteristics should these tools have to be effective and give dedicated support to the decision-making process by visualization of information? How can we ensure that an information visualization tool is competent and conforms to the needs in the decision-making process? In this work, we make a metric proposal for the evaluation of visualization tools. In our metric proposal, we consider criteria that are important for specific use in the decision-making process. The objective of our metric proposal is facilitating the selection of an information visualization tools since the selected criteria will permit the evaluation of tools to find those who fulfill the decision-making process needs.

We also made an architecture proposal for the development of visualization tools, considering the criteria defined in our metric proposal. In this way, we can ensure that our architecture proposal meets the needs of the decision-making process and that its use will allow obtaining benefits as knowledge generation, ease of analysis through the simultaneous presentation, interaction, collaboration, and real prospective in a multi-screen environment with information update in real-time. This work includes the review of related work in terms of existing visualization tools and existing evaluation methods. Likewise, we did the corresponding tests and analyzes that demonstrate its validity.

Finally, we conducted the conclusions, and in future work, we consider the possibility of continuing this proposal that requires a more extensive development.

List of Figures

Fig. 1 Traditional model of human perceptual system. 17
Fig. 2 Timeline showing the approaches reviewed. Blue, orange and green shadings denote language-based tools, model-based tools and interactive-graphical tools, respectively. 22
Fig. 3 Left most common used charts to show data. Right, charts to avoid. 29
Fig. 4 Gestalt principles. 30
Fig. 5 Pre-attentive attributes. 30
Fig. 6 Four phases of the decision-making process 34
Fig. 7 Evidence-based decision-making model proposed by Baba, V. V. 36
Fig. 8. Decision-maker and context factors within the EBM process 37
Fig. 9. Diagram of the MVVM pattern and its components. 42
Fig. 10. Bindings between the View and the ViewModel. 42
Fig. 11 View using operations from ViewModel. 43
Fig. 12 View handling notifications from the ViewModel. 43
Fig. 13. Timeline showing the reviewed visualization tools. Blue, yellow and green shadings denote software library/APIs, custom software and commercial software, respectively. 49
Fig. 14 Decision Theatre physical configuration (Ben, 1986). 60
Fig. 15 The Future Operations Centre Analysis Laboratory (FOCAL) at Australia's Defense Science and Technology Organization (DSTO) (Wark, y otros, 2005). 61
Fig. 16 Multi-Agent Architecture (Wark, y otros, 2005). 61
Fig. 17 ROLF Mark II (Brehmer, 2007) 62
Fig. 18 Concept of ROLF 2010 (Brehmer, 2007). 63
Fig. 19 Visualization Dome prototype exterior and interior (Gawlikowska, Marini, Chokani, & Abhari, 2018) 63
Fig. 20. Conceptual Framework of the Decision Theater for Participatory Planning (Boukherroub, D'amours, & Rönnqvist, 2018). 64
Fig. 21. Visualization integrated in the Evidence-based decision-making model 72
Fig. 22 Diagram of the proposed architecture using Service-Oriented Architecture and Model-View View Model patterns. 74
Fig. 23 Model View-Model View design architecture. 74
Fig. 24 Service Oriented Architecture Design. 76
Fig. 25. Diagram of the proposed architecture matching the Evidence-Based Decision-Making Model. 77
Fig. 26 Activity Diagram of our architecture proposal. 79
Fig. 27 Use case diagram of our architecture proposal. 80
Fig. 28. Component diagram of a system based on our architecture proposal. 81
Fig. 29 Sample image of mono-display format of phase one. 92
Fig. 30. Sample image of binational laboratory introductory video 92
Fig. 31. Sample image of binational laboratory usage video 93
Fig. 32. Sample images of slide show. 93
Fig. 33 Generated word cloud of advantages in Mono-screen phase from participants. 104
Fig. 34 Generated word cloud of advantages in Multi-screen phase from participants. 105
Fig. 35 Structure of an XAML document (Image taken from (Microsoft, Microsoft Developer Network, 2009) 124
Fig. 36 Visualization of a layout hierarchy, showing the parameters associated with each view. (Image taken from (Google, 2017)) 126
Fig. 37 Client–server architecture of UI protocol (Image taken from (Slováček, 2009)) 131
Fig. 38 Propagation of information through a platform (image taken from Macik M. K., 2011) 132
Fig. 39 Platform of automated UI generation(Image taken from (Macik M. C., 2014)) 134
Fig. 40. Image taken from google chart site, https://developers.google.com/chart/, visited on October 2018. 151
Fig. 41. Screenshot taken from examples of D3.js site, https://github.com/d3/d3/wiki/Gallery, visited on September 2018. 152
Fig. 42. Zoil Framework interface (https://www.youtube.com/watch?v=ngP0TEaOJwQ) 154

Fig. 43. FLUID interface protoype. *155*
Fig. 44. Munin sample of two devices connected and sharing data, providing collaboration among peers. *156*
Fig. 45. Screen shot for the DC-Tec Tool, showing data of an electric model in Mexico *157*
Fig. 46. QlikView's data visualization dashboard. *159*
Fig. 47. Microstrategy interface. Image taken form Microstrtegy web site (Microstrategy, 2018) (http://www.microstrategy.com) *160*
Fig. 48. Domo interface screen shot, taken from Domo website (http://www.domo.com, October 2018). *161*
Fig. 49. Tableau desktop interface on MacOS, showing a sample per regions data. Referencia de sitio. *162*
Fig. 50. SISENSE Dashboard example, taken from SISENSE website(https://www.sisense.com/) *163*

List of Tables

Table 1. SWOT analysis of the approaches reviewed in graphic user interfaces tools. 25
Table 2. Software libraries and APIs 51
Table 3. Custom software 52
Table 4. Commercial Software 53
Table 5. Dynamics taxonomy 69
Table 6. Rubric for the evaluation of visualization aspects 70
Table 7. Experts' profile, categories of tools and participants on the evaluation experiment. NA, Not Available, means there were no experts performing evaluation on the indicated tool category. 83
Table 8. Evaluation Rubric and added questions related to evaluation criteria for evaluation aspects. 85
Table 9. Questionnaire provided to the custom software category representative. 86
Table 10. Questionnaire provided to the commercial software category representative. 87
Table 11. Questionnaire provided to the libraries/API's category representative. 88
Table 12. Questions asked to participants after each phase one and phase two of the experiment. 94
Table 13. Overall average provided in the assessment by evaluators 97
Table 14. Welch two sample t-test and p-value. 98
Table 15. Commercial software group average scores per tool and criteria. 99
Table 16. Results of the Mono-Screen Vs. Multi-Screen & Dynamic Test Process 102
Table 17. T-test and p-value results of Multi-screen and Mono-screen phases. 106
Table 18 Advantages and disadvantages of language-based tools 129
Table 19 Advantages and disadvantages of model-based tools 136
Table 20 Advantages and disadvantages of interactive graphical specification tools 144
Table 21 SWOT analysis of the approaches reviewed in graphic user interfaces tools 146

Index

Abstract ___ ***7***

List of Figures ___ ***9***

List of Tables ___ ***11***

1. Introduction ___ ***15***

1.1 Problem Statement ___ **18**

1.2 Research Questions & Hypothesis ___ **19**

2. Theoretical Framework ___ ***20***

2.1 Graphic User Interfaces ___ **20**

2.1.2 Graphic User Interfaces Tools ___ 21

2.2 Information Visualization ___ **26**

2.2.1 Visualization Elements ___ 28

2.2.2 Multi-screen visualization ___ 31

2.4 Decision-Making ___ **33**

2.4.1 Decision-Making definition & process ___ 33

2.4.2 Evidence based decision-making ___ 35

2.5 Software Architecture ___ **38**

2.5.1 Service Oriented Architecture (SOA) ___ 40

2.5.2 Model View View-Model Architecture (MVVM) ___ 41

2.6 Metrics & Evaluation of Information Visualization ___ **44**

3. Related Work ___ ***47***

3.1 Information Visualization Tools classification ___ **49**

3.2 Metrics ___ **54**

3.3 Multi-screen visualization ___ **59**

4. Solution Proposal ___ ***66***

4.1 Metrics Solution proposal ___ **66**

4.1.1 Definitions of evaluation criteria ___ 66

4.1.2 Evaluation Rubric ___ 69

4.2 Multi-Screen & Dynamic Solution Proposal ___ **71**

4.2.1 Model ___ 71

4.2.2 Architecture ___ 72

4.2.3 Model View View-Model Component ___ 74

4.2.4 Service-Oriented Architecture Components ___ 75

4.2.5 Evidence-based decision-making model matching the proposed architecture ___ 77

UML Diagrams ___ 78

5. Test Process ___ ***82***

5.1 Metric Test Process ___ **82**

5.1.1 Participants ___ 82

5.1.2 Procedure ___ 84

5.1.3 Data collection method 85
5.2 Multi-Screen & Dynamic Test Process 90
5.2.1 Participants 90
5.2.2 Process 91
6. Results and Discussion 96
6.1 Metrics Evaluation Process Analysis and Results 96
6.2 Multi-Screen & Dynamics Test Process Analysis and Results 101
7. Conclusion & Future Work 108
References 110
Appendix 1 119
Graphic User Interfaces Review 120
Tools reviewed 120
Appendix 2 149
Information Visualization 149
Information visualization tools reviewed 149
Appendix 3 165
Energy Case Study 165
Appendix 4 169
Acronyms 169

1. Introduction

The 4th industrial revolution has brought about paradigm shifts in various areas, which we can perceive as technological advances that today allow us to overcome barriers with the development of new materials, methodologies, evolution and automation of processes, and connecting people, devices and data through platforms and technologies worldwide.

However, something that we can identify as positive brings with it disadvantages that we are aware of only in the medium or long term. Such is the case of the decision-making process, which faces new challenges due to information overload, group thinking, over-analysis, adoption of new technology, lack of clarity, adding more complexity and demanding faster solutions. In this way, the decision-making process is now placed in a context characterized by volatility, uncertainty, complexity and ambiguity in what we know as a VUCA context (Elkington, 2018).

For balancing the situation, it is necessary to keep well-defined objectives, a fair comprehension of the environment, clarify complex situations. It can adapt quickly to change, as proposed by the author (Johansen, 2007) with his VUCA-prime proposal to counteract or face risk factors identified. Tools such as information visualization can leverage the VUCA-prime components, significantly to clarify and improve the understanding of the information that is a vital element for the decision-making process within organizations.

Information visualization is vital to facilitate the analysis of complex information to make sense of it and understandable to stakeholders, albeit there are few shreds of evidence of how it provides insights, making organizations gain control over their most critical processes. The outcome of information visualization is the acquisition of knowledge that will enhance and provide support to the decision-making process in the analysis of complex problems (Al-Kassab, 2014).

The phrase “a picture is worth thousand rows” could be the adage of information visualization (Lurie N. H., 2007). Information visualization is known to be an extraordinary instrument that provides visual representations of data to facilitate human comprehension and analysis. Through analytical reasoning, the approach is the baseline to communicate the acquired knowledge among all levels of an organization to support rapid decision-making.

Due to the limited use that is made of the capabilities provided by information visualization, it is proposed to apply it as a tool to improve and maximize visual perception in such a way that it transmits information effectively in different management areas (Zhang, 2012). A work that extends Zhang's idea on the relevance and acquisition of information visualization in management areas is that from (Eppler, 2013) not just as a presentation of information but as a more powerful tool. A critical task in the decision-making process is when stakeholders have different areas of specialty, and they need to put together all their knowledge to generate new ideas or solutions to a given problem. The visualization of this generation of ideas will give to the stakeholders a way to have a better appreciation and will not only improve their analysis but also will enhance collaborative work. The implementation of visual representations increases communication, idea generation, knowledge transference, as well as the collaborative work among diverse processes, including decision-making (Eppler, 2013).

To take full advantage of the benefits of information visualization in decision-making activities is crucial to know how it works and how it impacts the process. According to (Thomas & Cook, 2006) information visualization enhances sense-making by increasing human cognitive capabilities when using visual resources to broaden operational memory, showing large amounts of data with relatively small representations reducing search while encouraging the recognition of patterns to facilitate the interpretation, and exploring the behavior of variables. The insights gained are translated into knowledge that will give them a set of criteria that will guide them into a consensus for a final choice of alternatives.

Human perception and cognition are limited, in a simple description, the perceptual system creates a mental representation, then evaluates the new representation by comparing it with known ones, and action is taken. Fig.1 illustrates the three stages of the perceptual system.

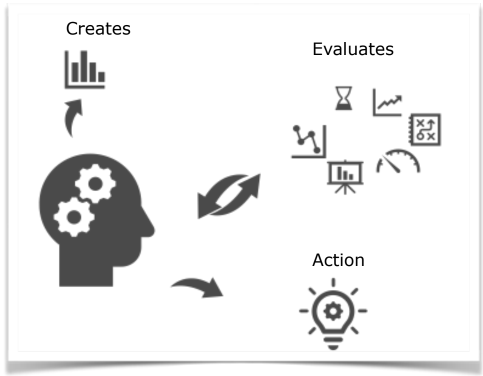

Fig. 1 Traditional model of human perceptual system.

For this reason, after identifying the limitations, there is a need to improve human perception through the use of tools that will maximize these capabilities, developing a more in-depth and faster understanding of information.

The use of visual representations allows the increase of cognitive reasoning by highlighting the critical aspects, whether negative or positive, through the exploration of data.

In decision-making, some models propose the use of facts to generate alternative courses of action engaging all those concerned into better judgements. An evidence-based decision-making model is known to be a dynamic process where its core is evidence itself as a basis for decision-making where individual preferences, values and judgments generate within an organizational context a set of evidence-based options (Wright, et al., 2016). In this model, evidence comes from a set of elements such as education, judgment and experience from an individual level that is influenced by institutional and organizational preferences and values in a given context considering possible constraints (Baba & HakemZadeh, 2012).

The highlights in the model are, that it is a dynamic process, although the shown diagram looks linear or one-way process, it is a cyclic process that needs to go from the beginning every time until all conditions are satisfied. Since it works from a multi-level perspective within an organizational or institutional context, it is used or implemented for the decision-making of complex problems. The knowledge generated during this process is not the solution to the stated problem or goal to achieve, instead, these transform in a series of different alternatives to offer decision options.

1.1 Problem Statement

Recalling the information visualization process and placing it in perspective within the above model there is a need for an information visualization tool that matches the evidence-based decision-making model activities to boost the idea, knowledge, and above all, fact generation to efficiently fulfill the decision-making goals. For this reason, we identify the following problems in this research area:

Dynamic Visualization. Given that the EBDM is a cyclic model, there is a need for a tool that allows information to be up to date and with real-time edition, keeping sources and views synchronized, while stakeholders make use of the tool.

Multi-level perspective. Recalling that EBDM model works in multi-level perspective gathering knowledge, experience, judgements from individuals in an organizational or institutional context, on a collaborative effort to create different solution alternatives given a complex problem, a tool should allow multiple views in parallel to facilitate the analysis and visualization of concurrent facts, KPIs, statistics and any other information related to a given situation.

Human-centered visual representations. Information visualization techniques adequate to the information to be presented. Related to the type of information to be presented, the tool should consider using the best graphic representation that will give better insights, an easier way to analyze and digest information to stakeholders.

If we can understand on detail each of the related processes, activities and needs related to the evidence-based decision-making model, we could maximize the benefits of information visualization and incorporate those characteristics that, as a complement, give us a tool with the ability to facilitate and support decision-making.

1.2 Research Questions & Hypothesis

According to the problems mentioned in the previous section, the following research questions arises:

Q1. What kind of architecture design can be better in order to have an adequate information visualization tool that enhance the insights of results presented to decision-makers?

Q2. What are the features needed and how can we measure them in order to know if they provide an effective visualization tool?

Due to the problem statement and the research questions presented in the previous section, the following hypothesis are defined:

H1. It is possible to design an architecture based on the fundamentals of information visualization with unique characteristics that allow improving and enriching the perceptions derived from visual analysis for decision makers.

H2. In an architecture design, dynamic and multi-screen visualization are features that enhance the information visualization tools hence give better support to the evidence-based decision-making activities in a multi-level perspective.

In this section we introduced the context of the problem statement, the research questions and hypothesis. In the next sections we will review the theoretical framework, related work, solution proposal, test process, results and discussion and conclusions related to this work.

2. Theoretical Framework

In this section, the theoretical concepts used during the development of this research are exposed. In this way, six concepts will be reviewed: graphic user interfaces (GUIs), information visualization, decision-making, software architecture, metrics, and information visualization evaluation.

The graphic user interfaces section has been included as a basis or preamble for developing information visualization. In the information visualization section, its background and definition are mentioned; it is also complemented with useful elements for its implementation and the alternative of multi-screen visualization.

The topic of decision-making is included to explain the need to implement information visualization in this area to enhance or improve results within the process.

The software architecture and the patterns defined, Service Oriented Architecture (SOA) and Model-View View-Model (MVVM), are essential to the research proposal. The subject of metrics and evaluation of the visualization of information is also critical to understanding the research proposal. We will also review existing visualization tools that are most used today.

2.1 Graphic User Interfaces

Throughout the development of computers, from batch-mode machines with perforated cards and printed outputs per line (1960–1989) (Myers, 1995) to today's digitized, user-interactive pan-sized touchscreens, user interfaces have evolved in leaps and bounds, facilitating and improving the user experience.

However, although today's user interfaces are simpler and more user-friendly than past interfaces, their implementations are less straightforward. Developing the graphical user interface (GUI) consumes 48%–50% of the total development time of an application (Kennard & Leaney, J , 2010). Therefore, minimizing the effort of creating a GUI using different techniques, tools, and approaches has become an urgent priority.

Since the mid-1990s, the excessive time spent in the development of GUIs has been alleviated by user-interface management systems, which are broadly classified into the following approaches language-based tools, model-based generation tools and interactive graphical specification tools (Kennard & Leaney, J , 2010). In language-based approaches, the syntax of the user interface is specified in a particular-purpose language, which limits the use of these tools to professional programmers only. Application frameworks generate specific application details through class definition, and some frameworks provide graphical or interactive editors for the visual part. Nonetheless, much of the functionality still needs to be programmed. In model-based approaches, automatic generation tools are created by models, and the GUI is performed by a high-level specification. However, as these approaches have not fully matured, the development language is complicated, and the interfaces generated are not yet sufficiently robust. Interactive graphical specification tools are direct graphic specifications that place objects on the screen by a pointer. These tools are very easy to use and require no special knowledge or skills, so they are accessible to all users. Moreover, advanced programmers can use them to specify the behavior of dynamic objects that change throughout the application's execution.

2.1.2 Graphic User Interfaces Tools

The importance of a GUI relies upon communicating information in two directions with the user. GUIs' creation is achieved with the use of tools, sometimes known as approaches, with a goal in common to generate a structure, foundation, or model with elements that facilitate the creation of the GUI. Some GUIs can be created during the design or run time of the application or program (Raneburger, Popp, & Vanderdonckt, 2012). Graphical User Interface Tools are identified by a specific function they provide, such as restate information already contented in an application and make it ready for presentation. This kind of functionality can be implemented by interactive graphical tools, model-based tools, or manual development. Another function is inspecting the code to find information that's relevant to be presented. Tools that can provide this functionality are the language-based tools (Cerny, Chalupa, & Donahoo, 2012).

We made a review of tools for developing GUIs by following the classification proposed by authors (Kennard & Leaney, J , 2010), language based, model bases, and graphic interactive tools. The review on detail is on Appendix 1. Here we will summarize the findings related to GUI creation.

Reviewed tools from each classification can be seen in Fig. 2, where blue color denotes the language-based tools, orange color denotes the model-based tools and green color denotes the interactive graphical specification tools. Note that in recent years, more tools that simplify the code generation have emerged. This visual representation clarifies the achievements to date and the interest in reducing the effort, time, and resources employed in the generation of GUI codes.

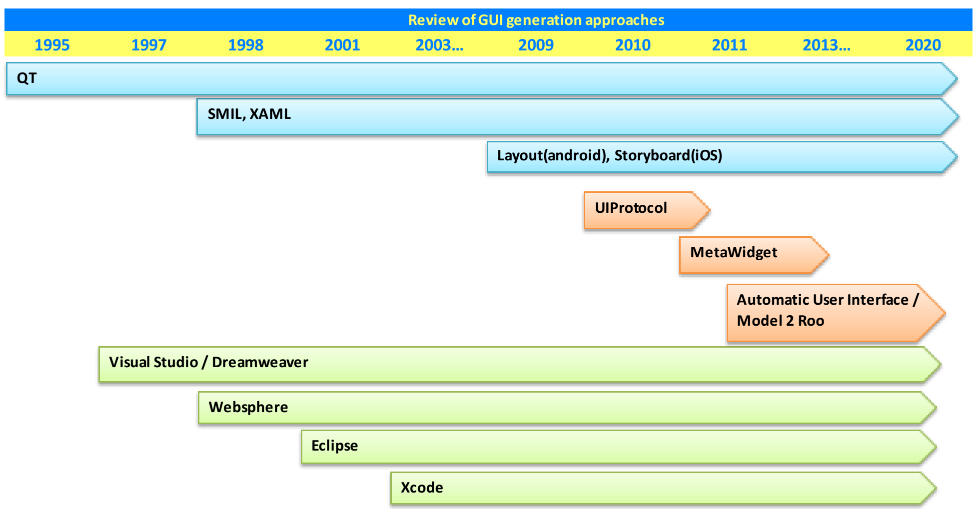

Fig. 2 Timeline showing the approaches reviewed. Blue, orange and green shadings denote language-based tools, model-based tools and interactive-graphical tools, respectively.

Language-based tools.

As stated above, special-purpose languages facilitate the syntax specification (input sequences or output actions) of user interfaces. One disadvantage is the large variety of languages, which presents a steep learning curve to programmers. Rather than learning each new language as it becomes available, some programmers prefer to work with graphical tools in the user interface. Among the language-based tools, we reviewed Qt, SMIL, XAML, Layout (Android), and Storyboard (iOS) in the appendix.

Model-Based Tools.

Model-based tools. In language-based tools, the programmer must specify the complete format of the user interface. This problem can be resolved by the model-based approach, which automatically generates the format from a higher-level specification. When the menus, buttons, and other interactive widgets are defined, the GUI is automatically generated, and can be changed or transformed by the user, providing flexibility and user-friendliness.

In the first study on model-based tools, (Kennard & Leaney, J , 2010) considered that a GUI should be generated in an adequate architecture with five main characteristics: 1) inspection of existing heterogeneous back-end architectures, 2) different existing practices for the application of inspection results, 3) mixing of recognition and multiple widget libraries for UIs, 4) mixing of support and multiple UI embellishments, and 5) application of mixed and multiple UI layouts. This approach offers flexibility for reviewing and ordering business objects after the first inspection. To improve the trustworthiness of the GUI, a reference to the user interface page should be maintained. The incorporation of free third-party pluggable widgets complements the interface construction by adding extra validation to the required format.

Although the final design was not successfully generated as expected, the proposed architecture satisfied its main objective; to facilitate the generation of a GUI. The authors also automated the GUI generation and reduced the number of lines of code. However, whether the proposed architecture reduced coding, development time could not be known, as the final results were not compared with the pre-solution results.

Interactive Graphical Specification Tools.

Interactive graphical specification tools. Permit users to add and format graphic elements on a UI. Most of these tools come under the umbrella of the Integrated Development Environment (IDE) framework. As most of them also change their behavior throughout the application execution, the added elements can be both static and dynamic. Therefore, the user can set the initial state of the graphic objects and their interactive dynamics depending on the final selection and user–app interaction.

The SWOT analysis (see Table 1) identifies the advantages and disadvantages of each approach presented.

***Table 1. SWOT analysis of the approaches reviewed in graphic user interfaces tools*.**

	S	W	O	T
Language-based	Consistency between functionality and graphical representation. Alleviates the difficulty of changing and modifying the code	Steep learning curve. Specialized programmers needed.	Powerful tools: give the programmer total control of the code, automation is possible.	The weaknesses drive some programmers toward better graphical tools
Model Based	Automatic generation from a model definition.	Because a higher specification is required, a specialized programmer is needed	As there is no unique model, most of the proposed solutions, systems, protocols, and platforms are developed ad-hoc to suit the user's needs	Modifications can be done in code. Consistency between application functionality and GUI can be compromised
Interactive graphical specification tools	Provide a graphic-friendly interface that is tractable to novices. Allows the integration of static and dynamic objects that can change their behavior during the application´s execution.	Some tools require significant hard disk space for installation. Some are free, but some licensed products are very expensive	The tools enable users to create cross-platform robust solutions by integrating many technologies	Increased licensing costs, require continuous upgrades

Language-based tools are powerful and complete. As the code is not easily manipulated, the application's functionality tends to be consistent with the graphical representation. The languages reviewed in this paper are license-free and installable with minimum requirements. However, when a new language is introduced, the steep learning curve presents a barrier to programmers. Moreover, a language-based solution can be constructed only by specialized programmers.

In model-based tools, the generated model should allow automatic creation of the GUI. One disadvantage is the lack of a unique model, so each model must be created to meet the requirements of the application. Again, the solution can be created only by specialized programmers. Moreover, when the modifications are directly programmed in the code, the consistency among the application functionality, GUI and model is easily disrupted.

In contrast, the GUI provided by interactive graphical specification tools is easily applied by novice or non-specialist users. Besides being friendly and intuitive, interactive graphical specification tools provide an editor that modifies the elements with code. The objects' behaviors can change during the application execution, conferring a dynamic property. However, the installation requires certain technical features such as frameworks or SDKs. The rich GI and other integration tools also demand significant hard disk space. Interactive-graphical tools are usually as license free or fully licensed versions. The limited functionality of license-free tools urges users to eventually upgrade to the fully licensed modality, which is generally expensive; moreover, the updates are frequent and sometimes require simultaneous application updates.

As mentioned before, the development of GUIs is the preamble to the subject of visualization information, and therefore its study and review are essential for understanding it. The central theme of this research focuses on the visualization of the information that will be explained below.

2.2 Information Visualization

The use of images to represent information (Few, 2007), is the simplest definition to understand data visualization. Data visualization has been present since the second century when the first table was created to organize astronomical information in Egypt

(Few, 2007). The need of having graphic information remains so far, not only to make the information accessible, but to make easier the way is translated and presented to be understood by the final user. Data visualization can be recognized as the presentation of graphics that helps to organize the information in a visual or graphical way. However, as formally defined by (Chen C. , 2010), it refers to the generation of graphical representations of information. The main feature that has to be noticed in the 21st century is that data visualization is achieved with computers, through already existing software or ad-hoc systems.

Different tools to create visualizations have been created since the term showed up. However, independently of the features and effective visualizations that can be created with any of these tools, organizations have started to notice the importance to the subject itself. Visual languages can be used as an effective communication medium in the management field (Zhang, 2012), hence visualization can be considered a visual language. Not just limited to quantitative but to qualitative visualizations it can be used in different managerial functions including idea generation, decision making, and planning. The reason visualization should be taken into consideration is because it could play an important role to support the decision-making processes inside the organization. Visual information can provide a deeper meaning of the information, as well as different perspectives that can allow seeing such information with different insights providing valuable data for critical processes and going beyond of what it's already visible (Al-Kassab, 2014).

According to (Chen C. , 2005), when data visualization started to take more importance and become part of information analysis, there were already unsolved problems related to data visualization. After more than ten years, we consider that some of these problems still remain. One of the most important is to see visualization not only as some "other tool" but as an important component that supports the understanding and comprehension of information to see what's beyond the numbers or graphical representations that we see. This is related to the proper training and education that should be given to professionals, so they can transmit the impact of visualization to internal or external members of the organization. Scalability, aesthetics and dynamics are relevant features to consider. Scalability referring to, growth of data, technology evolution (Marx, 2010)and the constant change in how people access information, (Feinleib, 2014). Aesthetics is as important as any other element in visualization, it's the way information is going to be presented in a graphical, visual form. It has to be visually appealing, clear, ordered and intuitive so users are able to interpret the information without extra details. Dynamics refers to information always changing through time, so the structures used to represent it should not remain static, and most important it should provide functionality to test hypothesis and projections beyond what we can see in present time. These features have been considered to develop and create software to generate visualizations in many areas and levels within an organization.

While information visualization was considered to be a subject in which only specialized professionals were able to create or produce graphical representations, now is considered to be a useful element in many areas as diverse as possible, from IT professionals to business roles, saving time and making processes more efficient (QlikView® architectural overview, 2014)

Information visualization tools have emerged to support the complexity derived from the development of visualizations such as analysis, modeling, coding, testing, and maintenance. The use of taxonomies has been useful to generate tools with desired capabilities that will significantly reduce the effort to create visualizations (Bassil & Keller, 2001). The existing variety of taxonomies, guidelines, or models to facilitate de information visualization is extensive. However, according to the authors (Kienle & Müller, 2007), software visualization tools should have at least the following quality attributes rendering, scalability, information scalability, interoperability, customizability, interactivity, usability, and adoptability. As for functional requirements, the following are desired views, abstraction, search, filters, code proximity, automatic layouts, and history. As the need for information visualization tools continue to increase, its evolution will continue to ensure a reduction in the complexity of its development.

2.2.1 Visualization Elements

The visualization elements mentioned in this section are meant to be a guide for designing or creating visualizations. A guide is a series of principles and guidelines, but not strict rules, that are suggested for creating visualizations (ENGELBRECHT, BOTHA, & ALBERTS, 2015). These display elements are context, chart type, distraction removal, focus, design, telling a story and are taken from (Knaflic, 2015).

The context refers to having a frame of reference for the information we want to present, knowing who our audience is, and clearly identifying who the stakeholders are and how we can use the information to convey a clear message that leads to the desired action, such as making an important decision.

To effectively present the information, it is crucial to choose a graph that represents the data and is easy for the interested public to interpret. Although there is a great variety of graphics today, it is essential to mention that those that are better known will be easier to interpret due to the familiarity that exists between our public and the graphic representation. In this way, it is suggested to use charts such as simple text, scatterplot, vertical bar, line, slope graph, and horizontal bar, depending on the type of presented information, these are shown in Fig.3. However, it is essential to mention that there are some types of graphics that, although they are known, their use causes confusion rather than clarification or facilitation in the assimilation of information, and it is widely suggested to avoid their use. Some of these charts are double y-axis charts, 3D charts, donut charts, and pie charts.

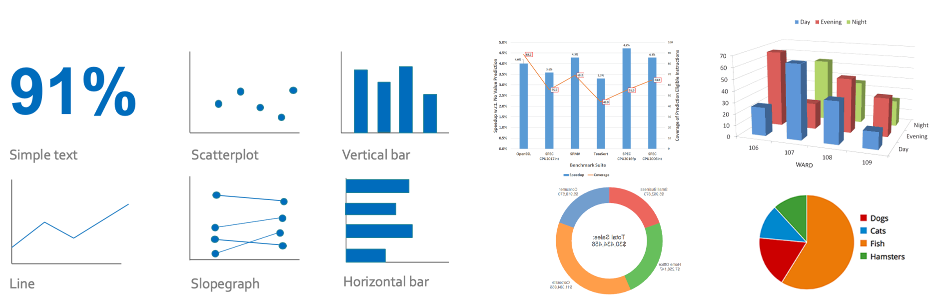

Fig. 3 Left most common used charts to show data. Right, charts to avoid.

Eliminating distractions in the visualizations is vital because any unnecessary element will produce unnecessary cognitive load making it more difficult for our audience to capture the message we want to convey. It is recommended to use the principles of gestalt to facilitate order and space in our visualization. These are shown in Fig. 4.

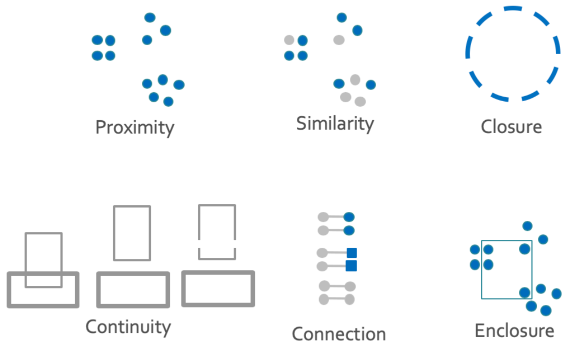

Fig. 4 Gestalt principles.

Focusing the audience's attention is as important as eliminating distractions as it will allow the audience to focus their attention on what is essential. For this, it is recommended to make use of the pre-attentive attributes as necessary. These are shown in Fig. 5.

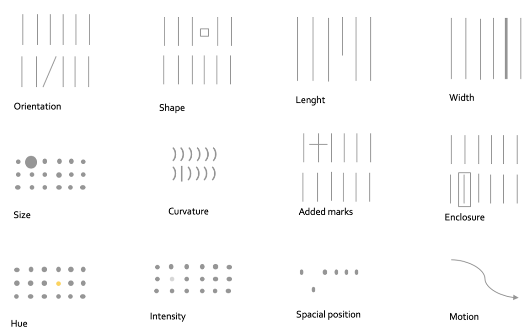

Fig. 5 Pre-attentive attributes.

Regarding the design, it is necessary to use already known elements to highlight the relevant information through a combination of the use of the font, size, color, italics, bold, etc. Similarly, use colors that allow to see clearly and highlight data without being too flashy or tiring for the eyes. Arrange each element within our visualization in an orderly manner, through the use of hierarchies, and eliminate unnecessary elements, avoiding saturation of elements that result in a saturated visualization. In the use of graphs, it is vital to name the graph concerning the information that is being presented, to identify the

axes correctly, and, if possible, by using text to place a brief conclusion. The design should be simple and focus on creating a clear and visually pleasing display.
Telling a story will be the most practical way to involve the audience in the problem or situation presented and implement the elements mentioned above. This story must have a beginning, a middle, and an end. In the beginning, the context will be introduced, and points of interest will be highlighted for the audience. In the middle, the previously introduced points will be developed, and evidence that supports the proposal or planning made at the beginning will be shown. In the end, action must be motivated so that our audience is clear that a decision must be made regarding what is presented. A story has a structure that allows us to capture and keep our audience interested by making all the information presented easily assimilated (Knaflic, 2015).

2.2.2 Multi-screen visualization

Arizona State University's (ASU) Decision Theater® is an immersive data visualization and computational modeling facility located at the ASU Tempe, Arizona and Washington D.C. campuses. The organization integrates data analytics, computational modeling and data visualization into a common operating software model to better understand and address complex problems.
The Decision Theater® at ASU provides an immersive visualization and collaborative physical environment. It purposely wields technology to promote and engage collaborative behavior to address public challenges (Hu, Johnston, & Hemphill, Fostering cooperative community behavior with IT tools: the influence of a designed deliberative space on efforts to address collective challenges, 2013). Its circular convening space visualizes complex systems across 7-screen panoramic 4K displays. It utilizes OpenSceneGraph and Google Earth for 3D spatial visuals, Tableau for rapid visualization of data, QGIS and ESRI for geographic data, and development of custom web-based visuals using open-source resources.
The ASU Decision Theater® partnered with ASU's Decision Center for a Desert City to model and simulate groundwater aquifer depths (Larson & Edsall, 2010) and the impacts of water-use decisions. *WaterSim* is a computer simulation model that represents groundwater supply and demand from residential, commercial and agricultural sectors

(Gober, Wantz, Lant, Tschudi, & Kirkwood, 2011). The model allows stakeholders to explore problems of deep uncertainty and look for robust policy options for water sustainability (White, Wutich, Larson, & Lant, 2015).

