# An Information Visualization Engine for Situational-Awareness in Health Insurance

**Flávio Gabriel de Melo Epifânio**

Thesis to obtain the Master of Science Degree in

**Information Systems and Computer Engineering**

Supervisor: Prof. Gabriel Cesar Ferreira Pestana

**Examination Committee**

Chairperson: Prof. Francisco João Duarte Cordeiro Correia dos Santos
Supervisor: Prof. Gabriel Cesar Ferreira Pestana
Member of the Committee: Prof. André Ferreira Ferrão Couto e Vasconcelos

**May 2018**

# Acknowledgments

Firstly, I would like to express my gratitude to my advisor Prof. Gabriel Pestana for his support and guidance through all stages of my Master Thesis and related research. His patience, motivation, and high standards, not only provided me with a positive example for my personal development, but also helped me to archive the results presented in this thesis.

I would also like to thank Future Healthcare (FH) by providing me tools and time to develop this work, without which, this work would never happen. In particular, I would like to thank FH’s CEO José Pina for his immediate association to this project, and to all IT co-works that had to lead with my limited time within the organization. I would also like to give a special mention to Tiago Dionizio, a FH co-worker, whose help modeling FH case study was priceless.

I must express my profound gratitude to Eunice Epifânio, my wife, for her unfailing support and encouragement through the process of researching and writing this thesis. I was continually amazed by her patient and encouragement. As someone with graphical design expertise, she also provided an important contribution to the visual component of this dissertation, producing and combining images. I’m also grateful to my family members and friends who have supported me along the way. This accomplishment would not have been possible without them.

Finally, but by no means least, I also thank God who have provided me moral and emotional support in my life and have guided me throw the hole process.

Thank you all.

# Abstract

Nowadays, private health insurance is experiencing a substantial transformation. Chronic medical conditions are responsible for two-thirds of current healthcare cost's increase and the number of chronic patients has increased greatly because of unhealthy behaviors. Health insurance players seek to provide solutions focused on health outcomes in detriment of classical models. These emergent models present new challenges on how to, simultaneously, prevent disease burdens and act preventively, promoting healthy behaviors to improve costumer's quality of life and well-being.

Information Visualization field has here a relevant role, providing value to the existing data where sensemaking fits into the whole process of working with vast volumes of datasets. Also, promoting a collaborative dataflow to keep all actors aware about events requiring their immediate attention assumes, therefore, a key relevance, contributing simultaneously to promote an active participation of healthcare professionals in monitoring the health status and well-being of each chronic patient, as well as the responsibility of the individual for their actions.

Following a user-centric Agile Design Science Research Methodology and Service Design Thinking principles, this thesis models an information visualization engine for situational-awareness in health insurance, using concepts of Dashboard Engine Frameworks in order to empower users through the creation of interactive dashboards, that streamlines visual representation of events in a collaborative decision making environment, as a way to proactively monitor unhealthy risk behaviors.

Finally, a designed prototype instantiates an innovative health insurance model addressed through two operational scenarios, demonstrating how the proposed engine can be applied to health insurance.

# Keywords

Information visualization, emergent health insurance models, risk indicators, gamification, spatio-temporal data-sensemaking, situational-awareness.

# Resumo

Actualmente, os seguros de saúde estão em transformação. As doenças crônicas são responsáveis por dois terços do aumento dos custos de saúde e o número de doentes crônicos tem aumentado, principalmente devido a comportamentos pouco saudáveis. Os actores dos seguros de saúde procuram apresentar soluções focadas na melhoria do estado de saúde em detrimento dos modelos clássicos. Estes modelos emergentes procuram agir de forma preventiva, reduzindo custos de saúde e promovendo comportamentos mais saudáveis, melhorando a qualidade de vida e bem-estar dos clientes.

O estudo da Visualização de Informação tem aqui um papel relevante, adicionando valor aos dados existentes ao trabalhar com grandes volumes de informação. Também, a promoção de um fluxo de dados colaborativo permitindo manter todos os atores conscientes acerca de eventos que requeiram atenção imediata, assume uma relevância fundamental, contribuindo simultaneamente para promover uma participação ativa dos profissionais de saúde na monitorização dos doentes crônicos, bem como a responsabilização do doente pelas suas acções.

Seguindo uma metodologia ágil de investigação centrada no utilizador, esta tese modela um motor de visualização de informação para consciencialização situacional nos seguros de saúde. Utilizando princípios de motores de *Dashboards* capacita os utilizadores para a criação de *Dashboards* interativos que simplificam a representação de eventos num ambiente colaborativo de tomada de decisões, como forma de monitorizar proactivamente comportamentos de risco prejudiciais para a saúde.

Finalmente, o desenho de um protótipo instancia um modelo inovador de seguros de saúde endereçado através de dois cenários operacionais, aplicando o modelo proposto aos seguros de saúde.

# Palavras-chave

Visualização de Informação, modelos emergentes de segures de saúde, indicadores de risco, *gamification*, significado espaço-temporal dos dados, consciencialização situacional.

# Table of Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| **AAA** | Authentication, Authorization and Accounting |
| **ADSRM** | Agile Design Science Research Methodology |
| **API** | Application Programming Interface |
| **ASF** | Autoridade de Supervisão de Seguros e Fundos de Pensões (Insurance and Pension Funds Supervisory Authority) |
| **BI** | Business Intelligence |
| **BMI** | Body Mass Index |
| **BPA** | Business Process Analytic |
| **BPMN** | Business Process Model and Notation |
| **CSV** | Comma-Separated Values |
| **D3** | Data-Driven Documents |
| **DOM** | Document Object Model |
| **DSR** | Design Science Research |
| **DSRM** | Design Science Research Methodology |
| **ETL** | Extract Transform Load |
| **FH** | Future Healthcare |
| **GUI** | Graphical User Interface |
| **IS** | Information Systems |
| **IT** | Information Technology |
| **JDBC** | Java Database Connectivity |
| **JSON** | JavaScript Object Notation |
| **MDX** | Multidimensional Expressions |
| **MVC** | Model-View-Controller |
| **MVP** | Model-View-Presenter |
| **MVVM** | Model-View-ViewModel |
| **MVW** | Model-View-Whatever |
| **PM** | Presenter Model |
| **REST** | Representational State Transfer |
| **SQL** | Structured Query Language |
| **UI** | User Interface |

# 1. Introduction

This thesis addresses the design and implementation of an information visualization engine that uses information visualization techniques and information awareness mechanisms to provide spatio-temporal data-sensemaking and streamline the way events are transmitted to the user, assuring situational-awareness.

The section outlines the fact that Information Visualization is an emerging field in studying ways to visually present abstract data in an understandable way, enhancing information discovery and interpretation. This thesis presents several information visualization techniques aspects and information awareness mechanisms regarding the visual presentation of data. These techniques have multiple applications in many domains; the engine that will be presented within the scope of this thesis, despite being designed to be used in any context, it was configured to address specific scenarios from the health insurance sector, in particular, a health management program in Future Healthcare (FH).

FH is a privately held international group focused in providing clients with the best health, life and well-being conditions, through health insurance products and services, enabling an easier access to private healthcare and well-being services. The modeled health management program, based on behavior communities, focuses on high risk clients (e.g. chronic patients), providing a telemedicine approach by monitoring health behaviors to prevent disease burdens and promote client’s quality of life and well-being. This work presents a set of operational scenarios addressing system’s information awareness mechanisms (e.g. gamification) to monitor and foment healthier behaviors.

## 1.1. Scope

Today, modern sedentary lives and people's unhealthy behaviors are directly related with health risk factors, responsible for the increased risk of morbidity and chronic diseases, such as high blood pressure, sedentary lifestyle, tobacco consumption, obesity and high cholesterol [1]. Accordingly to the World Health Organization, in 2015, 70% of the overall total deaths occurred due to chronic diseases. It is also estimated that two-thirds of current healthcare cost’s increase are related to chronic diseases [2]. Such unhealthy behaviors represent a high risk to healthcare players, such as health insurance providers. Also, nowadays the health care consumer is more informed, with a wider knowledge regarding disease typologies, treatment options and healthy habits and lifestyles. This reality is changing the dynamics between insurer companies and its customers.

Classic health insurance solutions are increasingly considered by costumers as commodity packages and at present time represents only 28% of health expenses [3]. Therefore, insurance companies seek to realign their models in order to provide solutions based more on health outcomes in detriment of classical models exclusively focused on cover unpredictable catastrophic expenses, providing a financial protection to such situations. The challenge faced by health insurance in health outcomes programs is to receive context awareness whenever the insured person presents a deviant behavior in order to address unhealthy behaviors as a risk factor that needs to be monitored to prevent disease burdens, improve quality of life and well-being [4]. However, the sustainability of these emerging models must be supported by a collaborative culture with information sharing and accountability of each intervenient stakeholder.

The ability to collect and store valuable data is increasing at a faster rate than the ability to analyze it [5]. The use of non-invasive wearable devices enabled the introduction of innovative and low-cost monitoring mechanisms, where daily activities can easily be tracked, addressing the surveillance of unhealthy behaviors and simultaneously acting as a self-awareness mechanism to mitigate the risk of adverse clinical situations. The bottleneck is no longer in data extraction process or data quality, but in data exploration, discover and analysis capabilities [6].

Data analysis processes are not covered in this work (it is assumed that data are already processed and are ready to be visualized). The scope is to understand how events and data should be presented to the user in such a way that the created information artifacts allow to: Identify and provide visual insights rewarding costumer's who have adopted healthy behaviors, monitoring the achievement of prescribed thresholds as a way to reduce burdens.

The Information Visualization field has here a relevant role, providing value to the existing data where sensemaking fits into the whole process of working with vast volumes of datasets [7]. This is particularly relevant when the user relies on the system situational-awareness capabilities to be aware about events requiring an immediate attention. Interactive graphical representations (e.g. visual analytics dashboards) also helps in minimizing the user effort to interpret and understand the reported information and make informed decisions [8].

Information has to be presented in an understandable way, providing quick insights, and triggering alert messages whenever the reported measurements exceed target values. In the health insurance domain, target values can be for example: prescribed physical activity thresholds based on the user clinical profile. The study of clinical indicators' thresholds is outside of the scope of this work. It's assumed that they are previously defined according to each client clinical profile, health insurance program characteristics and contract conditions agreed between the client and the insurance provider.

The large demand and increased use of interactive graphical representations (e.g. dashboard viewers [9]) are guiding the appearance of many commercial tools [10], competitive open source solutions [11] and research works [12]–[14] addressing the creation of effective interactive graphical representations without depending on expert users. The challenge is to provide improvements on the current solutions, enhancing situational-awareness, and empower users to (self-) build interactive visualizations through an information visualization engine framework without the need to have deep data analysis knowledge, without using programming environments or the need to use hard in-depth analysis platforms with focus on complex BI analysis or complex data analytics capabilities.

## 1.2. Research Challenges

The main research challenges addressed by this thesis consists in 1) the design of an information visualization engine framework promoting information awareness mechanisms, 2) the modeling of health insurance challenges regarding communities of chronic patients and health risk behaviors, 3) and implementation of a prototype for the FH health insurance case-study. To address these challenges, this work followed a user-centric approach with an iterative design process where system behavior and information visualization artifacts were designed based on the user profile, modeling health insurance user requirements.

The approach is to use Information Visualization principles as a way to streamline visual representation of critical data and in optimizing human interventions, mitigating simultaneously the risk impact and contributing to improve patient healthcare and well-being [15]. Information Visualization Engine Frameworks provide a simple and efficient way to enhance spatio-temporal data-sensemaking, empowering end-users to configure and customize informational artifacts to promote context situational-awareness and streamline the way events and data are presented to the user in a collaborative decision making environment. To have a clear and grounded understanding of the existing solutions, a survey and comparative analysis on analogous Information Visualization Engine's solutions was conducted.

Using the information visualization paradigm as guidelines, an expressive and complete metadata model was designed to proactively monitor client's unhealthy risk behaviors, keeping the users aware about events requiring their immediate attention, contributing to improve patient health status and well-being. The use of information visualization techniques aim to maximize identification of value and interest in data when building interactive graphical components. In order to empower end-users in building personalized dashboards, the information visualization engine also supports data integration, data aggregation, filtering and searching capabilities, streamlining the exploration and analysis of data, providing the ability to repeatedly redefine goals as new information insight is acquired [6].

Addressing situational-awareness capabilities the proposed solution provides a way to keep the user well informed about deviant behaviors. When a suspicious situation is detected, the system triggers an event to inform all intervenient stakeholders (based on the level-of-concern, i.e., event severity), creating in this way a collaborative dataflow to keep all intervenient actors well informed in order to reduce the risk of events, or to motivate the user by presenting rewards regarding healthy behaviors. Such motivational aspects can be managed using gamification techniques [4]. Gamification techniques provide the ability to improve user information awareness (self-awareness) concerning KPIs and alert situations that requires intervention. On the other hand, for healthcare professionals, data related with risky situations is closed associated to the level-of-concern of patients' health-outcomes (e.g. patient health status degradation or non-fulfillment of prescribed objectives) [15].

The proposed solution's architecture intends to be reusable allowing the system to be easily modifiable outside the scope of this thesis. Therefore, a study on architectural design patterns was conducted to support the design decision on the information visualization client view, providing flexibility, reusability and facilitate development process. The addressed informational artifacts will be materialized in an Information Visualization Engine Framework prototype for FH health insurance context, enabling the creation and usage of interactive dashboards.

The design prototype will be applied to a new health management program that aims to improve client health and well-being through preventive behaviors. Based on the achieved results, it is intended to conduct a formal evaluation of the proposed information visualization engine within the business context at FH in future work. Such study will integrate the company strategy in providing an innovative health program.

## 1.3. Document Structure

This thesis is divided into five separate chapters: Introduction, related work, proposal, health insurance case study at FH and thesis' conclusions. These sections' contents are briefly described in the following lines:

- **Chapter 2 (Related Work)** presents a literature research review, presenting design science research methodology and covering the key research areas: information awareness mechanisms, information visualization and service design thinking. This section also includes a study on architectural design patterns and a study on Dashboard Engine Frameworks' concept including a comparative analysis on analogous solutions;
- **Chapter 3 (Proposal)** presents, firstly, the proposed information visualization engine conceptual model, describing system's informational artifacts, structured into six layers, each one encapsulating specific aspects of the information visualization engine. Secondly, this chapter presents the proposed architecture with two levels of detail: an overview over the client-server solution and a detailed view over the server and client components.
- **Chapter 4 (Case Study)** presents a real case study to apply the information visualization engine for situational-awareness in health insurance. Presents an overview over the health insurance market and instantiates an innovative health management product at FH organization, a privately held international group focused in providing clients with the best health, life and well-being conditions. To do so, two operational system behavior scenarios are modeled, in order to identify the existing actors, user model profiling, business processes, and expected outcomes.
- **Chapter 5 (Conclusions)** presents the main achievements in designing and implementing an information visualization engine that uses information awareness mechanisms and visual information artifacts to streamline the way events are transmitted to the user. Also, discusses the proposed future research work.