The Decision Theater® approach to visualization and user engagement fosters learning and understanding of complex issues, as in the example of a *Watersim* model, which describes it as an "innovative mechanism to link knowledge and action for a sustainable environment" (White, Wutich, Larson, & Lant, 2015). The visualization model approach engages both policymakers as well as the general public in an equal manner. The Decision Theater® presented the model over the years to various groups, including middle school groups from around the Phoenix valley, policy and informatics classes at ASU, and water policymakers from various states. The ability to visualize and simplify, and also focus on various aspects of water policies (indoor vs. indoor water use, personal choices vs. public policy changes that affect water management) allows for broader conversations and engagement of such different audiences. Within ASU, the model is extensively used in classes studying public management, policy and business management. Exploratory studies done (Hu & Johnston, Using a wiki-based course design to create a student-centered learning environment: Strategies and lessons, 2012) engaged classes of 2010 and 2011 showed that *Watersim*, being a dynamic computer simulation, statistically increased the knowledge about water issues in Phoenix. The multi-screen environment of the Decision Theater® provided dynamic context and creative learning environment for students to individually and collectively apply systems thinking in information-rich environments with instant feedback channels. Interacting with the simulation also allowed for feedback loops previously inaccessible through typical learning forums about water policy. The Decision Theater's® physical space also contributes to creating an optimal learning environment that focuses on the learner and collaboration. According to (Graetz, 2006), changes need to be made to typical classroom design and learning environments that combine learnings from environmental, human factors and social psychology. The visualization model used to teach and help the students to understand a specific topic has proved to be efficient in engaging the student in learning as it enhances their performance (Yakovleva & Yakovlev, 2014). The use of immersive and interactive visualization are the elements that encourage the active participation of students instead of being only observers and are motivated to take part as mentioned in the method of action learning (Networks of Decision Theaters (Mexico City Campus), 2019).

2.4 Decision-Making

As defined by Turban, "Decision-making is a process of choosing alternative courses of action for the purpose of attaining a goal" (Aronson, 2005). This simple definition gives us a clear idea of a process that is executed day by day by managers of different areas within an organization. At the moment of the choice, the decision-maker is the manager.

2.4.1 Decision-Making definition & process

The process of decision-making comprises four main phases, sometimes referred to them as activities. These phases are finding occasions for making a decision, finding possible courses of action, choosing among courses of action, and finally evaluating past choices, as first stated by Simon (Simon, 1960). Some other authors (Aronson, 2005), took the Simon's definition and redefined the four phases as: intelligence, design, choice and implementation. Fig. 6 shows a conceptual diagram of the four phases and the interaction among them.

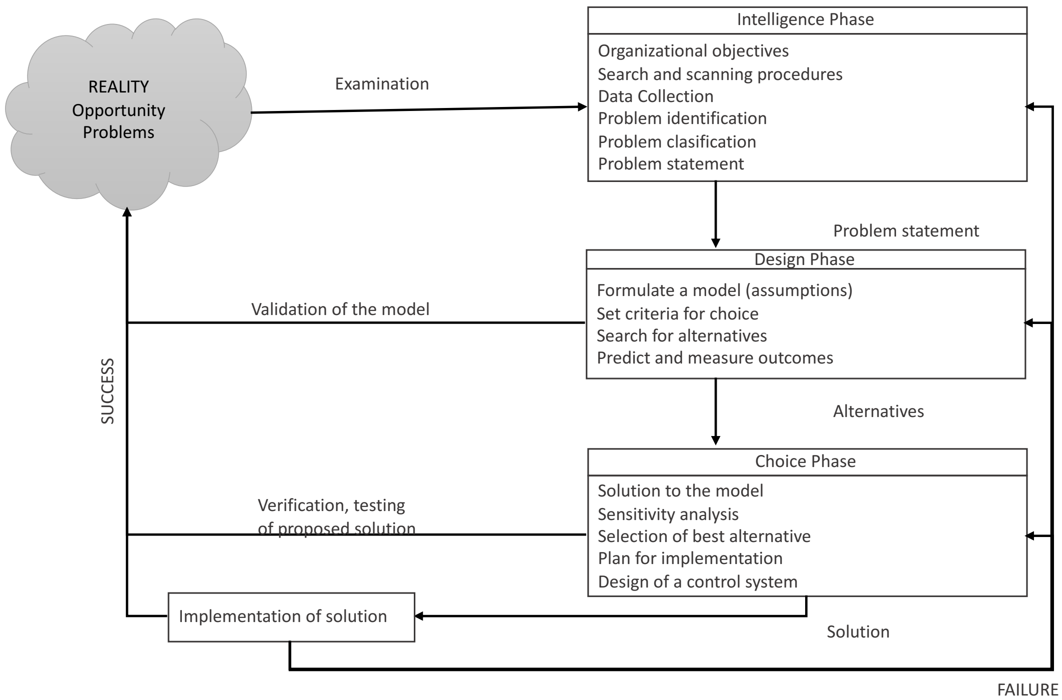

Fig. 6 Four phases of the decision-making process

As the decision-making process is not something that happens once as a series of steps to follow like in a cooking recipe, the process follows a constant loop of feedback as shown in the diagram between one phase to another. The intelligent phase is where a specific problem, or opportunity is identified according to the organization needs, goals and or objectives. Organization's productivity level can be a measure to recognize that are some opportunity areas of improvement within the organization. Some activities included in the intelligence phase are observation in the organization productivity level, analyzing organization procedures, collecting data, defining and stating problems that need to be solved. However, during this phase there could be some restrictions that can limit or make it difficult to achieve, some are, full data not available, accurate or enough, data recollection can take longer time than expected, assumption of future data being similar to historical data. In the design phase, once the problem has been identified and the data has been collected, the course of action are defined. The decision-making model is built here, identifying the variables and the relationship that exists among them. The choice phase is one of the most critical one, since the actual decision is made as long as with the course of action to follow to achieve the goals related to the decision made. In the implementation phase is the application or put in practice of the plan selected in the design phase. After all phases, intelligence, design, choice and implementation have been followed, the implementation phase is evaluated in the intelligent phase to verify that all the activities followed in the process are achieving their purpose. If failure occurs within the implementation phase, then the process goes back to the start, intelligence phase, to make a verification and review to adjust the course of action.

2.4.2 Evidence based decision-making

The difference among structured, semi structured and non-structured problems already shows enough complexity in the decision-making process. The identification, collection of resources, gathering the people, managers, representants, and managers and working in a collaborative way is a complex management decision. For a complex management decision process evidence need to be added or integrated as part of the process. The evidence in this case can be either quantitative, qualitative, theoretical and a mix of the three. Evidence-based decision-making was previously known as evidence-based medicine, since medicine is the field in which it was born. Evidence is known to be facts, organized sets of information or observations, that is used as support to justify beliefs or inferences. Two main characteristics of evidence itself is to be rigorous and relevant, in which case both have to be relevant to the context they belong.

Evidence-based decision-making consists of two different stages to achieve its practice. The first one is related to the academic field of a profession while a second is the one that implies using the evidence itself in practice with the objective to perform informed decisions based on them. Regarding to an organization the second stage is the one of interest and is better known as evidence-based management, in which to make informed decisions, the proposal of a model is made. This model proposal follows three main principles. The first one states that the process of decision-making is not seen from a rational view, instead the second principle suggests adopting a multi-level perspective to consider the integration of different and contextual factors, during the process. The third one suggests total transparency of decisions.

An evidence-based decision-making model seeks to provide evidence along with the context needed to interpret it in the best way. Although quantitative information is objective and clear, the qualitative information provided need an appropriate interpretation that are related to the preferences and values of stakeholders. Model must be accompanied with a proper methodology to gather the best evidence by selecting the right questions and looking to the best approach to answer them.

As the decision-making process itself, the evidence-based decision-making process is dynamic, focusing on the transformation of the discovered evidence into management decisions.

As proposed by (Baba & HakemZadeh, 2012) a model should follow a multi-level structure with a well-defined and structured individual level following a cross-level configuration between individual, organizational and institutional levels individually and in collaborative way. Fig. 7 Shows the evidence-based model proposed by Baba.

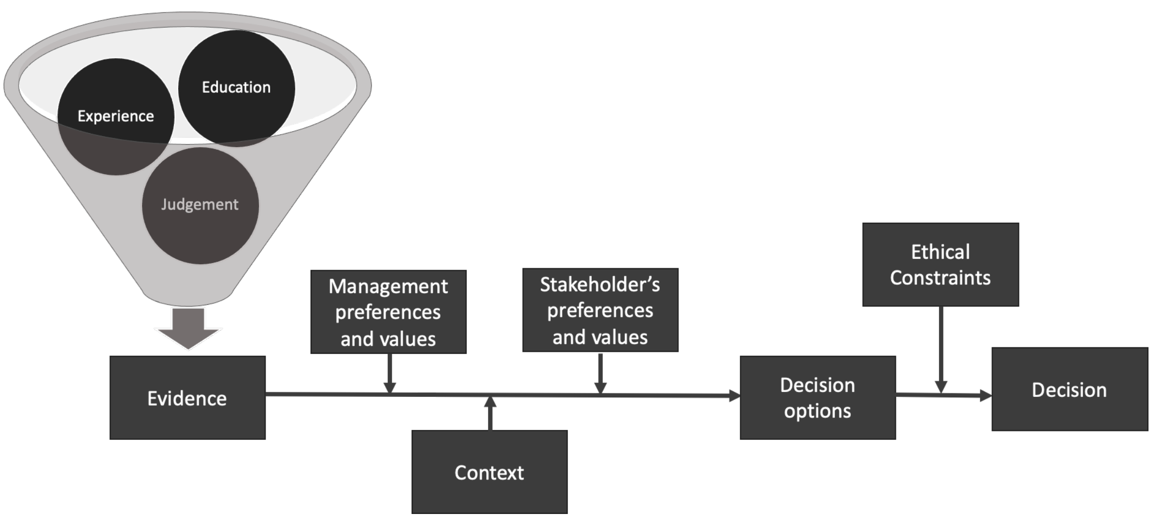

Fig. 7 Evidence-based decision-making model proposed by Baba, V. V.

Individually managers use the evidence to support their training, education, experience and judgement, although the process is influenced by manager's preferences and values. Other influences are given by the stakeholders which preferences are defined by a mix of institutional, organizational and individual levels.

An important role is played by the context in which the decision-making is made, but taking in consideration structural, environmental, cultural and political constraints which are directly related with the context. Also, the ethical constraints should be taken in consideration.

The authors (Wright, et al., 2016), propose an evidence-based decision process as shown in the Fig. 8.

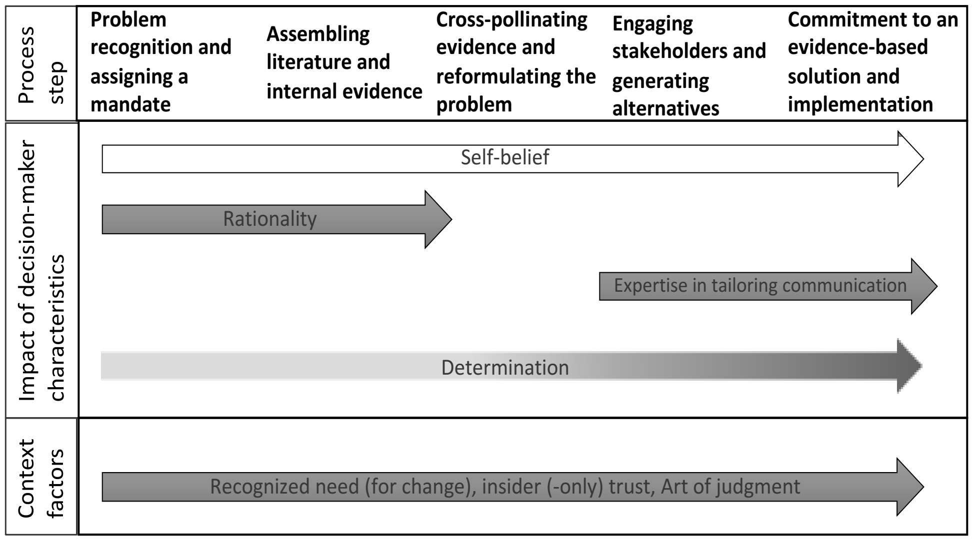

Fig. 8. Decision-maker and context factors within the EBM process

The process consists of five main phases, problem recognition and assigning a mandate, assembling literature and internal evidence, crossing evidence and reformulation of the problem, engaging stakeholders and generating evidence-based alternatives and committing to an evidence-based solution and implementation. The process also considers context factors, and the impact of the decision-maker, such as self-belief, rationality and expertise.

The evidence-based management incorporates many factors through a model or process as part of a more systematic process within the organization in a scientific form. However, it has to be understood that evidence itself has to be seen as the input within the information process that will transform it into better judgements in order to make better decisions (Rousseau, 2015). It is important to have in mind four main activities that are, using the best scientific findings, gathering organization facts, reflective judgement to reduce bias and improve decision quality without forgetting ethical issues that may impact the decisions of stake holders.

2.5 Software Architecture

The subject of software architecture has evolved since the emergence of the first ideas, such as the proposal of Edsger Dijkstra who in 1968 identified the need to create a structure before carrying out the development of a system. Concepts like this gave way to others such as P.I. Sharp, who in 1969 made a clear difference between engineering and software architecture, mentioning examples where the individual work of an engineer reflects good programming practices. However, when combining the work of several engineers in a single system, the result is a shapeless set of programs due to lack of an architecture. Fred Brooks, in 1975 designated the concept of system architecture as "the complete and detailed specification of the user interface" making an analogy of the software architect with the architect who builds a house. Fred Brooks also differentiated between architecture, which says what to do, and implementation, which refers to how to do it. It is around 1980 when the term software architecture begins to emerge formally, and in 1992 a work on software architecture is developed under the authorship of Perry and Wolf, who also make a comparison with building architecture and whose concept is still used (Reynoso, 2004) today.

Due to the nature of the evolution of software architecture, various authors have created different definitions trying to make clear what the particular function of software architecture is. As an example, we can mention the proposals of Perry and Wolf, Garlan and Shaw or those of ARPA Domain-Specific Software Architecture (DSSA) and that made by the authors Penedo and Riddle. However, in order to integrate and complement the existing works, the authors (Gacek, Abd-Allah, Clark, & Boehm, 1995) propose to take into account the different stakeholders who have different perspectives and therefore different needs. Thus, the authors identify five types of stakeholders: the client, the user, the architect and the systems engineer, the developer and the maintainer, each one of them with particular needs that will be met by a system whose integration of components, connections and restrictions provide a solid structure.

One of the definitions of software architecture reads as follows: “The software architecture of a system is the set of structures needed to reason about the system, which comprises software elements, relations among them and properties of both” (Bass, Clements, & Kazman, 2013).

When we speak of a set of structures, these refer to a certain number of elements and that are integrated through a defined relationship that together form the system. Although there are many types of structures, the most common are grouped into three classifications, module structures, component-and-connector structures, and allocation structures.

The module structures are named thus precisely because they group elements, of the same type, that compose them in modules or units. Its primary purpose is functional, although it is also used to indicate dependency and usability of other software elements and the relationship it has with other modules. Some of the most used module structures are decomposition structure, uses structure, layer structure, class structure and data model.

Component-and-connector structures are used to organize behavior in runtime as well as the interaction between the different elements that make up the structure. The components are responsible for the behavior at runtime, as well as the connectors of the interaction. Also, these types of structures serve to identify which are shared data stores, how data is transmitted through the system, which parts of the system are replicated, which parts of the system run in parallel, if any part of the system changes during the runtime and how that change is made. Some structures of this type are service structure and concurrency structure.

The allocation structures are essential to know how the system will use other types of structures that are not software, such as file systems, CPUs, networks, Etc. The allocation structures are used for the creation and execution of the system. Some of these structures are deployment structure, implementation structure, work assignment structure.

Architectural patterns are a composition of architectural elements whose design has been approved to solve common problems that arise. The solution provided by the pattern provides a set of types of elements, interaction mechanisms, a typological design of the components as well as their limitations and behavior. Architectural patterns are made up of a context, which is the recurring situation in which a problem occurs, the problem, the situation to be solved and the solution, which are the resources that together will solve the problem presented. Several patterns are often used simultaneously and adapted in a way that provides a single and effective solution. In a similar way to structures, patterns are classified, to facilitate their use and application, into three groups: module patterns, component-and-connector patterns, allocation patterns.

2.5.1 Service Oriented Architecture (SOA)

Service Oriented Architecture, SOA, is a pattern solution used for providing and consuming services over a network. In the early 2000s, SOA took on greater prominence after Web Service Technology Service was accepted as the standard for business application integration. SOA is useful in business scenarios where applications communicate with one another. Organizations implement SOA to develop and keep up to date over time (Gu & Zhang, 2010).

The context of the SOA pattern is characterized by having several services offered by different providers and which in turn are used by consumers who must be able to use them without the need for specialized knowledge. The problems that arise when exchanging services between providers and consumers is the variety of platforms, languages and interoperability through the internet as well as maintaining good performance, security and availability when doing this exchange. The SOA pattern solution provides a description of the collection of components belonging to providers and/or consumers without the difference between their platforms or languages being a limitation. The description of the components is available through interfaces that include information on the services they provide as well as the services they consume. An SOA application can also offer infrastructure services through the use of specialized components such as enterprise service bus (ESB), service registry, orchestration server (Bass, Clements, & Kazman, 2013).

The implementation of the SOA pattern implies that the architect must know the standards and protocols necessary to make the services accessible. Another critical issue to consider is security, as the service can be public or limited to a certain number of consumers. Consider minimizing attacks across the network as denial-of-service. It must also take governance into account to ensure that the demand for services is satisfied (Governor, Hinchcliffe, & Nickull, 2009).

2.5.2 Model-View-View-Model Architecture (MVVM)

The Model-View-ViewModel pattern was created to simplify the creation of interfaces and is derived from the Model-View-Controller (MVC) pattern that we will talk about first to have a better understanding.

The MVC pattern is used on different platforms, and its main objective is to separate the data from the user's presentation or View. The context of this pattern is that users often want to view the information in different ways. Due to this, the user interface needs to be modified continuously. The specific problem is to keep the user interface separated from the rest of the system in such a way that it can display different views of the data and keep abreast of changes, is easy to maintain and can be coordinated with requests made by users. The solution proposed by this pattern is to separate the functionality of the application into three different components. The Model which contains the application data. The View shows the user the interaction mechanisms to visualize the data. Furthermore, the controller that is the intermediary between the Model and the View is in charge of handling user inputs, updating the Model and the View (Bass, Clements, & Kazman, 2013). MVVM pattern is the update of the MVC pattern due to some disadvantages that the latter presents. The MVVM pattern was first known in 2005 when John Gossman posted it on his blog. However, it is also said that the creation of MVVM was inspired by the Model-View-Presenter (MVP) pattern, which is also considered as a variation of the same MVC. MVP was created by Martin Fowler and is similar to MVVM in that it separates View from behavior and state. Although the pattern proposed by Fowler seeks to make the View independent of the user interface, Gossman sought to simplify the creation of interfaces particularly on the Windows Presentation Foundation (WPF) and Silverlight (Smith, 2009) platforms.

The MVVM pattern essentially seeks to separate View from logic. It is made up of three Model components, View and ViewModel. The Model contains all the application data, entities, relationships, as well as functionality. The ViewModel, being the intermediary between the View and the Model, will facilitate the access of the View to the Model. The ViewModel and the View are entirely decoupled, however the ViewModel contains everything necessary to facilitate user interaction with the application through the View. This separation between the View and the ViewModel allows designers and developers to work independently to provide a friendly and functional interface together (McCarter, 2010). The Fig. 9 shows a diagram of the MVVM pattern.

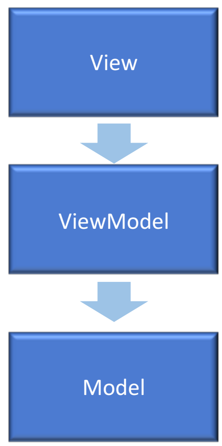

Fig. 9. Diagram of the MVVM pattern and its components.

To better illustrate how this pattern works and the exchange between the ViewModel and the View, look at the Fig. 10. Specifically, the purpose of the ViewModel is to maintain the state of the View. This state is preserved as data and is exposed as methods, properties and operations. Through a two-way binding, the data is loaded by the View and updated to the ViewModel as the user makes changes.

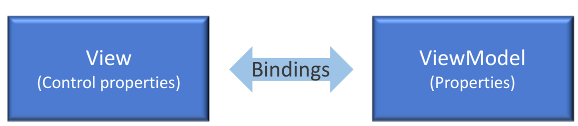

Fig. 10. Bindings between the View and the ViewModel.

The operations exposed by the ViewModel can be carried out using methods or commands. Either of the two forms is adequate, and architect have to bear in mind that the commands encapsulate the logic of the functionality and can be reused and therefore their complexity is greater. Methods, on the other hand, need an action or behavior to be invoked. See Fig. 11. On the other hand, the ViewModel notifies the View of an event that has occurred by implementing event handlers or triggers, see Fig. 12.

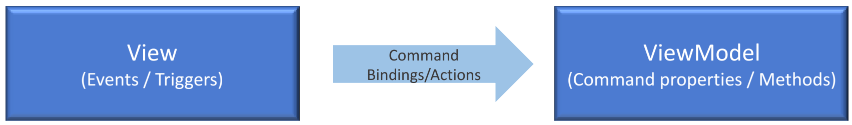

Fig. 11 View using operations from ViewModel.

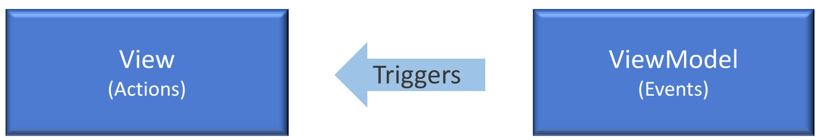

Fig. 12 View handling notifications from the ViewModel.

The separation of view and logic into different classes generates various benefits. Among these benefits are:

- The interface design can be carried out without the need to compile and execute the project by displaying data that allows the designer to have a better idea of the product as it will be shown to the user.
- It allows designers and developers to work simultaneously without interfering or depending on each other.
- The logic implemented in the ViewModel can be reused for the implementation of different views.
- Component independence makes it easy to implement unit tests (Anderson, 2012).

2.6 Metrics & Evaluation of Information Visualization

A metric is a measurement that is based on a scale to evaluate a product. The use of metrics is commonly used as a tool to ensure software quality (Estayno, Dapozo, Cuenca Pletch, & Greiner, 2009). An evaluation study is defined as an empirical inspection used to answer specific research questions through the performance of experiments or ordered procedures and the collaboration of participants. According to the defined objectives, the types of evaluation can be qualitative or quantitative (Elmqvist & Yi, 2015). The existing efforts to design information visualization evaluations are numerous and diverse but with a common objective, which is to ensure that the visualization of the information presented is optimal for the user in different application fields (Costagliola, De Rosa, Fuccella, & Perna, 2018). Several authors agree that most of the existing evaluations are based on HCI guidelines that turn out to be very generic, and therefore there is a gap that allows these evaluations to be adapted to the visualization of the information.

Although several authors agree that evaluations should be generalized, it is considered essential to identify the domain of the problem, as (Munzner T. , 2009) mentioned, due to the nature of the information with which one works and their behavior when interacting with it. Thus, within the decision-making area, it is essential to consider domain-related tasks that, through their implementation in the visualization of the information, allow to carry out better analysis and generation of knowledge to the interested parties. This definition of tasks can help to carry out the design of the system, adapting to the needs of the user. The tasks can also be used to carry out a heuristic evaluation where the precision of the implementation of the tasks will be verified and validated if the representations are adequate for the user and thus be able to make decisions (Amar & Stasko, 2004).

A scale must be validated to ensure a metric is reliable for evaluation and assure its quality. In physical studies, repeated measurements are carried out to prove reliability. However, when this is not possible to achieve, statistical instruments such as Cronbach’s alpha facilitate the means of reliability (Taber, 2018). Formally Cronbach´s alpha is defined as tau-equivalent reliability and is a weighted average of the correlations between the variables that are part of a scale. It can be calculated based on the variances or the correlations of the items (Cronbach, 1951). Formula based on variances is:

$$\alpha = \left[\frac{\kappa}{\kappa - 1}\right]\left[1 - \frac{\sum_{i=1}^{k} S_i^2}{S_t^2}\right]$$

Where S_i^2 is the variance of item *i*, $S_t^2$is the variance of the total observed values, *k* is total items. Formula based in correlations is:

$$\alpha_{est} = \frac{kp}{1 + p(k-1)}$$

Where *k* is the number of items and *p* is the average of the linear correlations between each of the items.

Ideally, it is expected that the scale elements are positively correlated. Hence a closer value to 1 shows high consistency between variables. However, for values below 1, a scale is considered reliable or valid when Cronbach's alpha value is above 0.7, and in some cases, even 0.6 is acceptable to show internal consistency (Taber, 2018).

Another useful instrument to measure a scale's reliability is the Average Variance Extracted (AVE) and Composite Reliability (CR), together as convergent validity usually implemented to assess the degree of shared variance between variables of a model. AVE measures the level of variance given by a construct against the measurement error level. Closer values to 1 are desired. However, those above 0.7 are considered very good, and a value of 0.5 is acceptable. CR is an estimate of reliability, such as Cronbach's alpha, and a minimum value of 0.7 is sufficient (Fornell & Larcker, 1981). The formula of AVE is:

$$AVE\xi_j = \frac{\sum_{k=1}^{k_j} \lambda_{jk}^2}{\left(\sum_{k=1}^{k_j} \lambda_{jk}^2\right) + \Theta_{jk}}$$

Where *Kj* is the number of indicators, λ_{jk} are factor loadings, Θ_{jk} is the error variance. The formular of the Composite Reliability is as follows:

$$\rho_{c\xi j} = \frac{\left(\sum_{k=1}^{k_j} \lambda_{jk}\right)^2}{\left(\sum_{k=1}^{k_j} \lambda_{jk}\right)^2 + \Theta_{jk}}$$

Where K_j is the number of indicators, λ_{jk} are factor loadings, Θ_{jk} is the error variance. For both formulas the error variance for the K^{th} indicator is:

$$\Theta_{jk} = \sum_{k=1}^{k_j} 1 - \lambda_{jk}^2$$

In the cases where the value of AVE is less than 0.5 and the value of CR is above 0.6 the convergent validity is still adequate (Huang, Wang, Wu, & Wang, 2013).

3. Related Work

The use of visualization tools to support decision-making has been more and more frequent in recent years. More than support tools, they have become a necessity due to various factors that require the use of technology to adapt and achieve objectives among groups of people who need to work with common goals and implement solutions quickly and efficiently. Some of these factors that interrupt or hinder traditional processes in the decision-making area are the rapid generation of information derived from technological advances, which in turn causes a high demand for exchange and feedback of information, multidisciplinary groups that work in a set of ideas with different perspectives and are located in different places. Moreover, people have differences related to their experience, personal prejudices and difficulties of adapting to changes that occur. Furthermore, availability of resources for the acquisition and training of the use of new technologies and a society that presents different challenges not only political and economic but also health such as the recent pandemic derived from COVID-19.

Despite any challenge or difficulty that may exist, organizations need to continue making more efficient use of their resources and processes that allow them to continue growing and generate resources that are beneficial to them. For this reason, decision-making is essential in critical moments that require rapid action and where resources such as information visualization facilitate processes and shorten response times.

To illustrate those mentioned above, we will describe some works where the authors have put their efforts to incorporate visualization in decision-making processes.

In the area of knowledge management, the objective is to carry out the transfer of knowledge between managers, and for this reason, knowledge visualization was created and to reduce or eliminate difficulties that arise in the process. For this (Burkhard, 2004) proposes a framework for knowledge visualization that consists of three perspectives. The knowledge type what, how, why, where, who. The recipient type, individual, group or organization. The type of visualization, which has yet to be developed but as a first step, uses the types of visualization used by architects sketch, diagram, image, object, interactive visualization as a base. However, the author considers that a taxonomy that suits the needs of knowledge visualization must be created since visual representations motivate people, present new perspectives, improve memory, support the learning process, focus attention and improve communication.

A clear example is mentioned in (Alonso, Herrera-Viedma, Cabrerizo, Porcel, & López-Herrera, 2007) to facilitate the consensus process in problem-solving in Group Decision Making where decision-makers do not communicate directly and are unaware of the selection of others. The creation of a visualization tool shows the consensus status through a diagram. A node represents each expert; the proximity between nodes suggests the similarity that exists in their selection. This representation, known as a consensus diagram, allows those involved to see the status of the consensus and facilitates its resolution.

In (Miettinen, 2014), it is considered that visualization helps decision-makers to generate different perceptions and analyze problems from different perspectives and be able to make a better comparison between the different alternatives. Therefore, its use in multiple criteria decision-making problems is vital for the visualization of alternatives, and they developed a survey of various visualization techniques to facilitate its implementation according to the characteristics of the problem that is presented. The survey classifies the visualization techniques in six groups commonly known, techniques using circles and polygons, icons, techniques based on hierarchical clustering, projection-based and others. This work was developed due to the lack of surveys available, highlighting the advantages and weaknesses of the reviewed visualization techniques. The decision-makers will select the visualization technique that better fits the analyzed problem.

In addition to the works mentioned above, there are others. However, we can identify a constant when using visualization as support for decision making. This constant refers to the development of an optimal visual representation that better supports decision-makers in the comparison of alternatives, knowledge transfer and a more efficient generation of processes. Therefore, the challenges presented in visualization are directly related to the design of a graphical interface that provides the necessary elements for excellent visualization of information. In this way, we will review existing visualization tools that are most used today.

3.1 Information Visualization Tools classification

To facilitate the related work, in this research we proposed a classification of tools given their predominant characteristics. The definition on each of the categories is provided as follows. These three categories are (1) software libraries or APIs, (2) custom software, and (3) commercial software. The classification of tools can be seen in Fig. 13 and the review of each tool on the appendix.

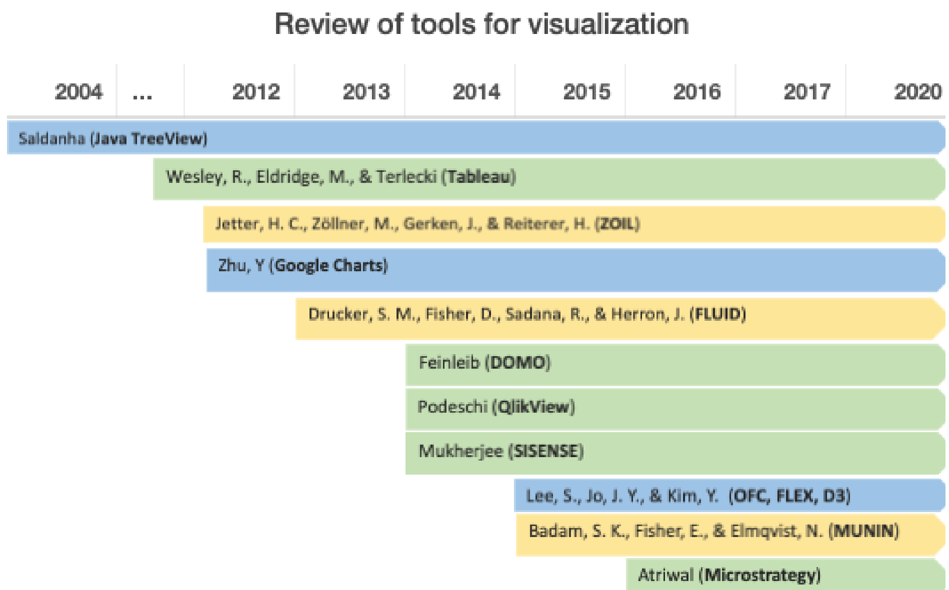

Fig. 13. Timeline showing the reviewed visualization tools. Blue, yellow and green shadings denote software library/APIs, custom software and commercial software, respectively.

Software Libraries/APIs

Software libraries and APIs are used to simplify the programming of an application. They provide functions, routines and methods that can be implemented in a particular software development for a specific behavior or functionality. The functionality provided covers common needs demanded for the development of software applications. They are not specific to a particular language and are available for use and implementations in a variety of computational languages. We present examples that can be implemented in different languages such as *Java, JavaScript* and others.

Custom software

In contrast with software libraries and APIs, custom software exists to cover particular and specific needs of a business. Following the software engineering cycle, developers gather with users to understand the specific needs of a business process and work in develop a solution that will help to fulfill the features needed. Custom software projects are as varied as the same diversity of business types, so we selected those that would stick the most to visualization solutions in our custom software review.

Commercial Software

Commercial Software also known as licensed software is the software that requires the user to pay a fee for using the full features it offers. Commercial software covers a specific market niche. Information visualization has become a trend among organizations through the use of different tools that seek to provide them with an ease of use, practical, and appealing way to visualize their information in order to maximize the organization's business value. According to the 2019 magic quadrant of Gartner (**https://www.gartner.com/document/3900992**, July, 1st 2019) there are 21 tools that provide this functionality, in this section we will analyze some of them from each of the 4 quadrants, niche players, visionaries, challengers and leaders.

Full review of the visualization tools can be found on Appendix 2. Following we will show a summary and analysis made on each of the categories.

Among the shared features in the libraries set of tools, we can find that most of them are multi-platform, free available for implementation, and with their implementation views and charts can be created. A great advantage of this libraries is that they can be integrated, to add functionality to other applications. This is translated in the simplification of development time, which in a regular software cycle it takes about 48%-50% of the total development time of an application (Navarrete-Corella, Noguez, & Molina, 2018). However, multiplatform feature is derived of the fact that with the exception of the Java TreeView library, the rest are libraries to be implemented in web application. Web applications themselves are multiplatform since they can be viewed in any browser existing in any platform with an Internet connection. Paradoxically, this feature makes it available in many platforms but also limits the accessibility since an internet connection is required. This limitation is a disadvantage found in the libraries set of tools. Other disadvantages found are that they are not identified as decision-making support tools, nor they have a multi-display capacity, and they don't provide a dynamic connection.

Table 2. Software libraries and APIs

LIBRARIES / APIS							
	PLATFORM	DATASOURCE	LICENCE	FEATURES	MULTIDISPLAY	DYNAMICA DATA CONNECTION	DM SUPPORT
Java Tree View	Multi-platform	Multiple	GPL	Views	No	No	No
Google Charts	Multi-platform	Web Services, CSV Files	Free	Charts	No	No	No
OFC	Multi-platform	Not Available	Free	Charts	No	No	No
FLEX	Multi-platform	Not Available	Free	Charts, Data integration	No	No	No
D3	Multi-platform	Files, requests	Free	Data Analysis, charts	No	No	No

In the Custom software, we found four different solutions. Each of them was created to cover a specific need or fulfill a specific target. They shared the multi-display feature which means that data can be display in multiple devices simultaneously, promoting collaboration among the people using these interfaces although only the DC-Tec tool was specifically created to support decision-making and also is the only one providing dynamic data connection. These tools' goal is specifically to facilitate the creation of graphical objects and make easier and more flexible the information visualization. Only DC-Tec tool is available also in normal browsers, Zoil and Munin, require more than one device connected. FLUID was created as prototype exclusively for mobile devices. All of the tools are available free but as they were created with a specific purpose, they cannot be used in other business areas.

Table 3. Custom software

Custom Software							
	PLATFORM	DATASOURCE	LICENCE	FEATURES	MULTIDISPLAY	DYNAMICA DATA CONNECTION	DM SUPPORT
Zoil	Multi-platform	Multiple	Free	Creation of graphical elements, data analysis	Yes	No	No
FLUID	Mobile	Not Available	Free	Charts	No	No	No
Munin	Multi-platform	Multiple	Free	Ubiquitous analytics and visualization	Yes	No	No
DC-Tec	Multi-platform	Google Sheets	Free	Real time data visualization	Yes	Yes	Yes

Commercial Software are the tools that offer a variety of features applicable to any business area. Multiplatform available, getting data from multiple data sources and with a wide set of charts and graphs are the most mature tools that we could found in our analysis. As being applicable in any business area within an organization, these tools not only provide graphic elements to visualize information, but they also provide a decision-making support. They are available for free in a trial version and to have a full license access they count with different payment plans. Different versions are available, desktop, on-line, on the cloud, making all the information available among the team member within a team in an organization. They also provide different levels of usability for different users such as IT experts, marketing, finances, communication and even high management members of an organization. We consider these are the more mature tools, since they not only visualization tools but also analytic tools and given their features, they are better known as business intelligence tools. Despite being multiplatform and count with a mobile version, neither of these have a multi-display functionality nor provide a dynamic data connection.

Table 4. Commercial Software

Commercial Software							
	PLATFORM	DATASOURCE	LICENCE	DEPLOYMENT	MULTIDISPLAY	DYNAMICA DATA CONNECTION	DM SUPPORT
QlikView	Multi-platform	Multiple	Free trial, Monthly/ one -time payment, quota based	Cloud, On Premise, Open API	No	No	Yes
Microstrategy	Multi-platform	Multiple	Free trial, quota based	Cloud, On Premise	No	No	Yes
Domo	Multi-platform	Multiple	Free trial, Standard, Profession al, Enterprise	Cloud, On Premise, Open API	No	No	Yes
Tableau	Multi-platform	Multiple	Free trial, one-time/ annual payment, quota based	Cloud, On Premise	No	No	Yes
Sisense	Multi-platform	Multiple	Free trial, Annual payment, quota based	Cloud, On Premise	No	No	Yes

In summary, we analyzed different types of tools with different specifications and qualities among the features seen as advantages found we can mention:

1. Multiplatform
2. Multiple data sources

3. Wide variety of charts and graphs

Over the biggest disadvantages found for the majority, we can mention:

1. No dynamic data connection
2. No multi-display functionality
3. Specific requirements needed, for instance Internet connection, more than one device connected, specific business, high prices.
4. Few of them consider the support to decision-making, since they have a more general application field.

Considering the advantages found on the reviewed tools, multiplatform, multiple data sources, variety of charts and graphs, and adding the features found as disadvantages, dynamic data connection, multi-display functionality, different business areas, on/off-line, a powerful tool can be created to support the decision-making process.

3.2 Metrics

Visualizations follow a simple rule, use a simple way to show more information. Nevertheless, it remains a challenge with all involved variables (Behrisch, et al., 2018).
Often human perception measures the quality of visualization. However, the use of quality measures is used instead to make a more precise evaluation (Sedlmair & Aupetit, 2015).
In an effort to formalize the process for measuring the quality of visualizations, different authors have proposed diverse ways of doing it. From metrics, guidelines, heuristic evaluations, literature reviews, Etc., the various works are countless. For this reason, we will mention some of the most relevant in recent years (Diehl, et al., 2018).
In this way, we can find several proposals of significant relevance, such as the nested model (Munzner T. , 2009). This work proposes to be based on a four-level model to carry out the design of the visualization. In addition to a guide for visualization design, it proposes different evaluation methodologies and a series of recommendations that apply to each of the levels to guarantee that the visualization is adequate. The first level, problem domain, is used to understand the domain in which one is working and the related

activities. For evaluation or validation at this level, as the author mentions, it is suggested to conduct interviews or record observations of the target audience. At the second level, the operations or data types related to the problem domain are identified to make their correct representation in the visualization. At this stage, it is recommended that problem domain personnel test the system to ensure that the operations and data types accurately represent the problem domain. At the third level, visual encodings and visual interactions are defined. At the third level, the use of guides based on perception and cognitive principles is recommended, as well as carrying out heuristic evaluations and reviews made by experts. In the fourth stage, the algorithm that will be used for the automatic creation of visual encodings and interaction is developed. The fourth stage validation suggests verifying that the algorithm is efficient and avoid the consumption of computational resources. It also suggests doing tests presenting paper impressions of the visualizations to the users to ensure that the objectives set as part of the evaluation at this level are met. To complement the model, the author suggests the following three recommendations. The first recommendation is to make a clear distinction between the four levels of the model, to identify which of them is the innovative contribution and facilitate the development of future work. The second recommendation is made for the case where the model is not fully developed, and the author focuses only on some levels. If this is the case, the author recommends making assumptions about the missing levels and documenting them to help readers understand the proposal made. The last recommendation and which is considered the most important is the first level, which is the identification of the problem domain. Thus, through this four-level proposal, with clear objectives and evaluation suggestions at each level, plus the three recommendations made, the nested model suggests carrying out a visualization design that meets the established purposes.

This heuristic evaluation proposal (Forsell & Johansson, 2010) establishes ten rules adapted to the information visualization area, based on previous work in the HCI area. Since most of the existing heuristic evaluations are focused on the HCI area, they are not adequately adapted to the specific needs of the information visualization area. Some of these needs are understanding the information for its proper use in visualization, and interaction required to generate different possible perceptions to help users in their analysis. A study was carried out in which participants verified the degree to which rules from six sets of heuristic evaluations cover previously identified usability problems in the information visualization area to find the essential heuristic rules. From a full set of 63 rules, a subset of 10 heuristic rules was obtained, which were identified as those that support the previously mentioned problems to a greater extent. These ten rules are information encoding refers to the objects that visually represent the information. Minimum actions refer to the actions necessary to carry out a task or fulfil an objective. Flexibility, this rule considers different ways to carry out a task. Guidance and help refer to actions that allow control of details and changes. Spatial organization refers to the workspace and the distribution of objects that occupy it efficiently. Consistency refers to consistency in the chosen design. Prompting, a suggestion of alternatives derived from a series of possible actions in a given context. To eliminate the strange is to do without unnecessary elements. Data set reduction, having characteristics that allow reducing the data set. This list of heuristic rules is a specialized alternative for information display evaluation with potential to be expanded and improved in future work.

Empirical evaluation is one of the most common in the visualization area, where it seeks to evaluate human performance, user experience and quality in the execution of algorithms (Isenberg, Isenberg, Chen, Sedlmair, & Möler, 2013). (Lam, Bertini, Isenberg, Plaisant, & Carpendale, 2012), have examined various existing works and has formalized them by classifying seven scenarios, which has been supplemented by the author, adding one more scenario, and referring to the scenarios as codes. These scenarios or codes are divided into two groups, understanding the visualizations, within which are User Performance (UP), User Experience (UE) and Algorithm Performance (AP). The second group is the understanding of data analysis processes, and this includes understanding environments and working practices (UWP), Visual Data Analysis and Reasoning (VDAR), Evaluating Communication Trough Evaluation (CTV), Evaluating Collaborative Data Analysis (CDA), Qualitative Result Inspection (QRI). These two groups generally represent the most widely used types of assessments in this area. The scenarios identified as the most used are QRI with a percentage of 46% and AP with a percentage of 35% from a total of 581 works analyzed and classified according to the defined codes.

Within the analysis carried out, it was observed that something in common among the reviewed works was the lack of specification or definition of the use of a methodology when carrying out the evaluation, reducing the relevance of the work. To avoid losing impact, the author recommended following a simple methodology by specifying details of the participants, details of collaboration, study protocols, controlled experiments, inspection of qualitative results.

Visualization patterns (Elmqvist & Yi, 2015) is a proposal for visualization evaluation. Under the same assumption that most of the existing works in evaluation are from the HCI area, this work seeks to make a set of solutions that can be used in the different cases of visualization evaluation, and that can be used in different application areas, as necessary. The visualization patterns propose five sets of patterns which are exploration patterns, control patterns, generalization patterns, validation patterns and presentation patterns. The exploration patterns are used in a first evaluation and allow identifying activities, factors, and the basis that will be necessary and giving reliable validity to the evaluation. The exploration patterns are factor mining, trial mining, human black box, do-it-yourself and wizard of oz. The control patterns are used as a second evaluation to validate that the factors identified previously are relevant and present significant results. The control patterns are luck control, time/accuracy elimination, deadwood detector, pair analytics. The generalization patterns allow us to identify that the evaluated elements can be applied to diverse situations. The generalization patterns are complementary studies, complementary participants, expert review, paper baseline. Validation patterns allow us to confirm that the evaluation design has been done correctly and identify possible errors in the evaluation design. The validation patterns are pilot study, coding calibration, prototype, statistics verification. The presentation patterns are used to show the results of the evaluation design. The presentation patterns are once upon a time, case study, visualizing evaluation. The author suggests implementing two or more patterns simultaneously to strengthen, clarify, and present robust results.

In this way, (Munzner T. , 2008) the author proposes a framework based on knowledge tasks by using two groups of tasks that focus on making complex decisions, under uncertainty, known as Rationale Based Tasks and on learning the application domain, known as Worldview Tasks. The rationale-based tasks focus on exposing the uncertainty in the data, as well as its effects on the results obtained; another task focuses on the creation of graphic elements that faithfully represent the relationships between the data and the last task focuses on being able to identify cause and effect relationships in the data set. The worldview tasks seek to find specific parameters of the problem domain, under the consideration of multivariate correlations, as well as the verification of hypotheses made through simulation.

Measuring the impact of the incorporation of uncertainty as a critical element for decision-making is the specific objective of this work (Deitrick & Edsall, 2006). In a case of policies for water use, specifically in the analysis of maps that show the water consumption of a specific population, elements are incorporated that graphically inform the interested parties of the uncertainty present in the information. The intention of incorporating uncertainty as a factor that is emphasized visually is to support the decision-maker during the analysis and improve the result by making conclusions, without these being correct or incorrect, but facilitating the final resolutions.

The visualization of uncertainty is vital in the decision-making area, making the information visualization process realistic and attached to the case study that is presented. However, to speak of specialized applications, that is to say in a particular field, we will cite the work of (Savikhin, Maciejewski, & Ebert, 2008) for its application in the economics area. In this work, in addition to recognizing the usefulness of the information visualization, the authors' objective is to highlight the interactive information visualization because through the study carried out it is demonstrated that the user can assimilate the information, make conclusions and obtain feedback efficiently. The study proposes to carry out three tests with users, using three types of information visualization. The first case is to present the information in tabular form. The second case is done by presenting the information with a simple visual representation through a static graphic in two dimensions. In the third case, the graphical representation in two dimensions is used in interactive mode and where the user can modify information values , and it is possible to see how the data is affected through these changes. For the three types of visualization, the same problem of bidding in an auction was worked. The results showed that the information presented in tabular form is easy to see but requires a mental effort to be able to carry out analysis regarding the information it contains.

In contrast, simple visual information, such as static images, is more comfortable to assimilate and makes it easier to draw conclusions compared to tabular information. However, the study showed that interactive visual analysis also allows obtaining immediate feedback, facilitating understanding and transformation of knowledge, facilitating decision-making. The case used is typical in the area of economics in which the user is carried away by what the naked eye can see, commonly incurring errors and bad decisions. However, as the study shows in its results, decision-making was significantly improved with the use of interactive visualization, followed by simple and tabular visualization. In this way, this work shows how decision-making is improved in a specialty area.

3.3 Multi-screen visualization

Multi-screen visualization environments can be described as spaces dedicated to specialized activities, equipped with hardware and software technology that allows viewing on multiple screens simultaneously, promoting the integration, participation and collaboration of several participants to facilitate decision-making. In this way, spaces have been developed that are known by various names such as decision theatre, decision center, decision laboratory, war room, operations center, and the cave that have a common purpose although diverse in physical and technological configuration.

In 1970 the Our Lady of The Lake University in San Antonio created the first Decision Theater and was used mainly as a learning space in management and as a research tool in decision-making. Later it was named the Business Decision Theater after a technological update in 2006 (Boukherroub, D'amours, & Rönnqvist, 2018). In the next figure, Fig.14, there is a diagram of the space (Ben, 1986).

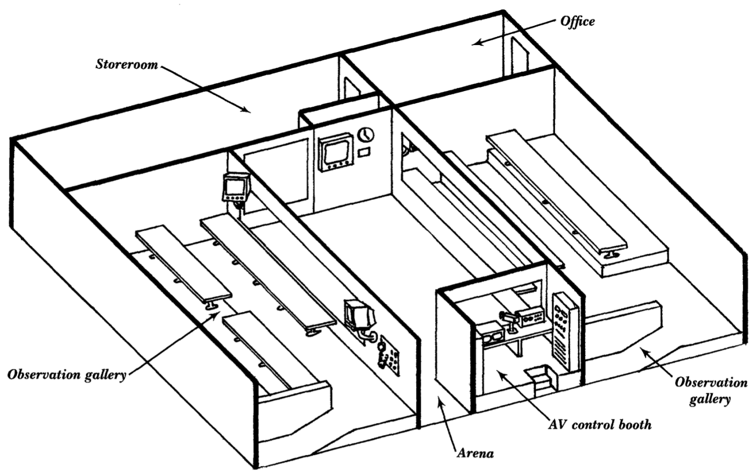

Fig. 14 Decision Theatre physical configuration (Ben, 1986).

With the name of Decision Theatre there are more that have been developed in the last years. Arizona State University Decision Theatre in Tempe, Arizona, another in Washington DC by the McCain Institute with such characteristics, a room with seven semicircular arranged screens equipped with an audio and video system used for displaying models, panoramic graphics and 3D video. Similar DTs have been developed in different cities around the world Tecnológico de Monterrey in Mexico (Balderas, Molina Espinosa, & Ruiz Loza, 2020), and more recently Huazhong University of Science and Technology in China.

In Australia, FOCAL, Future Operations Center Analysis Laboratory, was developed mainly for military use and training at Australia's Defense Science and Technology Organization (DSTO). The goal of FOCAL is to increase collaboration between military teams. Through the use of simulation technologies, virtual reality and 3D animation in real-time. It also incorporates other technologies such as voice recognition, natural language, gesture recognition, and gaze tracking to improve interaction between users (Wark, y otros, 2005). An image of FOCAL is shown in Fig. 15. FOCAL was developed with a must-agent architecture that, through a distributed model, facilitates interaction, information retrieval, information processing and display. The architecture diagram is shown in Fig. 16.

Fig. 15 The Future Operations Centre Analysis Laboratory (FOCAL) at Australia's Defense Science and Technology Organization (DSTO) (Wark, y otros, 2005).

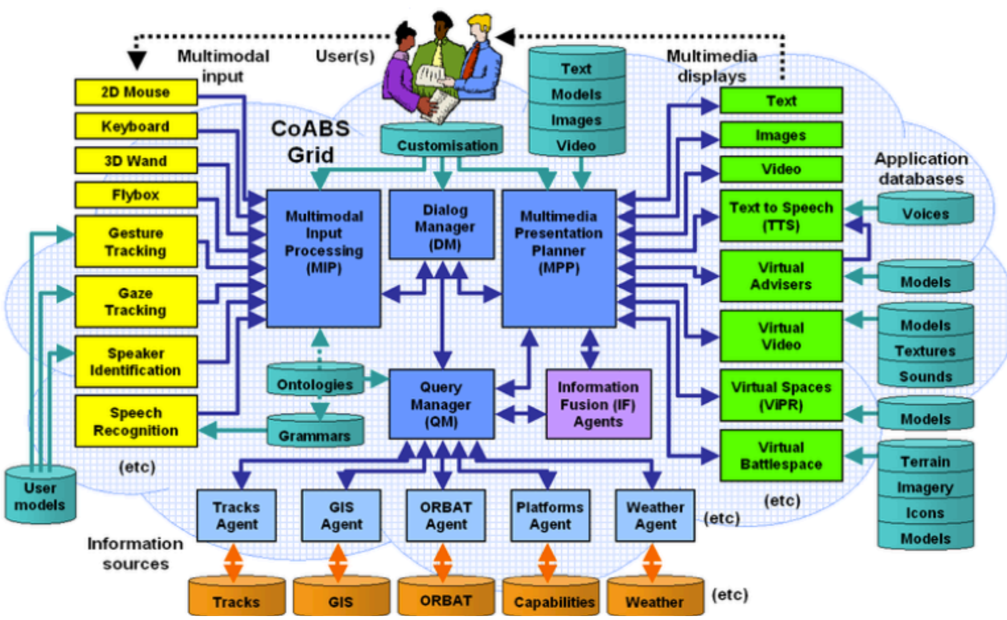

Fig. 16 Multi-Agent Architecture (Wark, y otros, 2005).

At the Swedish National Defense College, a futuristic concept was developed, the “Mobile, Joint Operational Command And Control Function for the year 2010”, ROLF 2010. Derived from its first two versions, ROLF Mark I, developed in 1997, and ROLF Mark II. The latter consisted of a group of four ceiling-mounted projectors that projected onto the table below, equipped with a touch screen, where interested people would gather. The general concept of the project was to improve the interaction between the members of a team. The objective was to improve the understanding and comprehension of the situations exposed through sensemaking. The ROLF Mark II image is shown in Fig.17. Regarding ROLF 2010, an image of the prototype can be seen in Fig.18 where three components stand out: the equipment, the seats and the technology.

Regarding the use of technology, they mention there must be 3D screens that will be part of the visual system in the center of the room and a system of screens located on the walls of the room where additional information will be projected to enrich the analysis carried out. Finally, the incorporation of a Decision Support System is necessary to handle complex situations in decision-making (Brehmer, 2007).

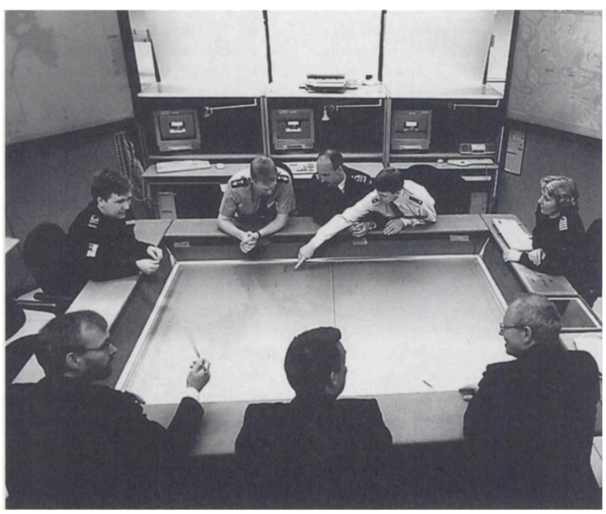

Fig. 17 ROLF Mark II (Brehmer, 2007).

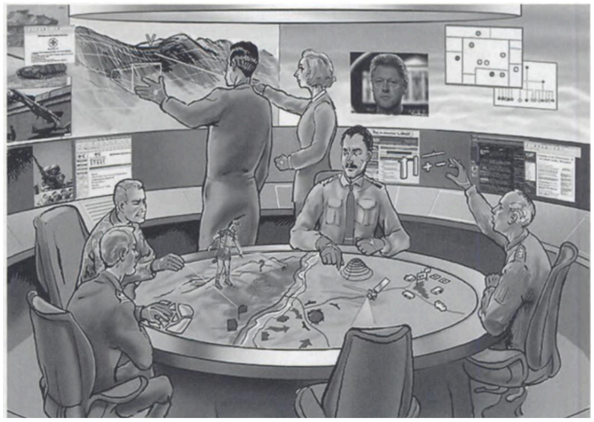

Fig. 18 Concept of ROLF 2010 (Brehmer, 2007).

A variation of a multi-screen visualization is the Visualization Dome. A prototype of the Visualization Dome was developed in the city of Lublin, Poland. The objective is to facilitate community communication and acceptance in the development of renewable energy sources (RES). The Visualization Dome has a diameter of 9.5 meters with a 360º projection. It is equipped with eight projectors and a surround audio system. Through an immersive experience, participants can interact through a touchscreen, in conjunction with the EnerPol software that includes more than 200 databases with anthropological, geographic, climatic information, etc., which is used for the modelling and display of information (Gawlikowska, Marini, Chokani, & Abhari, 2018). An image of the interior and the exterior of the Visualization Dome is shown in Fig. 19.

Fig. 19 Visualization Dome prototype exterior and interior (Gawlikowska, Marini, Chokani, & Abhari, 2018).

These are some examples of multi-screen display projects that have been developed in physical space. However, conceptual proposals have also been made, such as the generic framework for Decision Theaters for participatory planning as part of the decision-making process. This proposal considers five elements as essential in a Decision Theater. These are decision entities, decision support component, organizational system, technologies, and decision theater layout. Decision entities refer to the decision-makers or stakeholders involved in the process. The decision support component refers to the tools and experts that facilitate the generation of knowledge and insights during the decision-making process. The organizational system refers to the team that supports the DT such as facilitators, technicians and procedures for the functioning of the DT. The DT layout refers to the physical distribution of the room where the DT will be located and the configuration of the technological equipment that optimally allows the attendees' participation, interaction, and collaboration. Technologies are those devices or tools that support the other elements of the DT, and they can be Hardware or Software, for example, graphical user interfaces to carry out the visualization and interaction between the assistants and the data (Boukherroub, D'amours, & Rönnqvist, 2018). In Fig. 20 The proposed conceptual framework is shown.

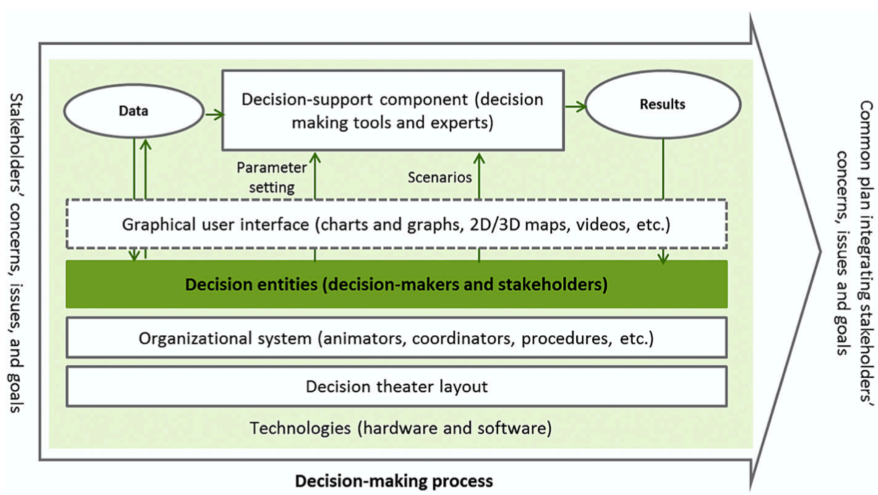

Fig. 20. Conceptual Framework of the Decision Theater for Participatory Planning (Boukherroub, D'amours, & Rönnqvist, 2018).

In this section we presented the related work, concerning traditional visualization tools, metrics used for the evaluation of visualization tools and work related to multi-display visualization. Next section is about our metrics solution proposal for the evaluation of visualization tools and our architecture proposal with Multi-Screen and Dynamic features.

4. Solution Proposal

Given the complexity of the decision-making process, it is necessary to implement visualization schemes to support the process. Nevertheless, to ensure the selection of the best visualization tool, we need its validation using metrics.

In the next section, we propose a set of metrics that we consider appropriate to give support and provide insights of information efficiently and accurately.

4.1 Metrics Solution proposal

We developed a measurement method based on the work of (Saket, 2016), (Nielsen, 2005) to perform a trustable evaluation of the products' quality. Specific criteria were used to evaluate if a particular product provides a specific feature that is needed. We propose the following criteria for evaluating aspects of visualization that are described below.

Proposed evaluation criteria:

1. Visual scalability: expressiveness
2. Visual scalability: effectiveness
3. Aesthetics
4. Dynamics

4.1.1 Definitions of evaluation criteria

*Visual Scalability***.** It is defined as the ability of the tools to generate graphical representations derived from large data sets efficiently (Andrews, Endert, Yost, & North, 2011), (Bieh-Zimmert O. &., 2014). Related to this term, we find graphical scalability that mainly refers to the number of pixels used in the generation of visual encodings (Yost, 2007). Pixel oriented approaches have played a significant role when creating visual structures from high-dimensional data and different visual encodings are needed to represent each dimension (Liu, Maljovec, Wang, Bremer, & Pascucci, 2016). The number of pixels used can provide an improvement in the quality of the visual encodings. A better quality of encodings can be applied to a large higher-resolution displays with the capacity of displaying more data and improving the human perception of decision-makers (Bieh-Zimmert & Felden, 2014). However, not all can be better represented by increasing or decreasing the number of pixels used for their representation. It is crucial to bear in mind that the representation of information in a graphic manner is a way to help the user to facilitate their understanding and interpretation, mostly when the amount of information has grown considerably (Lu, Chen, Lai, Lin, & Yuan, Frontier of information visualization and visual analytics in 2016, 2017). The expressiveness measures the graphical representation of the adequate amount of information enough to provide insights and discoveries derived from the visualizations. An expressiveness value closest to 1, means the visual representation does not exceed the amount of information presented, and that does not lack any data related to the related information. In contrast, an expressiveness value closest to 0 means the visual representation presents a deficiently or incomplete amount of information. Generating the visual representation of the information through the optimal use of computational resources and straightforward interpretation by the user are other elements that allow us to measure effectiveness (Saket, 2016), with one being the optimal value indicating that both elements are in balance.

Aesthetics. Aesthetics concerns the appreciation of things as they affect human senses. Although it is essentially an area of study in philosophy (Carlson, 2002), in computational fields aesthetics is considered one of the four dimensions of the user experience (UX) (Merčun T. , 2014). In visual representations of information, aesthetics is a crucial element to make the user feel interested and attracted to look into the graphics representing information (Luo, et al., 2019). Aesthetics can be uniform or non-uniform. Non-uniform aesthetics are easier to stay longer in the users' mind for their uniqueness characteristic against the uniform ones that being similar the user forgets quickly (Borkin, et al., 2013). To acknowledge the importance of an element we have to know the value it adds, so when a user spends considerable time interacting and revising a visualization, we already know that the purpose of the visualization is being reached. A decision-maker will spend time interacting with visualizations when these are easier to see and analyze because it will reduce the cognitive load. Lasting impressions in users formed after interacting with the visualization tell us about its value (Harrison, Reinecke, & Chang, 2015).

*Dynamics***.** Dynamics refers to information changing through time, so the structures used to represent it should not remain static, and most importantly it should provide the functionality to test hypothesis and projections beyond what we can see in present time. Dynamics is an undergoing challenge, as mentioned in (Lu, Chen, Lai, Lin, & Yuan, Frontier of information visualization and visual analytics in 2016, 2017). This concept is strongly related to interaction, as it is the way in how users explore data. The taxonomy of interactive dynamics for visual analysis defines three high-level categories:
Data and view specification, where the user can visualize, filter, sort, or derive information from the original data.
View manipulation, which permits to select, navigate, coordinate, and organize different elements within the visualization.
Process and provenance, where the user can make annotations and be able to share the reviews made for collaboration (Heer & Shneiderman, 2012).

Science of interaction (Cook & Thomas, 2005) should provide a taxonomy to support analytical reasoning with innovative techniques, from low-level to high-level interactions, that can adapt to different displays. Knowledge should be dynamic as part of the vision of the future of (Cook & Thomas, 2005), meaning knowledge can be modified or added as a result of the analysis. A system should detect changes or additions and show the effect. Since decision-makers have to consider different scenarios of presented data, its modification and update are essential to analyze these scenarios' various possible outcomes.
To complement the taxonomy mentioned above, we added data manipulation to write and update data while interacting with the graphical representations. We also considered the ability to record or save the changes in the data source. The proposed taxonomy is presented in Table 5.

Table 5. Dynamics taxonomy

Data & View Specification	Visualize Filter Sort Derive
View Manipulation	Select Navigate Coordinate Organize
Data Manipulation	Write Update Save
Process & Provenance	Record Annotate Share Guide

4.1.2 Evaluation Rubric

The use of rubrics provides a formalization for the evaluation of visualization tools. Along with criteria, engages the participants by giving a framework for data collection, analysis, and sense-making. It also facilitates the synthesis of data for reporting findings (King, McKegg, Oakden, & Wehipeihana, 2013).
The use of taxonomies and guidelines are standard evaluation techniques in visualization. However, since most of the outcomes give qualitative results, a rubric or evaluation framework is needed. Hence a rubric will open discussion and improve the design and functionality of the visualization tools (Gallagher, Hatch, & Munro, 2008).

Table 6 shows our proposed rubric with three levels of performance based on the definition of the evaluation criteria that we made before. The three performance levels were insufficiently scored with one point, average scored with two points and advanced scored with three points. The definitions for each of the levels and each of the criteria are also depicted in Table 6.

Table 6. Rubric for the evaluation of visualization aspects

Evaluation criteria / Support questions	1-Poor	2-Average	3-Advanced
Visual scalability: Expressiveness	The information presented is not enough, does not provide insights to the user. Expressiveness value closer to 0. The information presented is excessive, expressiveness value over 1.	The information presented is adequate but do not provide sufficient value for the analysis carried out. Expressiveness value of 0.5.	The information presented is adequate to the analysis that is being carried out. Expressiveness value closer to 1.
Visual scalability: Effectiveness	The presented information is hard to understand and interpret and the visual representation takes too long to load.	The presented information is hard to understand and interpret or the visual representation takes too long to load.	The presented information is easy to understand and interpret and the visual representation loads fastly.
Aesthetics	Limited variety of charts.	Visual representations can be created with variety of charts and colors.	Visual representations can be created with varied selection of charts and color. Organization of elements allowing the user easily to understand the content of the visualization.
Dynamics	Visual representations are not created or updated in real time.	Visual representations can be created and updated in real time.	Visual representations can be created and updated in real time. Users can modify data of visual representations and update the data source.

The criteria that we have selected for the evaluation of tools are considered essential for the effective visualization of information and the maximization of the benefits that it provides to the interested parties. Graphic representations of sufficient information for a straightforward interpretation of the participants in a short deployment time through the visual scalability feature. Wide variety of graphics, which can represent the data faithfully and are visually attractive through aesthetics. Visualization of content in real-time allows generating different alternatives for optimal decision-making through the dynamic feature. All the criteria together will guarantee that the information display effectively ensures that the results are optimal. Based on our metrics proposal, we created an architecture proposal. This architecture proposal will be described in section 4.2.

4.2 Multi-Screen & Dynamic Solution Proposal

In this section we will present our architecture proposal with Multi-Screen and Dynamic features. First, we will present the Evidence-Based Decision-Model and its modification with the integration of visualization. Second, we will present the architecture proposal and how this one matches the evidence-based decision-making model. In the last and third part we will present the UML diagrams of the proposed architecture.

4.2.1 Model

Following the principles of the evidence-based decision-making paradigm which its focus goes towards the support of complex decisions in management and recalling the assumptions:

- Evidence has a natural contingency,
- The implemented process, used to generate the evidence, should be replicable and transparent,
- Research consensus, among participants, gives higher accuracy of evidence,

The model proposed by (Baba V. V., 2012) and reviewed in section 2.4, considers a multi-level form by the particular levels corresponding to the individuals, organization and its interaction among them.

The complete model, process to gather evidence, considers preferences and values from managers and stakeholders, generation decision option, taking in consideration ethical constraints, is a dynamic process, that iterates between all the levels, several times until a final decision is made. During the iterations through the process, we consider the integration of visualization gives support when exploring the decision options. The visualization is integrated, as shown in figure 21.

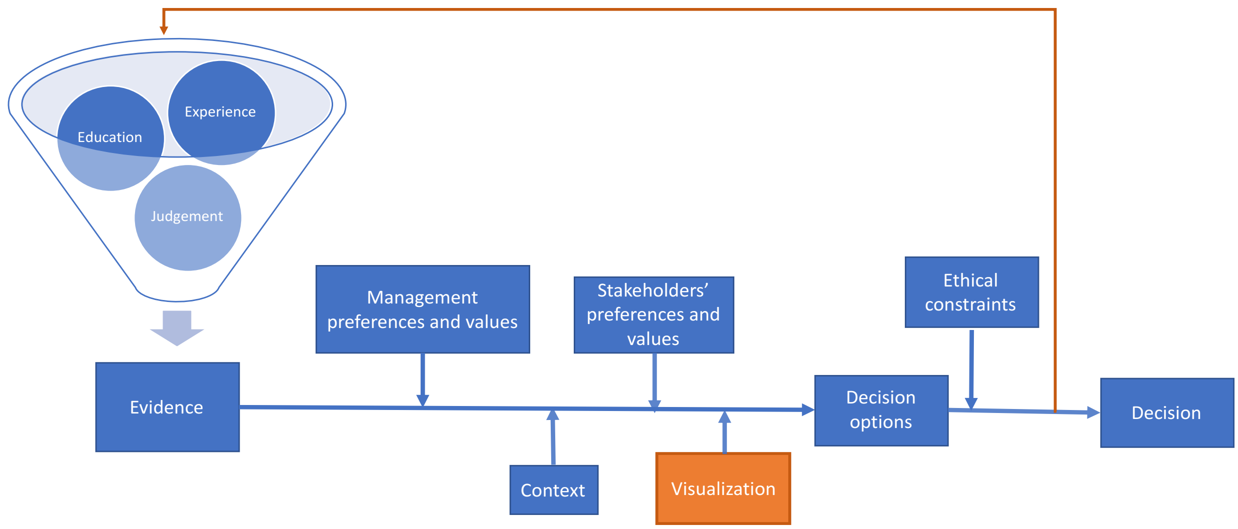

Fig. 21. Visualization integrated in the Evidence-based decision-making model

After considering the ethical constraints, the process goes back to the evidence, starting the complete process, and going through each step as many times as necessary, before making the final decision.

To make sure that the visualization as tool will give support to the evidence-based decision-making model, we followed our proposed set of metrics, from section 4.1, to measure and evaluate the features that will be necessary to fulfill this objective.

4.2.2 Architecture

Considering the opportunity areas found in the previous section, we make the proposal of an architecture to develop a visualization tool that will provide the features to fulfill the requirements and characteristics needed to give support to the decision-making process based on the Evidence-Based Decision-Making model.

The Multi-Screen & Dynamic visualization tool is a visualization tool designed to support the decision-making process in a decision-making center.
The features of this visualization tool are:

- Automated visualization scheme. The connection to a simulation model gives the facility of generating visualization scheme automatically. The visualization scheme will create different scenarios to provide the user better insight.

- Interpretation of the visualization model. Through the use of different charts, that are visually appealing and provide insights of the information. The charts show accurate and updated information coming from the visualization model.
- Dynamic visualization/interaction of information. The information displayed can be used for visual analysis, can be modified in real time and stays updated in every moment.
- Multi-screen visualization. The visualizations will be showed in a multi-screen format. The multi-screen format allows the simultaneous visualization in seven high-definition screens as part of the decision-making center structure and configuration. The seven screens can be used fully or partially, according to the user needs and the scenarios displayed.

An important feature of this tool is the connection to a simulation model. This simulation model has a predefined structure, and the connection will be made in an automatic way to read and load data. The tool will provide a dynamic interaction through the reading and data modification in real time, in such a way that the user can update the visualizations immediately with the facility of simultaneous visualization in the seven-screen high-definition format. The proposed architecture is a combination of two architectural patterns, the Server-Oriented Architecture Pattern (SOA) and the Model-View-ViewModel Pattern (reviewed in section 2.5.1 and 2.5.1 of Theoretical framework), as shown in figure 22.

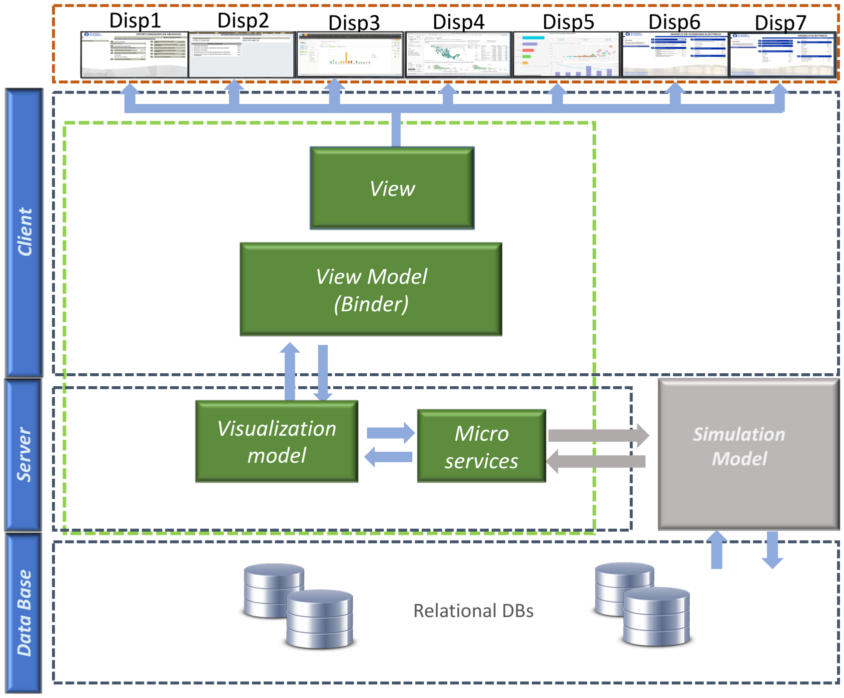

Fig. 22 Diagram of the proposed architecture using Service-Oriented Architecture and Model-View View Model patterns.