# 2. Related Work

This section provides an overview about the research work performed based mainly through a bibliographic search and by analyzing existing tools selected based on their proximity to cover some of the research challenges. First, the literature review presents a study on architectural design patterns to support architectural design decisions, focusing on flexibility, reusability and lightweight development processes. The related work also highlights Information Awareness Mechanisms focusing on gamification techniques and contextual situational-awareness to create informational artifacts addressing the operationalization of notification mechanisms and user awareness. Next, this section discusses Information Visualization techniques and frameworks as a way to improve user cognitive perception and decision making process. It is also included in this related work a survey on Dashboard Engines and a comparative analysis on analogous solutions. Finally, this literature review looks into this thesis' research methodology, presenting a study on Design Science Research.

## 2.1. Architectural Design Patterns

There are many different ways to design architectures and frameworks for large and complex graphical user interface (GUI) applications and a various number of patterns have been proposed by the software community to help this task. This section discusses a selection of Architectural Design Patterns that separates how internal informational entities are represented from the way that information is presented to the user.

***MVC***

**Model–view–controller (MVC)** architectural pattern, present in Figure 2.1, was introduced by Trygve Reenskaug [16] at Xerox PARC in the late 1970's. MVC was built as a design pattern for Smalltalk in order to implement GUI objects, but has been reused and adopted for other GUI application frameworks along the years. MVC became one of the first approaches to describe and implement software components based on their responsibilities, following a layered approach where the application is divided into three interconnected parts. The Model represents the knowledge that the view is responsible for presenting, while the controller mediates the relationship between model and view. The separation between the major application components allows components to be quickly and easily reuse improving code reusability, and allows parallel development by defining a clear separation between component responsibilities and communication. Although the clear layered concept, the concretization is still abstract generating different implementations depending on the language, platform and application purpose for this architectural approach.

***MVP***

**Model-view-presenter (MVP)** [17] architectural pattern appeared firstly in IBM during the 1990's as a generalization of the classic MVC. Compared with MVC, MVP offers a much greater separation between the presentation layer and the code required to implement the presentation functionality, since it completely separates the model from the view breaking the direct relationship between them, as can be seen in Figure 2.2. This separation generates a change of responsibilities where all presentation logic is passed from the view (turning it in to a passive interface) to the presenter. In MVP the presenter is used as a "middle-man" between the view and model. The presenter interprets user events and gestures and maps them to the appropriate commands for manipulating the model accordingly with the business logic definition. MVP adds a new layer of abstraction by decoupling the presenter directly from the view since it talks to it through an interface. This abstraction enhances portability across platforms or multitier partitioning.

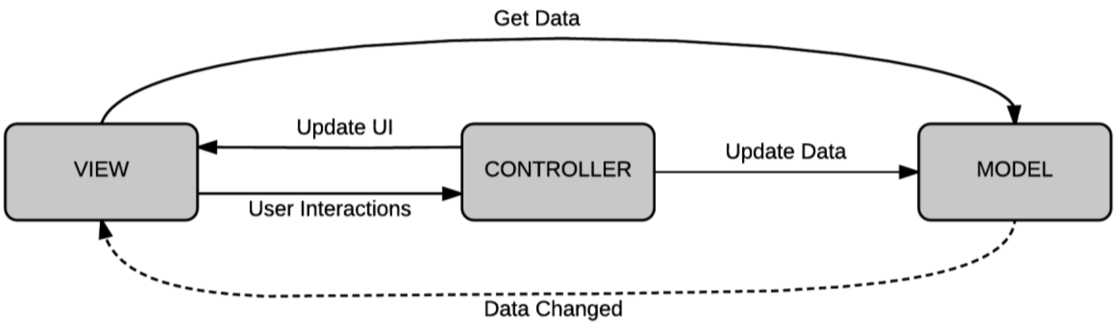


Figure 2.1: Model–view–controller (MVC) pattern.

***MVVM***

**Presentation Model (PM)** and **Model–view–viewmodel** (**MVVM**) [18] where presented in 2004 by Martin Fowler and 2005 by John Gossman, respectively. These two architectural patterns are very similar, since both present an abstraction of a view that contains and manages view's state and behavior. Martin Fowler's PM has a presentation model, while MVVM presents a view model as the abstraction to the view. This abstraction represents state, data and behavior of the view, but without the need to know the control widgets used to render the UI. To do so, data fields for all the dynamic information of the view, including control configuration that can change with user interactions (e.g. disable widget based on user input), must be present in the abstraction class. This layer manages the view's display logic by converting data model objects and exposing it in a structured and prepared way, to be easily managed and presented in the view layer. The view only needs to replicate the state of the presentation model to the UI. In order to maintain this replication between view and presentation model, a synchronization process must be implemented.


Figure 2.2: Model-view-presenter (MVP) pattern.

Compared to MVP, PM and MVVM improves separation between the view and associated presentation logic, since this logic code is completely independent from the view display components. Also, the creation of a containable and testable logical representation of the view, without the need to depend on testing visual elements, is a clear advantage. The downside is the needed synchronization mechanism between the abstract model and the view, but MVVM is simpler in this issue comparatively with PM. MVVM explicitly uses data binding as a simplification to the PM synchronization mechanism (see Figure 2.3). The data binding managed by the view model uses variables that are updated based on GUI changes and the other way around without the need of extra “behind code”. Unlike the Presentation Model in PM, the View Model in MVVM is similar to the Presenter in MVP concerning the encapsulation of commands.

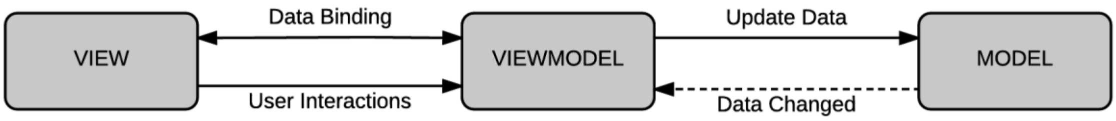


Figure 2.3: Model–view–viewmodel (MVVM) pattern.

***MVW***

**Model-View-Whatever (MVW)** architectural pattern firstly appeared with AngularJS [19] framework. This MVW pattern can be considered a generalization of the classic MVC or a different terminology of the more recent MVVM. In conceptual terms a typical MVW application consists primarily of a view, model, and controller or view model (directive), but in AngularJS these components can co-exist and there are other important components, such as services or filters. Thus, the layer with the presentation logic and responsibility to bind view and model is cataloged only by “whatever” (see Figure 2.4).


Figure 2.4: Model–view–whatever (MVW) pattern.

Regardless of the name, MVW pattern, like MVVM pattern, provides a clear separation of the responsibilities and concerns between the application layers, providing modularity, flexibility, and testability. Also, by taking advantage of the two way data binding, simplifies the event driven programming of the UI.

## 2.2. Information Awareness Mechanism

This section addresses Information Awareness Mechanisms, taking advantage of situational-awareness guidelines, as a way to keep the user well informed about relevant situations (e.g. patient health status degradation or non-fulfillment of prescribed objectives), which need to be rectified to reduce the risk of events or act accordingly with already occurred events, accordingly with different levels-of-concern. Also, user awareness can be used to motivate desired behaviors by presenting rewards (positive or negative), goals or challenges. Gamification plays here an important role [4], providing techniques to improve self-awareness in motivational/alert situations.

***Gamification***

The traditional gamer stereotype as a lazy, male, teenager playing endless hours in front of a computer screen it's refuted by Kumer et al. [20] that present gamers as highly motivated adults in mid-thirties (where forty-seven percent are women), with good education level and full time jobs. This shows the great potential of gaming design elements in business organizations to increase user engagement, motivation and participation.

Games were taken in consideration to create interface heuristics back in the 1980s and around the 2000s researchers started to become more interested in studying game design, game experience and game elements models due to its potential in creating engaging workplaces [21]. However, it was only in 2008 that "Gamification" appeared for the first time and only at 2010 that this term started to be adopted and widespread.

Poor Gamification design, just by adding simple game components and making business applications look like a game, without motivation as the core concern, can result, not only in a fail attempt to engage users, but can also damage the existing engagement with the product/service, resulting in many "gamified" applications failures [20], [21]. Despite the lack of rigorous evaluation studies comparing the use of different methods, principles or tools, concerning process and outcome quality [22], simple game mechanics (like a progress bar, a numeric cost indicator, or a simple leader board) with a clear purpose and business implications are more powerful than complex or visually immersive environments game components that lack a clear goal and definition [20].

In a recent study [4], it was demonstrated that gamification had a positive impact in promoting physical activity and keep all intervenient actors well informed about abnormal healthy or well-being situations. Gamification uses game design elements in order to increase user engagement to motivate desired behaviors, in non-gaming contexts, products and services [22]. According to [21], the design approach must start by analyzing the ability to visualize information related to the "gamified" business (e.g. evolution charts), with a clear categorization of information's relevance, the definition of business metric's thresholds (e.g. indicators) and the ability to identify achievements using a standard classification (e.g. rakings, badges or points).

In chronic patients, health indicators can be characterized by cost (e.g. hours without exercise), benefit (e.g. cholesterol) or on-target values (e.g. hearth rate, body temperature) for control or risk assessment. Each type of indicators has a different relevance based on the user's profile. The adoption of gamification techniques, such as recognition or rakings, can promote motivation sentiments enhancing self-awareness regarding good behavior or achievement of intermediate goals; but for healthcare professionals, indicators related with risk situations are more relevant. Therefore, notifications must be configurable based on the user profile/role.

***Situational and Context Awareness***

Situational-awareness addresses "knowing what is going on" providing a way to keep users aware of situations requiring a rapid intervention. In the health insurance sector, situational-awareness plays an important role [3]. This is particularly true when addressing the surveillance of behaviors of high-risk clients as a way to reduce disease burdens (i.e. preventing adverse clinical situations) and simultaneously to promote a better quality of life and well-being (by reducing health risk factors).

Supporting situational-awareness in complex domains requires the ability to present different types of data in a comprehensive format addressing quick insights [15]. In health insurance, when monitoring chronic patients, any awareness must be well characterized based on the situation triggering an alert. This may include information regarding the client health status (e.g. client's biometric data), changes to the client health factors (e.g. weight growth rate) or knowledge of the service quality (e.g. alert notification response rate). For instance, situational-awareness will inform a nutritionist if a patient outcomes are in line with expected results (e.g. a visual dashboard with diet indicators), or assist a client in accomplishing prescribed physical activities, with data collected from wearable devices.

In this domain, the framework presented by Endsley [23] addresses a model that approaches situation awareness in three levels: 1) perception of relevant elements in the environment, 2) comprehension of the significance of these elements regarding defined goals, and 3) projection of future environment state providing valuable information for decision making. Using this model, Franklin et al. [15] conducted a research to design dashboard visualizations in emergency departments or other clinical environments. They have shown that user-centric dashboard visualizations design, supporting situation awareness, can improve the decision-making process (and ultimately patient care) by increasing awareness, perception and comprehension. Situation awareness, using colors and graphical symbols, enabled healthcare professionals to be alerted for those situations requiring a rapid intervention (e.g. reported measurements values above acceptable thresholds).

## 2.3. Information Visualization

The research field of Information Visualization provides informational artifacts addressing a graphical representation of data in order to enhance human cognition [7]. This section identifies Information Visualization techniques that contribute to streamline the detection of patterns, increasing perceptual inference and improving understanding of the data, and addresses the use of different frameworks and libraries to enable the design of flexible, data-driven, interactive graphical elements. These Information Visualization frameworks will be particularly important to the architecture's solution design and development of the case-study prototype.

***Information Visualization History***

The history of information visualization goes back to the sixteenth century, when men's expeditions used maps and geometric diagrams for navigation and exploration, yet, it was only in the twentieth century that information visualization scientific field become more relevant [8]. In the 1970's "visualization" was the creation of visual images in human mind, but, along the years, visualization definition shifted to be an external informational artifact addressing a graphical representation of data in order to enhance human cognition [7].

The use of graphical representations takes advantage of the human visual perceptual capabilities that can process and acquire more information through vision than through all of the other senses combined [7], reducing search for information by streamlining the detection of patterns, increasing perceptual inference and introducing means of exploring and making sense of data [6], [8], [9].

Initially, information visualization field was used only by research experts, both to explore and analyze large amount of structured data, and as a way to communicate their findings [24]. With the increase need for more effective tools to monitor and analyze data in the business community, information visualization techniques reached a wider audience adoption, generating new challenges in order to address end-users requirement with different levels of expertise and professional backgrounds.

***Information Visualization Techniques***

When combined with other research areas, such as gamification, information visualization techniques has a relevant role in providing value to the existing data, improving self-awareness and information interaction using, for instance, Dashboard viewers [9]. Interactive and personalized dashboards are an information visualization technique to transform datasets into relevant information, supporting the decision making process and dynamically triggering alerts [24]. In a dashboard, data are organized to be monitored and visualized at a glance, making them the ideal communication mechanism for situational-awareness [13].

Taking advantage of different representations (e.g. colors, geometry, size) and visualization elements (e.g. charts, tables or indicators) the dashboard can alert and point the user to those situations requiring an immediate attention, improving the ability to take quick actions based on informed decisions [5]. End-user (taking into account the user profile and goals) should have the possibility to autonomously customize and personalize Dashboard's graphical elements without requiring any programming skills [11]. To improve interactions, the end-user should be able to operate with aggregations, filters and other operators to search and drill-down to more granular data [15].

The schema presented in Figure 2.5 outlines the reference model used to build the Information Visualization solution presented in this thesis. Based on the way users interpret and interact with visual elements in a Dashboard, the reference model [9] presents a three step process to describe how users can configure visual elements in a Dashboard to retrieve insights:

- **Data transformation's step** builds data tables from raw abstract data. Data tables represent data relations and attributes that offer structure and ease of manipulation through, for example, filters, calculated columns or data merge [6];
- **Visual mapping's step** combines structured data and a definition of visual elements (typically, generic visualization mechanisms) to build visual structures with their corresponding attributes;
- **View transformations' step** renders a view, from the visual structures, which can be interpreted by users. Each visual structure can serve one or more visualizations, allowing users to analyze data from different perspectives and levels of detail.

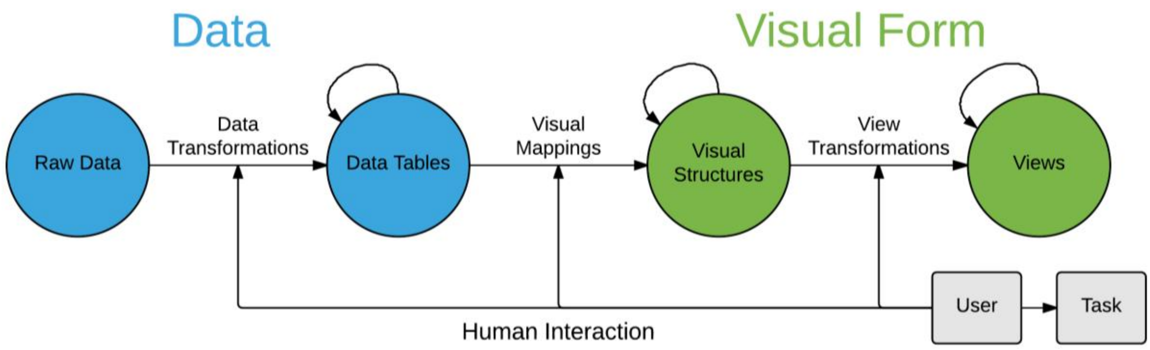


Figure 2.5: Visualization reference model, adapted from [9].