4.2.3 Model-View-ViewModel Component

This architecture design is integrated by the model component, a viewmodel component and a view component, that we will explain next. Its structure is show in Fig. 23.

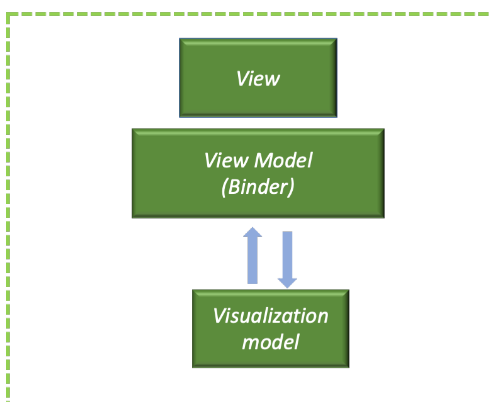

Fig. 23 Model View-Model View design architecture.

Model. This component will replicate or make a copy of the data model to be presented. A component allows the creation of Web services and networking tools using diverse programming languages and a collection of modules that handle various core functionalities. In this architecture, this component will handle the creation of the replica of the model provided by de Simulation Model component. The replica of the model will be mapped in a document file, that will be called at this moment, as part of the architecture, visualization model, and will be processed by other elements to be visually available.

ViewModel. This element will bind the model and has the responsibility for pushing changes to the server or any other component of the application. It will create objects, states and required transactions to manage the manipulation of the model and also to keep consistency and data updated and synchronized. Created objects will be ready to be shown.

View. Corresponding to the frontend, the interface in which the user sees finally the data, objects created by the ViewModel component will be used, processed and shown to the user. All data process will be available in an appealing look and feel view for the user. The view will provide interaction, in which users will be able to modify data, following model rules, and refresh the information to visualize changes made. Recall changes made will be handled by the View-Model.

4.2.4 Service-Oriented Architecture Components

The service-oriented architecture design where services are provided to the other parts by application components, through a communication protocol over a network and is integrated by three main structures, the database layer, the simulation model and the *Model-View-ViewModel* component. A diagram of this architecture can be seen in the Fig. 24.

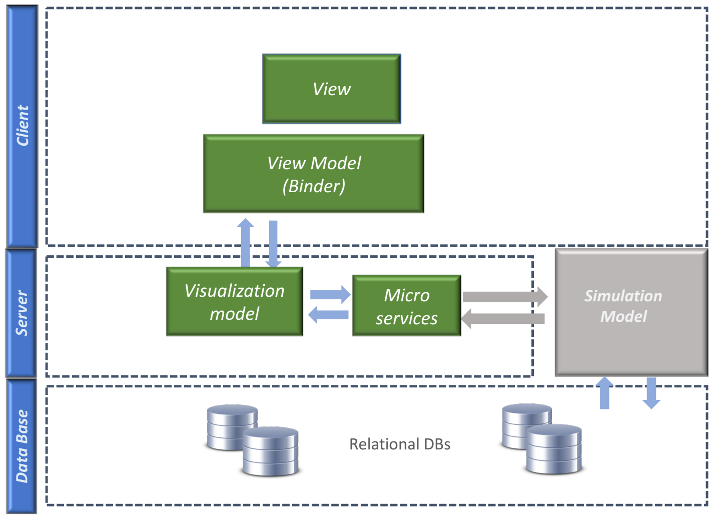

Fig. 24 Service Oriented Architecture Design.

Database component. This component will be any database to store the selected or filtered data, ready to be consumed and processed by the simulation model. Most common data base engines are relational, with some extra features as document store, graph DBMS, RDF store, these data bases are accessed by the structured query language (SQL) which is the standard for relational databases. Relational databases are common to use, however there are some other alternatives such as the NoSQL databases, cloud databases, object-oriented databases, which solve some of the issues of traditional databases and combined give a powerful functionality through a solid infrastructure.

Microservices. This component will run required processes to keep communication between the simulation model and the MVVM component.

Simulation Model. The simulation model will transform the data coming from the database, apply the business rules corresponding to the model, process it and generate the results or outputs.

4.2.5 Evidence-based decision-making model matching the proposed architecture

For the evidence-based decision-making model a diagram matching the proposed architecture is shown in Fig.25.

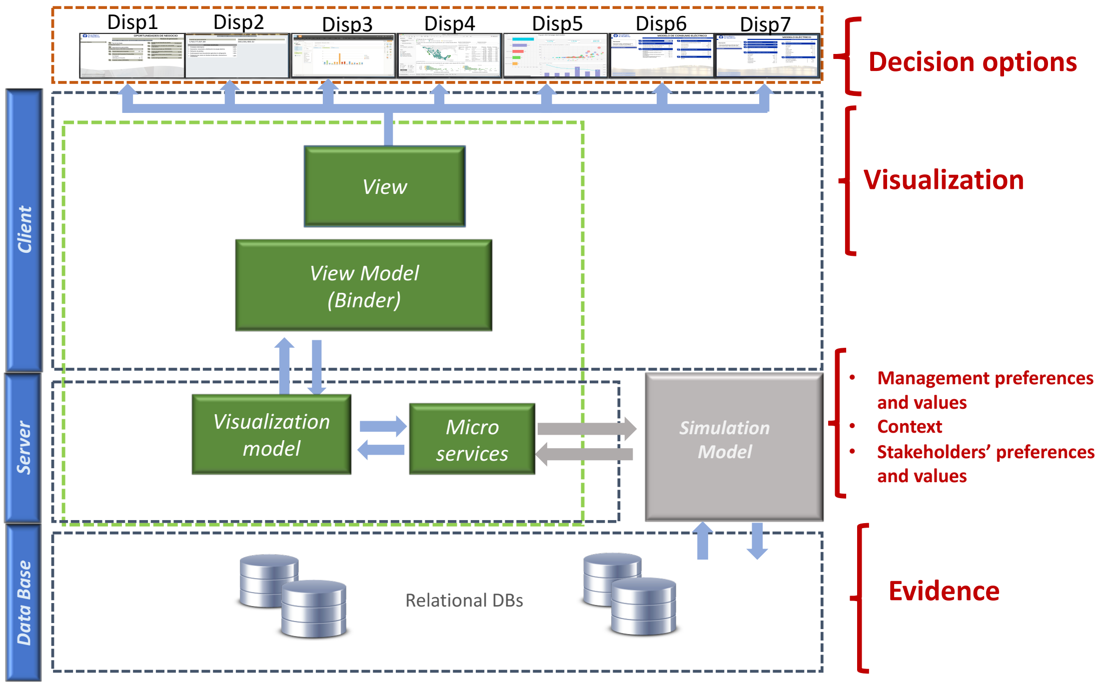

Fig. 25. Diagram of the proposed architecture matching the Evidence-Based Decision-Making Model.

Evidence. Data will be collected in data bases. This data corresponds to evidence of the model.

Management/Stakeholders preferences and values. These will be managed by the server side through the View-Model that will be in charge of synchronizing changes made with the simulation model and keeping everything updated.

Visualization. After synchronization is made visualization is prepared or generated.

Decision options. Decision options are shown to stakeholders in visual form in multi-screen format.

UML Diagrams

In this section, we present the UML diagrams to facilitate the view of a system based on our architecture proposal. We will present three diagrams, the component diagram, the activity diagram, and a use case diagram. These diagrams are structure and behavior diagrams that will help to have a better understanding of our proposal.

Activity Diagram

Another very important part of our proposal, which was not identified in the review of tools and work relationships, is the ease of updating the values of the variables of the visualization dynamically in order to facilitate the timely shot of decision making. The interaction was modeled through an activity diagram to incorporate the dynamic updating facility into the proposal. The activity diagram represents graphically activities and actions workflows of the system step by step. Fig. 26 shows the activity diagram of our architecture proposal.

A system based on our architecture proposal has three essential elements a chairman, a decision-maker and the system interface.

The system process is initiated by the chairman with a load model action, if the loaded model is correct then the system interface executes its corresponding actions.

Once a model is loaded the system interface will execute the show scenarios action. A decision-maker will choose a scenario to visualize it. If the decision-maker performs a modify variables action the system will confirm if there were changes in the variables values. If the system interface detects a change in the values, it will update them and reload the scenario. If there are no changes in the variable's values, it will only reload the scenario.

After the scenario is reloaded, the show scenarios actions are executed again by the interface in a cyclic process that will keep going when continuing. The process will stop when the continue option is no, then the scenario will be saved with all changes made.

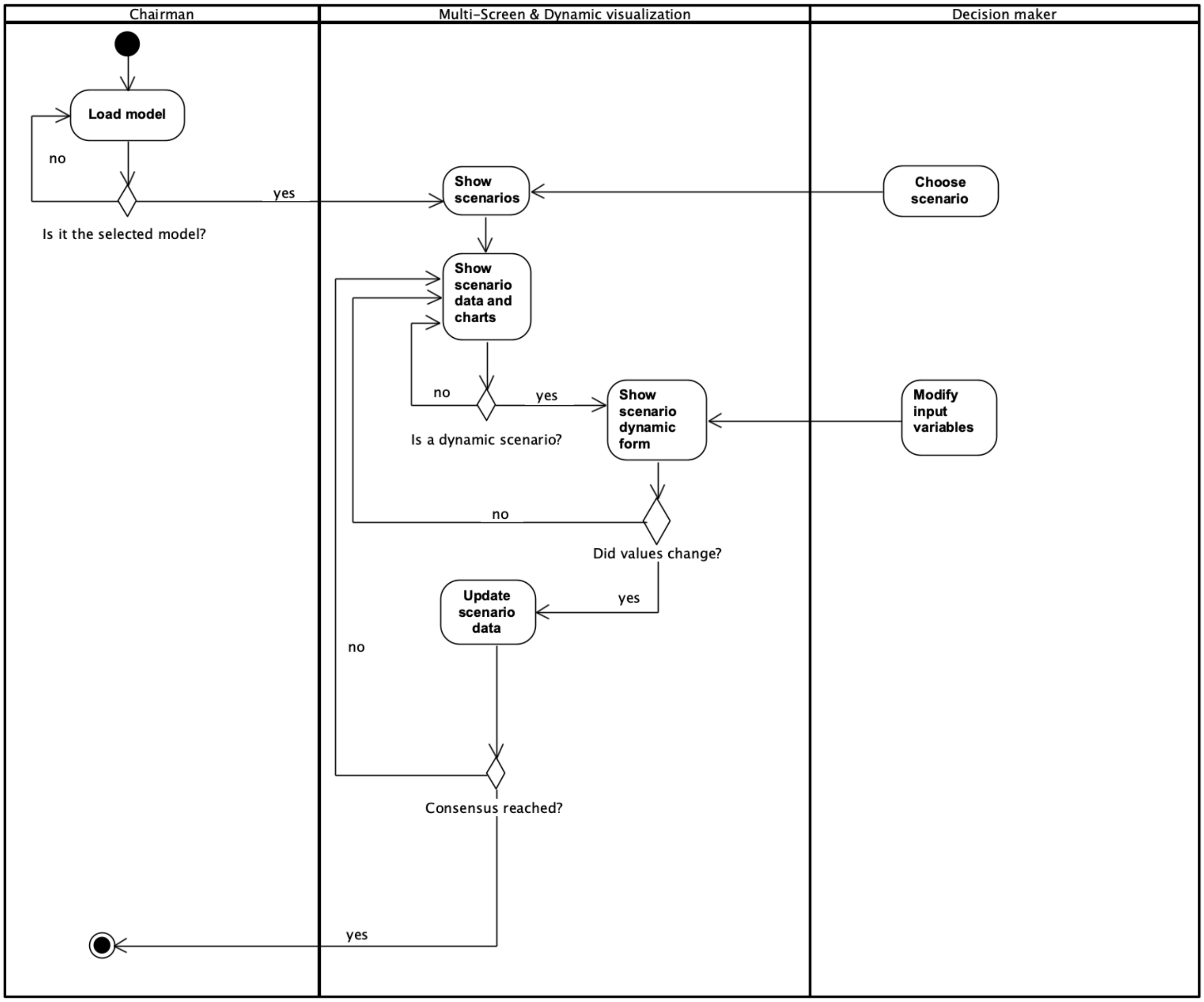

Fig. 26 Activity Diagram of our architecture proposal.

Use Cases

The use case diagram is used to represent the interaction of the user with the system based on our architecture proposal. Fig. 27 shows the use case diagram.

We have two actors, the decision-maker and the chairman. Both of them interacting with the visualization system.

The decision-maker can select a scenario, change scenario and modify variables within a scenario.

The chairman can load a model and save changes made in the scenarios.

The visualization system will execute the activities selected by the decision-maker and the chairman. And will update the model, when there exist changes, after it will reload the scenario.

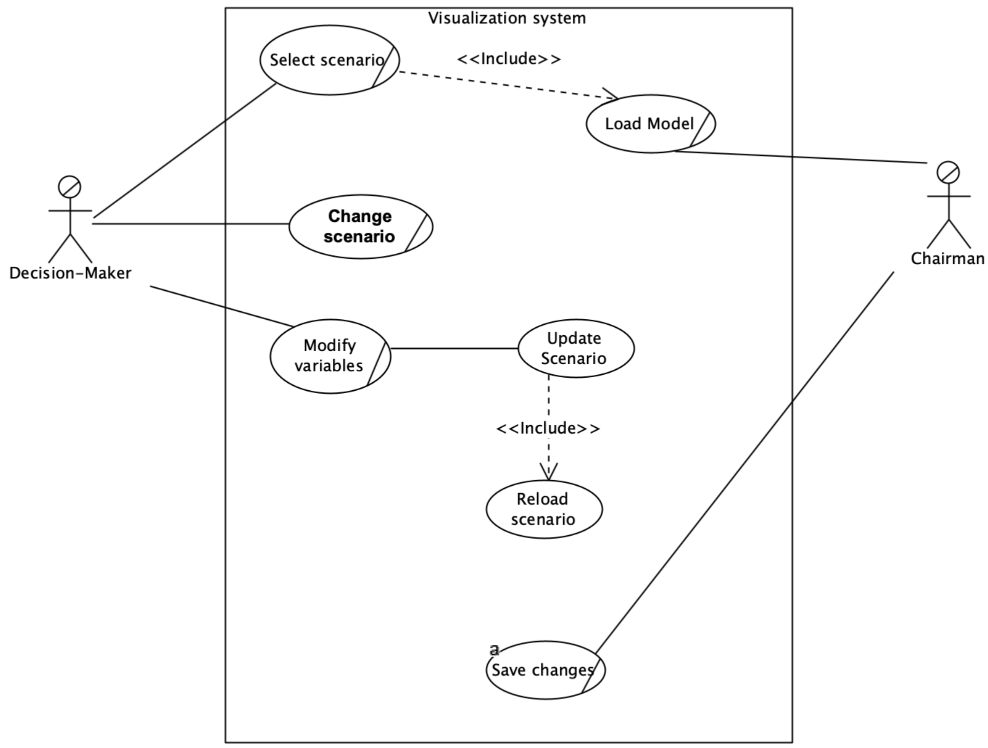

Fig. 27 Use case diagram of our architecture proposal.

Component diagram

This diagram shows each of the components that integrate the basic structure of the system based on our architecture proposal. This diagram is a roadmap to the implementation of our proposal. Fig 28. Shows the component diagram.

Authentication. This component is required so the decision-maker authenticates its identity and starts a session to work on the system.

Scenario selection. After a session is started, the decision-maker can select a predefined scenario or work on a dynamic one. Predefined scenarios are not modifiable, meanwhile the dynamic scenario does allow modify variables and update the visualization.

Visualization. Shows the charts according to the selected scenario. If the scenario is automatic, it refreshes every period of time to show the new chart.

Edit Values. This component will allow the decision-maker to modify values in the dynamic scenario.

Model. It contains the model to be analyzed. All scenarios and related data are loaded from the model.

Temp DB. Stores a copy of the model. Information will be loaded from this component.

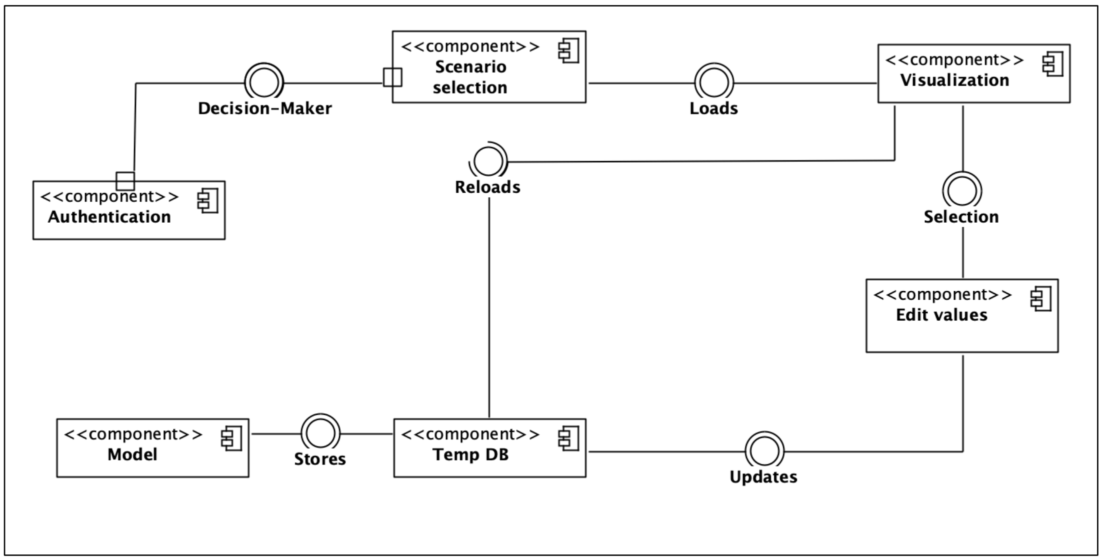

Fig. 28. Component diagram of a system based on our architecture proposal.

In this section we introduced our proposal solutions the metrics and the multi-screen & dynamic. In the next section we included the test processes of each of the proposals, the metric test process and the multi-screen & dynamic test process.

5. Test Process

5.1 Metric Test Process

This descriptive user study compared the visual scalability, aesthetics, and dynamics of the visualization tools reviewed in section 3.1. Recalling the proposed classification: (1) software libraries or APIs, (2) custom software, and (3) commercial software. Software libraries and APIs are often used to simplify the programming of an application. They provide functions, routines and methods that can be implemented in particular software development for a specific behavior or functionality. Custom software is used to fulfill the specific needs of a business. Developers gather with the user to understand the specific needs of a business process and develop a solution that will help fulfill the features needed. Commercial Software, also known as licensed software, is the software that requires the user to pay a fee for using the full features it offers. Commercial software covers a specific market niche. The evaluation was performed by a group of experts with two different profiles, software engineer and business analysts, the specific characteristic of each profile is described in the participants' section. The purpose of this study seeks to answer the research question: *What are the features needed and how can we measure them in order to know if they provide an effective visualization tool?*

5.1.1 Participants

Fourteen experts in the computational field, with expertise in two different areas, software engineering and business analysis carried out the evaluation. The criteria for recruitment were that a bachelor's degree in computer science was required for the software engineer profile. Additionally, for the software engineer profile, the participants should have five or more years of experience in front-end development with at least one programming language for web development such as Java, C#, React or Angular. It was preferred if the expert also had knowledge of Html and JavaScript, and integration with APIs. For the business analyst profile, the recruitment criteria required a bachelor's degree in computer science, information technology or similar and knowledge of the organization's business processes. It was also required five years of experience in business intelligence, analytics, and database technologies such as SQL.

A group of experts with the software engineer profile was required and split in two. First half was asked to evaluate software libraries/APIs category. Second half was asked to evaluate the custom software category. For the commercial software, experts with business analyst profile performed the evaluation. Table 7 shows the distribution of participants on each group of tools.

Table 7. Experts' profile, categories of tools and participants on the evaluation experiment. NA, Not Available, means there were no experts performing evaluation on the indicated tool category.

Expert profile	*Software libraries/APIs*	*Custom Software*	*Commercial Software*
Software Engineer	A total of four experts performed the evaluation.	A total of five experts performed the evaluation.	NA
Business analyst	NA	NA	A total of six experts performed the evaluation.

The four experts in software libraries had, on average, ten to fifteen years of experience in the industry. Their expertise was in software development. Their core professional activities are related to processes, requirements, design, development, testing, debugging, deployment and maintenance of front-end software applications to translate data into graphical interface through the use of technology such as Node.js, React.js, Redux, React Native, Angular, Asp.net, C# and Java. In addition to their software development skills, they also have a broad experience in database management systems by creating, managing and performing CRUD (Create, Read, Update and Delete) operations for consulting and manipulating data. Their bachelor's degree was in computer science, and some of them also had a master's degree in a computer science area. Some others had professional certificates in Advanced Angular, PSP software developer and Sun Certified Java Developer.

The five experts in custom software profile were academic with Computer Science PhD degrees and over 20 years of experience in teaching classes in high-education, and post-graduate programs in computer science area. Their research experience included developing projects in areas such as Data Science, Intelligent Systems, Databases, Mobile and distributed systems applied to some other fields such as medicine, education and innovation. Their work also included participation in MOOCs' development in Coursera and creation and administration of a decision-making laboratory including its infrastructure, interconnection, and the visualization schemes to support the decision-making process.

The four commercial software experts had worked for more than twenty years in the management, development and execution of software to support the business processes in different areas inside companies. Their expertise included strong skills in business processes, ITIL (Information Technology Infrastructure Library), Balanced Scorecard, Business Intelligence and Information Technology Service Management. This group of experts had a bachelor's degree in information technology and computer sciences. Two experts from the group had a master's degree, have worked in the industry and the academic field, teaching to high-education students in software management and business classes.

5.1.2 Procedure

For the evaluation process, the participants were provided with a document that contained the theoretical framework related to the assigned category, the criteria and metric definition, and a simplified description of the tools to be evaluated. The metric was supported with two questions per criteria to be answered by experts and facilitate their evaluation process. Additional to background information, the document contained an evaluation table in which the participants would provide, according to their expertise, the scores to each tool and per criteria (1-Poor, 2-Average, 3-Advanced). The evaluation table had extra space for the participants to provide observations, as required. The evaluation document was sent via email to participants, and they were asked to read it, perform it and return it thought the same medium.

5.1.3 Data collection method

The data collection method was performed using a rubric, which was defined in the chapter 4. Evaluators were also provided with a pair of questions per evaluation criteria to support their evaluation process. The rubric with the added questions is shown in Table 8.

Table 8. Evaluation Rubric and added questions related to evaluation criteria for evaluation aspects.

Evaluation criteria / Support questions	1-Poor	2-Average	3-Advanced
Visual scalability: Expressiveness *Can the information presented be viewed in its entirety?* *Does the information shown provide answers to the business questions?*	The information presented is not enough, does not provide insights to the user. Expressiveness value closer to 0. The information presented is excessive, expressiveness value over 1.	The information presented is adequate but do not provide sufficient value for the analysis carried out. Expressiveness value of 0.5.	The information presented is adequate to the analysis that is being carried out. Expressiveness value closer to 1.
Visual scalability: Effectiveness *Do the visual representations provide an easy way to interpret the information?* *Do the visual representations load easily and fast?*	The presented information is hard to understand and interpret and the visual representation takes too long to load.	The presented information is hard to understand and interpret or the visual representation takes too long to load.	The presented information is easy to understand and interpret and the visual representation loads rapidly.
Aesthetics *Do the graphs available fit the information presented?* *Can the analysis be made in collaboration and shared with other users for collaboration?*	Limited variety of charts.	Visual representations can be created with variety of charts and colors.	Visual representations can be created with varied selection of charts and color. Organization of elements allowing the user easily to understand the content of the visualization.

Dynamics *Can the user edit the data presented?* *Can new insights be generated from the information presented?*	Visual representations are not created or updated in real time.	Visual representations can be created and updated in real time.	Visual representations can be created and updated in real time. Users can modify data of visual representations and update the data source.

To validate the metric definition, we conducted a pilot test where one representative participant per each category was asked to complete the tool's assessment. Additionally, they were asked to provide feedback related to the evaluation criteria. Experts were asked with three questions and provided the value that better fit their opinion, following a 1 to 5 scale, being one the less value and five the maximum. The 1 to 5 scale provided was used to measure the usefulness and clarity of the criteria used for the evaluation. Additionally, they were provided with space to write additional observations and information regarding additional criteria and tools. In Tables 9, 10 and 11, we present the three questionnaires for each category with the answers given.

Table 9. Questionnaire provided to the custom software category representative.

	1	2	3	4	5	Observations
Was the background provided on custom software enough?			X			*No information provided.*
Do you consider all evaluation criteria are suitable to perform the assessment on the commercial software*?*					X	*No information provided.*
Are the supporting questions helpful in performing the assessment of these tools?					X	*No information provided.*
Do you consider any additional criterion for the evaluation should be added?	*No information provided.*					
Do you consider any additional tool (custom software) should be added?	*No information provided.*					

Table 10. Questionnaire provided to the commercial software category representative.

	1	2	3	4	5	Observations
Was the background provided on commercial software enough?				X		*It would be useful to have more technical information or more screenshots of the reports*.
Do you consider all evaluation criteria are suitable to perform the assessment on the commercial software*?*				X		*It would be also useful to evaluate the complexity of the data integration, data cleansing features and third-party connectors.*
Are the supporting questions helpful in performing the assessment of these tools?					X	*No information provided.*
Do you consider any additional criterion for the evaluation should be added?	*It would be also useful to evaluate the complexity of the data integration, data cleansing features and third-party connectors.*					
Do you consider any additional tool (commercial software) should be added?	*Business Objects*					

Table 11. Questionnaire provided to the libraries/API's category representative.

	1	2	3	4	5	Observations
Was the background provided on software libraries/APIs enough?			X			*No information provided.*
Do you consider All evaluation criteria are suitable to perform the assessment on the software libraries/APIs?	X					*No information provided.*
Are the supporting questions a helpful in performing the assessment of these tools?		X				*No information provided.*
Do you consider any additional criterion for the evaluation should be added?	*If yes, which ones?* *What dependencies a library might have? For example, flash and php only in case of OFC*					
Do you consider any additional tool (library) should be added?	*If yes, which ones?* *https://medium.com/dailyjs/data-visualization-libraries-for-react-developers-in-2019-a2b9c01262f8*					

The feedback provided by the evaluators during the pilot study suggested that they could assess the tools with few difficulties. For the custom software, they considered the evaluation criteria and the supporting questions adequate for the evaluation, although the background provided was not enough. For the commercial software category, they considered that background provided was adequate along with the evaluation criteria and the supporting questions. Finally, the background was enough for the libraries/APIs, but the criteria and the supporting questions were not considered adequate, according to the representative that provided the feedback.

To enhance the material given to evaluators and facilitate their assessment, we added some additional information. This information included description links to the related tools, and some additional information related to the tool's functionality. The added links allowed the experts, to have more information related to the tool and could see some demos. The feedback provided in the pilot test was useful to improve the methodology to later follow the assessment process with each group of experts according to the assigned category.

In chapter 6, results and discussion, the results found their analysis and the debate that will allow us to show the relevant data of this experiment will be discussed.

5.2 Multi-Screen & Dynamic Test Process

In this chapter we performed a case study based on our solution proposal stated in chapter 4, section 2, Multi-Screen & Dynamic Solution proposal. The case study was related to the energy sector and presented data based on electricity generation. We also implemented the metrics from section 4, metrics solution proposal. The use of our metrics solution proposal in the architecture is to ensure that it will fulfill the requirements and will generate visualizations able to give support to the decision-making process.

The objective of this study is to demonstrate that the proposed architecture design through its dynamic and multiscreen display characteristics, provide an information visualization tool that facilitates visual analysis, supports and improves the decision-making process. The purpose of this study seeks to answer the research question: *What kind of architecture design can be better in order to have an adequate information visualization tool that enhance the insights of results presented to decision-makers?*

5.2.1 Participants

There were sixteen participants divided into two groups for each phase of the experiment. First group of participants were between 25-35 years old, six men and two women, with bachelors, master and PhD degrees. The participants were working in private and public sector with experience in strategic planning, design of energy policies, energy policy operation, modeling, information visualization, data analytics, and consultancy. Second group of participants were in different range of ages, 25-35, 35-45, 45-55, and over 60, six men and two women, with bachelors, masters and PhD degrees. This second group of participants was working mostly in public sector with experience in strategic planning, design of energy policies, modeling, information visualization, data analytics, and consultancy. The first group was designated to work on phase one, mono-screen, second group was designated to work on the multi-screen, second phase.

5.2.2 Process

The experiment was be carried out in two phases, the first phase in single-screen format and the second in a multi-screen format. In both of the phases we presented visualizations based on the Energy sector information. Due to the pandemic caused by the COVID-19 virus, the sessions were carried out remotely through video calls and with the support of technology that allows showing the necessary material to carry out the corresponding evaluations, as well as the use of description of scenarios and use of prototypes that facilitated the use of the tools by the participants.

Both group pf participants were given a document with a case study related to the energy sector. They had to read and get familiar with this document before each of the sessions. The case study document explains, in general, the technologies considered for electricity generation in the model, which are based on the data published by the Ministry of Energy in its Energy Information System (Secretaría de Energía, 2018). The case study document is available in Appendix 3.

The first phase, mono display, was achieved with the use of Tableau Desktop Professional Edition 2020.2.2. This experiment showed a case based on the mathematical model of the Mexican electricity sector that contemplates evaluating the effects derived from the electricity sector's decision-making processes to have a better understanding of the impact of the actions during a period. In this case, we used the built-in story point tableau functionality, which allows ordering graphic representations sequentially and adding descriptive text so that the user can tell a story simultaneously as presenting the information (Mackinlay, Kosara, & Wallace). Below is the sequence of images corresponding to the visuals and the use of tableau story points in Fig. 29.

Fig. 29 Sample image of mono-display format of phase one.

In the second phase, through the use of storytelling, the transmission of videos, and the use of slides, the process that takes place during a session in the decision-making center was described to the participants. The elements used were:

Video:

- Laboratorio binacional / Teatro para la toma de decisiones <https://energialab.tec.mx/es/teatro-de-toma-de-decisiones>

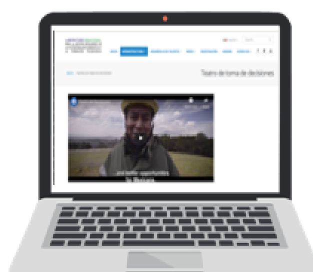

Fig. 30. Sample image of binational laboratory introductory video

- Collaborative Decision-Making Center <https://www.youtube.com/watch?v=AemRs7DdOT4&feature=youtu.be >

Fig. 31. Sample image of binational laboratory usage video

- Slide show (seven screen simulation, using Zoom)

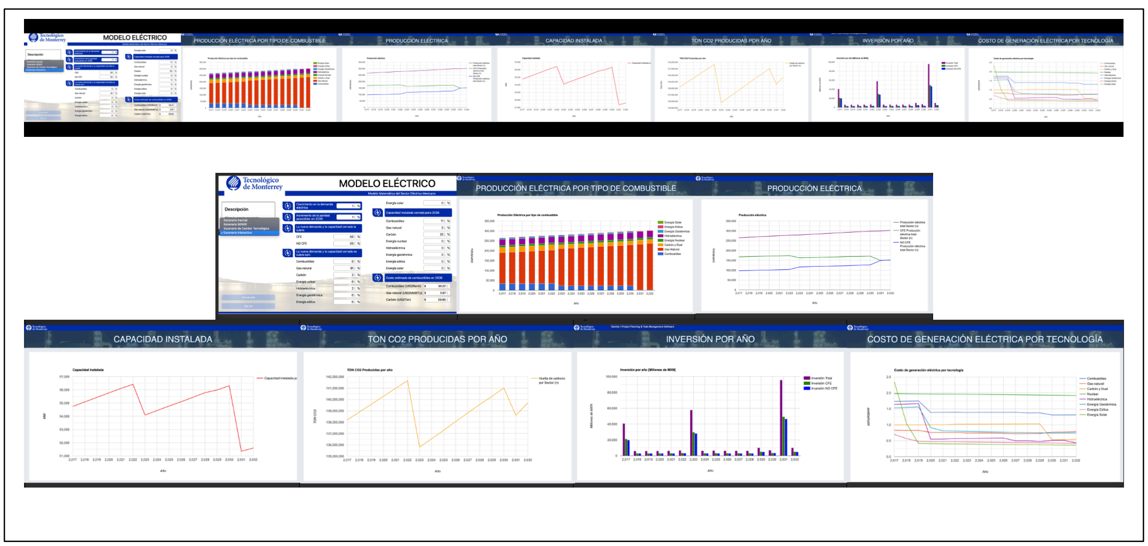

Fig. 32. Sample images of slide show.

At the end of each session, each group of participants was asked to answer a questionnaire of nine questions.

We used a Likert scale with values from 1 to 5, 1 representing total disagreement and 5 representing total agreement.

Question from 1 to 6 are related to the metric definition. Questions from 7 to 9 are related to decision-making activities. In Table 12 we present the questions asked to participants.

Table 12. Questions asked to participants after each phase one and phase two of the experiment.

	Metric	Question
	Visual scalability: Expressiveness	1. The tool reduces the participant's cognitive load, facilitating the assimilation of information through the visualization of corresponding graphic representations. 2. The graphic representations are congruent with the information reviewed.
	Visual Scalability: Effectiveness	3. The tool displays the data in a fast and efficient way through the use of technological resources. 4. Graphical representation reduces the time invested for analysis and obtaining results.
	Dynamics	5. The tool offers the possibility of interacting with different scenarios derived from the same information. 6. The tool supports visual analysis by updating information in real-time (From changes in input variables, changes/benefits/effects are reflected in output variables).
	Decision-making activities	7. The tool facilitates collaboration and consensus by allowing the joint participation of the group of participants. 8. The tool facilitates obtaining results, confirmation or rejection of hypotheses, by having the support and participation of a moderator during the session. 9. Describe the advantages, disadvantages, or suggestions perceived when using the tool in the presented model's information display.

This chapter presented the case study of the Energy sector and the implementation of both of our solution proposals. First one the use of metrics to validate and ensure a visualization tool supports a decision-making process. The second solution proposal the architecture of a tool for visualization in a decision-making center.

In chapter 8, results and discussion, the results found, their analysis and the debate that will allow us to show the relevant data of this experiment will be discussed.

6. Results and Discussion

In this chapter we present the results of the evaluation processes presented in chapter 5, the results of the metrics evaluation process and the multi-screen & dynamic evaluation process.

6.1 Metrics Evaluation Process Analysis and Results

We used validation instruments for the exploratory and validation phase in the metrics evaluation process, which we will explain next.

For the exploratory phase, we performed factor analysis using the "psych" package in R Studio to obtain Cronbach's alpha value of 0.81. This value was confirmed using SPSS Statistics software™. The obtained value is above 0.7, showing right consistency of the proposed scale (González Alonso & Pazmiño Santacruz, 2015), (Molina, Aranda, Flores, & López, 2013).

In the validation phase, the analysis was performed using SPSS Statistics Software™ and Microsoft Excel™ to obtain the Average Variance Extracted (AVE) value of 0.44 and a value of 0.79 for the Composite Reliability (CR) being adequate according to (Fornell & Larcker, 1981).

We now present the results of the evaluation performed by each expert group and the results to perform the hypothesis validation. Table 13 presents average scores for each of the criteria and set of tools of each group of visualization tools.

Table 13. Overall average provided in the assessment by evaluators

Criteria	*Libraries or APIs*			*Custom Software*			*Commercial Software*		
	Mean	SD	N	Mean	SD	N	Mean	SD	N
Visual scalability: Expressiveness									
Can the information presented be viewed in its entirety?	2.35	0.61	20	2.31	0.67	25	2.52	0.71	25
Does the information shown provide answers to the business questions?	1.93	0.80	20	2.44	0.51	25	2.8	0.41	25
Visual scalability: Effectiveness									
Do the visual representations provide an easy way to interpret the information?	2.15	0.75	20	2.16	0.50	25	2.64	0.57	25
Do the visual representations load easily and fast?	2.05	0.39	20	2.49	0.51	25	2.73	0.46	25
Aesthetics									
Do the graphs available fit the information presented?	2.4	0.60	20	2.45	0.69	25	2.76	0.44	25
Can the analysis be made in collaboration and shared with other users for collaboration	1.55	0.80	20	2.4	0.84	25	2.64	0.64	25
Dynamics									
Can the user edit the data presented?	1.75	0.79	20	2.16	0.50	25	2.44	0.77	25
Can new insights be generated from the information presented?	1.87	0.83	20	2.35	0.67	25	2.64	0.49	25

Recalling that the evaluation was made on a scale ranging from 1, being poor, 2, being average, and 3 being the top value representing advanced, we made an in-depth analysis on each of the criteria evaluated per group of tools.

For the questions “Can the information presented be viewed in its entirety?” and “Does the information shown provide answers to the business questions?”, the commercial software tool got the closest value to 3, showing they provide an advanced expressiveness feature. For the questions “Do the visual representations provide an easy way to interpret the information?” and “Do the visual representations load easily and fast?” the commercial software also got the closest values to 3, meaning this group of tools has an advanced effectiveness feature. For questions “Do the graphs available fit the information presented?” and “Can the analysis be made in collaboration and shared with other users for collaboration”, the commercial software category also obtained the highest values, showing advanced features in aesthetics. For the “Can the user edit the data presented?” and “Can new insights be generated from the information presented?” questions, commercial software category obtained 2.44 and 2.64, the closest values to 3, however they do have flexibility to work online by copying or duplicating the information, however the data cannot be modified or updated in real-time.

In the overall evaluation, the commercial software got the highest values, close to 3, on each question. Commercial software tools provide the most advanced features in the criteria related to expressiveness, effectiveness, aesthetics and dynamics. Among all of the criteria evaluated in the commercial software category, the lowest value obtained was the dynamic criteria. The overall lowest scores where for custom software and software libraries/APIs being visual scalability: effectiveness and dynamics, the weakest features for custom software and for software libraries/APIs visual scalability: expressiveness, aesthetics and dynamics got the lowest scores.

A more in-depth analysis using the Welch Two-sample t-test, confirms the veracity of the evaluation since they show a p-value below 0.001 for the comparison of commercial software against custom software and the comparison of commercial software against libraries/APIs. Although the p-value was higher for comparing custom software against libraries/APIs, it is still considered significant since its value is below 0.01. The degrees of freedom are given by the Welch–Satterthwaite equation and defines the probability distribution to calculate the p-values. The t-test, degrees of freedom and p-value analysis are shown in Table 14.

Table 14. Welch two sample t-test and p-value.

Welch Two Sample t-test			
	t-test	DF	p-value
Commercial vs. *Custom Software*	5.4572	13.972	8.51E-05
Commercial vs. *Libraries/APIs*	5.6205	9.158	0.0003056
Custom Software vs. *Libraries/APIs*	2.8486	8.9801	0.01917

The higher scores obtained for commercial software are due to the advanced features they provide on each metric evaluated. The average score for expressiveness is 2.66, for effectiveness the average score is 2.68, for aesthetics the average score is 2.7, and for dynamics, the average score is 2.54. Commercial software more decisive feature is aesthetics since they provide a variety in charts and colors adjustable to data presented, with the possibility of creating unique visualizations that support the user understanding and interpretation. These tools provide enough expressiveness, answering the business questions with a straightforward interpretation by representing data graphically and loading it quickly. Their lowest feature is dynamics, although they provide facilities to work with data online, copy and retrieve information cannot be modified and updated in real-time. All of the characteristics within these tools, made experts qualify them with the highest scores.

This comprehensive study and analysis helped us know which features are the most important to consider when choosing an appropriate tool to support and enhance the decision-making process. We classified different types of tools in three main groups, libraries/APIs, custom software, and commercial software. According to the scores obtained, commercial software got the highest scores, followed by custom software and libraries/APIs being second and third. Commercial software category provides advanced features that comply with the criteria definition, scalability: expressiveness, scalability: effectiveness, aesthetics, dynamics. Within the commercial software evaluated, it can be seen in Table 15, the average scores per criteria.

Table 15. Commercial software group average scores per tool and criteria.

	Qlikview	Microstrategy	Domo	Tableau	Sisense
Visual Scalability	2.7	2.59	2.54	2.7	2.84
Aesthetics	2.5	2.8	2.6	2.9	2.7
Dynamics	2.7	2.5	2.4	2.5	2.6
Total Average Score	2.63	2.63	2.52	2.7	2.71

In the previous table, it can be observed that the tools with higher scores are Tableau and Sisense. Tableau more advanced feature is related to aesthetics. Meanwhile, Sisense most robust feature is visual scalability. Both tools had a total average score closer to 3, 3 being the highest. Tableau and Sisense have a set of features including various charts, grouping and manipulating data while creating visualizations. Therefore, these are tools that better fulfill the criteria to give support to the decision-making process.

While usability testing and controlled experiments were the more common techniques for evaluating tools to find out if they were adequate to users (Plaisant, 2004), there have been some other works for evaluating information visualizations through the use of heuristic evaluation tree (Zuk, 2006), use of guidelines (Shneiderman & Plaisant, 2006) and use of methods such as thinking aloud tests, and formal experiments while comparing different visualization (Andrews K. , 2006). One of the most recent developments for evaluation of visualization is the heuristic evaluation, mostly based on Nielsen, Molich and Pierotti guidelines (Tarrell, 2014), with four specific heuristics, perception, cognition, usability, and interaction; all organized in a framework to facilitate the evaluation as in a checklist. Most of these previous works are either explicitly centered in the creation or development of the visualization and in the case of evaluating tools, the main goal is to find out if the visualizations are appropriate and the users find them easy to use and understand. Unlike these applicable works in general areas, our evaluation method includes the use of evaluation metric, with detailed criteria defined, and looking for a future specialized application field related to decision-making. The use of detailed criteria and evaluation metric makes our work distinct from previous researchers' traditional heuristic evaluation. However, one of the study's limitations was to find enough professionals able to provide time to perform the assessment. Another limitation was that some tools, such as those from libraries or APIs and Custom software, were hard to test since their implementation was complicated and required time to integrate and deploy, compared to commercial tools that are easy to download and install. Finally, as not all of the tools have a download version available, the analysis had to be made through literature review.

6.2 Multi-Screen & Dynamics Test Process Analysis and Results

We used validation instruments for the exploratory and validation phase this evaluation process, which we will explain next.

For the exploratory validation phase, we performed factor analysis using the “psych” package in R Studio to obtain Cronbach’s alpha value of 0.7 for the Mono-Screen phase experiment and a value of 0.71 for the Multi-Screen experiment. The overall value for both experiments was of 0.83. The obtained values are above 0.7, showing right consistency of the proposed scale (González Alonso & Pazmiño Santacruz, 2015), (Molina, Aranda, Flores, & López, 2013).

We now present the evaluation performed by each group and the results to perform the hypothesis validation. Table 16 presents average scores for each of the questions on each of the phases of the experiment.

Table 16. Results of the Mono-Screen Vs. Multi-Screen & Dynamic Test Process

	Mono-screen			**Multi-screen**		
Visual scalability: Expressiveness	*Mean*	*SD*	*N*	*Mean*	*SD*	*N*
The tool reduces the participant's cognitive load, facilitating the assimilation of information through the visualization of corresponding graphic representations.	3.6	0.5	8	4.9	0.35	8
The graphic representations are congruent with the information reviewed.	4.6	0.5	8	4.7	0.5	8
Visual scalability: Effectiveness						
The tool displays the data in a fast and efficient way through the use of technological resources.	4.0	0.8	8	5.0	-	8
Graphical representation reduces the time invested for analysis and obtaining results.	3.7	0.99	8	4.9	0.4	8
Dynamics						
The tool offers the possibility of interacting with different scenarios derived from the same information.	2.3	1.4	8	4.3	0.9	8
The tool supports visual analysis by updating information in real-time (From changes in input variables, changes/benefits/effects are reflected in output variables).	1.7	1.3	8	4.9	0.4	8
Decision-making activities						
The tool facilitates collaboration and consensus by allowing the joint participation of the group of participants.	4.4	0.8	8	4.5	0.5	8
The tool facilitates obtaining results, confirmation or rejection of hypotheses, by having the support and participation of a moderator during the session.	3.2	1.2	8	4.2	0.7	8

The evaluation was performed on a Likert scale ranging from 1, being totally disagree, to 5 being totally agree, we will present the analysis of the results.

For the question "*The graphic representations are congruent with the information reviewed.*", the Multi-screen phase has a mean value of 4.7 and Mono-screen phase has a mean value of 4.6. Not too much difference, however, Multi-screen is still higher. With these two questions it can be said that Multi-screen has a better performance for visual scalability: expressiveness. In question "*The tool facilitates collaboration and consensus by allowing the joint participation of the group of participants.*", the Multi-screen phase has a mean value of 4.5, Mono-screen phase has a mean value of 4.4. These two questions were the only ones evaluated with the closest values for both of the phases, however the evidence shows a significant difference in rest of evaluated questions.

The total average score for the Multi-screen phase was 4.7, the total average score for the Mono-screen phase was 3.4. Although they had similar values in two questions "*The graphic representations are congruent with the information reviewed.*", "*The tool facilitates collaboration and consensus by allowing the joint participation of the group of participants.*", the Multi-screen phase generally shows higher values.

Although we used the proposed metric definition, we added a fourth criterion, "Decision-making activities", to complement the metric since we wanted to know which one of the phases, aside from having the required features, gives better support to the decision-making process.

In the overall results it can be observed that Multi-screen phase had the higher values for all of the four criteria evaluated. Hence, the Multi-screen tool presented offers better capabilities on visual scalability: expressiveness, visual scalability: effectiveness, dynamics and decision-making activities.

For the opinion analysis, we asked each group the next question "*Describe the advantages, disadvantages, or suggestions perceived when using the tool in the presented model's information display*." We used the "wordcloud" library of R Studio to perform the user perception analysis.

For the Mono-screen phase we generated a word cloud with the identified benefits from the evaluators. The generated word cloud is in fig. 33.

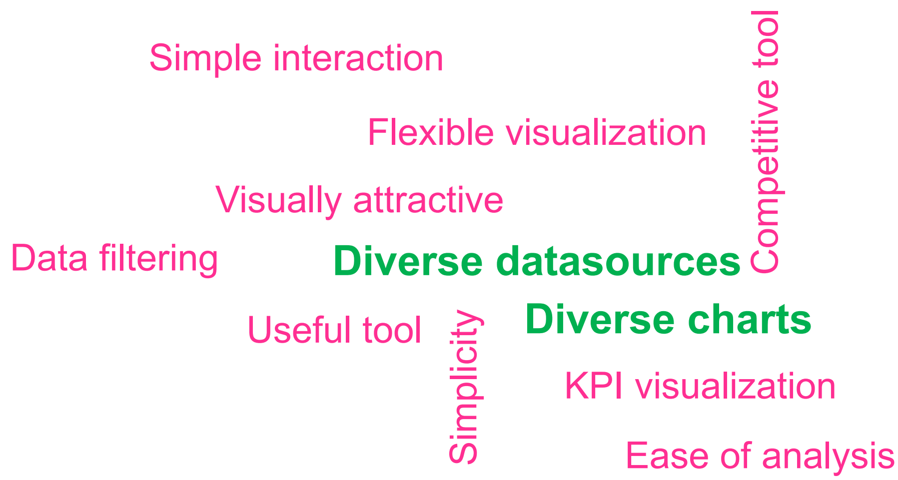

Fig. 33 Generated word cloud of advantages in Mono-screen phase from participants.

The featured features of the Mono-Screen phase are diverse data sources and diverse charts. Secondly were the simple interaction, flexible visualization, visually attractive, competitive tool, data filtering, useful tool, simplicity, KPI visualization and ease of analysis. As disadvantages the evaluators identified six features the tool should have, simultaneous visualization, real time analysis, better interaction, reduce charts and make them according to the business cases, should allow visualization of multiple scenarios and do not have dependency on external data sources.

For the same question, the word cloud of the Multi-Screen phase is shown in the Fig. 34.

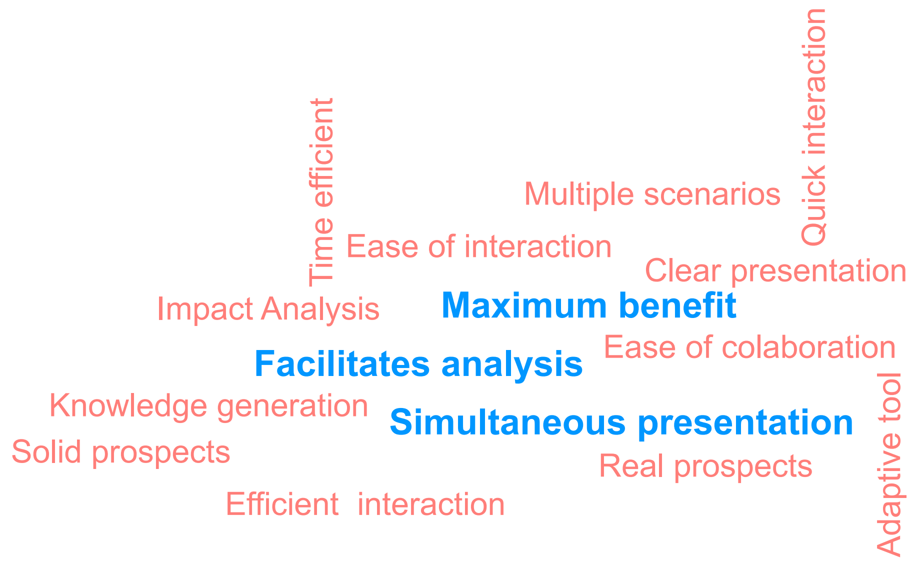

Fig. 34 Generated word cloud of advantages in Multi-screen phase from participants.

There are three features that were more relevant than the rest referring to the tool facilitates analysis, simultaneous presentation, maximum benefit. Rest of the identified advantages were time efficient, ease of interaction, multiple scenarios, quick interaction, clear presentation, impact analysis, knowledge generation, ease of collaboration, solid prospects, adaptive tool, efficient interaction, real prospects. The evaluators identified four features to be improved connection to data sources, colors and fonts, add maps functionality, consider limitations of disabled people, like color blind ones.

Our null hypothesis states that "*In an architecture design, dynamic and multi-screen visualization are features that enhance the information visualization tools hence give support to the evidence-based decision-making activities in a multi-level perspective equally than a mono-screen visualization".*

Our alternate hypothesis states “*In an architecture design, dynamic and multi-screen visualization are features that enhance the information visualization tools hence give better support to the evidence-based decision-making activities in a multi-level perspective than a mono-screen visualization*”

An analysis using the Welch Two-sample t-test, confirms the veracity of the evaluation since they show a p-value below 0.05 for the comparison of Multi-screen phase against the Mono-screen phase. Meaning the null hypothesis is rejected and confirming the alternate one “*In an architecture design, dynamic and multi-screen visualization are features that enhance the information visualization tools hence give better support to the evidence-based decision-making activities in a multi-level perspective than a mono-screen visualization*”

The t-test, degrees of freedom and p-value analysis are shown in Table 17.

Table 17. T-test and p-value results of Multi-screen and Mono-screen phases.

t-test	*DF*	*p-value*
3.2197	8.2106	0.01183

The metric evaluation process was used to confirm the defined criteria was useful to determine if a visualization tool can fulfill specific requirements for enhancing the decision-making process. The selected criteria were visual scalability: expressiveness, visual scalability: effectiveness, aesthetics, and dynamics. We reviewed among three different type of tools that we classified in three categories as libraries/APIs, custom software and commercial software. The tools evaluated showed significant values on the evaluation, however the dynamics feature is limited since it doesn’t really permit the edition of values while working with the visualization, this can be seen in the group of tools with the highest values on this criterion, the group of commercial software.

In our second evaluation process, multi-screen & dynamics, we took one tool form the best evaluated group in the previous process to use it in as the mono-screen phase. We had a second group in a multi-screen phase. The use of multi-screen visualization improves the visual scalability: expressiveness, since it displays the information in simultaneous multiple screens. Also, we added a fourth set of questions related to the decision-making activities.

In the multi-screen & dynamics phase we used a multi-screen visualization tool, based on our architecture proposal, that provides visualization in a multiple and simultaneous view allowing the participant to modify the scenario and see the immediate update in the charts given its dynamics feature.

The results confirmed that a multi-screen visualization tool gives better support to decision making activities.

In this section we presented the results and discussion of the two experiments performed according to our two proposals stated in chapter 4, the Metrics solution proposal and the Multi-Screen & Dynamic proposal to give support to the decision-making process. In the next section we will present conclusions and future work.

7. Conclusion & Future Work

The objective of this thesis work focused on the use of information visualization tools as support in the decision-making process based on evidence. We did a study on existing tools; we also did a study on the evaluation of visualization tools. There are several evaluation methods for information visualization tools; however, not finding any specific to Evidence-Based Decision-Making, we made a proposal taking into account the needs of the process. Our metric proposal considers four essential criteria. Visual Scalability: Expressiveness refers to the efficient graphic deployment of the information that is being analyzed. Visual Scalability: Effectiveness refers to the proper use of resources for the deployment of the information. Aesthetics, which refers to the proper use of colors, graphs, typography, and space distribution so that the displayed visualization is pleasing to the user and facilitates their assimilation. Dynamics, referring to the modification and updating of real-time information, allowing the decision-maker to generate new scenarios for the best analysis and understanding of the revised case. Within the review of existing visualization information tools and with the use of our metric evaluation proposal, we do not find any specific use tool for decisions based on evidence.

In this way, we developed an architecture proposal that fits the previously defined criteria. We develop the necessary tests to verify that our proposal is valid and supports and improves the decision-making process based on evidence. The results showed that our proposed architecture, following our metric's evaluation criteria, was significantly better. There are two important features in our Multi-Screen & Dynamic Visualization Tool architecture proposal. The first one is the deployment of information on multiple screens. This feature has simultaneous visualization on multiple screens, facilitating the analysis of the information presented. The deployment of simultaneous information allows decision-makers to analyze various information segments related to an environment that encourages collaboration and facilitates consensus. The second important characteristic is that the information is dynamic; it allows the generation of scenarios and its modification for the impact analysis and consideration of different alternatives. This characteristic that other tools do not have simplifies the decision-making process by not having to wait for new scenarios to be generated because they are created dynamically while analysis is carried on.

In the survey made to our study participants, positive comments focused on mentioning that our proposed tool provides maximum benefit, facilitates collaboration, efficient time, facilitates the analysis, helps to do impact analysis, and aids to the knowledge generation. In the aspect of improvements, they suggest having a variety of data sources, improving colors and sources used, and considering the possible visual limitations of the participants.

In future work, we will consider modifying the decision-making model as an improved alternative since now we work with scenarios generated from the model, and possibly this may limit the analysis given the demand and the constant change of the information. The development of the tool will be carried out, and its implementation will be sought in various organizations for testing that allows its improvement and consolidation as a tool for visualization of information in decision making.

References

Cacho, A. E. (2015). A smart destination initiative: The case of a 2014 FIFA World Cup Host City. *Smart Cities Conference (ISC2)* (pp. 1-6). Guadalajara: IEEE.

Lam, H., Bertini, E., Isenberg, P., Plaisant, C., & Carpendale, S. (2012, December). Empirical Studies in Information Visualization: Seven Scenarios. *IEEE Transactions on Visualization and Computer Graphics, 18*(9), 1520-1536.

Carlson, A. (2002). *Aesthetics and the environment: The appreciation of nature, art and architecture.* London: Psychology Press.

Larson, K. L., & Edsall, R. M. (2010). Larson, K. L., & Edsall, R. M. (2010). The impact of visual information on perceptions of water resource problems and management alternatives. *ournal of Environmental Planning and Management*, 335-352.

Castrejón, J. C.-L. (2011). Model2Roo: A model driven approach for web application development based on the Eclipse Modeling Framework and Spring Roo. *Electrical Communications and Computers (CONIELECOMP), 2011 21st International Conference* (pp. 82-87). Cholula: IEEE.

Lee, H. Y., & Wang , N. J. (2013). The implementation of integrating e-Procurment, e-Contracting and e- Invoice platforms for the B2B E-MarketPlace web-based system . *International Journal of Engineering Research* , 300-309.

Lee, S. J. (2015). Restful Web Service and Web-Based Data Visualization for Environmental Monitoring. *International Journal of Software Innovation*, 75-94.

Chen, C. (2005). Top 10 unsolved information visualization problems. *IEEE computer graphics and applications*, 12-16.

Chen, C. (2010). Information visualization. *Wiley Interdisciplinary Reviews: Computational Statistics*, 387-403.

Cerny, T., Chalupa, V., & Donahoo, M. J. (2012). Towards Smart User Interface Design. *2012 International Conference on Information Science and Applications* (pp. 1-6). IEEE.

CFE. (2007). Programa de Obras e Inversiones en el Sector Eléctrico 2007-2016. México.

Liu, S., Maljovec, D., Wang, B., Bremer, P. T., & Pascucci, V. (2016, December). Visualizing high-dimensional data: Advances in the past decade. *IEEE transactions on visualization and computer graphics, 23*(3), 1249-1268.

Cook, K. A., & Thomas, J. J. (2005). *Illuminating the path: The research and development agenda for visual analytics (No. PNNL-SA-45230). Pacific Northwest National Lab.* Richland, WA: Pacific Northwest National Lab.

Costagliola, G., De Rosa, M., Fuccella, V., & Perna, S. (2018, October). Visual languages: A graphical review. *Information Visualization, 17*(4), 335-350.

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, 297-334.

Lu, M. C. (2017). Frontier of information visualization and visual analytics in 2016. *Journal of Visualization*, 667-686.

Lu, M., Chen, S., Lai, C., Lin, L., & Yuan, X. (2017, November). Frontier of information visualization and visual analytics in 2016. *Journal of Visualization, 20*(4), 667-686.

Lu, M., Chen, S., Lai, C., Lin, L., & Yuan, X. (2017, November). Frontier of information visualization and visual analytics in 2016. *Journal of Visualization, 20*(4), 667-686.

Luminiţa, Ş. &. (2012). Optimizing time in business with Business Intelligence solution. *Procedia-Social and Behavioral Sciences*, 638-648.

Luo, X. Y. (2019). Enhancing statistical charts: toward better data visualization and analysis. *Journal of Visualization*, 1-14.

Luo, X., Yuan, Y., Zhang, K., Xia, J., Zhou, Z., Chang, L., & Gu, T. (2019, August). Enhancing statistical charts: toward better data visualization and analysis. *Journal of Visualization, 22*(4), 1-14.

López-Landa R.A., N. J. (2012). EMF on Rails. *Proceedings of 7th International Conference on Software Paradigms Trends* (pp. 24-27). Rome: ICSOFT.

Lurie, N. H. (2007). Visual representation: Implications for decision making. *Journal of marketing*, 160-177.

Lurie, N. H. (2007). Visual representation: Implications for decision making. *Journal of marketing*, 160-17.

Baba, V. V., & HakemZadeh, F. (2012). Toward a theory of evidence based decision making. *Management decision*, 832-867.

Al-Kassab, J. O. (2014). Information visualization to support management decisions. *International Journal of Information Technology & Decision Making*, 407-428.

Alonso, S., Herrera-Viedma, E., Cabrerizo, F. J., Porcel, C., & López-Herrera, A. G. (2007). Using Visualization Tools to Guide Consensus inGroup Decision Making. *nternational Workshop on Fuzzy Logic and Applications* (pp. 77-85). Berlin: Springer.

A Shukla, S. D. (2016). Shukla, A., & Dhir, S. (2016). Tools for data visualization in business intelligence: Case study using the tool qlikview. *Information Systems Design and Intelligent Applications* , 319-326.

Abeywickrama , D. B., Hoch , N., & Zambonelli , F. (2014). An integrated Eclipse plug-in for engineering and implementing self-adaptive systems. *WETICE Conference (WETICE), 2014 IEEE 23rd International* (pp. 3-8). Parma: IEEE.

Amar, R., & Stasko, J. (2004). A knowledge task-based framework for design and evaluation of information visualizations. *IEEE Symposium on Information Visualization* (pp. 143-150). IEEE.

Anderson, C. (2012). The Model-View-ViewModel (MVVM) Design Pattern. *Pro Business Applications with Silverlight 5*.

Andrews, C., Endert, A., Yost, B., & North, C. (2011, October). Information visualization on large, high-resolution displays: Issues, challenges, and opportunities. *Information Visualization, 10*(4), 341-355.

Andrews, K. (2006). Evaluating information visualisations. *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization* (pp. 1-5). Venice, Italy: ACM.

Aronson, J. E. (2005). *Decision support systems and intelligent systems (Vol. 4).* New York: Prentice Hall.

Atriwal, L. N. (2016). Business Intelligence Tools for Big Data. *ournal of Basic and Applied Engineering Research*, 505-509.

Avazpour, I. G. (2015). Generating reusable visual notations using model transformation . *International Journal of Software Engineering and Knowledge Engineering* , 277-305.

Balderas, D., Molina Espinosa, J. M., & Ruiz Loza, S. (2020). Decision-making laboratory for socio-technological systems. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 1557-1568.

Baba, V. V. (2012). Toward a theory of evidence based decision making. . *Management decision*, 832-867.

Badam, S. K. (2015). Munin: a peer-to-peer middleware for ubiquitous analytics and visualization spaces. *IEEE transactions on visualization and computer graphics 21(2)*, 215-228.

Bao, F. &. (2014). Visual framework for big data in d3. js. *2014 IEEE Workshop on Electronics, Computer and Applications* (pp. 47-50). Ottawa, ON, Canada: IEEE.

Bass, L., Clements, P., & Kazman, R. (2013). *Software Architecture in Practice.* Addison-Wesley .

Bassil, S., & Keller, R. K. (2001). Software Visualization Tools: Survey and Analysis. *Proceedings 9th International Workshop on Program Comprehension. IWPC 2001* (pp. 7-17). Toronto, Canada: IEEE.

Behrisch, M., Blumenschein, M., Kim, N. W., Shao, L., El-Assady, M., Fuchs, J., . . . Keim, D. (2018). Quality metrics for information visualization. *Computer Graphics Forum*, 625-662.

Ben, R. (1986). Decision Theatre: Curtain Gogln Up on an Innovative Approach to Management Education. *Business Horizons*, 70-77.

Bieh-Zimmert, O. &. (2014). Shaping Unlimited Patterns: A Vision for State-of-the-Art Visual Scalability. *Proceedings of the 6th International Conference on Management of Emergent Digital EcoSystems* (pp. 72-77). Al Qassim, Buraidah, Saudi Arabia: Association for Computing Machinery.

Bieh-Zimmert, O., & Felden, C. (2014). Shaping Unlimited Patterns: A Vision for State-of-the-Art Visual Scalability. *Proceedings of the 6th International Conference on Management of Emergent Digital EcoSystems*, 72-77.

Borkin, M. A. (2013). hat makes a visualization memorable? *IEEE Transactions on Visualization and Computer Graphics*, 2306-2315.

Borkin, M. A., Vo, A. A., Bylinskii, Z., Isola, P., Sunkavalli, S., Oliva, A., & Pfister, H. (2013, October). What makes a visualization memorable? *IEEE Transactions on Visualization and Computer Graphics, 16*(9), 2306-2315.

Boukherroub, T., D'amours, S., & Rönnqvist, M. (2018). Sustainable forest management using decision theaters: Rethinking participatory planning. *Journal of Cleaner Production*, 567-580.

Brehmer, B. (2007). OLF 2010: A Swedish Command Post of the Future. *Decision Making in complex environments* , 129-142.

Burkhard, R. A. (2004). Learning from Architects: The Difference between Knowledge Visualization and Information Visualization. *Proceedings. Eighth International Conference on Information Visualisation* (pp. 519-524). London, UK: IEEE.

Deitrick, S., & Edsall, R. (2006). The influence of uncertainty visualization on decision making: An empirical evaluation. *Progress in spatial data handling* (pp. 719-738). Berlin, Heidelberg: Springer.

Diehl, A., Abdul-Rahman, A., M., E.-A., Bach, B., Keim, D., & Chen, M. (2018). VisGuides: A Forum for Discussing Visualization Guidelines. *Eurovis*, 61-65.

Dirección General de Industria, Energía y Minas de la Comunidad de Madrid. (2007). El Carbón. España.

Drucker, S. M., Fisher, D., Sadana, R., Herron, J., & Schraefel , M. C. (2013, April). TouchViz: a case study comparing two interfaces for data analytics on tablets. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Duru, H. A., Çakır, M. P., & & İşler, V. (2013). How does software visualization contribute to software comprehension? A grounded theory approach. *International Journal of Human-Computer Interaction*, 743-763.

El-laithy, R. A. (2012). Study on the use of Microsoft Kinect for robotics applications. *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION* (pp. 1280-1288). Myrtle Beach, SC, USA : IEEE.

Elkington, R. (2018). Leadership Decision-Making Leveraging Big Data in Vuca Contexts. *Journal of Leadership Studies*, 66-70.

Elmqvist, N., & Yi, J. S. (2013). Patterns for visualization evaluation . *Information Visualization*, 250-269.

Elmqvist, N., & Yi, J. S. (2015, July). Patterns for visualization evaluation. *Information Visualization, 14*(3), 250-269.

ENGELBRECHT, L., BOTHA, A., & ALBERTS, R. (2015). Designing the Visualization of Information. *International Journal of Image and Graphics*, 1-20.

Eppler, M. J. (2013). Visualization in management: From communication to collaboration. A response to Zhang. *Journal of Visual Languages & Computing*, 146-149.

Estayno, M., Dapozo, G., Cuenca Pletch, L., & Greiner, C. (2009). Modelos y Métricas para evaluar Calidad de Software. *XI Workshop de Investigadores en Ciencias de la Computación.*

Feinleib, D. (2014). *The Big Data Landscape. In: Big Data Bootcamp.* California: Apress, Berkeley, CA.

Few, S. &. (2007). *Data visualization: past, present, and future.* IBM Cognos Innovation Center.

Fornell, C., & Larcker, D. F. (1981). Evaluating structural equation models with unobservable variables and measurement error. *Journal of marketing research*, 39-51.

Fornell, & Larcker, D. F. (1981). Evaluating structural equation models with unobservable variables and measurement error. *Journal of marketing research*, 39-50.

Forsell, C., & Johansson, J. (2010). An heuristic set for evaluation in information visualization. *Proceedings of the International Conference on Advanced Visual Interfaces*, (pp. 199-206).

Gallagher, K., Hatch, A., & Munro, M. (2008). An Evaluation Framework and Its Applicatio. *IEEE Transactions on Software Engineering*, 260-270.

Gacek, C., Abd-Allah, A., Clark, B., & Boehm, B. (1995). On the definition of software system architecture. *Proceedings of the First International Workshop on Architectures for Software Systems*, 85-94.

Gartner, I. (2018, November 12). *Magic Quadrant for Analytics and Business Intelligence Platforms*. Retrieved from Gartner: https://www.gartner.com/document/3861464?ref=ddrec&refval=3847505

Gartner, I. (2019). Retrieved July 1, 2019, from Magic Quadrant of Gartner: https://www.gartner.com/document/3900992

Gawlikowska, A. P., Marini, M., Chokani, N., & Abhari, R. S. (2018). Visualisation and Immersion Dome Experience for Inspired Participation. *Journal of Sustainable Development of Energy, Water and Environment Systems*, 67-77.

Gober, P., Wantz, E. A., Lant, T., Tschudi, M. K., & Kirkwood, C. W. (2011). WaterSim: a simulation model for urban water planningin Phoenix, Arizona, USA. *Environment and Planning B: Planning and Design*, 197-215.

González Alonso, J., & Pazmiño Santacruz, M. (2015). Cálculo e interpretación del Alfa de Cronbach para el caso de validación de la consistencia interna de un cuestionario, con dos posibles escalas tipo Likert. *Revista publicando, 2(1)*, 62-67.

Google, I. (2017, March 2). Retrieved March 2, 2017, from https://developer.android.com/guide/topics/ui/declaring-layout.html

Governor, J., Hinchcliffe, D., & Nickull, D. (2009). *Web 2.0 Architectures: What entrepreneurs and information architects need to know.* O'Reilly Media, Inc.

Graetz, K. A. (2006). The psychology of learning environments. *Educause Review*, 60-75.

Gu, C., & Zhang, X. (2010). An SOA based enterprise application integration approach. *010 Third International Symposium on Electronic Commerce and Security*, 324-327.

Hanrahan, Pat; Stolte, Chris; Mackinlay, Jock;. (2018). *Whitepapers: visual analysis for everyone.* Retrieved from Business Intelligence and Analytics Software: https://www.tableau.com/learn/whitepapers

Harrison, L., Reinecke, K., & Chang, R. (2015). nfographic aesthetics: Designing for the first impression. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp. 187-1190). ACM.

Heer, J. &. (2012). Interactive dynamics for visual analysis. *ACM Queue*, 30.

Heer, J., & Shneiderman, B. (2012, February). Interactive dynamics for visual analysis. *ACM Queue, 10*(2), 30.

Hoschka, P. (1998). An introduction to the synchronized multimedia integration language. *IEEE MultiMedia*, 84-88.

Hu, Q., & Johnston, E. (2012). Using a wiki-based course design to create a student-centered learning environment: Strategies and lessons. *Journal of Public Affairs Education*, 493-512.

Hu, Q., Johnston, E., & Hemphill, L. (2013). Fostering cooperative community behavior with IT tools: the influence of a designed deliberative space on efforts to address collective challenges. *The Journal of Community Informatics*.

Huang, C.-C., Wang, Y.-M., Wu, T.-W., & Wang, P.-A. (2013). An Empirical Analysis of the Antecedents and Performance Consequences of Using the Moodle Platform. *International Journal of Information and Education Technology*, 217-221.

Isenberg, T., Isenberg, P., Chen, J., Sedlmair, M., & Möler, T. (2013, December). A systematic review on the practice of evaluating visualization. *IEEE Transactions on Visualization and Computer Graphics, 18*(9), 2818-2827.

Ivanov, L. (2011). The I-phone/I-pad course: a small college perspective . *Journal of Computing Sciences in Colleges* , 142-148.

Jetter, H. C. (2012). Design and implementation of post-WIMP distributed user interfaces with ZOIL. *nternational Journal of Human-Computer Interaction*, 737-747.

Jetter, H. C., Zöllner, M., Gerken, J., & Reiterer, H. (2012). Design and Implementation of Post-WIMP Distributed User Interfaces with ZOIL. *International Journal of Human-Computer Interaction*, 737-747.

Jha, D. K. (2004). *Nuclear Energy.* India: Discovery Publishing House.

Johansen, B. (2007). *Get there early: Sensing the future to compete in the present.* Berrett-Koehler Publishers.

Juárez, J. (1992). *Centrales Hidroeléctricas.* México: Universidad Autónoma Metropolitana.

Kennard, R., & Leaney, J . (2010). Towards a general purpose architecture for UI generation. *Journal of Systems and Software*, 1896-1906.

Kienle, H. M., & Müller, H. A. (2007). Requirements of Software Visualization Tools: A Literature Survey. *2007 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis* (pp. 2-9). Banff: IEEE.

King, J., McKegg, K., Oakden, J., & Wehipeihana, N. (2013). Evaluative rubrics: A method for surfacing values and improving the credibility of evaluation. *Journal of MultiDisciplinary Evaluation*, 11-20.