The reference model provided valuable inputs to the design of the domain model associated to the information visualization engine proposed in this work. Card et al. model provides an understanding of the different stages and informational entities used along the iterative process in transforming raw data into visual representations that are relevant to the end-user.

***Information Visualization Frameworks***

Several open-source frameworks and programming libraries have been developed to enable visualization elements' creation, using simple and lightweight programming. These frameworks provide a relevant input to address the creation of informational artifacts addressing the information visualization engine solution. Two main information visualization enabler's frameworks were selected: AngularJS[1] and Angular-nvD3[2].

AngularJS [19] is an open source, client-side JavaScript framework, created in 2009, that supports development of high-productivity, flexible, reusable, and testable web applications. AngularJS promotes the MVW architectural pattern and offers a two way data binding, providing a way to simplify event driven programming, an important simplification to build an information visualization engine platform that aims to provide an interactive data exploration to the user and to improve self-awareness. Also, AngularJS speeds up the construction of informational entities that represents data model state, such as properties related with concepts like relevance or severity, and user interactions that generate model changes.

Angular-nvD3 is an AngularJS directive for a re-usable charting library enabling easily use and customization of charts through a JSON API. This re-usable charting library is based on D3 [25], an embedded domain-specific language offering efficient manipulation of documents based on data. D3 is a JavaScript library that allows code reuse, supports large datasets and dynamic behaviors for user interaction, providing a combination of powerful visualization components and a data-driven approach enabling arbitrary data binding to Document Object Model (DOM) manipulation. Angular-nvD3 speeds up the construction of information visualization components, such as tables, charts or indicators based on a dynamic data model, also supporting user interactions in a flexible way.

[1] https://angularjs.org/
[2] http://krispo.github.io/angular-nvd3

## 2.4. Service Design Thinking

Service Design Thinking is an emerging approach, yet, without any common accepted definition. It highlights the value of users as part of the research process, with an active participation in the service design process, as a way to gain comprehension and understanding of user needs [26]. The Service Design Thinking approach follows five principles [26]:

- **User-centered** (design for user needs): The user-centered principle offers a "common language" during the development process. This principle provides a way to receive input from users, making service design based on user needs rather than the internal needs of the business.
- **Co-creative** (all stakeholders collaborating in the design process): User needs can differ based on the user model/role. Thus, using a co-creative principle, services should be designed in collaboration with the actors to address different needs (e.g. providing motivational self-awareness features to clients or to healthcare professionals).
- **Sequencing** (modeling correlative actions and interactions): The design process requires the understanding of service's actions and interactions to address the user needs, creating value for users. Where, prototyping plays an essential role as a way to test service impact on users and add additional value based on iterative user feedback.
- **Evidencing** (artifacts as outcome of intangible services): Evidencing, the (intangible) design process focus on modeling complete and flexible informational artifacts rather than fragmented components of the identified artifacts, making users aware of the provided services.
- **Holistic** (considering service environment): Finally, service design should be holistic, considering that, special events (variations in general processes) must be considered and treated as standard events. This is particularly important when modeling an event driven Information Visualization Engine platform for an interactive data exploration aiming to improve self-awareness.

In this work, Service Design Thinking principles are used as guidelines when designing and modeling a set of services that ensures the required system behavior in terms of guided surveillance, supporting different awareness mechanisms, triggering alerts and events within a collaborative dataflow to keep all actors well informed.

## 2.5. Dashboard Engine Framework

This section presents a research study on dashboard engine frameworks. Firstly, a dashboard concept survey explores the definition and evolution of dashboards. Secondly, a dashboard engine survey presents dashboard engine key features/aspects and a solution's survey that describes and compares seven dashboard engine framework solutions using thirteen comparison criteria.

### 2.5.1. Dashboard Concept Survey

In order to explore dashboard's concept, this survey starts by approaching the concept evolution. This historic review focuses not only on the existent dashboard solutions' types through time, but also on the technologies that limited or enhanced dashboard's evolution and relevance. Finally, dashboard concept's is clarified by presenting a dashboard definition and description.

***Concept Evolution***

Dashboards relevance has increased in past years, but dashboards appear several years ago. Few [5] presents an historic view where dashboards are the oldest brother of Executive Information Systems created in the 1980s. Executive Information Systems had the purpose of providing financial information in a simple way, but faced a big limitation on data quality and lack of capability to processed data in the desired way. Only after a decade, with BI early stages, the limitations started to diminish, but the new focus of the IT community in the data gathering process cause a slowdown on providing methodologies to make important information available for easy user interpretation. In the same time, at the 1990s, a new type of dashboards toke place: The Balanced Scorecards (nowadays called as Business Performance Management). These Balanced Scorecards uses KPIs and aren't restricted only to the financial domain, providing information for other business' domain metrics. Near the 2000s Balanced Scorecards started to gain biggest relevance and the current dashboard's concept recognition arrived.

As presented before, the progresses over the years in BI platforms capability of data gathering and transformation to a usable form using high-performance databases generated an increase in size, quality and complexity of data, but the ability to use that information efficiently in order to support decision makings kept limited [5], [6]. BI intends to provide an interactive process of exploring and analyzing data in order to gain insights and draw conclusion upon the existing information, but the lack of engagement and interaction with human perception, exploring human intelligence, is damaging its purpose [5]. It was necessary to change the approach and provide solutions to efficiently support informed decision makings. Dashboards can't offer a complete solution, but deftly offers a unique and effective one, regarding the presented problem.

***Dashboard Definition***

In simple words, Dashboards transform datasets into consumable information. Few [5] conducted several researchers on dashboard's related work and in 2004 presented a widely accepted dashboard definition, stating that "*A dashboard is a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance.*".

Dashboards are, first of all, visual displays used as communication mean. The use of visual representations take advantage of human visual perceptual capabilities to acquire and interpret information [7]. Visualizations' design should follows Information Visualization and Information Awareness principles and techniques, presented in the last sections of this related work research.

Dashboards are made to address tasks, goals or objectives, and must have a direct relation to business activities. They must provide the ability to access different types of information, understand data relations and enable user interaction (interactive dashboards). By providing interactive exploration and taking advantage of different representations, such as colors or sizes, the visual elements composing the dashboard, such as charts, tables or indicators, can improve users' data comprehension and reduce search for information [5], [9], [13].

Dashboards should be composed by a set of organized visual components, presented in such a way that information can be monitored and analyzed in a simple way [5], providing quick and easy access to information, "at a glance". Details are important, and the capability to analyze detailed information also, but dashboards should enhance user self-awareness/situation awareness, alerting/pointing to important information and improving the ability to make quick decisions based on simple and clear metrics.

Few [5] presented a dashboard categorization in order to explore techniques and visualizations to enhance usability and effectiveness of the different types of dashboards, users and their goals. Selecting "role" as the most useful variable to categorize a dashboard, Few, presented three major dashboard categories:

- **Strategic dashboards** are the mostly recognized ones. Its purpose is to provide a quick overview to business management users about organization's high-level measures of performance, comparing it in time and to defined goals. Simpler visualizations, with little interactions and moderate update frequency are the typical requirements for this type of dashboards that usually supports long-term decisions.
- **Analytical dashboards** typically help to understand what needs to be explored and it's used to make questions to data, explore scenarios and examine the causes of some business values. These dashboards don't need a real-time update frequency, but requires more complex visualizations and comparisons possibilities, supporting user interactions enabling data exploration.
- **Operational dashboards** are used to monitor operations. Requires a high data update frequency and awareness capabilities, enabling quick responses to generated alerts. Visualizations should be simple, clear, and build to highlight important data that are diverging from target, acceptable thresholds or different alert's nature. Also, must provide user interactions enabling data detail exploration.

### 2.5.2. Dashboard Engine Survey

With the recognition that dashboards and data analysis provide an important tool for companies, a new need to provide quick access to customizable dashboard to end-users appeared [13]. In this section dashboard engine frameworks are analyzed, describing the key features/aspects differentiating such type of engines, presenting an analogous solutions survey and comparative analysis of those solutions.

***Dashboard Engine Key Features/Aspects***

Through the years, many dashboard engine tools assumed that dashboard design, from the end-user point of view, remained constant with small variations [5]. To create new dashboards or customize existing ones, users depended entirely on external tools or dashboard expert designers [13]. Also, multiple actors that had to be involved, making dashboard's creation process inefficient due to the complex communication coordination of all actors [24]. Users depended on business analysts to bridge the gap between the business needs (user needs) and the dashboard specifications.

Dashboards must be an end-user-centric informational artifact and a dashboard's builder engines must empowers end-users in the building process. Nowadays, research projects [12], [13], [24] focus on engines with the ability to allow end-users to build dashboards and visualizations, and multiple commercial [10], [27], [28] and open source solutions [11], [29] enables end-users to build, customize and compose dashboards and visualizations without any programming skills and without depending on expert dashboard designers. Regardless this new approach of "independence" from IT or dashboard designers experts, to achieve an effective dashboard, the creation process still should be a collaborative effort and created dashboard interfaces must support a diverse of user profiles and workflows [15]. Thus, dashboard engines need to provide a creation model that gives "freedom" to the end-user, but also takes advantage of different user profiles and expertise.

Using, North and Shneiderman [14] three level's flexibility classification, a dashboard engine should be flexible in data selection, allowing different datasources, should be flexible in visualization's type, allowing the use of different visualizations, and should be flexible in data relations, allowing user's interaction in one visualization to affect other visualizations. A dashboard engine must be flexible and generic enough to be able to create dashboards supporting different categories (such as the described in the last section). This type of engine must be capable of supporting the creation of dashboards for different needs, data domains, data types, interactivity types, data update frequency, visualization complexity and other variables. Nevertheless, dashboard engines with greater flexibility and expressiveness shouldn't forget the tradeoff between flexibility and accessibility, minimizing the need of a greater user expertise in data manipulation and visual representation [24] in flexibility solutions.

***Solutions Survey and Comparative Analysis***

A set of analogous solutions were selected due to their ability to build online interactive dashboards and visualizations, without using programming environments or hard in-depth analysis platforms with focus on complex BI analysis. The selection included three research project solutions (Snap-Together Visualization (Snap) [14], Exploration Views (EV) [13] and VizDeck [12]), one open-source solutions (SpagoBI[3]) and three commercial solutions (Microsoft Power BI[4], Tableau Desktop[5] and Pyramid Analytics BI Office[6]) with released versions until May 2017.

[3] spagobi.org
[4] powerbi.microsoft.com
[5] tableau.com
[6] pyramidanalytics.com

A detailed version of solution's analysis can be found in the appendix section, whereas this section focuses on describing the conducted comparative analysis and its results. To notice that, although the inclusion of research project solutions in the current survey provides a relevant input in analyzing different dashboard engines approaches, the comparison analysis scores are relevant only on commercial an open source solutions with enough ability to fulfill the majority of the selected criteria.

The comparative analysis followed Gartner's [30] approach, using a score table (Table 2.1) and a list of criteria to compare the selected analogous solutions, building the comparison grid present in Table 2.2. The score table intends to measure criteria requirements, classifying if all, some, or none of criteria requirements are meet. Also, depending on relevant remarks on quality, complexity or flexibility of implemented solutions an increasing score can be applied. The base scale goes from 1 (not achieved) to 3 (meet requirements) and, with a possible increase of 0 to 1, the final score scale goes up to 4 (increased value).

Table 2.1: Comparative score table adapted from [30].

| Score | Description |
|---|---|
| 1 | Not achieved: most or all defined requirements for a capability are not achieved |
| 2 | Partially achieved: some requirements are achieved |
| 3 | All achieved: meets requirements |
| 4 | Increased value: meets requirements and exceeds in quality, simplicity and/or flexibility |

From the initial fifteen Gartner's critical capabilities criteria, only ten were considered and four were merged into two. This selection excluded criteria relevant for full BI Platforms, but not Information Visualization Engines (e.g. Self-Contained ETL and Data Storage) and criteria witch evaluation out of the scope of this thesis (e.g. Ease of Use). Some other criteria [6], with focus, not only on functional aspects, but also on non-technical aspects were considered, allowing the enrichment of the already selected criteria and resulting in one new criterion: Licensing. Finally, two more criteria were added: Self-Service Visual Preparation and Event-Based Awareness. Self-Service Visual Preparation replicates Gartner's Self-Service Data Preparation, but focusing on building visualizations to use in dashboards. Event-Based Awareness capability addresses the ability to configure and visualize/notify (be aware) alerts depending on different levels of concern. The criteria selection process resulted in a final list of eleven comparison elements (see appendix section for a detailed version).

The solutions' survey and comparison analysis aims to demonstrate proposed solution's relevance, identifying existing user requirements and opportunities of improvement. Accordingly with the comparison survey grid, represented in Table 2.2 (detailed in appendix Table 0.1), some conclusions can be drawn. Criteria C9 (Publish and Embed Analytic Content) is a relevant requirement addressed by several analogues solutions to provide the possibility of creating and modifying analytic content, visualizations and applications, embedding them into a business process, and/or an application or portal. Although, the proposed engine doesn't address this criterion in this essay is considered future work, further discussed in relevant section. In criteria C6 (Metadata Management), C8 (Interactive Dashboards), C10 (Collaborative Work) and C11 (Event-Based Awareness) all analogous solutions have global scores below 3 ("meet requirements"), exposing an improvement opportunity for the proposed information visualization engine.

Table 2.2: Information Visualization Engine analogous solutions' comparison analysis using eleven comparison criteria and a 1-4 score scale.

| | Proposed Engine | Tableau | PowerBI | Pyramid | SpagoBI |
|---|---|---|---|---|---|
| C1: Licensing | - | C[1] | C[1] | C[1] | OS[1] |
| C2: Admin, Security and Architecture | Y[2] | 3,00 | 3,00 | 2,88 | 2,50 |
| C3: Data Source Connectivity | Y[2] | 3,00 | 3,00 | 3,00 | 3,00 |
| C4: Self-Service Data Preparation | Y[2] | 3,17 | 2,83 | 3,00 | 2,50 |
| C5: Self-Service Visual Preparation | Y[2] | 2,90 | 3,00 | 3,20 | 2,60 |
| C6: Metadata Management | Y[2] | 2,13 | 2,50 | 2,50 | 1,88 |
| C7: Interactive Visual Exploration | Y[2] | 2,63 | 3,13 | 2,88 | 2,50 |
| C8: Interactive Dashboards | Y[2] | 2,60 | 2,80 | 2,50 | 1,90 |
| C9: Publish and Embed Content | N[2] | 3,00 | 3,00 | 3,00 | 3,00 |
| C10: Collaborative Work | Y[2] | 2,50 | 1,00 | 2,75 | 2,38 |
| C11: Event-Based Awareness | Y[2] | 2,33 | 2,50 | 2,33 | 2,00 |
| | **Final Score:** | **2,73** | **2,68** | **2,80** | **2,43** |

Legend: [1] C: Commercial; OS: Open source; [2] Y: Yes, addressed by the proposed engine; N: No, not addressed by the proposed engine.

C6 (Metadata Management) criterion represents the quality of the metadata model in terms of expressiveness, completeness and standardization. That is, the ability to create and customize, in a centralized way, semantic model objects, and the ability to extend this model with new metadata fields. The proposed information visualization engine addresses an expressive and complete model, allowing the extension of metadata, through user metadata, an innovative feature comparing to other solutions.

C8 (Interactive Dashboards) criterion represents the construction and customization of highly interactive dashboards with visual exploration. The proposed information visualization engine address a flexible and customizable way to build dashboards, providing a differentiator ability to interact and customize visualizations' views instantiated at the dashboard. This approach allows to address a sub criteria (see criterion C8.3 in Table 0.1 - measure the possibility of changing visualization type directly in dashboard design) that all compared solutions failed to meet.

C10 (Collaborative Work), a criterion with increased recognition over the years [10], [29], represents the possibility of sharing information in form of comments/annotations or discussion threads/chats. The proposed information visualization engine addresses improvements in sharing information flexibility, enabling the creation of notes, comments or ratings through all levels (all informational entities) of the engine.

C11 (Event-Based Awareness) criterion represents how the engine addresses situational-awareness and the ability to configure, visualize and notify events and alert situations that require user attention. The solutions' survey showed that existing solutions lack in capability to address user awareness improvement, and that the proposed solution can provide a valid improvement. The proposed information visualization engine addresses informational-awareness in order to provide a flexible way to configure and address alerts depending on different levels of concern.

## 2.6. Design Science Research

Design Science Research (DSR) has played an important role in Information Systems (IS) research since its appearance and the IS community acceptance to DSR is increasing. The timeline presented in Figure 2.6 intends to show the evolution from first contributions, to current date, focusing in several important additions in DSR maturity and development.

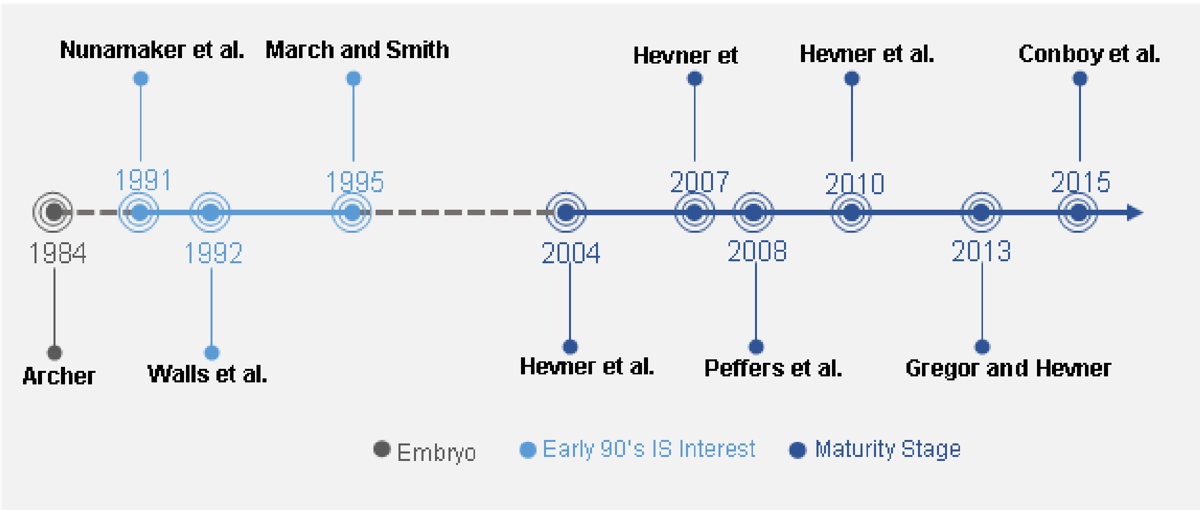


Figure 2.6: DSR evolution timeline, adapted from: [31]–[33]

A DSR preliminary proposal can be found in 1984, but was in the early 1990s that using DSR with proven benefits started to be seen as new approach with increased interest by the IS community [34]. At that point there was already an agreement in the basic differences between DSR and other research paradigms. March and Smith [35] addresses DSR output as one important differentiator characteristic, saying that "*Whereas natural science tries to understand reality, design science attempts to create things that serve human purposes*".

However, it was only in 2004 that the DSR was accepted as a research methodology (DSRM). The trigger to this recognition was the paper published by Hevner et al. [33]. This paper presented DSR as a way to formulate hypothesis to addresses important unsolved problems in unique or innovative ways, as well as to approach well-known problems in a more effective or efficient way. The generated IT artifacts (one of the main outcomes from DSR) were defined as constructs, models, methods or instantiations to create informational artifacts relate to IS in the organization [31].

Hevner et al. [33] describes IS research procedures in seven guidelines (present in Table 2.3). These guidelines provide understanding of a well conducted research process and are, in some way, a common sense notions of business utility. (1) DSR address a problem with the production of artifacts, such as constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype systems) [35]. (2) Those artifacts must be relevant to address, as said above, “important unsolved problems”, and (3) must be measured in a rigorous evaluation to guarantee utility, quality, and efficacy of the solution. (4) Effective DSR should make knowledge contributions and clear novelty contributions to the real-world applications environment [31] ensuring relevance by feedback from application in the appropriate environment, and (5) ensuring rigor by applying suitable methodologies and foundations when developing and evaluating the result. (6) This process must be a search process ensuring that DSR contribution must be functional, “utilizing available means to reach desired ends”. (7) Finally, the research must be presented, in an effective manner, both to technical and management audiences.

Table 2.3: DSR Guidelines, 2004, source: [33]

| Guideline | Description |
|---|---|
| 1: Design as an Artifact | Design science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation. |
| 2: Problem Relevance | The objective of design science research is to develop technology-based solutions to important and relevant business problems. |
| 3: Design Evaluation | The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods. |
| 4: Research Contributions | Effective design science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies. |
| 5: Research Rigor | Design science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact. |
| 6: Design as a Search Process | The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment. |
| 7: Communication of Research | Design science research must be presented effectively to both technology-oriented and management-oriented audiences. |

Hevner et al. [33] also defined an IS research framework in 2004, furthered extended in a 2007 paper about the Three Cycle View of DSR, presenting three cycles, as we can see in Figure 2.7. These cycles can be clearly mapped to the seven described guidelines and provides extra depth to the guidelines and proposed process that focus on the existing bridges between Design Science activities, contextual environment and knowledge base.

The Relevance Cycle addresses problem relevance, defining requirements and aims to recognize opportunities and problems to improve the Environment. But also provides criteria for evaluation acceptance and active user (“Application Domain People”) involvement in the design iterations, providing necessary feedback throughout the whole process [32] making DSR an user-centric paradigm. The Design Cycle is the core iterative cycle. At this stage an artifact must be generated and demonstrate/evaluate. The feedback from this step provides the ability to refine the artifact design in a iterative way. It can be a formal evaluation process, a simpler demonstration proving the success of the artifact or both [34], but either way, is an essential part to achieve an artifact that meets the defined objectives. The Rigor Cycle represents the bridge with the knowledge base where past knowledge feed design cycle ensuring innovation. Also contributes with new additions to the knowledge base based on experience from the iterative cycle and results of the artifact design. This contribution is accomplished in the proper communication of the research.

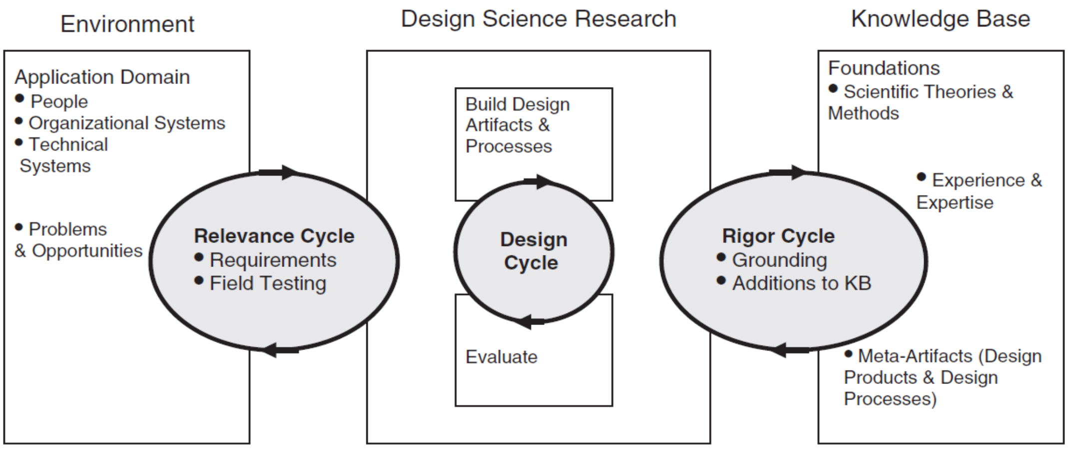


Figure 2.7: DSR Three Cycle View, 2007, source [33]

After the 2004 publication, and with newly and renovated recognition in the IS research, new important contributions appeared in DSR [33]. Despite this success, the process model wasn’t consensual and that’s why Ken Peffers et al. [34] proposed a synthesized generalizable process model for DSR in IS. This effort was built on prior approaches (as we can see in Table 2.4), generating a useful general model [31]. DSRM Process Model consists in a six step nominal sequence: Problem definition and motivation, define solution objectives, design and development, demonstration, evaluation and communication.

Peffers et al. generalized model has been further used and refined over time by other researchers. Gregor and Hevner [31] harmonize some different opinions in prior publications and provides guides to understand DSR knowledge contributions, developing a DSR knowledge framework and publication schema that contributes to the current work. Accordingly to them, different types and levels of contributions can be achieved depending on problem maturity (current problem context maturity) and solution maturity (current maturity of potential starting point artifacts). They propose a DSR knowledge contribution framework, focusing on knowledge starting points in current research to define project goals and new contributions of four types: routine design (apply known solutions to known problems), exaptation (extend known solutions to new problems), improvement (new solutions to known problems) and invention (new solutions to new problems). Since the proposed work's contributions are located in the improvements section, these types of contribution are further discussed.

Improvement contributions require broad understanding of problem domain in order to build novelty solutions to important problems. The reasons for the needed improvement should be clear and grounded on the knowledge base. Also, needs to be able to clearly demonstrate that the proposed solution really improves on current solutions and previous knowledge over design artifact evaluation. Depending on the nature of the research, improvements can be in efficiency, productivity, quality, competitiveness, market share, or other types of measures. This is what we intend to describe in the next section concerning our essay.

Table 2.4: DSRM Model State of the Art, 2008, source: [34]

| Objectives for a design science research process model | Archer (1984) | (Takeda et al. 1990) | Eekels and Roozenburg (1991) | Nunamaker et al (1991) | Walls et al (1992) | (Rossi et al. 2003) | (Hevner et al. 2004) |
|---|---|---|---|---|---|---|---|
| **1. Problem identification and motivation** | Programming<br>Data collection | Problem enumeration | Analysis | Construct a conceptual framework | Meta-requirements<br>Kernel theories | Identify a need | Important and relevant problems |
| **2. Objectives of a solution** | | | Requirements | | | | Implicit in "relevance" |
| **3. Design and development** | Analysis<br>Synthesis<br>Development | Suggestion<br>Development | Synthesis, Tentative design proposals | Develop a system architecture<br>Analyze and design the system.<br>Build the system | Design method<br>Meta design | Build | Iterative search process<br>Artifact |
| **4. Demonstration** | | | Simulation, Conditional prediction | Experiment, observe, and evaluate the system | | | |
| **5. Evaluation** | | Confirmatory evaluation | Evaluation, Decision, Definite design | | Testable design process/product hypotheses | Evaluate | Evaluate |
| **6. Communication** | Communication | | | | | | Communication |

In 2015, Conboy et al. [32] addressed an agile approach for DSR. Figure 2.8 represents the Agile DSRM (ADSRM), with DSRM as base (black lines), and new agile additions (red lines). The methodology steps are described below in a brief overview that aims to present the base model applied in this research. ADSRM first step is problem identification and motivation. In this step the new agile approach proposes two changes: A new problem backlog component and the possibility of feedback from further steps to this problem backlog and problem identification, allowing the iterative design cycle to provide changes to the problem space. This enriches the model with the possibility to mature the problem space after design processes, prototypes, and/or feedback from users, depending on the needs, interests and concerns of user profiles.

The first sub-process includes definition of the problem, as guidance for the following processes iterations and importance of the solution in a way that becomes clear the solution relevance, results acceptance and the understanding of the problem. This is included in the DSR cycle, described before, and defined as the Relevance Cycle.

As presented in Table 2.4, the sub-process of defining objectives (step two) isn't consensual. Although, defining objectives, either in an explicit process step or an implicit part of other process step is definitely a relevant part of the iterative process and must reflect the defined design problem. Agile methods also focus on non-functional requirements and quality constraints, resorting to active user involvement throughout design iterations. In particular, [32] highlights the addition of use cases (taking into account user experience and semantic associated with user profile) during the design and development in order to identify divergences not otherwise outlined in initial requirements.

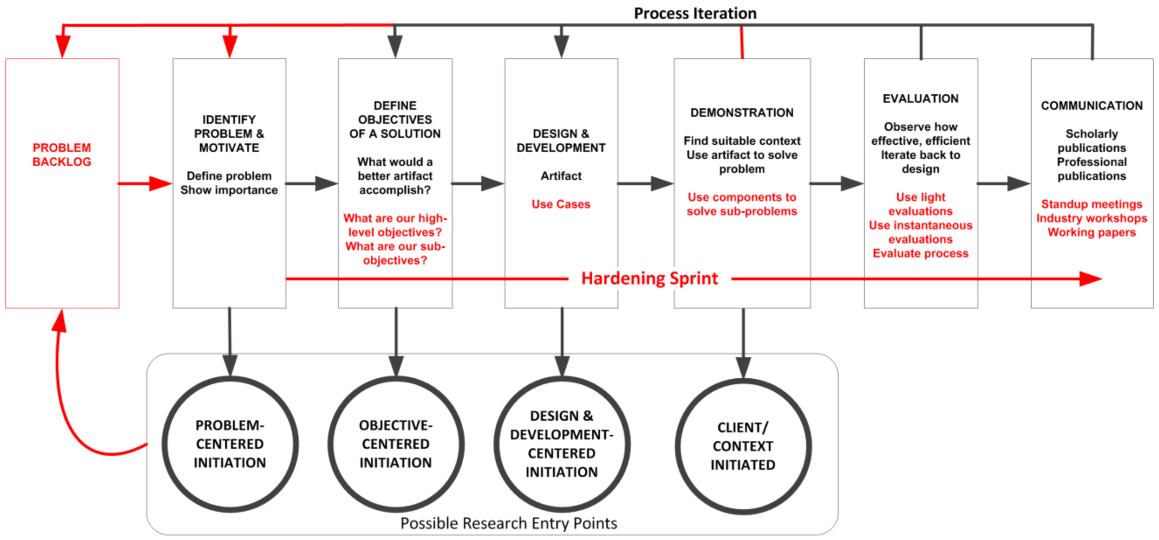


Figure 2.8: ADSRM, 2015, source [32]

The following steps are design and development, demonstration and evaluation. These sub-processes are the core of the methodology and the center iterative cycle in DSR. At this stage, design and development step, must generate an artifact since it represents the key contributions of the research [31], [33], [35]. Then the artifact must be evaluated. It can be a demonstration in form of an experiment, simulation, case studies, proof, or other meaningful activities, to evidence the possible design application, and how that can be done. It can also be a formal evaluation process, requiring relevant evaluation techniques and metrics depending on the type of problem and artifact, or both, but either way, it is an essential step to achieve an artifact that meets the defined objectives. The feedback from this step provides the ability to refine the artifact design in an iterative way.