Knaflic, C. N. (2015). *Knaflic, C. N. (2015). Storytelling with data: A data visualization guide for business professionals.* John Wiley & Sons.

Krunic, M. L. (2013). An approach to model driven development and automatic source code generation of GUI controls. *ntelligent Systems and Informatics (SISY), 2013 IEEE 11th International Symposium* (pp. 63-38). Subotica: IEEE.

McCarter, R. (2010). *Design Patterns - Problems and Solutions with Model-View-ViewModel.* Retrieved from docs.microsoft.com: https://docs.microsoft.com/en-us/archive/msdn-magazine/2010/july/design-patterns-problems-and-solutions-with-model-view-viewmodel

Macik, M. C. (2014). Context-sensitive, cross-platform user interface generation. *Journal on Multimodal User Interfaces*, 217-229.

Macik, M. K. (2011). UI generation for data visualisation in heterogeneous environment. *International Syposium on Visual Computing* (pp. 647-658). Berlin: Springer.

Mackinlay, J., Kosara, R., & Wallace, M. (n.d.). White paper:Data Storytelling Using visualization to share the human impact of numbers. *Tableau* .

Maresca, P., Stanganelli , L., & Coccoli, M. (2011). Managing a software project leveraging students' cooperation: on the road to Eclipse (OTRE) experience. *Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement* (pp. 96-100). Brindisi: ACM.

Martínez-Ruiz, J. M. (2006). Transformation of XAML schema for RIA using XSLT & UsiXML. *In Proc. of XlX Congreso Nacional y V Congreso Internacional de Informática y Computación de la ANIEI, Avances en Tecnologías de la Información CNCIIC* .

Marx, L. (2010). Technology: The emergence of a hazardous concept. *Technology and Culture*, 551-577.

Merčun, T. (2014). Evaluation of information visualization techniques: analysing user experience with reaction cards. *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization* (pp. 103-109). ACM.

Merčun, T. (2014). Evaluation of information visualization techniques: analysing user experience with reaction cards. *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization* (pp. 103-109). November: ACM.

Microsoft. (2009, January 01). *Microsoft Developer Network*. Retrieved March 21, 2017, from Microsoft Developer Network: https://msdn.microsoft.com/en-us/library/ff629155.aspx

Microsoft. (2014, December 23). [MS-XAML-2009]: XAML Object Mapping Specification 2009 . *[MS-XAML-2009]: XAML Object Mapping Specification 2009* . Microsoft.

Miettinen, K. (2014). Survey of methods to visualize alternatives in multiple criteria decision making problems. *OR Spectrum*, 3-37.

Molina, J. B., Aranda, L. L., Flores, M. H., & López, E. J. (2013). Utilización del alfa de Cronbach para validar la confiabilidad de un instrumento de medición de satisfacción del estudiante en el uso del software Minitab MISP. *Utilización del alfa de Cronbach para validar la confiabilidad de un instrumento de medición de satisfacción del estudiante en el uso del software Minitab MISP*, (pp. 14-16).

Mukherjee, S. (2014). Ovum Decision Matrix: Selecting a Business Intelligence Solution, 2014–15. *Ovum*.

Munzner, T. (2008). Process and pitfalls in writing information visualization research papers. *Information visualization*, 134-153.

Munzner, T. (2009, October). A nested model for visualization design and validation. *IEEE transactions on visualization and computer graphics, 15*(6), 921-928.

Myers, B. A. (1995). User Interface software tools. *ACM Transactions on Computer-Human Interaction (TOCHI)* (pp. 64-103). New York: ACM.

Navarrete-Corella, C., Noguez, J., & Molina, M. (2018). Review of Graphic User Interfaces Generation Tools.

Navarrete Corella, C., Noguez, J., Molina Espinosa, M., Magaña, A., & Benes, B. (2021). Criteria Definition Towards a Metric for Visualization Tools. *Journal of Information Visualzation*.

Networks of Decision Theaters (Mexico City Campus). (2019). Retrieved from ASU in Mexico: https://mexico.asu.edu/network-decision-theaters-mexico-city-campus

Nielsen, J. (2005). Ten usability heuristics.

Plaisant, C. (2004). The challenge of information visualization evaluation. *Proceedings of the working conference on Advanced visual interfaces* (pp. 109-116). Gallipoli (LE), Italy: ACM.

Park, T. H., & Wiedenbeck, S. (2010). First Steps in Coding by Informal Web Developers . *2010 IEEE Symposium on Visual Languages and Human-Centric Computing* (pp. 79-82). Madrid: IEEE.

Pemex. (2017, Mayo 8). Retrieved from Combustóleo pesado: http://www.pemex.com/comercializacion/productos/Paginas/refinados/combustoleo-pesado.aspx

Pemex. (2017, Mayo 2017). Retrieved from Gas Natural: http://www.gas.pemex.com.mx/PGPB/Productos+y+servicios/Gas+natural/

Podeschi, R. (2014). Experiential Learning using QlikView Business Intelligence Software. *Proceedings of the Information Systems Educators Conference.* Baltimore.

QlikView® architectural overview. (2014). *Qlik Resource Library: Visual Analytics Insights.* Retrieved from Qlik: https://www.qlik.com/us/resource-library

Raneburger, D., Popp, R., & Vanderdonckt, J. (2012). An Automated Layout Approach forModel-Driven WIMP-UI Generation. *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems* (pp. 91-100). ACM.

Reynoso, C. B. (2004). Reynoso, C. B. (2004). Introducción a la Arquitectura de Software. *Universidad de Buenos Aires*.

Richardson, J., Sallam, R., Schlegel, K., Kronz, A., & Sun, J. (2020). *Magic Quadrant for Analytics and Business Intelligence Platforms.* Gartner.

Rousseau, D. M.-L. (2015). Evidence-Based Management. *Wiley Encyclopedia of Management*, 1-5.

Rutledge, L., Hardman, L., & Ossenbruggen, J. (1999). THE USE OF SMIL: MULTIMEDIA RESEARCH CURRENTLY APPLIED ON A GLOBAL SCALE. *Proceedings of Multimedia Modeling*, (pp. 1-17). Amsterdam.

Slováček, V. M. (2009). Development framework for pervasive computing applications. *ACM SIGACCESS Accessibility and Computing*, 17-29.

Saldanha, A. J. (2004). Java Treeview—extensible visualization of microarray data. *Bioinformatics*, 3246-3248.

Saket, B. E. (2016). Beyond usability and performance: A review of user experience-focused evaluations in visualization. *Proceedings of the Sixth Workshop on Beyond Time and Errors on Novel Evaluation Methods for Visualization* , 133-142.

Savikhin, A., Maciejewski, R., & Ebert, D. S. (2008). Applied visual analytics for economic decision-making. *Symposium on visual analytics science and technology* (pp. 107-114). IEEE.

Secretaría de Energía. (2018, Marzo 2015). Retrieved from Sistema de Información Energética: http://sie.energia.gob.mx/

Secretaría de Energía de Argentina, S. D. (n.d.). Energía Geotérmica. Argentina.

Sedlmair, M., & Aupetit, M. (2015). Data-driven evaluation of visual quality measures. *Computer Graphics Forum*, 201-210.

Shneiderman, B., & Plaisant, C. (2006). Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization* (pp. 1-7). Venice, Italy: ACM.

Simon, H. A. (1960). *The new science of management decision.* New Jersey: Prentice-Hall.

Smith, J. (2009). *Patterns - WPF Apps With The Model-View-ViewModel Design Pattern.* Retrieved from docs.microsoft.com: https://docs.microsoft.com/en-us/archive/msdn-magazine/2009/february/patterns-wpf-apps-with-the-model-view-viewmodel-design-pattern

Sopan, A. F.-M. (2013). Exploring data distributions: Visual design and evaluation. . *International Journal of Human-Computer Interaction* , 77-95 .

Sousa, F. R., Cordeiro, L. C., & de Lima Fihlo, E. B. (n.d.). Bounded Model Checking of C++ Programs Based on the Qt Framework . *IEEE 4th Global Conference on Consumer Electronics* (pp. 179-180). Osaka: IEEE.

Szelitzky , E., Aluţei , A.-M., Chetran , B., & Mândru , D. (2011). Data Glove and Virtual Environment - a Distance Monitoring and Rehabilitation Solution . *Proceedings of the*

3rd International Conference on E-Health and Bioengineering - EHB 2011 (pp. 1-4). Iasi: IEEE.
Tableau: Business Intelligence and Analytics Software. (2018). Retrieved from Business Intelligence and Analytics Software: www.tableau.com
Taber, K. S. (2018). The use of Cronbach's alpha when developing and reporting research instruments in science education. *Research in Science Education*, 1273-1296.
Tarrell, A. F. (2014). Toward visualization-specific heuristic evaluation. *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization* (pp. 110-117). Paris, France: ACM.
Thomas, J. J., & Cook, K. A. (2006). A visual analytics agenda. *IEEE computer graphics and applications*, 10-13.
Thomas, M. (2014). iOS app programming using an inverted classroom in a small department . *Journal of Computing Sciences in Colleges* , 179-185.
Tong, W. (2010). Wind Power Generation and Wind Turbine Design. *Wit Press*.
Verma, N. &. (2016). On being actionable: Mythologies of business intelligence and disconnects in drill downs. *Proceedings of the 19th International Conference on Supporting Group Work*, 325-334.
Wark, S., Broughton, M., Nowina-Krowicki, M., Zschorn, A., Coleman, M., Taplin, P., & Estival, D. (2005). The FOCAL Point – Multimodal Dialogue with Virtual Geospatial Displays. *Proc. SimTecT*.
Wesley, R. E. (2011). An analytic data engine for visualization in tableau. *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data* (pp. 1185-1194). ACM.
White, D. D., Wutich, A. Y., Larson, K. L., & Lant, T. (2015). Water management decision makers' evaluations of uncertainty in a decision support system: the case of WaterSim in the Decision Theater. *Journal of Environmental Planning and Management*, 616-630.
Wiegand, L. K. (2013). Visualizing library statistics using Open Flash Chart 2 and Drupal. *Code4Lib Journal*, 1-11.
Wright, A. L., Zammuto, R. F., Liesch, P. W., Middleton, S., Hibbert, P., Burke, J., & Brazil, V. (2016). Evidence-based management in practice: opening up the decision process, decision-maker and contex. *British Journal of Management*, 161-178.
Yakovleva, N. O., & Yakovlev, E. V. (2014). Interactive teaching methods in contemporary higher education. *Pacific Science Review*, 75-80.
Yepes, J. C. (2013). Implementation of an Android based teleoperation application for controlling a KUKA-KR6 robot by using sensor fusion. *Health Care Exchanges (PAHCE)* (pp. 1-5). Medellin: IEEE.
Yost, B. H. (2007). Beyond visual acuity: the perceptual scalability of information visualizations for large displays. *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 101-110). ACM.
Zhang, K. (2012). Using visual languages in management. *Journal of Visual Languages & Computing*, 340-343.
Zhu, Y. (2012). Introducing google chart tools and google maps API in data visualization courses. *IEEE computer graphics and applications*, 6-9.

Zuk, T. S. (2006). Heuristics for information visualization evaluation. *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization* (pp. 1-6). Venice, Italy: ACM.

Appendix 1

Graphic User Interfaces Review

This section presents a summary of the approaches used for the generation of graphical user interfaces. In order to carry it out, the research and analysis of different works developed on different graphical interfaces in recent years were made. We include the timeline that allows us to graphically visualize how the evolution of the different approaches shown here has been carried out. Afterwards, we describe the work that the authors have developed for the construction and implementation of solutions. The solutions have a common objective, to make the process for the generation of graphical user interfaces more efficient. Within the reviewed approaches, we will mention the following: language-based tools, model-based approach, and interactive-graphical tools. This review provides a general description of the objectives they seek, the solution they pose, and the results obtained. We include a comparative table in which the solutions provided by the authors are enlisted and also allows us to analyze the advantages and disadvantages of each one concisely. Also included are our observations of the analysis performed and the areas of opportunity encountered.

Tools reviewed

In the next section, these different approaches will be discussed in detail and exemplified by relevant research. The developed tools will be reviewed in a timeline, as shown in Fig. 2. Note that in recent years, more tools that simplify the code generation have emerged. This visual representation clarifies the achievements to date and the interest in reducing the effort, time, and resources employed in the generation of GUI codes.

Language based tools

Qt.

Qt was pioneered in 1991 by Haavard Nord and Erik Chambe-Eng while working on a database application at Trolltech. Its first release was made available for Unix and Windows platforms in 1995 and was intended as a free license edition. Today, both commercial and free license versions are available. The Qt Project, with its latest release

QT 5.15.1, is a collaboration of many companies and individuals. The project follows a meritocratic governance model with many contributors, services, technologies, and community partners from around the globe.

The Qt framework integrates different technologies: QML for front-end development, JavaScript for enhancing user interface development, and C++Qt for back-end development. Among the diverse applications of Qt are console and desktop applications, widgets, Qt quick applications and connections to databases. The applications can be distributed to different platforms: Windows, Linux, MacOS, Android, iOS, and others. Qt is most commonly developed in C++ and QML, but other languages can be implemented through third-party bindings. Qt is freely available as a limited version, or fully available in paid license.

The following studies highlight the potential utility of the Qt framework.

Human–computer interaction (HCI) fields such as haptic applications, biomechanics, virtual reality, robotics, and rehabilitation have attracted increasing attention in recent years.

(Szelitzky , Aluţei , Chetran , & Mândru , 2011) created a glove that captures human hand movements by sensors and constructs a 3-dimensional (3D) model of the hand from the data transmitted to a software unit. Their proposed e-rehabilitation concept enables the therapist to supervise a patient’s rehabilitation exercises from a distant physical space. To avoid the expense of commercially available gloves, they fitted a regular glove with the required equipment to achieve the project goal. The software-unit integration was developed in the C++ language on the Qt platform, which allows easy integration of the software-unit elements. The unit comprises a USB communication unit that transmits the data from the glove to a PC, a user–PC interaction unit, and a control through which the human interacts with the 3D model. The GUI of the 3D model (buttons, menus, windows, and other interactive components) was constructed in the Qt modeling language QML and enriched by third party scripts compiled in languages such as Python. Data transmission was managed by a network communication unit. Despite some calibration problems with the glove and sensors, the movements of the glove were successfully reproduced in the 3D model. The lack of voice was considered as a deficit to be solved in future development. If successful, the proposed e-rehabilitation would remove the need for the patient to visit the rehabilitation center. Moreover, the e-rehabilitation was developed from low-cost materials and the software integration was consistent and easily achieved on the Qt platform.

In another application of Qt, (Sousa, Cordeiro, & de Lima Fihlo) developed an operational model that verifies systems correctness through pre and post-conditions and simulation features. The verification process starts with a C++ program created in Qt, which is then converted to an intermediate representation. Finally, an operational model written in C++ contains the structures needed for verifying the properties related to the Qt framework. This is supplemented with an ESBMC++ program that checks specific properties of the operational model related to arithmetic underflow and overflow, division by zero, out-of-bounds indices, pointer safety, dead-locks, and data races. The success rate of the operational model exceeded 94%. However, Qt proved insufficiently robust in this application, and required additional components.

SMIL.

Synchronized Multimedia Integration Language. SMIL (pronounced as smile) was developed by the World Wide Web consortium (W3C) in 1999. The latest version recommended by the W3C is the 2008 version SMIL 3.0. Similarly to HTML, SMIL is used for writing multimedia presentations. Multimedia elements include text, images, audio, video, and other objects. A SMIL document specifies the order, position, and the appearance time and duration of each element added to the document.

The most relevant characteristics of SMIL were described by (Hoschka, 1998). Synchronization enables users to add objects in a parallel or sequential way. Hyperlinks supplement objects of interest with additional information. If some elements are not supported by the user's device, SMIL provides additional information explaining why the object cannot be reproduced and what it was referred to. Layout defines the settings relative to position, size, color, and background. Finally, different objects can be included as different versions become available.

Multimedia presentations in SMIL alone can be very rich and dynamic. However, in 1999, (Rutledge, Hardman, & Ossenbruggen, 1999) explored whether different architectures can improve the versatility and capabilities of SMIL. REMDOR extends SMIL by incorporating a Quality of Service (QoS) through a partial-order/partial-reliability model. The Berlage architecture generates automatic SMIL presentations from a file specifying a media output. Some of the tools using SMIL-based multimedia presentations are the free and commercial versions of Real Player G2 and GriNS, and the free Java-based browser player SOJA.

We mention three SMIL applications that are available to users in diverse areas. First, infotainment applications provide interactive information in multimedia presentations. For instance, Amsterdam provides screen displays, timelines, and navigation links. Second, accessible multimedia such as Physics Interactive Video Tutor and web-based multimedia players provide information to users with visual or listening deficiencies. These applications provide text or audio descriptions in video presentations. Users can also access wider descriptions of the material while pausing the video. The third SMIL application is conceptual multimedia art. Through the interactive capabilities of SMIL, artists can visualize and play different media integrating images and sounds into virtual reality scenarios. The artist Maja Kusmanovic has accomplished this in GoTo. In a virtual reality scenario, users interact with a story developed by the artist, providing a dynamic, fully immersive experience.

XAML.

XAML (pronounced as zammel) stands for Extensible Application Markup Language (Microsoft, [MS-XAML-2009]: XAML Object Mapping Specification 2009 , 2014). XAML was created by Microsoft and first released in June of 2008. It represents structured information and is part of the Windows Presentation Foundation (WPF) that handles the visual presentation of windows-based applications and web browser-based client applications. User-interface documents are created from XAML elements such as controls, text, images, shapes, and animation. XAML comprises a schema information set, which defines a specific vocabulary, and a XAML information set, containing the information itself. The vocabulary or structure of a XAML document depends on the information carried in the document, so there is no generic vocabulary. The particular vocabulary is defined through the schema, which defines all the object types that carry the information, and whether the objects in an instance are mandatory or not. The information in the XAML information set (also known as an intance of the XAML document) is formatted or structured according to the schema.

An XAML document is structured as shown in Fig. 35 (Microsoft, Microsoft Developer Network, 2009)

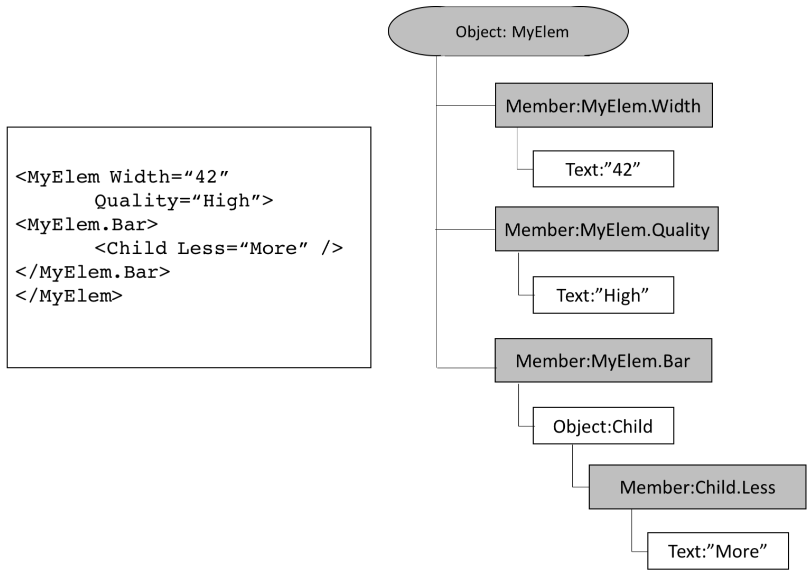

Fig. 35 Structure of an XAML document (Image taken from (Microsoft, Microsoft Developer Network, 2009)

Although released in 2010, the latest XAML specification continues to be used and integrated as the visual presentation in Microsoft products such as Silverlight.

When rebuilding or migrating an application into another technology, the application must usually be recoded in a different programming language. (Martínez-Ruiz, 2006) developed a XAML application to resolve this problem. They proposed the use of a model language, UsiXML, in which the schema definition allows the GUI of the application to be specified as a XAML document. This process abstracts the basic elements of the UI. The UI elements are technology independent, meaning that a button remains a button in Java, PHP, a web platform or a desktop application. In this way, all components related to the user interface are described within the scheme, translated into the XAML implementation, and then re-integrated into the new technology or language. Although this approach has not yet reached maturity, it is a very practical approach that conserves much time and effort by avoiding code regeneration of the entire app.

(Avazpour, 2015) created a proof of concept framework that eases the visualization of complex information, in CONcrete Visual assistEd Transformation (CONVErT), the user introduces an already designed drawing split into different views for representing different points of data. From each view, a new and different reusable visual notation is created, and the user selects the representation consistent with the visual notation. From this model, a view is created, and the created notations are saved in a repository. The view can be reused whenever the notations are mapped with the corresponding input, enabling multiple visualizations. Once the mapping is complete, the notations that generate the composition are saved in the repository, and the visualization file is generated. Along with the mapping transformations defined in the notations, the visualization files realize the visualization.

The input data in this model are displayed in a tree layout. The user can drag and drop the elements of the notations to later generate the model transformation defined by the notations.

Avazpour's solution allows users to experiment with interactive visualization. The notations are easily created by introducing the drawings, either in simple text SVG or with more complex XAML graphics.

The tool can generate model-to-view or model-to-model transformations and also define the interaction logics. Test cases demonstrated that the framework can generate reusable visual notations.

2.1.4 Layouts.

Layouts (Android). The Android operative system was created in 2003 by a group of developers working on a Linux based operative system for mobile platforms. The company was bought by Google Inc. in 2005. The first version of Android, known as Android 1.0, was launched in September 2008. Since then, the Android operative system has gained a significant portion of the smartphone market; consequently, a large community of developers is interested in creating apps for this platform. The latest release, Android 11, was launched in September of 2020.

Mobile applications for the Android operative system are developed in the Java programming language through Android SDK. The UI Android defines the visual structure using Layouts, which define the activities or widgets within the application. Layouts can be declared by means of an XML file or dynamically during the runtime. In dynamic mode, the initial state of the application UI is composed using an XML file, and the elements are manipulated throughout the execution time. Even elements declared in the XML file can be modified. By declaring graphical objects in the XML file, one can separate the presentation from the application. The layout-related elements allow the definitions of size and position. As the view of an application can change during its execution, many child layouts with independent characteristics can be created, as shown in Figure 36 (Google, 2017).

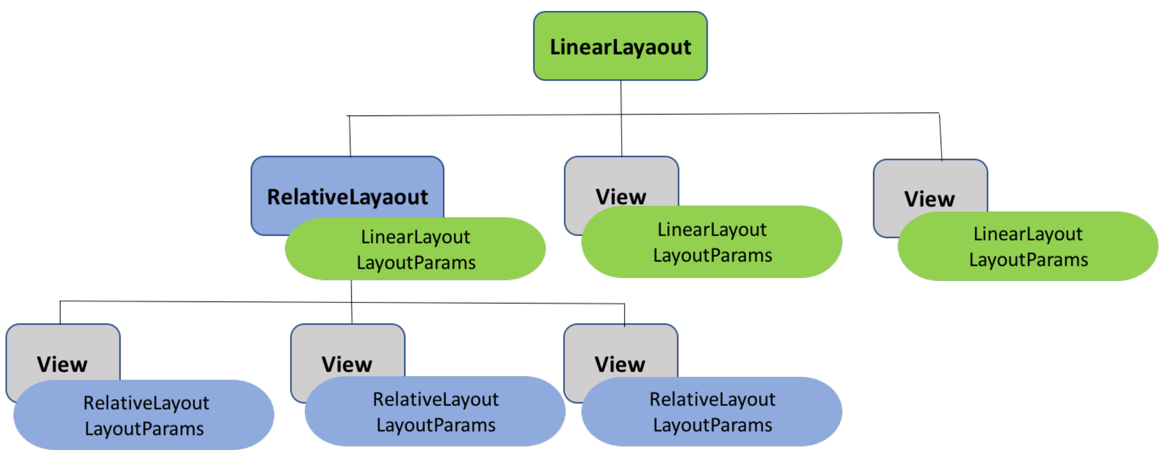

Fig. 36 Visualization of a layout hierarchy, showing the parameters associated with each view. (Image taken from (Google, 2017))

The use of layouts is exemplified in two works related to Android development.

As an open-source operative system, Android has rapidly increased its market share owing to its wide applicability, stability, and speed. Using an Android platform, (Yepes, 2013) manipulated an industrial robotic arm, KUKA KR6 for use in the telerobotics field. Teleoperated systems aided by mobile devices have steadily grown in recent years, especially in biomedical applications for patient rehabilitation, management of biological hazardous materials, and medication storage. Yepes' proposal is based on a master–

slave system whereby the user controls the robotic arm through a tablet or smartphone. The master system, created in an Android application, sends the orders captured by the touch screen to the slave system, which executes the movements of the robotic arm. The GUI of the master system consists of a series of graphical elements, enabling easy manipulation of the basic robotic-arm movements.

Yepes' project demonstrated the effectiveness of combined mobile devices and Android-based systems in the telerobotics field. Users assessed the system as an easy and accurate way of moving a robotic arm. The concept is potentially applicable to more specific tasks in telemedicine.

As another application, (Cacho, 2015) created an Android app for smart tourism destinations. In 2014, Natal City in Brazil hosted the FIFA World Cup. To prevent the problems related to transportation, safety, and water consumption incurred by numerous tourists visiting the city, smart technology can propose alternatives that enhance the competitiveness of destinations. Such process automation transformed Natal into a Smart Tourist Destination. Cacho's application integrated three components: a mobile tourist guide, a tourist information system, and a business intelligence infrastructure. The mobile tourist guide is an Android-based app through which tourists can consult different urban attractions, find related information, and seek arrival directions in a map. When tourists use the mobile app, their information is fed to the tourist information system available in a cloud infrastructure. Finally, the business intelligence system gathers the accumulated information and disseminates it to policy makers. By consulting the information through graphics or maps, policy makers can better understand the behavior of tourists, and affirm their affluence, most frequently visited locations, time spent at each location, and trajectories. The developed app provides tourists with an interface that improves their experience in the city.

Storyboard (iOS).

Storyboard (iOS). iOS is a mobile platform created by Apple Inc. When first released in June of 2007, iOS was intended only for iPhone users, but later versions are compatible with other Apple devices such as iPad and iPod Touch. Thus far, the company has released 10 versions of this operating system, the latest (v. 14.1) in October of 2020. Each version improves the product and fixes the bugs present in previous versions. Applications of this exclusive operating system are developed on SDK, in a customized language called Objective-C. However, the GUIs of iOS mobile applications are developed on Storyboards. The application is visually represented through a ViewController. As the views corresponding to different scenes are connected by objects, one can define the transitions between each view. Storyboards can be created and managed through Xcode, which provides a visual editor on which all elements (such as buttons, table views, text views) that compose the Storyboard can be managed and added. Storyboards allow users to visualize the UI of an iOS application in a graphical way, as it would finally appear in the device. In 2011, Microsoft released its Xamarin product for creating Storyboard iOS applications. Storyboards can be created manually or through the abovementioned Xcode or Xamarin designers.

Below we review two applications of iOS in the education field. In these studies, students learned how to create their own apps using the platform.

In an experimental course of developing iOS applications at California State University, Stanislaus, (Thomas M. , 2014) adopted the flipped (inverted) classroom strategy. Their students used on-line resources only, avoiding the need for buying material. A main goal was to reduce the instructor's workload.

In the flipped classroom model, the students took home lectures and reading materials as homework. Next, day, in the classroom, they were invited to ask specific questions or discuss specific concerns related to the lectures. As the students did not own iOS devices, they developed their apps and executed the simulators in the laboratory, where Xcode was installed on MacOS computers. In the final weeks, the students worked on a project either individually or with a partner. Before developing their apps, students were asked to deliver a design document to coordinate their activities outside the classroom.

At the end of the course, students reported their experiences of the course modality in a survey. Most of the students responded with positive comments, reporting that the learning experience was dynamic, and the laboratory practices were entertaining. A few students disagreed; among the most common complaints was the need to physically attend the university due to the technical/software requirements. Finally, the instructors reported no reduction in their workload.

This work demonstrates the ease with which students can adapt to and learn the platform for developing iOS apps, given the right tools. However, the requirement of MacOS and iOS devices for the app development and testing proved a great disadvantage for students, requiring them to spend more time at university to achieve the goals defined in their developed projects.

Table 18 summarizes the main features, advantages and disadvantages of the language-based tools reviewed above.

Table 18 Advantages and disadvantages of language-based tools

Language	Description	Advantages/Disadvantages
Qt	Framework that integrates different technologies: QML, JavaScript, C++QT. Diverse applications can be developed and exported to different platforms. Provides a complete set of tools for the creation of different applications. Applications can be developed in a native language and supplemented with third-party languages.	Multiplatform framework uses C++ extensions that facilitate the development for people already familiar with the language. Supports many compilers such as Visual Studio. Free and licensed versions are available.
SMIL	W3C format for multimedia on the Web. SMIL is an XML-based language for the specification of multimedia presentations. SMIL covers all aspects of an interactive multimedia document, including the spatial layout, the temporal synchronization, and hypermedia links.	Based on XML, easy to understand and implement. Time dynamics can be considered in the project development.
XAML	XAML (short for Extensible Application Markup Language) is a markup language used to represent structured information. The structure or vocabulary depends on the represented information and is defined in a schema.	Being based on XML, XAML is easy to implement. Used mainly on Windows platforms, but portable to some other platforms.
Layaout(android)	Element that represents the UI design as an activity, fragment or widget. Establishes a visual representation that facilitates the communication between users and the application interface.	Most mobile devices use the Android operative system. SDK is freely available.
Storyboard(iOS)	Visually represents the UI of an iOS application. The Storyboard is composed of a sequence of scenes, each scene representing a view controller and its views. The scenes are connected by Segue objects that perform the transitions between two view controllers.	The development language is object-oriented and based on C++. Not difficult to learn, but limited to Mac users, which introduces some difficulty.

(Slováček, 2009) separated the implemented logic from the client side and concentrated it at the server side. This avoids the need for developers to modify the UI when implementing the logic and when using scripts to improve the interactivity and attractiveness of the GUI. Defining the application logic at the server-side improves and simplifies the server maintenance and the implementation of future changes.

As an example, Figure 37 shows the architecture of a UI protocol.

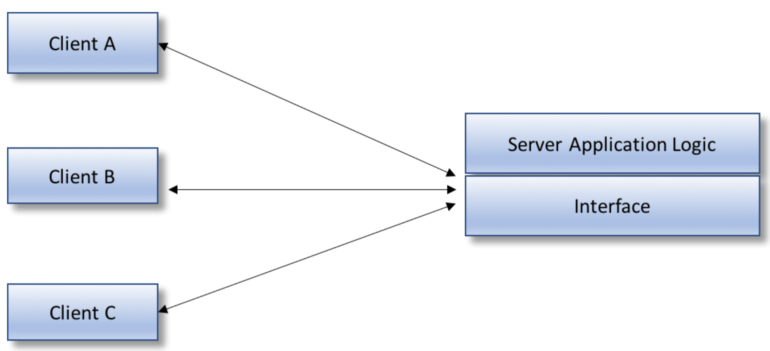

Fig. 37 Client–server architecture of UI protocol (Image taken from (Slováček, 2009))

This solution allows the use of scripting languages to make practical and fast changes at the server side without recompiling the code. Moreover, a link allows the developer to deploy without knowing how they are represented in the interface, enabling dynamic behavior. All changes are either updated or automatically reflected in the interface of the application. The server-centric logic minimizes the amount of information exchanged between the GUI designer and the application logic developer, simplifying the long-term maintenance and scalability of the application.

A standard client–server communication protocol based on XML can manage several interacting applications on the same server. This allows UI developers and designers to focus on their area of expertise (logic or interface development) rather than entering an area outside of their specialty.

(Macik M. K., 2011) recognized the difficulties of data visualization and interface creation on the plethora of existing devices. To resolve this problem, they generated abstract contextual models that integrate user models, devices and the environment. This abstract model is converted to a specific UI model by a user interface generator (UIGE generator). The UI model is sent to the UIP server, which replicates the generated information and sends it to each of the UIP clients, which deploy the information according to the characteristics of each device. Figure 38 schematizes the implementation of this process on a platform.

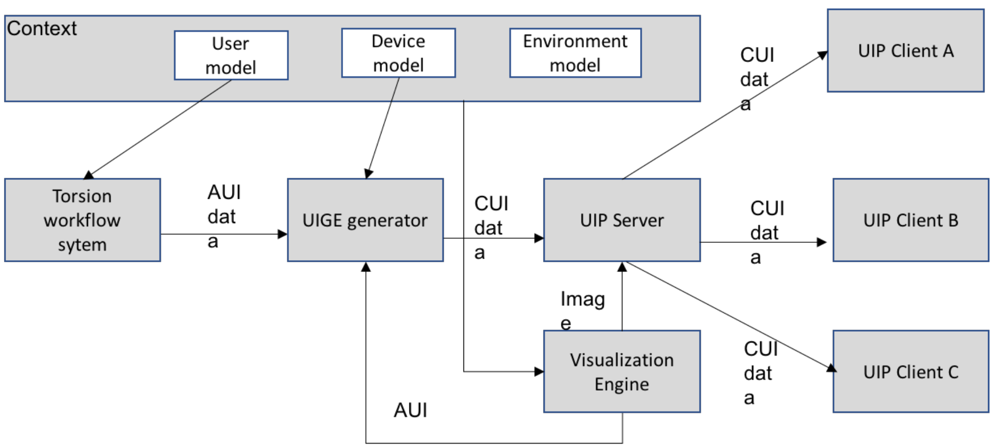

Fig. 38 Propagation of information through a platform (image taken from Macik M. K., 2011)

The information processing leads to one of the following outcomes:

- Success. All AUI elements, regardless of their importance, are represented in the resulting CUI.
- Partial success. All critical UAI elements (elements that cannot be omitted) are represented in the resulting CUI.
- Failure. Some or none of the critically important AUI elements can be represented in the CUI.

If the UI generation is at least partially successful, the generated UI is transferred to the target UI platform via the UIP and processed. If the UI generation is wholly successful, the display allows a variable data representation. To maintain a universal focus of the solution, the visualization engine is integrated with the interaction in a context-independent way.

The functionality of automatically generating a UI with data visualization was demonstrated in a simple case study. In the near future, the authors will assess users' perceptions of the resulting data presentation. A module that rescales the resulting CUI to fully utilize the available space would extend the UIGE implementation.

Adopting a model-based approach (Krunic, 2013) generated a system that allows the automatic generation of a GUI given the product specifications. This system is designed in two stages: automatic model development (AMD) and automatic source-code generation (ASCG). The AMD stage processes the initial specification (which is stored in a CSV file) and transforms it into an XML data model. In the ASCG stage, the GUI generation code is composed in Java and carried out by an ASCG. Finally, consistency between the generated controls and the original data model emerging from the CSV file is checked to ensure the quality of the generated code. The great disadvantage of this approach is that both AMD and ASCG work exclusively with special function registers, which limits their implementation. One context-model approach automates the UI generation (Macik M. C., 2014). The cost and effort of UI generation grow with increasing numbers of required characteristics and supported users. Overall, developing applications for different devices is a very challenging task, as the applications must be natively run on each device, which also increases the development costs. The platform proposed by Macik promises to resolve these problems. Based on the client–server architecture, it processes the data separately from the user interfaces. The data structures are contained in data models that are ultimately transferred to customers on demand. The data models remain consistent as the updates are propagated to the application back-end, where they are automatically distributed to the clients. This solution supports desktop computers, tablets, smartphones, Smart TV emulators and webs. The solution is presented as a block diagram in Figure 39.