ADSRM proposes simpler, dynamic and iterative evaluation methodologies and metrics to improve the iterative process. Also, promotes evaluation, not only of the artifact, but equally of the agility supplied by the artifact and by the process from which was created. Since a "lighter" evaluation can put in jeopardy an essential part of DSR, the rigor [33], there was a need to balance agility and rigor. This balance is provided in the form of a "special" iteration (the Hardening Sprint) that freezes the problem (losing agility to gain rigor), freezes the process (transforms agile methodology of "people over process" to "process over people") and adds rigor-driven parts to the process (for example, add extra measures in evaluation step) in order to enhance the rigor.

Finally, the communication step include the communication of the problem, its relevance and opportunities, the artifact itself, the innovation or the design rigor in many forms as papers and publications. Although we addressed design as the main iterative cycle, communication it's also essential, since accordingly with Gregor et al. [31], "*improvement DSR is judged first on its ability to clearly represent and communicate the new artifact design*". Having described the process steps, we must clarify that there is no obligation to precede in a sequential order from 1 to 6 due to the multiple entry points for different approaches, as seen in Figure 2.8.

This thesis follows the agile methodology approach [32] that gives more detailed importance for user participation (user-centric) over each iteration process and manages iteration cycles in a much more agile form, but without losing the rigor needed to address the creation of informational artifacts that presents relevant contributions to the knowledge base.

# 3. Information Visualization Engine

The proposed solution follows a client-server web-based architecture supported by a metadata info-structure. The main features were designed to assure scalability and a high level of configurability, for a multi-user environment with authorization procedures based on the user model (user roles/profiles). This work assumes that all incoming data are ready to be visualized, focusing on how to present the information to better understand the health-status of the patient.

The design followed a user-centric approach, providing a solution to enhance end-users' flexibility in the creation and customization of an interactive dashboard as a way to present events and data to different user profiles in an understandable way, providing quick insights and minimizing user's effort to interpret and understand the reported information. The informational artifacts are presented in each tab element of the dashboard (i.e. data addressing the information needs and interests of the user), and each artifact is expressed as a set of customized content instances. Based on the user profile/role the awareness level might change, e.g., the number of events triggered to keep the user informed about those situations requiring an immediate attention. Such level of awareness is mostly expressed though the configuration of Content/Indicators elements responsible to dynamically trigger notifications whenever required. The dashboard can hold a set of different Content Types, each one customized based on the user preferences and access level.

## 3.1. System Conceptual Model

The system's informational artifacts are structured into six layers, each one encapsulating specific aspects of the information visualization engine. The schema in Figure 3.1 outlines this layered approach following a classical 3-tier architecture model: presentation layer (D) responsible to present the data graphically; business logic layer (C) responsible to transform the data based on existing business rules; and data layer (B) responsible to implement the business rules or any other business logic to be applied to raw data. The schema is complemented with two additional layers encapsulating functionalities related to security (E) and collaboration aspects (F). Finally, a metadata layer (A) is responsible to handle and manage the info-structure of metadata used to characterize some of the informational artifacts within the domain model. To streamline the association of the layers in this schema with the informational artifacts in the system domain model, a color code (A:Blue, B:Orange, C:Beige, D:Brown, E:Gray, F:Pink) was adopted in Figure 3.2.

**Metadata Layer.** Addresses the characterization and management of the informational artifact's metadata in a centralized way for all reported artifacts. The *Entity* artifact (see Figure 3.2) encapsulates base attributes (e.g. entity version attribute) and functionalities (e.g. adding a note to a *Dashboard* or to a *Content* is generalized by adding a note to an *Entity*) responsible to characterize artifacts holding data from each layer of the architecture model (Entity Type). Each *Entity* attribute is characterized by the association of a *Metadata* description to a value. For example, an Indicator has a *Metadata* attribute named "Type" that can have the following values: "Benefit", "Cost" or "On-Target", and a set of Boolean attributes indicating if it is required, has no default value, isn't read only, is functional or userMd. These last two ones are particular relevant for customization issues. When the userMd is set to true it means that the user can create and customize existing metadata attributes, providing a flexible and simple way to extend the metadata model. When functional attribute is set to true the user is able to add functionalities to the engine (e.g. to inject a value in a SQL query).

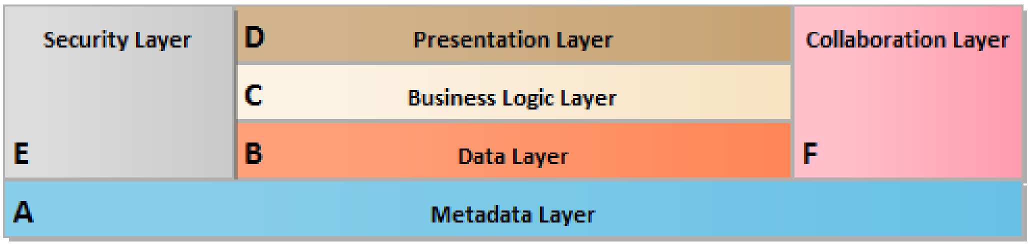


Figure 3.1: High level overview of the model internal structure

Each *Metadata* attribute can be assigned to a group, providing the ability to relate them in different ways. The group relation is used by the information visualization engine to address related fields in the same section/tab (e.g. adding a user metadata to the a content in the "Options" *Metadata Group* would dynamically add that field to the "Options" tab in the content configuration view, whereas, adding the new metadata to a new *Metadata Group* "FH" would dynamically add a new tab "FH", in the content configuration view, with a single field).

**Data Layer.** Addresses the informational artifacts required to manage data connectivity and information gathering, providing the ability to retrieve and structure data for visual mapping. The proposed model follows Card et al. [9] approach using raw abstract data to build more complex structured data, separating the following two concepts: *DataSources* for data connectivity and access to heterogeneous data, and *DataItems* to structure the source data in a set of *Columns* with functionalities to filter and configure how to visualize the data to work with.

*DataSources* can be of four types: REST, JDBC, MDX and CSV. Each *DataSource* type has different metadata fields to characterize the data connection. The *DataItem* configuration depends on the *DataSource* type (e.g. on a *JDBC DataSource* the *DataItem* uses an SQL query, but on a *REST DataSource* the *DataItem* selects the root node). *Columns* are used to describe heterogeneous data in a structured way. For instance, to build the contents presented in the case study, a *JDBC DataSource* was configured and used to build a Weigh Metrics *DataItem* (see Figure 4.2). Data selection originated the creation of five *Columns* used on further layers to build visualizations, such as the BMI metric column used to build the Weigh Evolution chart and the BMI indicator (see Figure 4.4).

**Business Logic Layer.** Addresses the management of informational artifacts used to fill the workspace of the tab elements in each configured dashboard. After defining *Content's* connection to a specific *GraphicalObject*, a default visual structure is presented (e.g. graphical mode or tabular mode). In the graphical mode the user can select a set of multiple graphical elements (e.g. pie chart, donut chart, bar char, multi bar chart, line chart, cumulative line chart, scatter chart, stacked area chart, indicator card, combo box or list box). *GraphicalObjects* (visual components) offers an abstraction for the implemented visualization mechanism.

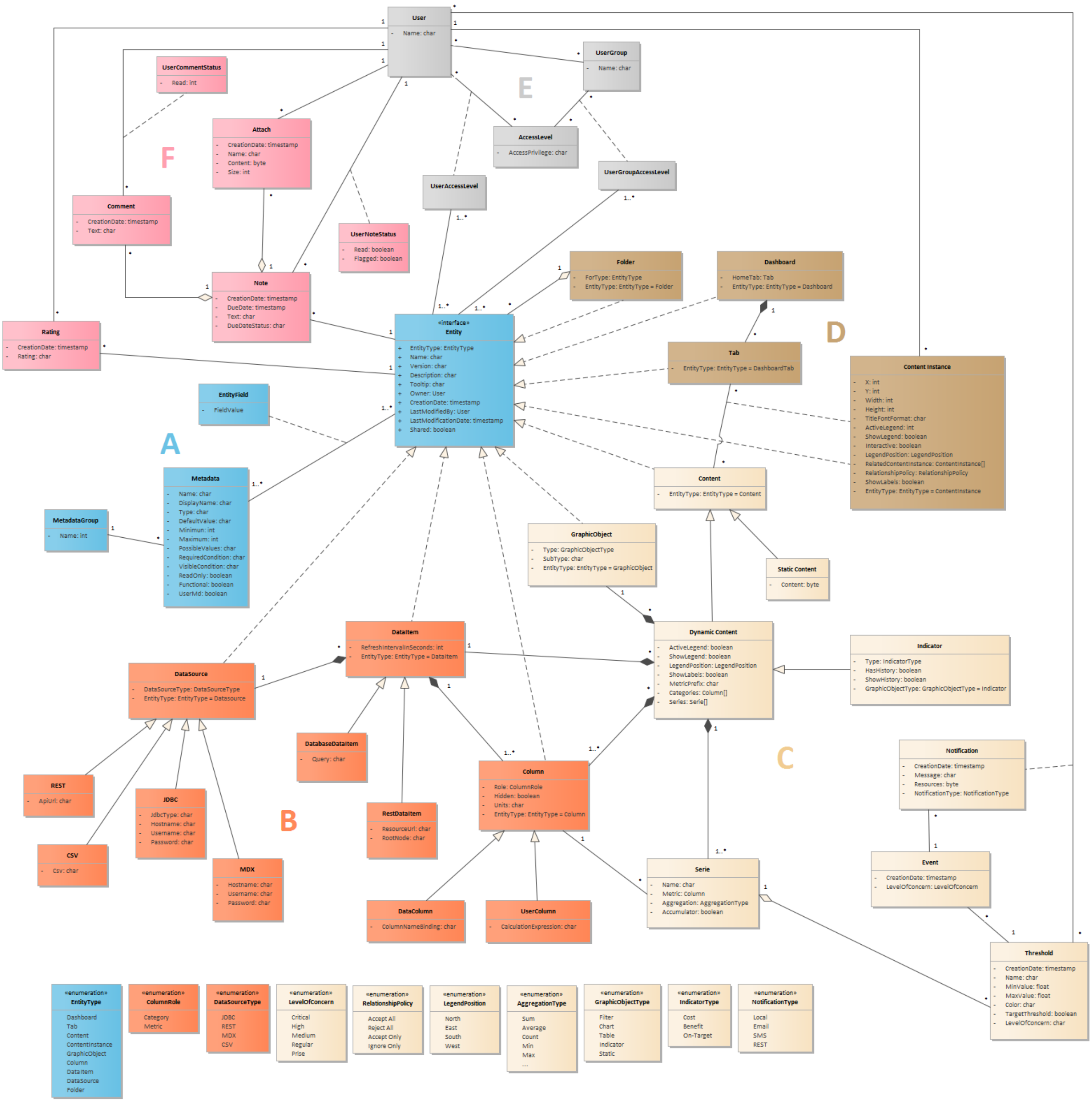


Figure 3.2: Metadata Model of the Information Visualization Engine

Using *Content's* connection to a specific *DataItem*, a set of functionalities is available to customize how the values of the data are going to be visualized. *DataItems* and corresponding structural data components (i.e. *Columns*) are associated to *Contents* using categories and series. Categories are a set of *Columns* used to define the "X-axis" values and metric aggregation dimension. By defining more than one category we can configure a multi-value/level content (the levels created by this set of columns can be used as custom hierarchies for data visualization). *Series* define the "Y-axis" values using a metric (*Column*), an aggregation function and an accumulation flag. The metric value is calculated based on the aggregation function (with or without accumulation – e.g. Daily steps sum with accumulation shows in day two the sum of steps in day one and day two) and the selected set of categories.

*Contents* provide the configuration of *Thresholds* and *Alerts* with a spatio-temporal context using colors and shapes to draw users' attention to relevant data and enhancing information-awareness. A *Threshold* is characterized by a value interval defined with a minimum and maximum acceptable value, and an indication of the target threshold. For each value range, an associated color is defined (e.g. red, yellow, green), providing a visual representation (i.e. color-code) to the values of the incoming data. *Thresholds* can raise *Alerts* and *Notifications* to "subscribing" *Users*. Based on the *Threshold* values, events are generated and identified with different symbols/colors.

*Contents* have a specialization named *Indicator*. The standard configuration supports three types of indicators: cost, benefit or on-target indicators. This typification enriches the content definition characterizing value variation (e.g. using color - green for benefit or red for cost) for awareness enhancement (e.g. indicator showing a positive value, using green as the threshold's configured color, but a negative increase variation). *Contents* also enable the use of gamification techniques to support situational-awareness (e.g. an *Indicator* used as leaderboard/ranking or a *Content* providing evolution information). The access to these informational entities is based on the user profile.

**Presentation Layer.** Addresses the informational artifacts required to create, view and customize an interactive context dashboard. A dashboard provides a tab-based structure where visualizations (*ContentInstances*) can be added and customized based on the user preferences, empowering end-user's customization and interactivity. The separation between the *Content* (visual structure) definition and *ContentInstance* (view) enhances reusability and collaborative work, since the same *Content* can have several instances viewed by multiple users from different perspectives.

*ContentInstances* in each tab can interact with other *ContentInstances* hold by the same tab. Such tab-based approach combined with the user role/profile provides a way to manage access to specific information on each tab, allowing users to analyze and monitor data from different perspectives. In our case study (e.g. Figure 4.4), the presented dashboard has multiple tabs (see Figure 4.1) with different accesses based on user role/profile and where the same *Content* can be instantiated with different configurations. For instance, different health professional actors (e.g. nutritionist or nurse) could access the Health Professional tab and see the Cholesterol *Indicator* in different ways, for example, a classic indicator view or a line chart evolution view (by activating *Indicator's* "ShowHistory" metadata). *Folders* provide an organized tree list view of the informational entities created, where the listed information depends on user role/profile.

**Collaboration Layer.** Addresses the informational artifacts used to enhance collaborative work, sharing information in the form of comments/annotations, discussion threads/chats or ratings. The proposed model presents an flexible solution in collaborative work capabilities by providing the possibility of add *Notes* (with *Attachments*) and *Comments* through all levels of the application, using different informational entities, such as *Dashboards*, *Tabs*, *Contents*, *Columns*, and more.

Collaboration work is not only a capability with increased recognition in last year's [10] in any analytic platform (e.g. visual analytic dashboards), but it is a feature also relevant to improve situational-awareness by providing the status of read (notes/comments) and flagged (notes), which might include visual insight of new comments (that may need intervention, for example, a new medical prescription based on some indicator) or relevant (flagged) information enabling an memory aid. On the other hand, *Ratings* can be used based on gamification techniques to reward good behavior (e.g. physical trainer rating 5 starts to a weight evolution chart that the client will see in his dashboard).

**Security Layer.** Addresses informational artifacts that instantiates the user model and access levels' configuration. The access levels can be configured by user group/role or for specific users restricting access privileges to different informational entities, such as dashboards, tabs, visualizations, data structures, and more. To notice that the user informational entity, although cataloged in security layer, plays an important role in the presentation layer (providing the ability to configure visualization instances content instances by user) and plays an important role in the business logic layer (allowing the configuration of different notifications for different users).

## 3.2. System Architecture

In this section, decisions behind the information visualization engine system design are presented. The design was driven to address a user-centered design approach, supporting rapid and agile iterative prototyping to create informational artifacts delivering a functional prototype for a real-world health insurance context. The aim of this section is to explain the rationale behind the adopted solutions, providing an analysis on the key architectural components and technological decisions.

### 3.2.1. Architecture Overview

The information visualization engine follows a client-server architecture (see Figure 3.3) comprised of an AngularJS JavaScript client, a Java backend server and a metadata info-structure database for engine's configurations. This architecture manages multi-user access based on the user mode (user roles/profiles).

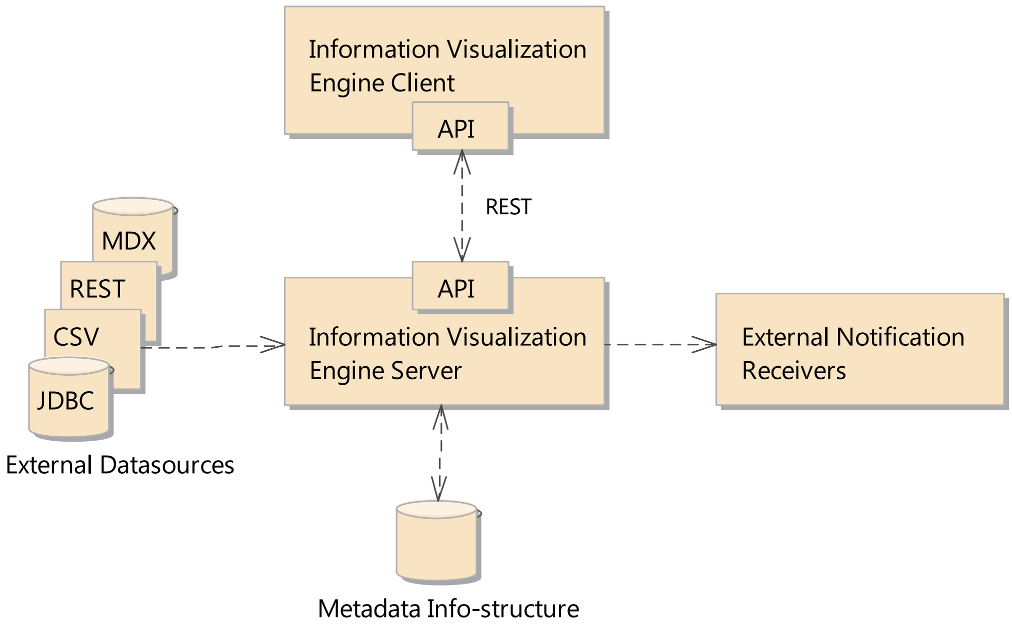


Figure 3.3: Information visualization engine client-server architecture overview

The server provides access to external data sources and manages the metadata info-structure that supports the designed model using a created API. Also, enables the dispatch of external notification alerts based on thresholds and notifications' configuration, providing an important contribution for situational-awareness. The communication between server and client is supported by a RESTful API with JSON messages that are serialized and de-serialized in each communication.

The client addresses the creation of visual analytics dashboards for situational-awareness following a layered approach with a clear separation of the responsibilities and concerns between the application layers, providing modularity, flexibility, and testability to the implementation. Uses an event driven approach to enable situational-awareness with the construction of informational entities representing data model state, related with concepts like relevance or severity in a spatio-temporal dimension/context (when and where it occurred) providing the ability to draw users' attention to relevant data through color, or shapes.

### 3.2.2. Client-Side Architecture View

This section addresses client's architecture view focusing on the main software components present in each architecture layer (see Figure 3.4 for high-level view and Figure 3.5 for a detailed view). Also, addresses the differentiator technologies used and how they provide solution for user requirements and the graphical objects built to support the information visualization engine.

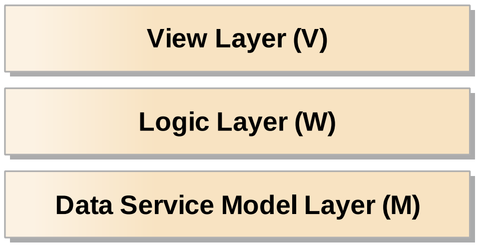


Figure 3.4: Client architecture layers

The proposed client architecture uses a layered approach that follows a Model-View-Whatever (MVW) architectural pattern. By using MVW architectural pattern it's achieved a clear separation of the responsibilities and concerns between the application layers, providing modularity, flexibility, and testability to the implementation. This approach takes advantage of some simplifications that take an important role when implementing interactive dashboards GUI and when managing informational artifacts' data model state and data representation.

MVW data binding, implemented through AngularJS framework's two way data binding, provides a way to simplify management of event driven programming. This event driven programming simplification is important in a platform that aims to provide an interactive data exploration to the user and to improve self-awareness. Also, AngularJS two way data binding speeds up the construction of informational entities that represents data model state, such as properties related with concepts like relevance or severity, and user interactions that generate model changes.

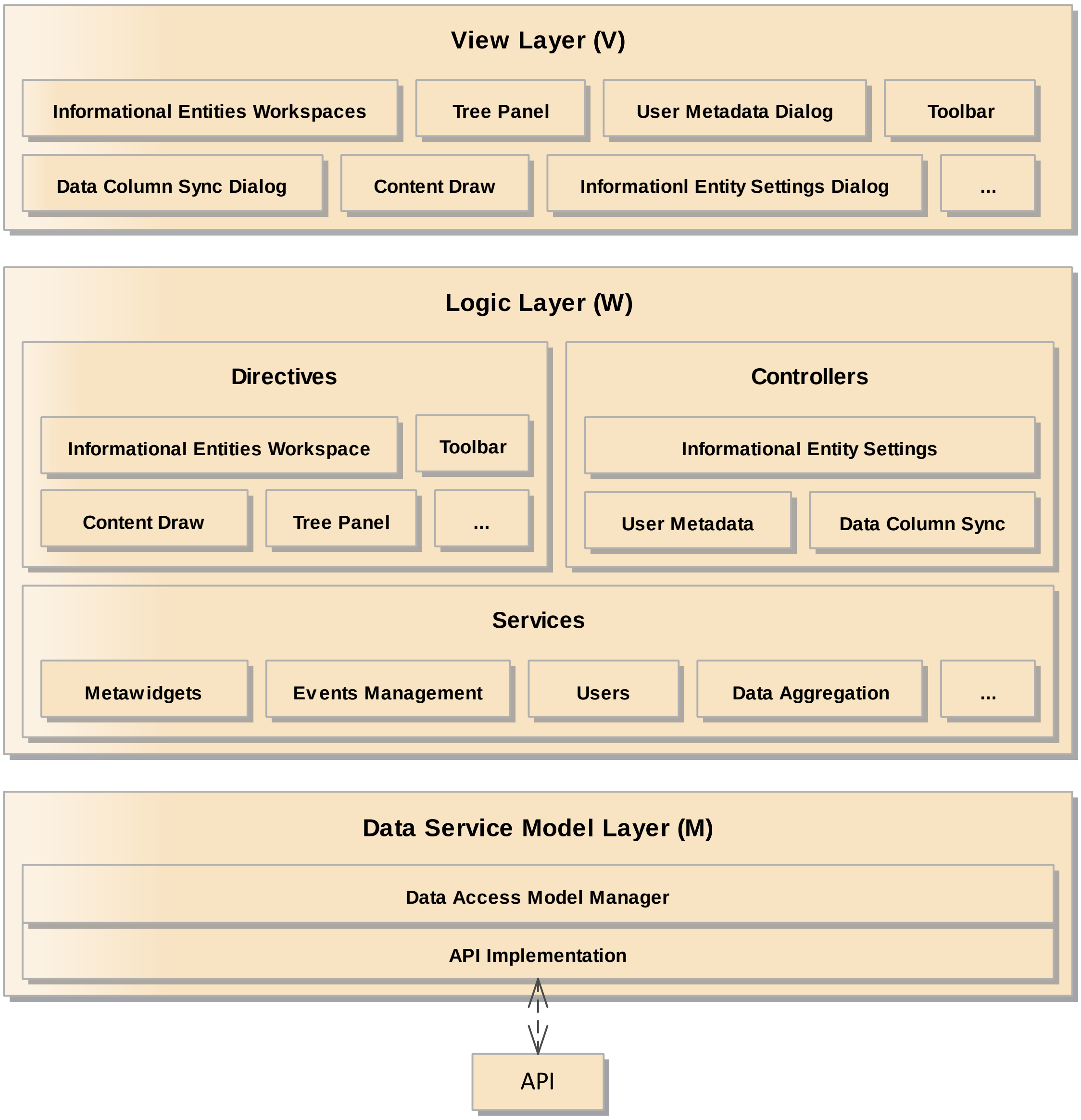


Figure 3.5: Client architecture detailed view

**View Layer.** Responsible to render information for client's interaction. Each view template is built in an extended HTML that provides expressive, reusable, and maintainable templates. Figure 3.5 shows some of the most relevant templates in view layer, such as templates to build several types of dialog models, the tree panel, the templates used to draw contents and content instances or the workspace editors (including each one a toolbar) such as the dashboard or the content editor.

**Logic Layer.** Aims to remove the business logic from the views providing implementation of template actions and behavior with direct correspondence (in directives) to all discussed view templates. The same directive can be used in several view templates, improving reusability of behavior and associated templates. Apart from directives, this layer is also composed by a set of controllers and services, as can be seen in Figure 3.5. Controllers implement behavior associated with dialog models templates, such as a generalized settings definition or concrete management of user metadata or data columns synchronization process. Services support several application processes such has aggregation calculations, management of events between directives, metawidgets implementation or users management. These services support directives and controllers implementations following the architecture modularity, flexibility and reusability already discussed.

**Data Service Model Layer.** Layer composed by a set of services that support the data model and dependency management. The Dashboard Engine Client depends on the ability to communicate with the Dashboard Engine Server, and, in order to provide that ability, the API Implementation service, implements the API exposed by the server. The API implementation receives and generates JSON messages that are deserialized or serialized by core framework services. This API Implementation component not only provides communication, but also hides communication details from the rest of the application software components. Another core service is the Data Access Model Manager. This component manages domain model dependencies, defines the data model structure, manages request’s cache and exposes to all other service components the API in a simpler way.

### 3.2.3. Server-Side Architecture View

This section addresses server’s architecture view, describing a backend server, a RESTful API exposing a set of services, a metadata info-structure database (supporting the system domain model), a notification system used to enhance user awareness and the external connectivity capability to access heterogeneous data sources.

The server component was implemented in Java, using Spring framework, where the goal was to keep the implementation lightweight, distributing processing power to client-side (avoiding to overload the server-side) and enhancing an agile response to user feedback and changing requirements, using a user-centric design approach.

The metadata info-structure provides a way to persist and manage metadata characterizing each informational entity of the system´s metadata model. This database info-structure takes form in a semi-structured JSON type format data, independent of the persistent engine, avoiding a traditional relational model. This offers an important level of abstraction supporting a dynamic metadata model that can be extended by the user (an innovative feature comparing to other information visualization engine solutions). The ability to characterize the created informational artifacts with metadata plays a central role in system’s capability for the execution of information visualization techniques, which will contribute to a greater visibility of the situation that require immediate intervention of the decision maker (spatio-temporal situational-awareness).

In order to interact and take advantage of the metadata info-structure, a RESTful API was designed using Service Design Thinking principles. This informational artifact follows a user-centric and co-creative design guideline in order to expose a set of services available in a collaborative/multi-user information sharing environment. Ensures the required system behavior in terms of guided surveillance, supporting different awareness mechanisms, triggering alerts and events within a collaborative dataflow that aims to keep all intervenient actors well informed.

The designed API follows a standardized URL pattern. Table 3.1 shows samples of main actions/services exposed by the API regarding an *Entity* informational artifact. Any informational entity generalized from an *Entity* (see the metadata layer in the domain model section for more detail) will have these services available through this generalized URL pattern.

Table 3.1: Main URL pattern samples.

| Operation | URL | Result | Description |
|---|---|---|---|
| GET | /entity/ids?type=<EntityType> | List of {id, name} | Returns list of IDs of entities of type <EntityType>, sorted by name. |
| GET | /entity?type=<EntityType> | List<EntityData> | Returns list of entities of type <EntityType>, sorted by name. |
| POST | /entity?forType=<EntityType> | EntityData | Creates new entity (Parameter forType must only be specified when creating a Folder) |
| GET | /entity/{id} | EntityData | Returns entity data. Number of returned fields depends on user permissions. |
| PUT | /entity/{id} | EntityData | Updates entity data. |

On the other hand, Table 3.2 shows the services available specifically for *Metadata*, fulfilling the domain model specification. Provide services to manage metadata fields (categorizing different types of Entities) and also to add new metadata fields, extending the base metadata model, allowing its enrichment.

Table 3.2: Metadata URL pattern samples.

| Operation | URL | Result | Description |
|---|---|---|---|
| GET | /entity/metadata?type=<EntityType> | EntityMetadata | Returns metadata required to create an entity of type <EntityType> |
| GET | /entity/{id}/metadata | EntityMetadata | Returns metadata required to view/edit an existing entity. |
| POST | /entity/{id}/metadata | EntityMetadata | Adds user metadata to an existing entity. |
| PUT | /entity/{id}/metadata | EntityMetadata | Updates user metadata of an existing entity. |
| DELETE | /entity/{id}/metadata/{name} | | Removes user metadata from an existing entity. |

To notice that this work defines the user model and access levels, but the user management implementation is outside of the scope of this thesis and can be achieved through external open-source frameworks (e.g. openAM[7]). Also, accounting (to fulfill AAA[8] security protocol) is outside of the scope of this thesis.

[7] https://github.com/OpenIdentityPlatform/OpenAM/

# 4. Case Study: Health Insurance at FH

The solution proposed in this thesis was applied to a case study in the field of health insurance. The case study addresses an innovative and ambitious health management program, following new emergent models focus on health outcomes, providing an alternative to classic insurance products. The health management program was designed based on the business model of FH[9]. It addresses health insurance challenges regarding chronic patients and health risk behaviors, managing different behavior communities that characterizes costumer's behaviors and rewards

FH is a private company established in 2003, operating in the health insurance domain with headquarters in Lisbon and an office in Zurich. FH is focused in providing clients with innovative health insurance products and services enabling an easier access to private healthcare and well-being services. FH manages in Portugal a medical network with over 25.000 health providers, including 80 private hospitals, over 3000 well-being providers (e.g. nutrition, senior care or nursing) and over 20.000 medical doctors with different specialties, such as cardiology, pneumology or sports medicine.

## 4.1. Health Insurance Market

The growing demand for healthcare insurance, especially in the private sector, has reached in 2016 a 1.3 trillion Euros in global revenues (and forecasted to double by 2025) [3]. However, the healthcare insurance market faces new challenges due to changes in the insurance risk burden dimension. Singhal et al. [3] have analyzed a set of healthcare risks, classifying them into eight categories: cheap routine care, preventive care, chronic care, catastrophic care due to chronic conditions, elective procedures, expensive discretionary care with no medical justification, unpredictable catastrophic care, and end-of-life care.