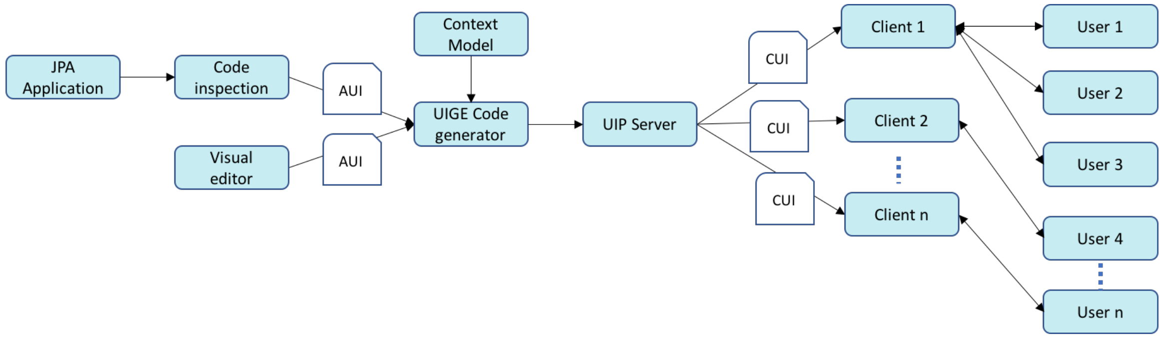

Fig. 39 Platform of automated UI generation(Image taken from (Macik M. C., 2014))

After implementing this platform, the resulting automation allows correlation between the data model and the adaptive UI. Any necessary changes are applied without requiring manual changes in the data model. Additionally, AUIs and CUIs are designed by a manual editor. The approach is applied in two contexts: application and interaction. The platform requires only half the code required of manual development, which is beneficial from a development and maintenance perspective.

Another approach is the one presented in (Castrejón, 2011), that through the development of model-driven software (MDSD) that proposes using model abstraction to carry out the implementation, rather than carry it out in detail. They applied model-driven technologies (MDA) that specify the system functionality and are independent of the platform, EMF, ATL, and Spring Roo. These tools also optimize (as far as possible) the technological changes and adaptations expected in future developments.

The proposed solution, Model2Roo, is intended to develop web applications by transforming models constructed from meta-models in Spring Roo scripts, and using tools associated with the EMF framework. Model2Roo enables developers to ignore the implementation details and focus on the characteristics of the application modeling domain to be developed. Model2Roo can be incorporated into the Eclipse platform using a plug-in. In this implementation, Model2Roo operations become part of the EMF framework, conferring large utility benefits to users. This solution maintains integrity between the data model and the general application, reduces the development time and maintenance, and reduces the number of lines of generated code. However, its use is limited almost exclusively to the development of web applications. Extending its usability to other types of applications would be an interesting future improvement.

Noting the increasing popularity of developing web applications by the rapid application development methodology, (López-Landa R.A., 2012) proposed a model-driven engineering architecture named EMF on Rails for developing web applications and related families. In MDE, the applications are abstracted such that developers are provided with domain-specific languages, allowing the generation of models that will construct the application. The architecture of Lopez-Landa et al. combines existing frameworks for web-application development (such as Spring Roo, Ruby on Rails, and Django) with the MDE. The proposed architecture embraces model-based construction of code generators for families of web applications, and the construction of web applications from domain-specific models by these generators.
To automate the customized environments and generators of web application families, EMF on Rails integrates the Spring Roo project and the Eclipse Modeling Framework. Later, these generators produce web applications of a specific domain. All processes begin from a defined meta-model or DSL, which will determine the final web application. After defining an add-on model that describes the domain-specific functionality, a skeleton set of Java classes is created by an ATL transformation and a code generator. The developer needs to complete this set with a specific functionality. A repository, to be created in future work, would enable reuse of the classes and easy generation of families of web applications. Finally, to create a web application, the designer needs only to describe the application from a model definition (in practice, a meta-model). A Roo script is then generated from the add-on annotations and the final web application is generated in Spring Roo. In evaluation experiments, EMF on rails conserved time and effort and did not require a system expert. Unlike other proposed solutions, EMF on rails requires effort and expertise only when defining the models and constructing the web application from the generator. Once these tasks are complete, single web applications are very easily automated, saving not only time and effort, but also cost.

Table 19 summarizes the main features, advantages and disadvantages of the model-based tools reviewed in this section.

Table 19 Advantages and disadvantages of model-based tools

Authors	Description	Advantages/Disadvantages
Myers	Through the UI administration system, a UI can be generated by different tools. Tools are classified into language tools, model tools, application frameworks, and direct graphic generators.	Most of the mentioned tools are already deprecated, but their tool classification system is still used today.
Kennard, Leaney	Developed a five-layer architecture platform covering the important characteristics (in their view) of a UI generator.	Their process provides no definition for stopping the initial inspection. The process reduces the number of lines of code but not necessarily the development time.
Krunic, Letvencuk, Povazan, Krunic	The specifications are read from a CSV file, and are later transformed to an XML data model from which the source code is generated	Limitations are imposed by the XML data model and the ASCG used for data generation
Macik, Cerny, Slavik	Their platform separates the UI from the data model and achieves aspect-oriented transformation of AUIs using AF.	Advantages are the multiplatform implementation and code reduction. The development time is not specified.
Castrejon, Lopez-Landa, Lozano	Scripts generated by the model definition are used to produce Java code for web applications.	Compatible with Java only
López-Landa, Noguez, Guerra, de Lara	Integrates the Spring Roo project and Eclipse Modeling Framework to create web applications and related families. Generates models by abstracting applications.	Automation of applications from the model definition. Limited to Java programming.
Macik, Klima, Slavik	Transforms an abstract model into a UI concrete model. The UI is generated and visualized in different devices.	The studied case is very simple. A more complex case would be more interesting.

Slováček, Macík, Klíma	A protocol based on a client–server standard communication XML. The implementation is separated from the client and focused in the server	Event managers can be developed in different script languages. The event managers can perform changes on the server side without the need for recompiling the code

Interactive Graphical Specification Tools

Visual Studio.

Created by Microsoft and first released in 1998 as Visual Studio 6, Microsoft Visual Studio IDE is among the most popular interactive graphical specification tools. In its 2002 release, it evolved into Visual Studio.NET, with important changes in the compilation of applications and support for 64-bit architectures. The latest release is Visual Studio 16.8.3 made in October 2020, on which users can develop Android, iOS, Mac, Widows, Web and Cloud applications. Visual Studio is a complete set of tools for developing a variety of web, console, desktop, web services, and mobile applications in a variety of languages such as visual basic, visual C++, and visual C#. It has a code editor, a debugger, and a designer. Visual Studio 2017 is a highly versatile tool, as its IDE works on the same framework in any language, enabling mixed-language solutions. It is available to the community for free or as paid editions such as Professional and Enterprise. It is also available for both Windows and MacOS.

To show the dynamics of Visual Studio, we review (El-laithy, 2012), who integrated Microsoft X-box Kinect in robotics applications. Along with various sensors, they incorporated Microsoft X-box Kinect's depth camera as part of an autonomous ground vehicle that undertakes different tasks. Thus far, the vehicle can navigate in open areas with the aid of a global positioning system (GPS) and laser scanner sensors. The depth camera detects obstacles to be avoided in an indoor room, where GPS is not available. Linux and Google both offer open-source development kits, but Microsoft has released its own kit and rendered it available through Visual Studio 2010, providing assemblies and libraries for robotics developers. (El-laithy, 2012) tested several algorithms and selected the one giving the best results in indoor and outdoor static-object detection, indoor and outdoor robotic runs, and indoor-and-outdoor skeletal tracking comparison. One disadvantage of outdoor application is the performance degradation of the depth camera under excess light, but high contrast is very useful during the nighttime. In contrast, the indoor navigation performance was highly efficient. Some improvements are required to achieve the project goals. Fortunately, more open-source platforms are becoming available for this type of work, such as Microsoft's Visual Studio 2010 Express and new releases for Windows.

(Duru, Çakır, & & İşler, 2013) proposed software visualization as a tool for software comprehension. Given the high inherent costs of software maintenance (a considerable percentage of this cost is invested merely in understanding the source code), software visualization is considered to reduce the effort of code analysis. Software visualization tools provide various comprehension strategies and help to create visual representations that allow users to review and analyze large amounts of information. Besides facilitating the analysis process, these tools lower the operation costs.

However, the contribution of software visualization to software comprehension is unclear. How software visualization is related to software comprehension, and how it assists programmers to understand the source code when modifying or correcting software, are not properly understood. Eye-tracking technology might assist by showing the fixation sequences and scan paths while completing a programming task. To test their idea, (Duru, Çakır, & & İşler, 2013) used a web-based application developed in the Microsoft.Net framework. Programmers (13 software engineers) performed seven application-related tasks, either with or without the software visualization. The visualization aids were the NDEPEND software visualization tool and the Tobii T1750 eye-tracking system. The development environment was Visual Studio.NET. Qualitative and quantitative analysis were performed by the ground theory methodology. The group using NDEPEND showed significant improvements in understanding the source code, and successfully completed the web-application tasks, whereas the control group achieved fewer of the tasks. Although visualizations can be a valuable resource, enabling quick and improved understanding of the source code during software modification tasks, it does not fully clarify the code. Ongoing research will better support the application of software visualization tools to software maintenance.

Eclipse.

Another popular tool is Eclipse, an open-source project in which anybody can collaborate to build a development platform. Created by IBM in 2001, it was followed three years later by the Eclipse Foundation, a non-profit organization that facilitates an open community working on a common Eclipse project. More than just a developing platform, Eclipse provides extensive frameworks, tools, and runtimes for building, deploying, and managing software projects. The latest version is Eclipse 4.17 , released on September of 2020. This open-source, license-free IDE is commonly used for developing Java applications, although plug-ins enable access to other features and applications created in other programming languages, such as JavaScript, PHP, Python, Ruby, and Scheme. To exemplify the use of Eclipse IDE, we discuss two related works in the following paragraphs.

In a collaborative effort involving six universities in Italy (Maresca, Stanganelli , & Coccoli, 2011), Eclipse and the Jazz framework were enhanced by integrating IBM's Rational Asset Manager. The project itself comprised eight subprojects and four supporting projects, and was undertaken by 44 undergraduate students, three graduate students, one PhD candidate and two teachers. The initial goal was to develop an application for managing the annual conference of the Italian Eclipse Community. The main objectives were to rule the software development process, render it an observable process, and to identify suitable software metrics for evaluation purposes. By involving students from different universities across Italy, the project facilitated the teaching of students and engaged them in the software development process.

Another objective of this project is to collect, organize, and share the assets that emerge while the students develop and work on their assigned projects, promoting the reuse of code and thus saving time, effort, and risks. Collaborative work is suitable for large organizations or public institutions, in which geographically separated individuals must sometimes share information and results. In a collaborative environment, workers are not restricted to the same place at the same time. In the Italian scenario, the IDE of Eclipse successfully managed all tasks among all teams.

Eclipse was also employed by (Abeywickrama , Hoch , & Zambonelli , 2014), who introduced the SOTA model for adaptive systems. The SOTA framework analyzes self-awareness and self-adaptation in adaptive systems. It reaches goals by self-adaptive patterns and provides feedback loops for improved goal achievement. To satisfy the demand for high-standard software engineering tools and methods, they developed a plug-in for Eclipse named sim-SOTA. The simSOTA plug-in, which can be added to Eclipse through Eclipse's update site, allows developers to architect, engineer and implement self-adaptive systems based on the feedback loop proposed by the authors. (Abeywickrama , Hoch , & Zambonelli , 2014) validated their proposal in an e-mobility electric-car case study. Their solution produces both domain-independent and domain-specific model patterns, which are translated into Java code by model-to-model and model-to text transformations aided by JET (provided by IBM). In this way, the behavior of the case study could be monitored and the SOTA patterns could be validated. In other case studies, the provided patterns proved their usefulness, and the plug-in from Eclipse made the work easier.

Dreamweaver

Dreamweaver was launched in 1997 and developed by Macromedia until its acquisition by Adobe Systems. The latest version is Dreamweaver 20.2, released in June of 2020. This web development tool was created by Adobe Systems for both MacOS and Windows operating systems. It combines a graphic and a code editor, and a control system for web developers. It also supports a variety of languages, such as ActionScript, C#, CSS, ASP, XML, Java. A trial version is available, but the complete version with all features has a monthly cost.

(Park & Wiedenbeck, 2010) conducted a Dreamweaver web development study with novice/informal developers, who were interested in creating web content but were not related to or familiarized with the field. After taking a web development course, the participants developed two web applications in two different scenarios. In the first scenario, the code for the website development was required to be manually inserted in a text editor. For the second website, the students were asked to select a web development tool (all participants selected Dreamweaver). In the first scenario, all participants developed their solution, although some participants struggled with formatting problems. The participants were able to reuse some code or simplify it using Cascading Style Sheets. However, the Dreamweaver results were not as expected. Although the tool was supposed to simplify the work by visually displaying the incorporation of graphic components, the generated code was unclear to the students. Consequently, some of the participants preferred the text editor and others used the tool for input content but modified the elements manually to avoid the automatic generation of obscure code. Although Dreamweaver was not the most suitable tool for novice web designers, exploring different tools might stimulate their interest and encourage them to grow their development skills.

Websphere.

Websphere is an application infrastructure that provides the necessary features for developing, hosting, maintaining, and disseminating services and applications. It was released by IBM in 1998 for creating applications and integrating them with other applications. The latest version is 9.0, released in June of 2016 with a latest fix pack released in April 2019. Developers, administrators, business planners and other technical professionals can apply various Websphere tools to integrate different solutions. Websphere can run on different platforms such as Solaris, Linux, Unix, Windows, AIX, and OS. To facilitate application development, Websphere is based on open industry standards such as J2EE and Eclipse. Some of the tools offered by Websphere are Rational Application Developer, Websphere Studio Device Developer, and Websphere Integration Developer. The Websphere website provides not only the tools and guidelines for creating applications, but also the documentation, tutorials and updates on information related to the Webshpere infrastructure. It also provides a free download for product evaluation.

(Lee & Wang , 2013) provided a web-based system that integrates e-procurement, e-contracting, and e-invoice platforms, enabling efficient information sharing among the members of a supply chain. The platform integration is intended for the business-to-business (B2B) market. Given the increasing frequency of information exchange and the reduced cost of electronic documents over paper invoices, an e-marketplace based on a computer network can be an effective trading platform. In particular, companies and clients need a common space to meet and exchange their products and services. To implement their technological solution, Lee and Wang investigated Active Server Pages (ASP), Hypertext Preprocessor (PHP), and Java Server Pages (JSP). Although all of these technologies are competitive and offer a variety of good features, Lee and Wang ultimately developed their system in JSP, a Java-based scripting-server side language that creates dynamic web pages. JSP was more promising than ASP and PHP, being both an open-source technology and adopted by leading companies such as IBM. The presentation layer constructed in JSP, along with JavaBean/EJB for the web-system, follows a three-tier architecture that runs in a Websphere Application Server, using Oracle as the database. This mix of technologies achieves a multiplatform solution that can run under Solaris, Windows, and Unix systems. The created B2B e-MarketPlace facilitated the exchange of information between suppliers and buyers, providing significant benefits to both parties.

XCode.

Xcode was introduced in October of 2003 as part of MacOS version 10.3. Originally created for developing macOS applications, the 2007 release included the new Objective-C language, which enables the development of iOS applications. Xcode is an IDE created and distributed by Apple. Its various tools allow the development of applications for different Apple products: macOS, iOS, watchOS, and tvOS. Using either Swift or Objective-C as the main programming languages, users can manipulate objects and elements indirectly by coding or directly in the graphical editor. In the latest version, 12.1.1 made in October 2020, available Xcode provides easy integration for source control to github account. Xcode can be freely downloaded from the itunes appstore, but only for macOS operative system.

In a course on iOS application development taught at Iona College (New York, USA) (Ivanov, 2011), students developed an iPhone application in Xcode. Throughout the course, the students were expected to learn problem solving and software design principles, integration of DB design, networking, computer graphics elements, and the Objective-C programming language. Using the knowledge acquired in their lessons, the students developed a scientific calculator app. Later they developed a game requiring their newly acquired knowledge of single- and multi-touch handling, animation and NSTimer. They also learned how to integrate sound with video recordings. In more challenging projects, they applied accelerometers, location services, and MapKit. Two disadvantages of iOS development are the limited application memory and multi-threading support. Despite these limitations, the students were able to learn and use the iPhone capabilities and develop full-featured applications with the tools provided by the Xcode IDE.

The main features, advantages, and disadvantages of the above-reviewed IDEs are listed in Table 20.

Table 20 Advantages and disadvantages of interactive graphical specification tools

IDE	Description	Advantages/Disadvantages
Visual Studio	Used for developing computer programs for Microsoft Windows, websites, web apps, web services, and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store, and Microsoft Silverlight. It can produce both native code and managed code.	The VS licenses are very expensive, as the proprietary is Microsoft. VS can be used in various OS such as Windows and Mac. A lighter free version is available but lacks some of the features.
Dreamweaver	Adobe Dreamweaver is a web design and development tool that combines a visual design surface known as Live View with a code editor with standard features. It is available for macOS and Windows. Combined with an array of site management tools, Dreamweaver allows the design, coding, and management of websites. Mobile content can be encoded in different languages: ActionScript, C#, ASP, XML, Java, etc.	Available for the most widely used operating systems (Windows and Mac OS). Additional support for many scripting languages. The complete version is licensed.
Websphere	Websphere is an application infrastructure that use open standards such as Java EE, XML, and Web Services. It is supported on the following platforms: Windows, AIX, Linux, Solaris, IBM I, and z/OS. The open standard specifications are aligned and common across all platforms. Platform exploitation is done below the open standard specification line.	A complete suite that provides different solutions and services. License required.

Eclipse	Most widely used Java IDE. Eclipse is written in Java and is primarily used for developing Java applications, but applications in other programming languages are enabled by installing plug-ins. Allowed languages are C, C++, JavaScript, Python, R, Ruby, Scala, Scheme.	This Java-based IDE, is widely used. Its SDK is free and open source. Not limited to Java development.
Xcode	An integrated development environment for macOS containing a suite of software development tools developed by Apple. Software can be developed for macOS, iOS, watchOS, and tvOS. Supports source code for C, C++, Objective-C, Objective-C++, Java, AppleScript, Python, Ruby, ResEdit (Rez), and Swift, with a variety of programming models, including but not limited to Cocoa, Carbon, and Java. Third parties support GNU Pascal, Free Pascal, Ada, C#, Perl, and D.	Limited to MacOS. License free

Table 21 SWOT analysis of the approaches reviewed in graphic user interfaces tools

	S	W	O	T
Language-based	Consistency between functionality and graphical representation. Alleviates the difficulty of changing and modifying the code	Steep learning curve. Specialized programmers needed.	Powerful tools; give the programmer total control of the code, automation is possible.	The weaknesses drive some programmers toward better graphical tools
Model Based	Automatic generation from a model definition.	Because a higher specification is required, a specialized programmer is needed	As there is no unique model, most of the proposed solutions, systems, protocols, and platforms are developed ad-hoc to suit the user's needs	Modifications can be done in code. Consistency between application functionality and GUI can be compromised
Interactive graphical specification tools	Provide a graphic-friendly interface that is tractable to novices. Allows the integration of static and dynamic objects that can change their behavior during the application´s execution.	Some tools require significant hard disk space for installation. Some are free, but some licensed products are very expensive	The tools enable users to create cross-platform robust solutions by integrating many technologies	Increased licensing costs, require continuous upgrades

The SWOT analysis (see Table 21), identifies the advantages and disadvantages of each approach presented.

Language-based tools are powerful and complete. As the code is not easily manipulated, the application's functionality tends to be consistent with the graphical representation. The languages reviewed in this paper are license-free and installable with minimum requirements. However, when a new language is introduced, the steep learning curve presents a barrier to programmers. Moreover, a language-based solution can be constructed only by specialized programmers.

In model-based tools, the generated model should allow automatic creation of the GUI. One disadvantage is the lack of a unique model, so each model must be created to meet the requirements of the application. Again, the solution can be created only by specialized programmers. Moreover, when the modifications are directly programmed in the code, the consistency among the application functionality, GUI and model is easily disrupted.

In contrast, the GUI provided by interactive graphical specification tools is easily applied by novice or non-specialist users. Besides being friendly and intuitive, interactive graphical specification tools provide an editor that modifies the elements with code. The objects' behaviors can change during the application execution, conferring a dynamic property. However, the installation requires certain technical features such as frameworks or SDKs. The rich GI and other integration tools also demand significant hard disk space. Interactive-graphical tools are usually as license free or fully licensed versions. The limited functionality of license-free tools urges users to eventually upgrade to the fully licensed modality, which is generally expensive; moreover, the updates are frequent and sometimes require simultaneous application updates.

Besides the three approaches reviewed here, we should consider other ways of displaying and analyzing information. For instance, (Sopan, 2013) visualized information in tables whose columns contain the distribution information. Traditionally, the information in each column represents a different range of values. They applied ManyNets, which provides distribution columns to facilitate the display of information. However, as this tool itself is insufficient for correlating, sorting, and filtering the distribution information, they extended the ManyNets tool functionalities for better handling of distributions. In their solution, users can correlate the information in different columns, enabling a deep and complex analysis of their data. Even with the extended functionality, the authors found deficiencies in navigating among the information and the processing speed. Here, we found that a single tool cannot usually achieve a set of goals and must be enhanced or adjusted to a user's needs or a specific IT evolution. ZOIL, an evolutionary tool developed by (Jetter, Zöllner, Gerken, & Reiterer, 2012), is a post-wimped distributed interface that not only displays information, but also enables interaction across different devices and components such as displays, tablets, and platforms. Within the interactive spaces of ZOIL, which supports all interactive devices, users can share, discuss and update information in real time. This tool exemplifies the development of tools alongside the changes in technology. Although our developments must consider the best existing practices, we also should be aware of the potential changing needs of our tools and ensure that our developments can acclimatize to technological changes.

Appendix 2

Information Visualization

This section presents a review of tools used for information visualization. We start with a timeline to graphically visualize the evolution of the different tools. The review of tools will be presented according to the aforementioned classification of software libraries or APIs, custom and commercial software. We will describe each tool based on the work of different authors who either implement an existing one or developed their own to create a specific solution. We include a general description of the objectives they seek, the solution they pose, and the results obtained. We also include a comparative table in which the solutions provided by the authors are enlisted and also allows us to analyze the advantages and disadvantages of each one concisely. Also included are our observations of the analysis performed and the areas of opportunity encountered.

Information visualization tools reviewed

In the Fig. 13 a timeline is shown with the tools reviewed and are used to generate visualizations.

We will make a review on each of these tools. The review also will classify each of the tools in any of three categories that we consider according to the features they have. These three categories are (1) software libraries or APIs, (2) custom software, and (3) commercial software.

Software Libraries/APIs

Software libraries and APIs (Application Program Interface) are used to simplify the programming of an application. They provide functions, routines, and methods to be implemented in particular software development for a specific behavior or functionality. The functionality provided covers everyday needs demanded for the development of software applications. They are not specific to a particular language and are available for use and implementations in various computational languages. We present examples that can be implemented in different languages such as *Java, JavaScript* and others.

Java TreeView

Some of them are of specialized field as the *Java Treeview* which its main objective is to be a tool for discovering patterns in genome-wide datasets (Saldanha, 2004). Adapted from the original *Windows Treeview*, it provides three different visualizations that can be explored and used in the website ((http://jtreeview.sourceforge.net)) which provides examples and tutorials on how it can be implemented. It is written in *Java* and it allows to add different types of data as well as views related to the data. The fact that it has been developed with Java, makes The *Java TreeView* library accessible in a variety of platforms such as *Linux*, *Windows* and *MacOS*. It provides a custom and ready to use functionality but also it can be extended. The generated views can be exported to different image formats and data to a tab-delimited text file for further analysis. Specialized tools can make us think that it is difficult to use and for sure it cannot be used by us if we work in a different field other than specialized IT. Fortunately, there exist some other tools that are available for different areas of expertise and they are easy and ready to be used.

Google Charts

Google Charts is a free cross-browser tool available on web and provided by *Google* ™. It has different charts available that can be created from data collected from one or different data sources. Charts created with this tool can be embedded in web pages because it is based on *HTML5* and *SVG* which make it simple and easy to display in various browsers without the need of installing plug-ins. Some other features that *Google Charts* presents are the interactivity of their charts, cause when the user clicks on one entity more detailed information is presented. Also, category pickers, range sliders or auto completers can be used as controls within the charts to create interactive dashboards. Related to data, this can be read in two different forms, raw data and by Data Table that can easily allow the use of different charts and dashboards along with different functions to interact with data such as sorting, modifying, finding and filtering. As being a google resource it can be connected to other different google products such as *Google Spreadsheets*, and Fusion Tables (Lee S. J., 2015). The tool has been reviewed as easy to use, with clear documentation, with available forums for related questions, having brand-recognition, able to be embedded in projects with java or *JavaScript*, database and web services support (Zhu, 2012). In Figure 40 it can be seen a view of the *Google Chart* tool, available in its web site, https://developers.google.com/chart/.

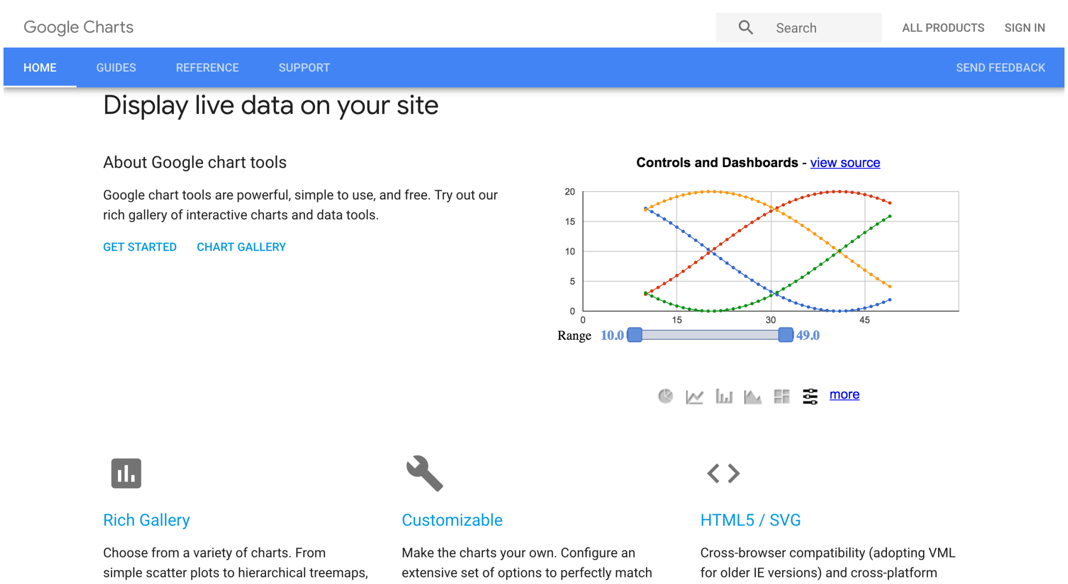

Fig. 40. Image taken from google chart site, https://developers.google.com/chart/, visited on October 2018.

OFC

Another free tool available on web is the *Open Flash Chart* (*OFC*). It is a simple easy tool to use that has up to twelve chart types. As some other tools it uses SVG image format, to make the charts as images available for interactivity with the user. *OFC* can be used to create appealing dashboards as the one described in (Wiegand, 2013).

FLEX

Adobe Flex it's a free, open-source application framework, was developed to create Rich Internet Applications. It provides a graphical user interface that permits the user to add elements by dragging and dropping. It facilitates data integration since it uses *MXML* and *XML* formats. Data can be read, and then translated into a graphic way either as an image, animation or a chart. It can be visualized in any web browser with *Flash* compatibility, which can be a disadvantage since it requires the user to install additional components (Lee S. J., 2015).

D3.js

Written in *JavaScript D3.js* (Data Driven Document) is a free library specialized in the creation of visualizations that can be represented either as *SVG* images or embedded in html files with *HTML5* and Cascade Style Sheets (*CSS*) (Lee S. J., 2015). The created visualizations, following web standards, with its data-driven approach to DOM manipulation it allows the creation of visualizations according to data that needs to be represented (Bao, 2014). With the extra functionality offered, the representations of data can go beyond traditional charts. The *D3 API*, besides the data visualization, and geospatial functionality provides supplementary functionality for the creation of animations, to perform analysis, and also contains data utilities and DOM utilities (*JQuery*). As it supports standard web technologies *D3.js* has been used in different fields other than data visualization, data science, and even by artists.

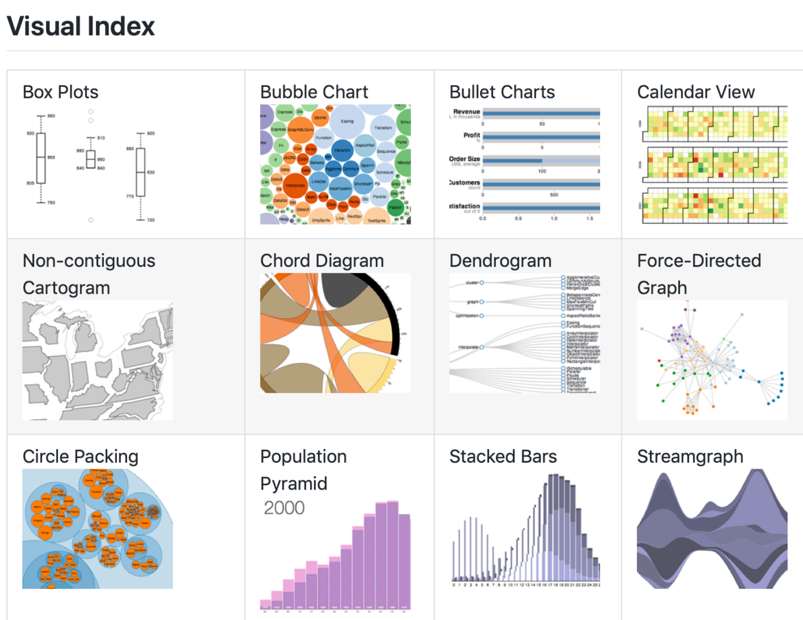

Fig. 41. Screenshot taken from examples of D3.js site, https://github.com/d3/d3/wiki/Gallery, visited on September 2018.

Custom software

In contrast with software libraries and API, custom software exists to cover the particular and specific needs. Developers gather with the user to understand their specific needs of a business process and develop a solution that will help fulfill the features needed. Following, we will describe some of them.

Zoil

Zoil is an object-oriented framework that was created for distributed user interfaces, meaning the interaction with different devices like monitors, displays, platforms, displays and the user to enrich the user experience within interactive spaces with ubiquitous computing technology (Jetter, Zöllner, Gerken, & Reiterer, 2012). It defines six important principles to consider when designing or creating interactive spaces, allowing the direct manipulation of the user interfaces, zooming, visualization tools, to make annotations, and supporting collaboration among different devices. *ZOIL*'s future work will focus on having a low threshold and keep or increase the high ceiling for the implementation of collaborative post *WIMP* distributed user interfaces.

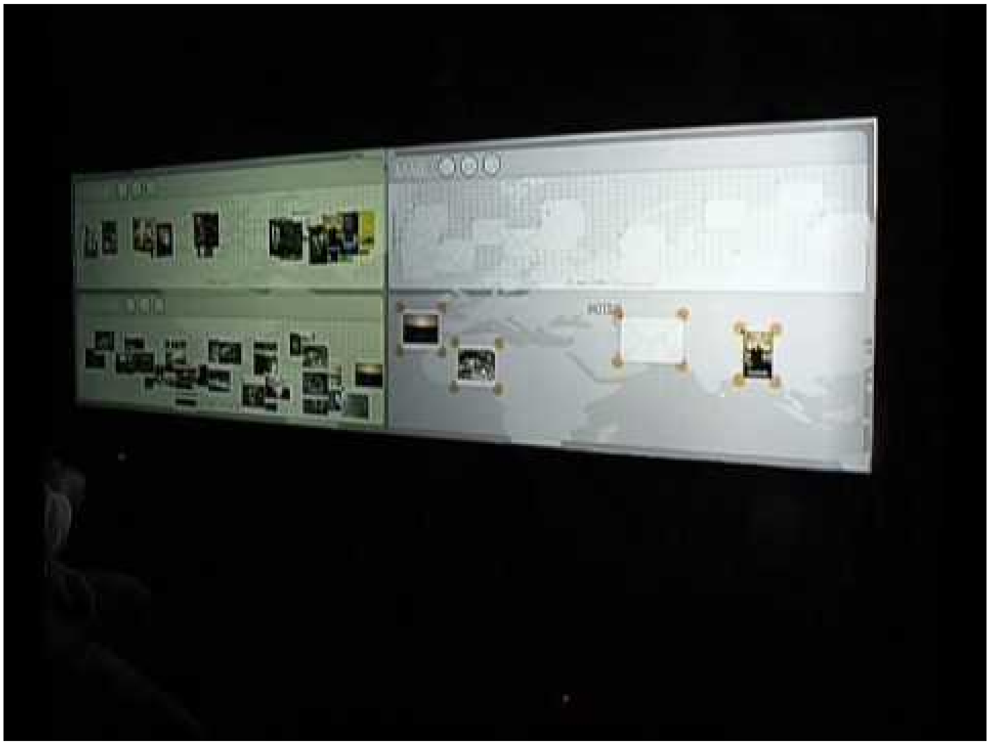

Fig. 42. Zoil Framework interface (https://www.youtube.com/watch?v=ngP0TEaOJwQ)

FLUID

It is an interface that was created to show the advantages of gestural interfaces over traditional *WIMP* design-based interfaces. It is a visualization tool available for tablets with data selection and interaction functionality such as filtering and ordering. With this tool the authors seek to simplify the process of getting and visualizing data, with the suppression of menus and other graphical elements that can be found in traditional interfaces, they maximize the available space on the screen to allow the user to have a better and clearer perception of the data visualization (Drucker, Fisher, Sadana, Herron, & Schraefel , 2013).

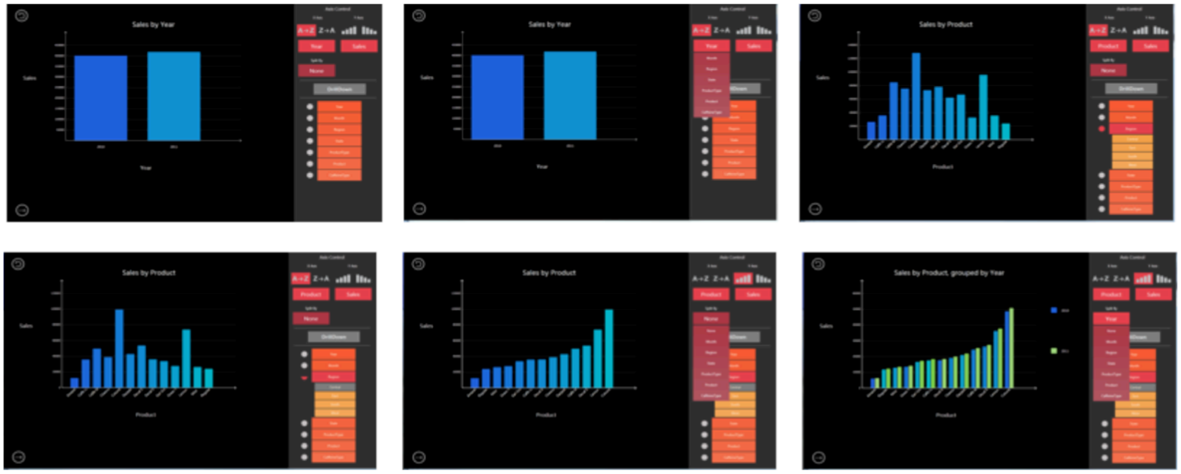

Fig. 43. FLUID interface protoype.