The classic insurance business model was designed to cover unpredictable catastrophic expenses, providing a financial protection to such situations. Yet, classic insurance risk represents only 28% of health expenses [3], whereas the largest spending reason in healthcare are derived from chronic conditions [2]. The change in insurance risk burdens represents a high-risk to health insurance providers. According to the Portuguese Insurance and Pension Funds Supervisory Authority (ASF)[10], although pre-existing health conditions (e.g. chronicle diseases) of the insured person (at contract's issuance date) are covered by the health insurance, it is a usual exclusion condition to figure in any health insurance contract. Apart from the high-risk for health providers, new insurance models might represent an opportunity to develop innovative products and services addressing consumer niches, not only providing access to better healthcare and financial protection, but also focusing in health outcomes providing advice and guidance to prevent disease burdens and promoting healthy behaviors, acting preventively in mitigating the risk for healthy complications.

[8] Authentication, Authorization and Accounting
[9] http://future-health.care/en/
[10] http://www.asf.com.pt/EN

Information technologies can play here an important role. The use of wearable devices enables clients to share day-to-day data, providing health insurance providers the means for customizing health management programs, addressing telemedicine programs in monitoring the client's progress in real time. In health insurance, when monitoring chronic patients, any awareness must be characterized based on the situation triggering an alert. This may include information regarding the client health status (e.g. client's biometric data), changes to the client daily environment (eating habits) or knowledge of the service quality (e.g. alert notification response rate). Enabling a nutritionist to be informed about each patient results (e.g. a dashboard with a nutritional program), or assist a client in achieving prescribed physical activities, with data collected from wearable devices, are some examples of applicability.

## 4.2. Operational System Behavior Scenarios at FH

This section models a set of operational system behavior scenarios using storytelling techniques [36] and a BPMN representation of the business process on FH's innovative health insurance program. All scenarios address system's awareness capabilities in order to monitor high-risk clients (e.g., adult chronic patients) unhealthy risk behaviors, enabling quick interventions as a way to mitigate the risk of adverse clinical situations and promoting simultaneously client's quality of life and well-being.

To instantiate these scenarios, the proposed solution was configured to address a multi-user environment within the innovative health insurance program. This multi-user environment, supporting a collaborative data flow, uses a tab-based structured layout to address the defined user model (see Figure 4.1 for the tab-based layout and Table 4.1 for user profile's model definition for the addressed scenarios).

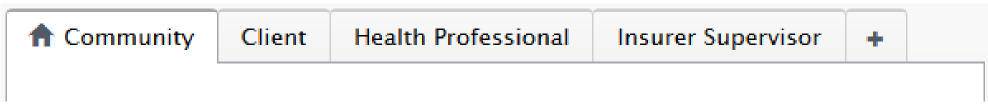


Figure 4.1: Tab-based structured layout addressing the modeled multi-user environment within FH case study health program.

The tab-based layout includes: A public community tab with aggregated metrics regarding community data, including threshold's information that can represent challenges to beat as a way to be promoted to higher communities or targets that when missed generates community's relegation; A client, health professional and insurer supervisor tabs addressing client detailed data presented in different views accordingly with the user profile (e.g. sensible clinical data for health professionals, engine event's details for insurance supervisor).

The addressed program is based on clustering of customers based on heath behaviors in form of behavior communities. Each community will have different rules and thresholds, leading to different rewards (negative and positive). Customer behavior's changes will generate customer transitions between communities as a promotion (on positive behaviors) or relegation (on negative behaviors). Although this work does not cover machine learning algorithms, these standardized clusters provide the bases for future work on this domain, since the application of machine learning algorithms usually needs cluster classifications. The inclusion of such algorithms will allow, in future work, an adaptive model using dynamic learning capabilities in creating and managing communities.

Table 4.1: Operational system behavior scenarios user profile definition

| **Role** | **Description** |
|---|---|
| Health Professionals | Health professionals with privileges to access sensible and confidential medical information. |
| FH Supervisor | Users with edition privileges that can build and edit dashboards. Operations that regular FH Staff cannot do. |
| FH Staff | Users with view privileges to client behavior or public information. |
| Client | Client of FH company with access only to data related to him or community aggregated data. |

When modeling the scenarios it is assumed that the interactive dashboard was configured accordingly to the client's clinical profile (see Figure 4.2 for engine's backoffice). This assumption relates to the creation of an instance of the standard interactive dashboard, together with the informational entities (*DataSource, DataItem/Column)* at the data layer, the business rules at the business logic layer (*Content*) and the graphical visualization at the presentation layer (*Dashboard*, *Tab*, *ContentInstance*).

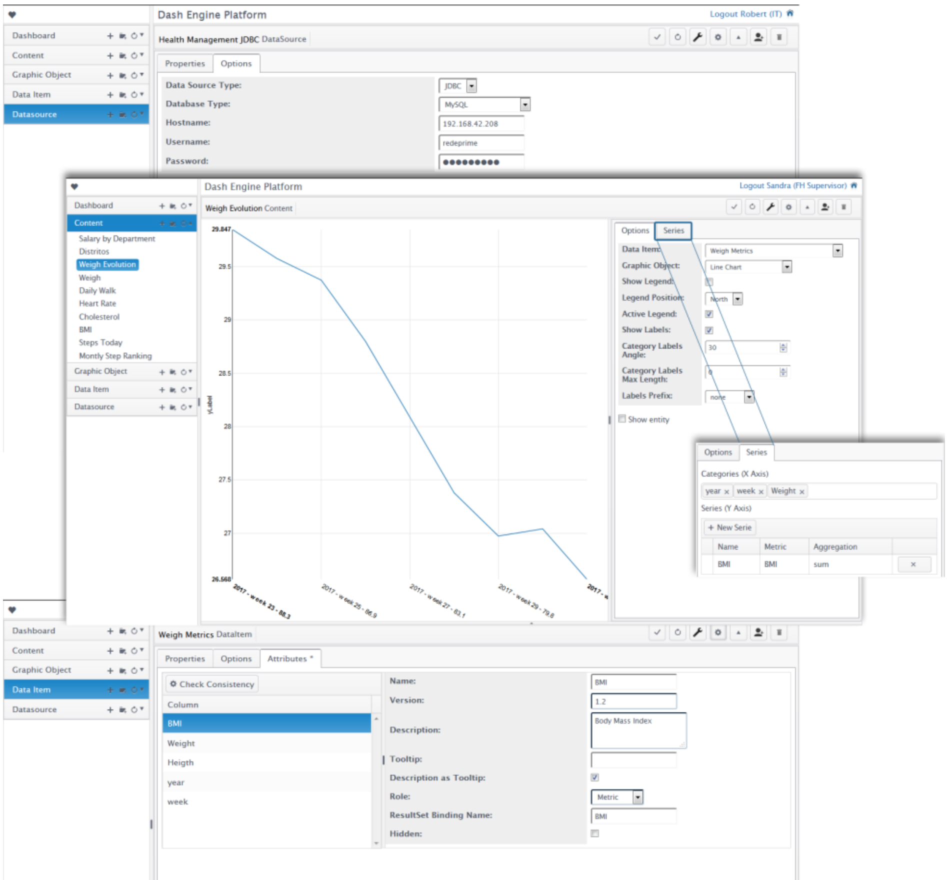

Figure 4.2: Information Visualization Engine backoffice configuration views of *DataSources* (top) *DataItems* (bottom) and *Contents* (center).

Figure 4.2 presents a set of print-screens of the engine's backoffice configuration environment. On the top side it presents the *Datasource,* on the bottom the *DataItem* and at center the Content configuration views. The *DataSource* workspace shows the connection configuration to a database supporting the Health Management Program and the *DataItem* workspace shows the configuration of Weigh Metrics data structure with five *Columns*, showing in particular, "BMI" Data Column with the associated metadata fields. At the center, the "Weigh Evolution" *Content* workspace presents a preview of the current configurations (providing a fast-feedback configuration cycle) and two tabs to configure *Content* visualization attributes and data selection/calculation using categories and *Series* through *DataItem's Columns*.

### S1: Mitigate insurance risk regarding recurrent unhealthy behaviors

The first health insurance scenario models health behaviors of Philip's *persona*, instantiated with multiple views (accordingly with different user roles). Modeling different views of the same dashboard highlights the multi-user engine capabilities, configurable accordingly with the user profile/role. Each actor role view is mapped to a dashboard tab. This scenario is formally represented (BPMN) in Figure 4.3 and instantiated in Figure 4.4.

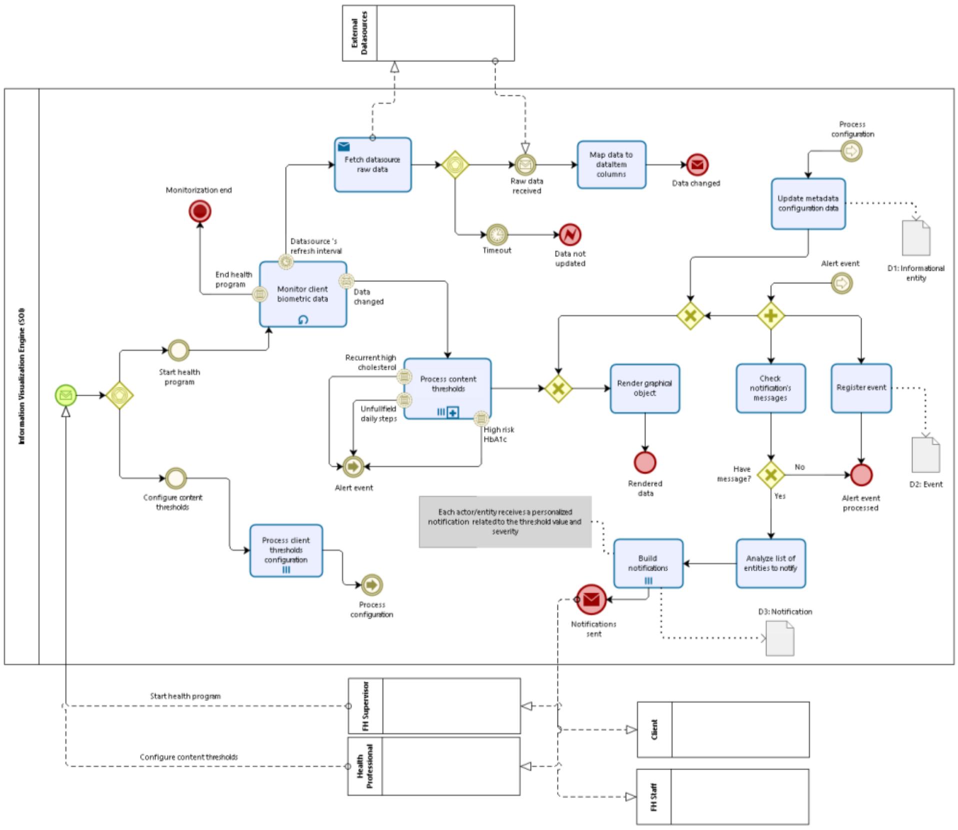


Figure 4.3: Scenario S1 BPMN collaboration diagram.

**Scenario.** Six months ago, Philip started a telemedicine program provided by FH and supported by an Information Visualization Engine for biometric data awareness. Philip is a male taxi driver of 57 years old, diabetic (type I), smoker, with moderate drinking habits and has been deteriorating his diabetes condition lately. Monica, a health professional nurse, became aware of the deviant behaviors in

Philip's health situation by an event triggered by the engine regarding an increase in HbA1c[11] values. Due to the negative tendency in HbA1c, combined with an increase in cholesterol values, a food diet and a 10.000 daily step exercise, were prescribed and configured in Philip's dashboard tab. Phillip missed target exercise for two full weeks, triggering recurrent notifications to FH's call center operator that makes a call to reinforce Philip's commitment, but without success. Two months later, the tendency in cholesterol metrics continued negative, raising a new event for Anthony, the nutritionist, which prescribed a new food diet to complement the prescribed one. Despite the events received by Philip and the collaborative effort of the health professionals to address his unhealthy behaviors, Phillip still failed to change to healthier behaviors, leading to negative rewarding.

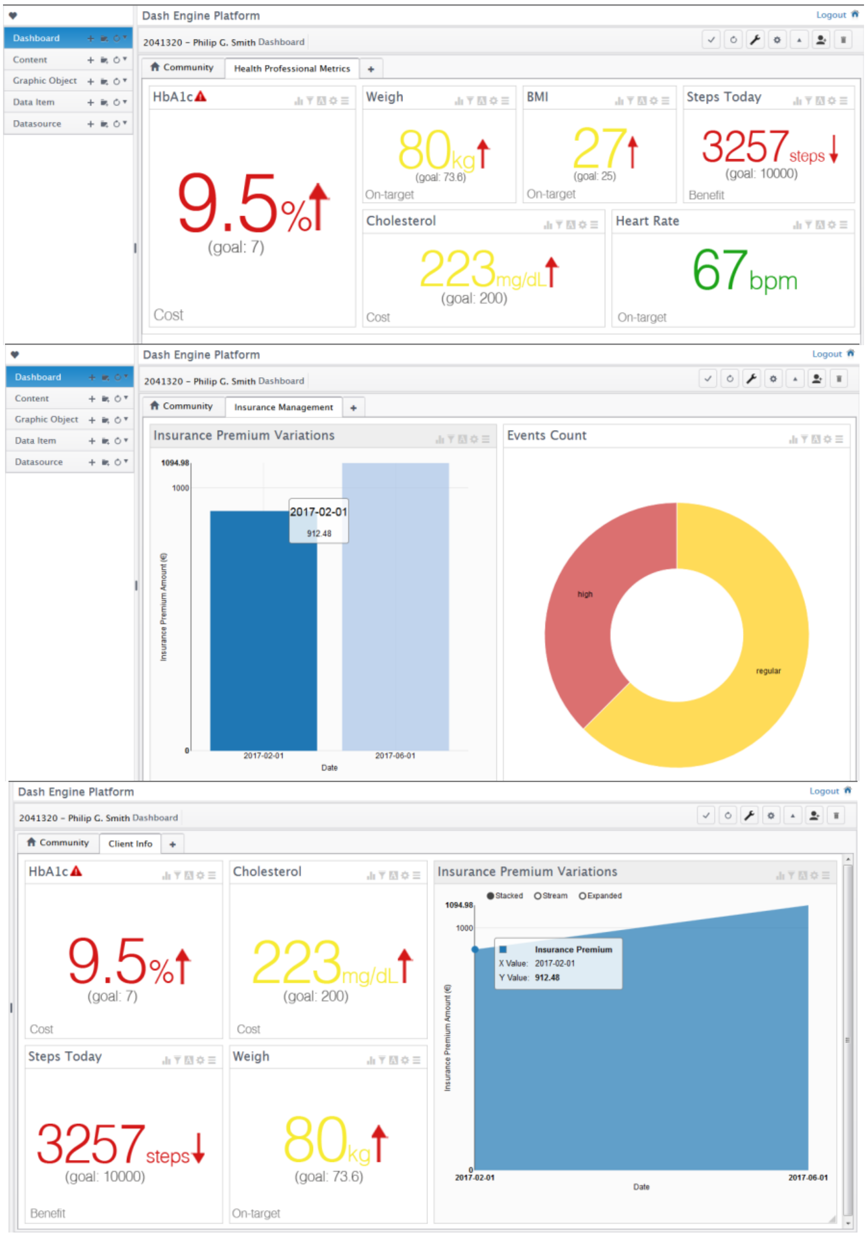


Figure 4.4: Scenario S1 dashboard layout addressing different user roles: Health professional (top image), FH Supervisor (center image) and Client (bottom image).