Munin

A software framework developed to increase and support collaborative work is *Munin*. It is a three-layered system to support the replication of data, extends support and services, and provides distributed visualization. It's an open source project available on git-hub that provides ready to consume services, but also new ones can be added as needed. The main feature of *Munin* is that it has a peer-to-peer architecture, which doesn't require a central server to manage and render data, these processes are carried among the peers connected to the system. It supports several devices connected, collaboration, post-wimp interaction trough gestures, tangible and full body interaction devices and different visualizations. The shared state provided by *Munin* allows the shared tables to be queried, linked and combined in different tables, to make available different insights of data. *Munin* was built using the Java platform which makes it multiplatform available, including mobile devices, although it's not available for iOS platform yet (Badam, 2015). In the next figure the *Munin* interface is showed connecting and sharing information among two devices.

Fig. 44. Munin sample of two devices connected and sharing data, providing collaboration among peers.

DC-Tec

Another custom tool to cover specific needs is *DC-Tec*, decision center tool developed by the Instituto Tecnológico y de Estudios Superiores de Monterrey university in Mexico City. The need of this tool was born to support the creation of knowledge and its transference to solve different problems related with the society in Mexico. The tool was developed to be used in the decision center of Tecnologico de Monterrey and the main features it offers are described next. The *DC-Tec* tool has two main functionalities, execution and visualization. The execution module, connects to the data, giving format and organizing in specific way, ready to be display in the visualization module. The visualization module shows the information with predefined scenarios and also gives the facility of a dynamic scenario that can be personalized and edited by the specific user needs. One important particularity is that the information visualized can be modified and updated in real time, allowing other users, connected remotely verify the new data and the projections performed, according to the model rules. The *DC-Tec* tool can be used in the decision-making center, which is seven flat displays connected, allowing the visualization in seven screens simultaneously, giving a wider area to analyze and examine data concurrently. The information provided by the specific model, in this case related with the energy sector, can be viewed graphically through charts, that are also created dynamically as the data is consulted or modified. This tool has been constructed with technology that made easier the integration, development and use. Its base technology is composed of *JavaScript,* *JSON*, *Firebase and Google Sheets*, which make it accessible through any browser. In the figure below, we show a screen shot of one of the views of the *DC-Tec* Tool.

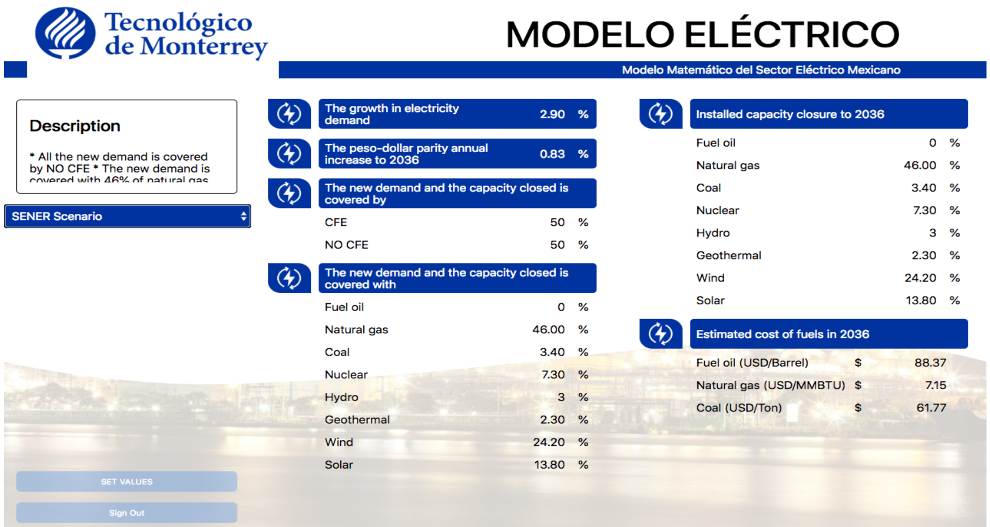

Fig. 45. Screen shot for the DC-Tec Tool, showing data of an electric model in Mexico

Commercial Software

Commercial Software, also known as licensed software, is the software that requires the user to pay a fee for using the full features it offers. Commercial software covers a specific market niche. Data visualization has become a trend among organizations by using different tools that seek to provide them with an ease of use, practical, and appealing way to visualize their information to maximize the organization's business value. According to the 2019 magic quadrant of Gartner (Gartner, Magic Quadrant of Gartner, 2019) , 21 tools provide this functionality, in this section, we will analyze some of them from each of the four quadrants, niche players, visionaries, challengers and leaders. Since there are many tools, and we cannot include all of them in the evaluation, we selected a representative tool from each quadrant considering not being on the extremes, nor the lowest or the highest. Microsoft Power BI is recognized as the best one for its integration with other Microsoft products such as Azure cloud, Microsoft Office, and its desktop version availability. Among Microsoft Power BI capabilities are data preparation, visual-based data discovery, interactive dashboards and analytics (Richardson, Sallam, Schlegel, Kronz, & Sun, 2020) . We consider that choosing the best tool, from leaders' quadrant, could bias the evaluation study. For this reason, the chosen tools were QlikView, Tableau, MicroStrategy, Domo and Sisense.

QlikView

With a user-driven experience, *QlikView* is a Business Intelligence platform designed to give users tools to explore data, make discoveries, and uncover insights related to business concerns with conducted searches making available data through interactive dashboards. Data association is the way *QlikView* enhance the way a user can explore and interact with data, showing results that would not appear with traditional query-based tools. Its integrated platform has three components to be used by either IT professionals, Business analysts or developers and Business users. The *QlikView* desktop is used by business users, to consume and transform data into graphical presentations. *QlikView* publisher has two main functionalities that is to create the connection with data sources and to create rules to simplify data and make easier the load for distribution across other *QlikView* servers or static distribution files. *QlikView* Server is the analytics engine which contains the management environment to handle the communication with the client (*QlikView* desktop) and can provide access to documents through its own web server. *QlikView* is a tool that provides different features to adapt and carry user's needs according to their profile plus it provides several resources online through its community site that makes easier the understanding and learning of this tool. It also provides a program to facilitate the training for professors and students in the academic field to improve the analytical skills (Podeschi, 2014). *QlikView* name is related to the mouse click as a metaphor to say how practical it is to use, and that the information is available in a fast an easy way through a "click" (A Shukla, 2016), (Luminiţa, 2012).

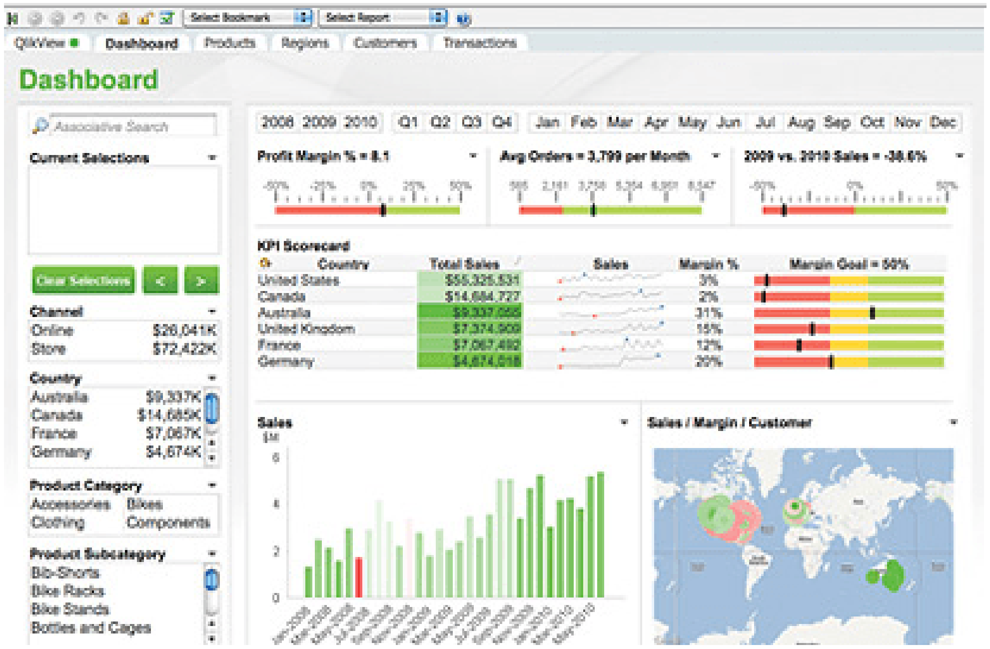

Fig. 46. QlikView's data visualization dashboard.

Microstrategy

MicroStrategy is a Business Intelligence platform with more than 20 years on the market that has been evolving to adapt to medium and large companies need. With their last version, *Microstrategy 2019*, it supports production reports, dashboards, scorecards, ad hoc reporting, *OLAP* and reporting. It can fulfill any user needs, whether it is a data consumer, data explorer, data analyst or data scientist. And data can be consumed from a wide variety of data sources, from a *CSV* file to a more complex one such a relational database, Hadoop or from Cloud storage (Atriwal, 2016). One feature recently added is the support for free-form visual discovery and data preparation to adapt to the self-service model, so users are more independent from IT experts and they can analyze, visualize and share data that is relevant to other team members. With the *SDK* it provides, custom visualizations can be created and embed also some third-party libraries, such as *D3.js, Highcharts* and *Google*. Maps and multimedia can also be embedded with the multimedia widget it provides. Facing the challenges of the *BI* world, *Microstrategy* has been evolving to offer more and better features making it be on the challenger's quadrant of the Gartner magic quadrants for *BI* platforms.

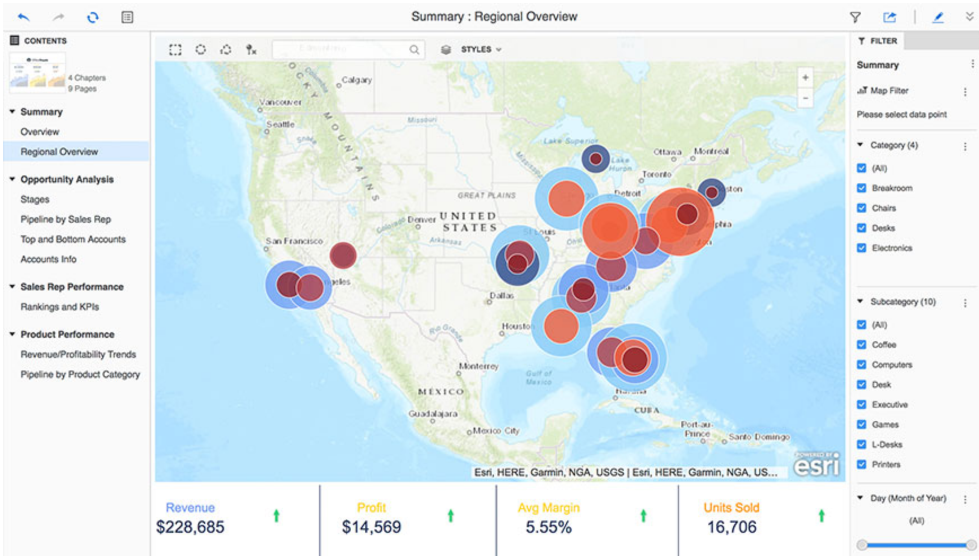

Fig. 47. Microstrategy interface. Image taken form Microstrtegy web site (Microstrategy, 2018) (http://www.microstrategy.com)

Domo

Domo is a Business Intelligence cloud-based tool (Feinleib, 2014). It provides a wide variety of features to users, to allow them build insights of their data in order to support the decision-making activities related to a company business. *Domo* provides connection to data from several sources, among which we can find *CSV*, *spreadsheets, json, XML* files and more complex data sources such as *Amazon* Web Services (AWS), *Google services*, *SQL* server, *MongoDB, SAP, PeopleSoft,* etc. With a drag-and-drop functionality to prepare data it also allows real-time data updates from the data-sources. For data visualization, it has a storytelling feature, through its Card Builder functionality, which based on the data source selected by the user gives different visualization suggestions within the more than 50 chart types it has that can be customizable according to user needs. Visualizations generated can be shared and modified by team members. To enrich collaboration, *Domo* provides a chat to allow team members to communicate with each other and discuss among the results obtained with the analyzed data and not only to share it. Being a multiplatform tool available for *Windows* and *MacOS*, it also gives connectivity through mobile devices such as Android or iOS-based smartphones. Users reviews claim that *Domo* is a tool that provides all the functionality necessary to gather, analyze and visualize data, all in once in an easy way and proves to be a decision-making support tool that provides accurate information when required (Verma, 2016). In Figure 48, a screen shot of *Domo* interface is shown.

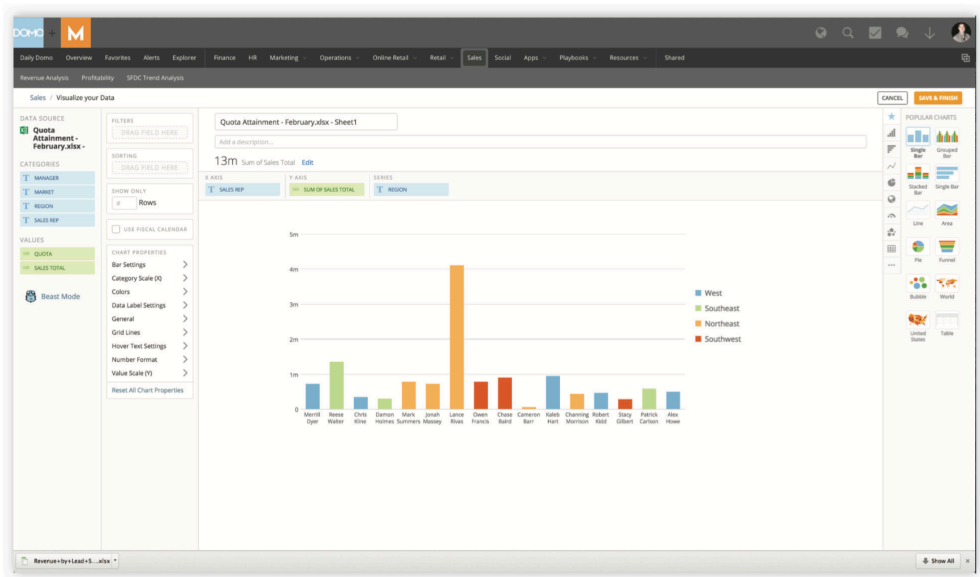

Fig. 48. Domo interface screen shot, taken from Domo website (http://www.domo.com, October 2018).

Tableau

Tableau is a project that was born from the Polaris project. Is a graphical system that provides different functionalities to the user to be able to work with data (Wesley, 2011). It has an online/offline version available, it allows to connect to different data sources and provides different ways to explore and analyze data. *Tableau* creators realized different needs that users have, so they made it available through three different versions, *Tableau* online to make sharing and collaborating easier and without server maintenance required; *Tableau* Server for sharing data and dashboards with different deployment options, on-premises and on the cloud; Tableau desktop is installed in any personal computer, allowing the user have a copy of the data to work with, and later dashboards, or derived data can be shared through Tableau online or Tableau server; Finally the latest version released *Tableau* Prep to connect, filter and prepare data to make it accessible in a faster and more dynamic way. However, the availability of *Tableau* is limited. It has a trial version available and different licenses available for professional use. It also has academic programs for students and professors. It also counts with a non-profit program of different social global issues through its *Tableau* foundation in which different participants work on projects such as health, poverty, climate, education, etc. to promote a real change in the world with data. All of this information is available in their website. *Tableau* is promoted as a business intelligence tool to understand and communicate data. In the next figure, a screenshot of the *Tableau* desktop interface in Spanish can be seen. *Tableau* is also available in a wide variety of languages. (Tableau: Business Intelligence and Analytics Software, 2018), (Hanrahan, Pat; Stolte, Chris; Mackinlay, Jock;, 2018).

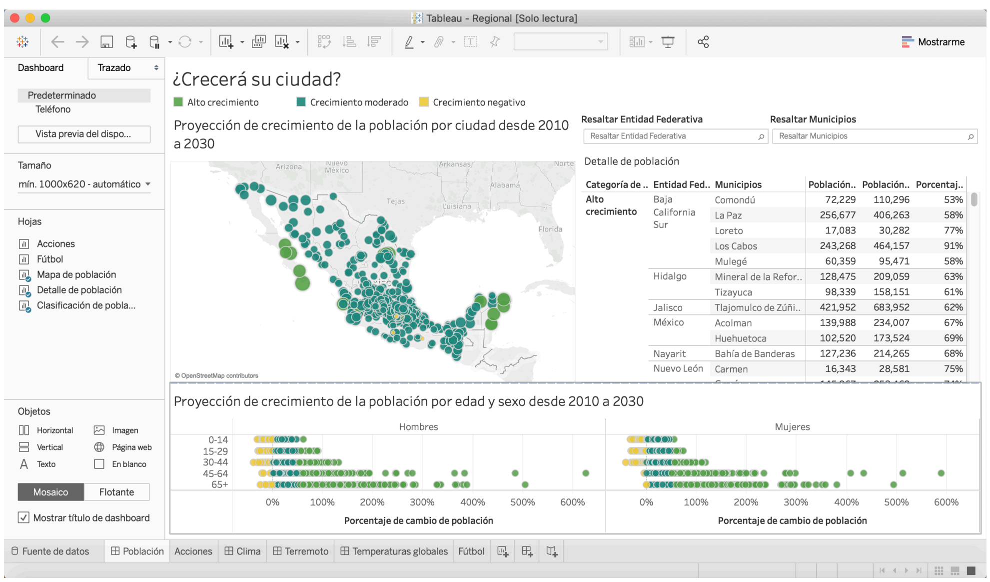

Fig. 49. Tableau desktop interface on MacOS, showing a sample per regions data. Referencia de sitio.

Sisense

In 2014 Ovum Consulting and Analyst firm considered BI software *SISENSE* to be in growth to fulfill market's needs such as data visualization, collaboration, Big Data, Analysis (Mukherjee, 2014). In 2018 Gartner's Magic Quadrant for Analytics and Business Intelligence Platforms positions *SISENSE* in the visionary quadrant not just by the ease of use but because it offers innovative solutions as reported by their customers' experience (Gartner, Magic Quadrant for Analytics and Business Intelligence Platforms, 2018). *SISENSE* is a platform that provides an integrated *BI* solution with capabilities to process, analyze and visualize data in a way to provide medium and large companies a tool that facilitates the way business decisions are made. It provides a way for preparing data in an unspecialized form to allow any user to bring data from one, two or more different sources and exploration can be done on the fly and on different levels as required. For analysis it can perform multiple data aggregations in one same calculation meanwhile it generates the view thanks to its core In-Chip technology that improves the performance to return results just is seconds and reduces costs in hardware. It's ElastiCube repository was designed to facilitate collaboration among users and it keeps all data used in a single repository, this way everybody has the same version of data. Part of its innovative features are that it provides, as some other tools do, the possibility of getting notifications through alerts on web or mobile devices. However, the innovation relies in the use of AI by providing the connectivity with the *Amazon* Echo device and ask the *Alexa* assistant about alerts of the state of KPIs in real time. *Sisense Boto* also incorporates machine learning to connect to a messaging platform and be able to ask questions related to the data of interest. Finally, it provides tools to automate frequent tasks, integration with third party applications and the possibility to personalize the look and feel of the company to make it look as proprietary tool.

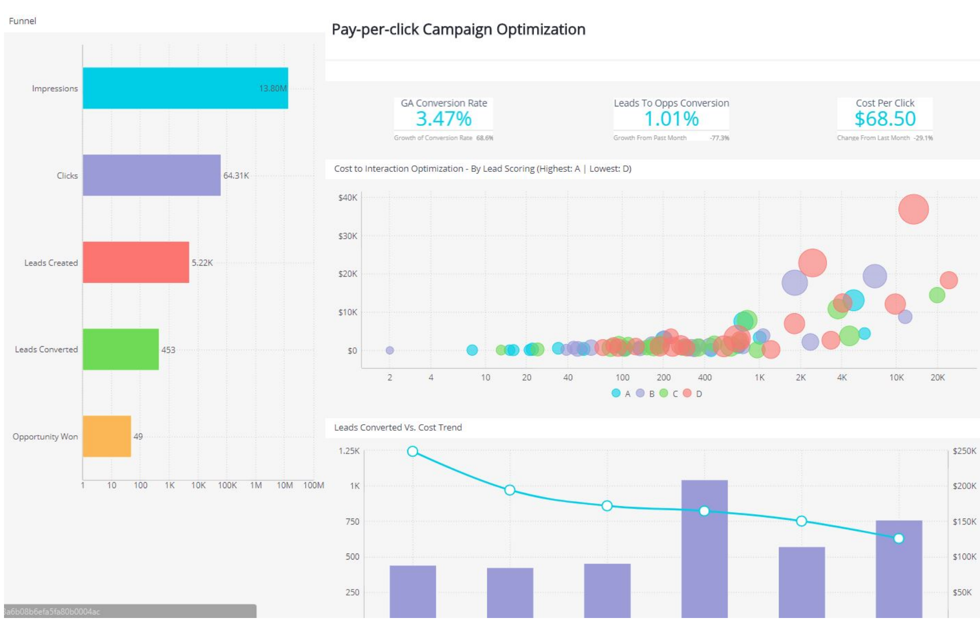

Fig. 50. SISENSE Dashboard example, taken from SISENSE website(https://www.sisense.com/)

Appendix 3

Energy Case Study

The following paragraphs explain, in general, the technologies considered for electricity generation in the model, which are based on the data published by the Ministry of Energy in its Energy Information System (Secretaría de Energía, 2018).

In the Mexican electricity sector, for electricity generation, the methods, technologies and fuels described below are used. The display structure, which is based on the mathematical formulation of the model, considers the following fuels and their corresponding electricity generation technologies:

Fuel oil

Fuel oil is a product obtained from the refining of crude oil. This fuel is made up of carbon molecules, it is characterized by its high viscosity and by being insoluble in water. Its main uses are in electricity generation, mainly in steam thermoelectric plants, and in the industrial sector (Pemex, 2017).

Natural gas

Natural gas is a fuel made up mainly of hydrogen and carbon molecules in a gaseous state; the natural gas most used in electricity generation is dry gas, which is mostly made up of fuel in combined cycle plants and in gas turbines for methane molecules (CH4) (Pemex, 2017). Natural gas is used to generate electricity (CFE, 2007).

Coal

Coal can be defined as a sedimentary rock impregnated with a large amount of carbon from adjacent organic matter (Dirección General de Industria, Energía y Minas de la Comunidad de Madrid, 2007). It is a common fuel for electricity generation, widely used in thermoelectric and dual-type plants (Secretaría de Energía, 2018).

Nuclear

Nuclear energy consists of releasing a large amount of energy from the internal structure of atoms, which can occur through two types of complex chemical reactions: fission and nuclear fusion. Nuclear fission is the process that occurs when an atomic nucleus decomposes, this generates two atomic nuclei of equal size and allows the release of a large amount of energy. Nuclear fusion is a reaction in which two atomic nuclei combine, releasing a large amount of energy and thus generating a larger stable nucleus (Jha, 2004). The energy released in the nuclear fission and fusion processes can be used to generate electricity in nuclear power plants.

Hydroelectric

A hydroelectric power generation plant is one that takes advantage of the potential energy contained in water to convert it into electricity, through a turbine-generator system (Juárez, 1992).

Geothermal

Geothermal energy is that type of energy that comes from the interior of the crust of the planet earth, being generally in the form of steam at high temperatures, and through thermodynamic cycles it is converted into electricity (Secretaría de Energía de Argentina, S. D.).

Wind

Wind energy is considered as that type of energy that is obtained by taking advantage of the movement of air due to pressure gradients in the atmosphere. This type of energy can be transformed into electricity by means of wind turbines, which are responsible for converting the kinetic energy of the wind into mechanical energy, and later, with the help of a motor-generator system, transforming it into electrical energy (Tong, 2010).

Solar

Solar energy is that which is obtained from solar radiation and is converted into thermal energy or electrical energy, through active or passive technologies. Active technologies are those that take advantage of the sun directly, some of them are solar panels (electrical energy) and solar collectors (thermal energy). Passive technologies are those that are usually implemented in architecture, to provide thermal comfort and reduce energy consumption in buildings. For this model, the Solar Energy section only considers that solar energy that is transformed into electrical energy.

It is important to mention that within the development of the Mexican electricity sector model, two groups of power generators were considered, emulating the segmentation carried out by the Ministry of Energy in its Energy Information System: Federal Electricity Commission (CFE) and Private Industry (NO CFE).

The results or outputs that are considered within the model of the Mexican electricity sector are the following:

• Electricity production

It is the amount of electrical energy, measured in Giga watt-hours (GW-hr), that was consumed in a given year at the national level.

• Installed capacity by type of technology

It is the installed capacity available in a specific year by type of technology, it is measured in Megawatts (MW).

• New installed capacity to meet the new demand

It is the capacity that is installed to cover the increase in the demand for electrical energy, its unit of measurement is the Megawatt (MW).

• Production associated with the new installed capacity to satisfy the new demand

It is the amount of electricity produced by the new capacity added to the electrical system.

• New installed capacity by replacement

The installed capacity due to the closure of electricity generation plants, its unit of measurement is the Megawatt (MW).

• Production associated with the new installed capacity by replacement

It refers to the electricity produced by the new installed capacity due to the replacement of closed capacity, it is measured in Giga watt-hours (GW-hr).

• Closed capacity

It identifies the number of power generation plants that closed in a given year, its measurement unit is the Megawatt (MW).

• Production associated with closed capacity (GW-hr)

It refers to the amount of energy that stopped being generated by the capacity closure, to measure it the Giga watt-hour (GW-hr) is used.

• Total electricity production by type of fuel

It shows the proportion of electrical energy that is produced with renewable sources, hydrocarbons and others, its reference unit is the percentage (%).

• Investment

It is the monetary investment that is required to install new capacity in the Mexican electricity system, the investments are measured in millions of pesos (millions of MXN).

• Cost of electricity

The cost of producing electrical energy measured in pesos per kilo watt-hour (MXN / kw-hr) of electrical energy.

• Electricity sale price

The cost of electricity for consumers is measured in pesos per kilo watt-hour (MXN / kw-hr).

• Income

The amount of money collected from the sale of electrical energy measured in millions of pesos.

• Gross profit

The economic benefits generated by the sale of electricity measured in millions of pesos.

• Jobs

It refers to the number of jobs generated in the Mexican electricity sector.

• Carbon footprint

The tons of carbon dioxide emitted into the environment as a result of electricity generation measured in tons of CO_2.

• Clean energy certificates

It is the number of clean energy certificates (CELs) generated by the production of electricity with renewable energy sources.

Appendix 4

Acronyms

	Description
EBDM	Evidence-Based Decision-Making
SOA	Service Oriented Architecture
MVVM	Model-View-ViewModel
GUI	Graphic User Interfaces
IDE	Integrated Development Environment
SWOT	Strengths, Weaknesses, Opportunities, and Treats
SDK	Software Development Kit
ASU	Arizona State University
MVC	Model View Controller
AVE	Average Variance Extracted
CR	Composite Reliability
API	Application Program Interface
HCI	Human Computer Interface
UX	User Experience
DBMS	Data Base Management System
RDF	Resource Description Framework
SQL	Structured Query Language
UML	Unified Modeling Language
CRUD	Create, Read, Update, Delete
ITIL	Information Technology Infrastructure Library
KPI	Key Performance Indicator

Published papers

"Use of a decision-making laboratory to support student’s visual analysis for the solution of a transportation problem in Mexico City." *2019 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2019. Doi: 10.1109/FIE43999.2019.9028550

Use of decision center to support student's visual analysis for the solution of a transportation problem in Mexico City

Christelle Navarrete Corella,
Ph.D. student
Escuela de Ingeniería y Ciencias
Tecnológico de Monterrey
Mexico City, Mexico
christelle@itesm.mx

Julieta Noguez, Ph.D.
Escuela de Ingeniería y Ciencias
Tecnológico de Monterrey
Mexico City, Mexico
jnoguez@tec.mx

Martín Molina, Ph.D.
Escuela de Ingeniería y Ciencias
Tecnológico de Monterrey
Mexico City, Mexico
jose.molina@tec.mx

Erzhena Soktoeva
Decision Theater
Arizona State University
Tempe, USA
erzhena@asu.edu

Rahul Salla
Decision Theater
Arizona State University
Tempe, USA
rahul.salla@asu.edu

line 1: 6th Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address

Abstract*— Universities' educational models seek to help students to develop skills that allow them to become leaders who embrace the different challenges and opportunities of the 21st century. Leaders must make numerous decisions that can be, to a greater or lesser degree, important, as well as easy or difficult, depending on the consequences they imply. To choose the best option from among the possible ones, information is needed. To encourage important information to be available when making decisions, a decision-making center laboratory has been created, which allows viewing in 7 high-definition flat displays simultaneously and 32 spots for people to allow collaborative work to support the decision-making process. In the process of students' education, a set of challenges were designed. The use of the decision-making center laboratory allows the participants to visualize, analyze and interact with information that is displayed. The case study consisted in the analysis and visualization of open data concerning the urban mobility problem in the financial and corporate center of Mexico City. It was designed for students enrolled in the last third of Computer Science major. This paper presents the implementation process and the encouraging results of this process through the use of a decision-making center laboratory as a tool to create and use specialized visualizations regarding an urban mobility problem in Mexico City. With the use of different and diverse tools as the decision-making center laboratory and real-life challenges we seek to enrich student's outcomes and enhance the experimental learning.

***Keywords*—decision center, decision-making, visual analysis, information visualization.**

I. Introduction and Problem Statement

Universities' educational models have become a challenge in the last century. The easy access to information has made students be more aware and interested in not just learning but to put in practice the concepts learned in their classes. The use of technology has become a key element to transform traditional classes into a new paradigm oriented to collaborative activities in which students learn by doing and hand-on experiences [1]. Tec21 model is the educational model through which Tecnologico de Monterrey University seeks to adapt teaching techniques to student's needs and improve their learning process [2]. Universities around the globe have been trying to modify their teaching strategies to enhance the alumni way of learning [3] and to incorporate the results of their practices in another student's education [4]. One of the main features of Tecnologico de Monterrey's educational model is the learning based on challenges component, in which the students work in facing real world problems in order to develop the required competencies that will allow them to generate different solutions in a collaborative way transforming their environment and adding value to it. To carry out this strategy, a challenge was developed, in a week called *i week*, in which students work full time for a week in knowing, analyzing and solving real world situations or needs, that would allow the students of the computer science major apply their knowledge to identify high impact problems through the use of open data that provides information related to excessive vehicular traffic in Mexico City. With the information obtained they would be able to relate data, generate hypothesis and complement it with additional facts. The analysis and integration of this information will allow them to generate visualization schemes in order to check or verify different options of solution. The different options will be displayed in the decision-making center laboratory, that is part of Tecnologico de Monterrey infrastructure. The decision-making

Miranda J., Navarrete Ch., Noguez J., Molina-Espinosa J.M, Ramírez-Montoya M.S, Navarro-Tuch S.A., Bustamante-Bello M.R, Rosas-Fernández J.B., Molina A., 2021. "The Core Components of Education 4.0 in Higher Education: Three Case Studies in Engineering Education." Computers and Electrical Engineering. Special Issue VSI-tei. ISSN: 0045-7906. JCR, FI=2.663 Q2. (In press)

The Core Components of Education 4.0 in Higher Education: Three Case Studies in Engineering Education

Jhonattan Miranda[1], Christelle Navarrete[1*], Julieta Noguez[1], José-Martin Molina-Espinosa[1], María-Soledad Ramírez-Montoya[1], Sergio A. Navarro-Tuch[1], Martín-Rogelio Bustamante-Bello[1], José-Bernardo Rosas-Fernández[2], Arturo Molina[1]

[1]School of Engineering and Sciences, Tecnologico de Monterrey, Mexico City, Mexico
{jhonattan.miranda, jnoguez, jose.molina, solramirez, snavarro.tuch, rbustama, armolina}@tec.mx, a00711169@itesm.mx[*]
[2]Secretary of Education, Science, Technology and Innovation, Government of Mexico City, Mexico, jbrosas@cdmx.gob.mx

Abstract

Technological progress and its rapid evolution have positively affected the industrial sector and different productive/service sectors. One of the service-sectors that have benefited the most has been Education. In this sector, the implementation of current and emerging technologies combined with innovative pedagogical procedures and best practices is known as Education 4.0, which, in this paper, is described and mapped according to the well-known periods of the four industrial revolutions and related to higher education. Likewise, four core components of Education 4.0 to be used as a reference for the design of new projects in educational innovation are proposed (i) Competencies, (ii) Learning Methods, (iii) Information and Communication Technologies, and (iv) Infrastructure. Finally, three case studies applied to Engineering Education illustrate how the proposed components are considered in educational programs' designs.

Keywords: Education 4.0, Engineering Education, Educational Innovation, Higher Education.

I. INTRODUCTION

In recent years, the term "Education 4.0" has been assigned high relevance by different authors who have conceptualized it according to the well-known four industrial revolutions periods. Accordingly, how the evolution of technologies has influenced the education sector through time has been analyzed. Hence, pedagogy, teaching philosophies, educational models, information sources, learning methods, and students' and educators' roles have been included in the conceptualizations. Nowadays, it is observed that methodologies,

"Criteria Definition Towards a Metric for Visualization Tools." *Journal of Information Visualization, 2021* (Accepted with changes).

Criteria Definition Towards a Metric for Visualization Tools

Summary

This study's primary goal is to identify the critical features a visualization tool should have to enhance the stakeholders' decision-making activities in decision-making centers. Decision-making centers have been used in the analysis of information related to real-world problems. Information visualization plays an essential role in decision-making centers to facilitate the visual analysis process to stakeholders. Hence our study aims to find the optimal features a visualization tool for use in a decision-making center. Although experts performed the study in business analysis and software engineering in this first phase, we will focus on finding the most relevant tasks for specific application fields in the decision-making area to enhance the overall process. Based on decision-making center specific features, a methodology, based on visualization metrics, was created to evaluate 14 tools classified in three categories named as software libraries or APIs, custom software, and commercial software. According to results obtained, the selected evaluation criteria were features included in every set of tools and were well evaluated, showing these are critical elements in visualization tools to enhance benefits in processes such as decision-making.

Keywords: Evaluation criteria, Information visualization, Metrics, Visualization tools, Decision-making center.

Introduction

The use of images to represent information is the simplest definition to understand data visualization [1]. Since the second Century, data visualization has been present when the first table was created to organize astronomical information in Egypt [1]. The need for having graphic information remains so far, not only to make the information accessible but to make it easier to translate and presented to be understood by the final user. Data visualization can be recognized as a graphics presentation to help organize the information visually or graphically. However, as Chen [2] formally defines, it refers to generating graphical representations of information. The main feature that has to be noticed in the 21st Century is that data visualization is achieved with computers, through already existing software or ad-hoc systems.

Different tools to create visualizations have been created since the term showed up. However, independently of the features and compelling visualizations that can be created with any of these tools, organizations have noticed the importance to the subject. Visual languages can be used as an effective communication medium in the management field [3]. Hence visualization can be considered a visual language. Not just limited to quantitative but also to qualitative visualizations, these tools can be used in different managerial functions, including idea generation, decision making, and planning. The reason visualization should be considered is that it could play an essential role in supporting the organization's decision-making process decision-making. Visual information can provide a deeper meaning of the

Curriculum Vita

Christelle Navarrete Corella was born in Mexico City, Mexico, on May 1st, 1980. She earned the Computer Systems Engineering degree from the *Instituto Tecnológico y de Estudios Superiores de Monterrey,* Campus Ciudad de México in December 2022. She was accepted in the Engineering Sciences doctorate program on august 2016. She earned the Engineering Sciences Doctorate degree from the *Instituto Tecnológico y de Estudios Superiores de Monterrey*, Campus Ciudad de México in June 2021. She also worked in software development for companies Redisegno S.A. de C.V. from 2003 to 2007, NAAT Consulting in 2008, TV AZTECA in 2008, and Rhyno Systems S.A. de C.V. in 2013.
Areas of interest: Softrware Development, GUIs, UX, Information Visualization.

[1]**The style file phdThesisFormat.sty used to set up this dissertation was prepared by the Center of**

Intelligent Systems of the Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey Campus