---

[11] HbA1c refers to glycated hemoglobin, which identifies average plasma glucose concentration, providing a longer-term trend of how high your blood sugar levels have been over a period of time.

In Figure 4.4, the first layout (top image) instantiates health professionals view, configured with detailed metrics regarding relevant risk situations, supporting a collaborative multichannel awareness mechanism that contributes to improve patient health status and well-being. The second layout, the center image, instantiates FH supervisor view, providing a way to monitor deviant behaviors metrics (i.e. generated events) and rewards/penalties resulting from it. The third layout (bottom image) instantiates client's view, providing awareness mechanisms that assure user self-awareness as a way to involve him/her in his own health management. Gamification techniques can play here an important role, promoting healthy behaviors though user rewards and mitigating risks through penalties.

Table 4.2: Scenario S1 indicators list addressing multi-user and multichannel awareness.

| Content | Type | Awareness |
|---|---|---|
| HbA1c | Cost Indicator | Risk indicator related to client's chronicle condition (diabetic). Presents a high risk metric value, with negative trend and associated event. |
| Cholesterol | Cost Indicator | Risk indicator, presenting a high risk metric value, with a negative trend. The recurred aggravation generated an associated event. |
| Steps Today | Benefit Indicator | Indicator addressing client's health behaviors. Unhealthy behavior is reflected by a low metric value, without a positive trend. The recurred behavior generated an associated event. |
| BMI | On-target Indicator | Presents a moderate risk metric value with a negative trend. |
| Hearth Rate | On-target Indicator | Presents a positive metric value without any trend information. |
| Events Count | Chart | Presents generated events count by severity using a pie chart. |
| Insurance Premium Variations | Chart | Presents health insurance premium values variations in form of a line chart. |
| Weight | On-target Indicator | Presents a moderate risk metric value with a negative trend. |

The configured interactive dashboard provides awareness related with risk situations using visual informational artifacts (i.e. indicators and contents) listed in Table 4.2. Indicators are special types of contents. Each indicator shows a measure/value (from the configured series' metric and aggregator function), the indicator type (cost, benefit or on-target), specified goal (based on the value of the threshold configuration) and a variation arrow (showing metric trend). Values use thresholds configured color and the variation arrow makes use of green (benefit) or red (cost) color to express trend awareness.

For example, in Figure 4.4, the "HbA1c" indicator shows a high-risk situation through measurement's red color and a negative tendency (metric is increasing) using a red arrow. Although, dedicated predictive analysis is outside the scope of this work, the use of the trend together with the target goal composes mechanisms supporting the implementation of a predictive model, providing understanding on what is likely to happen next.

**Expected Outcome.** After the system has generated several events related with three indicators (HbA1c, Cholesterol and Daily steps) and multiple notifications (to the health professional nurse, the FH staff, the nutritionist, and simultaneously to the client), Phillip failed to behave as expected to improve his well-being. System's awareness mechanisms regarding Philip's health risk behaviors and further failure to meet the prescribed targets lead to a penalty in insurance product premium due to the client's deviant behavior, mitigating health insurance risk burdens that may occur due to Phillip's unhealthy behaviors.

### S2: Motivational awareness for healthier behaviors

The second health insurance scenario addresses user awareness regarding Samantha's *persona* health behaviors, instantiated in two different moments in time highlighting client's health behavior changes representation. Figure 4.5 is the visual outcomes of the two moments. Also, this scenario is formally represented by a BPMN collaboration diagram in Figure 4.6.

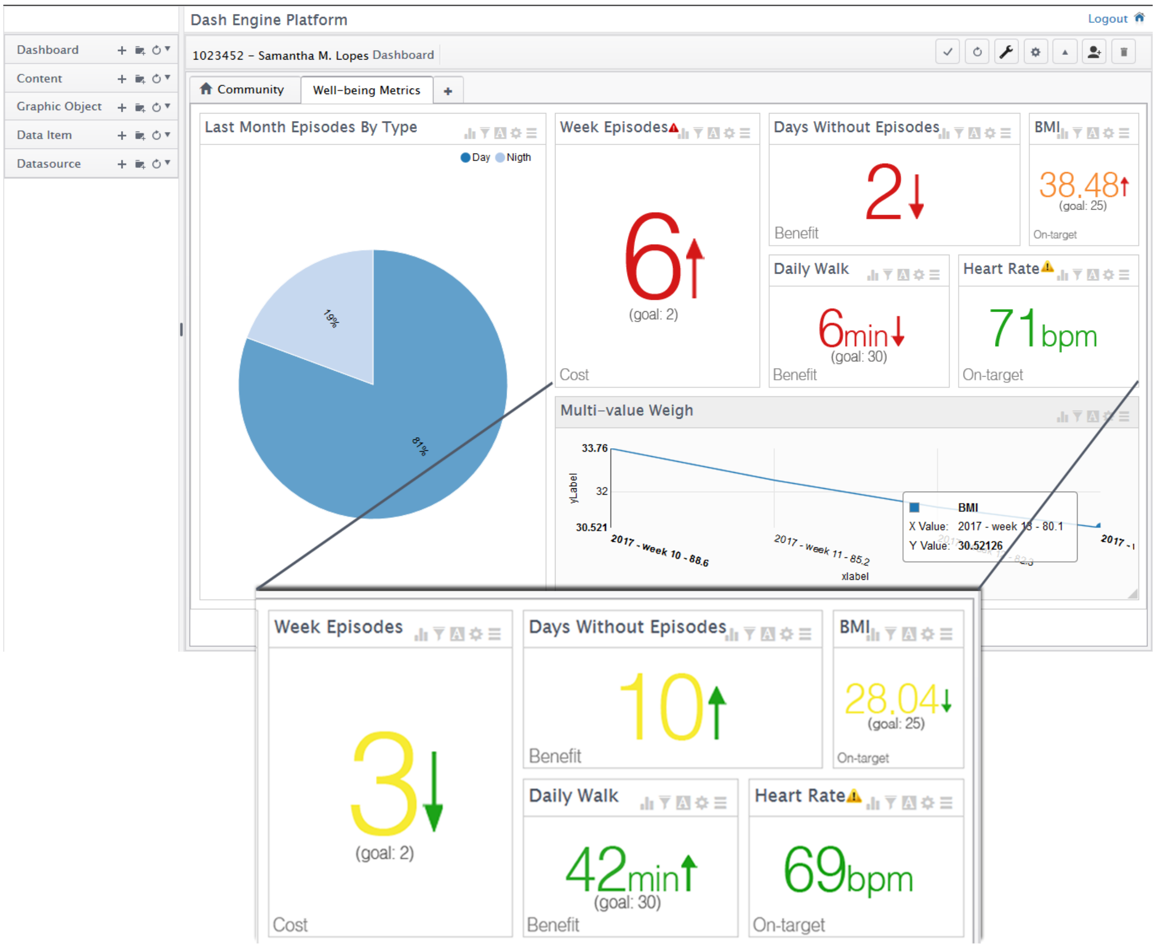


Figure 4.5: Scenario S2 dashboard layout addressing Samantha's initial well-being metrics (top) and after two months (focused on bottom), showing healthier behaviors effects.

**Scenario.** Samantha, an overweight, 47 years old, female, stockbroker, suffers from asthma and has been experiencing an increase number of asthma crises due to an abrupt attempt to follow an intense exercise program for weight loss. The recurrent asthmatic episodes present a high risk of deterioration to the lung function over time. Samantha has an FH insurance product that includes a health management program, monitoring registered asthma attacks and health behaviors (supported by an Information Visualization Engine). Samantha was prescribed a continuous 30 minutes daily walk, avoiding abrupt exercise effort. Although the prescribed walk, Samantha continued to over exercise, raising multiple events on high hearth rate due to abrupt running exercise. Due to the generated events, Martha, a physical trainer, was notified. Martha reinforced the risks of over exercise due to Samantha's clinical profile. Also, due to the unchanged asthma crisis events, Joseph, a pneumologist, received a notification regarding Samantha's risk behavior, generating a medication prescription change. Now, for the past two months, Samantha started to follow the exercise prescription, lowering the experienced asthma episodes, improving Samantha's well-being and life quality, Also, she received a bonus discount in the policy premium as a prize for the continuous healthier behavior.

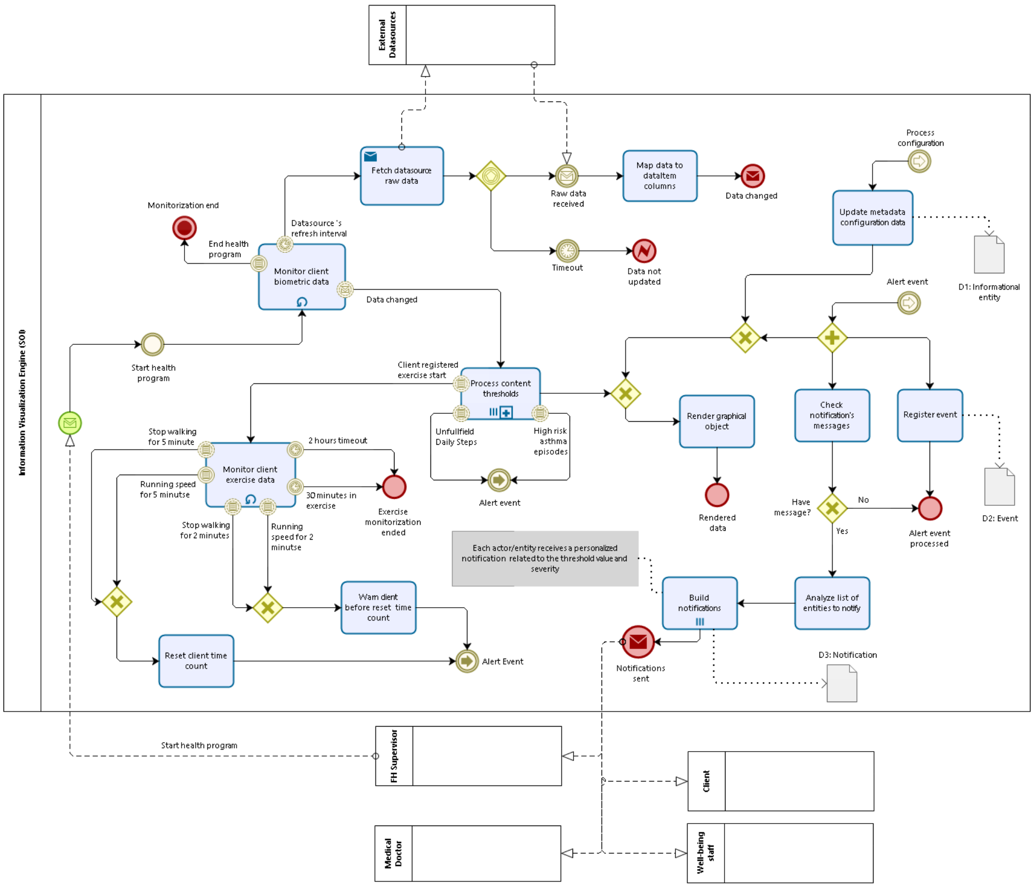

Figure 4.6: S2 scenario BPMN collaboration diagram.

Both BPMN collaboration diagram in Figure 4.3 (Scenario S1) and Figure 4.6 (Scenario S2) identifies three data objects (D1 to D3). Informational Entity (D1) data object addresses the defined metadata model (see Figure 3.2 for domain model detail) and represents the informational entity state. Event (D2) and Notification (D3) data objects represents the informational artifacts outcome of the implemented awareness mechanisms. When suspicious situations are detected (taking into consideration its level-of-concern) the system triggers events to inform all intervenient stakeholders (trough configured notifications), creating a collaborative community acting proactively to assure the person health status and well-being.

An alert is represented with a related symbols and colors accordingly with the level-of concern (e.g. “regular” level-of-concern alert on “Daily Walk” indicator and “high” level-of-concern alert on recurrent risk situation regarding “Asthma Episodes” metrics. The use of different levels of severity empowers awareness mechanisms, enabling user awareness regarding deviant behaviors (accordingly with defined goals), and secondly, the need to act accordingly as a way to mitigate deviations.

The instantiate interactive dashboard provides awareness related with risk situations using visual informational artifacts listed in Table 4.3. This table describes indicators/contents referring two moments (separated by two months). Modeling different moments in time of the same dashboard highlights the potentials of the awareness capabilities was a way to address client’s health behaviors, mitigation risk situations an improving client’s well-being. The second layout shows a “Weigh Evolution” line chart with a multi-value/level X-Axis’ values: year, week and weight. The “Weigh Evolution” content configuration is presented in Figure 4.2.

Table 4.3: Scenario S2 indicators list addressing multi-user and multichannel awareness.

| Content | Type | Awareness |
|---|---|---|
| Episodes By Type | Chart | Content related to client’s chronicle condition (asthmatic). Presents last month asthma episodes count by type. |
| Week Episodes | Cost Indicator | Indicator related to asthmatic condition. At moment 1 presents a high risk metric value, with negative trend and associated event. At moment 2 presents a low risk metric value with a positive trend. |
| Days Without Episodes | Benefit Indicator | Risk indicator related to client’s chronicle condition (asthmatic). At moment 1 presents a stable (no variation) high risk metric value. At moment 2 presents a low risk metric value with a positive trend. |
| Daily Walk | Benefit Indicator | Indicator addressing client’s health behaviors. At moment, 1 unhealthy behavior is reflected by a low metric value. Also, shows a negative trend and associated events. At moment 2 shows a positive metric value with a positive trend. |
| BMI | On-target Indicator | Presents a high-risk metric value with positive trend at both moments. |
| Hearth Rate | On-target Indicator | At moment 1 presents a recurrent high metric value, and at moment 2 shows a more controlled metric value. |
| Multi-value Weight | Chart | Presents weight metric values history in form of a multi-value line chart. |

**Expected Outcome.** A newly created and configured dashboard, addressing FH’s health insurance user model, generated daily events regarding high risk behaviors (high heart rate and elevated daily exercise) and two notifications for health professional and medical doctor. Samantha’s self-awareness regarding her unhealthy behavior generated a behavior change, improving her well-being. System’s awareness mechanisms regarding Samantha’s health behaviors lead to a bonus in health insurance product premium (due to target goals fulfillment), promoting positive health behaviors, and ultimately, better life quality and well-being.

# 5. Conclusions

This thesis outlines the main achievements in designing and implementing an information visualization engine (using a client-server web-based architecture) that uses information visualization and gamification techniques to streamline visual representation of events in a collaborative decision making environment. This work followed a DSRM using agile iterative design cycles in order to refine the designed informational artifacts, focusing on end-users.

Information Visualization techniques, when working with large volumes of data, helps to provide value to existing data to support the decision-making process, where user relies on the system situational-awareness capabilities to keep all actors aware and well informed about undergoing events, as a way to mitigate risks of recurrent deviant behaviors. Gamification techniques helps to address motivational factors of each individual, enhancing his/her commitment with prescribed behaviors and healthy lifestyles, reducing the risk of undesired events, leading to positive or negative rewards. Service design thinking principles were used as guidelines to model a set of services that ensures system behavior capabilities concerning situational-awareness mechanisms within a collaborative dataflow (based on the user model). Regarding solution's client side definition, a dashboard engine solutions survey and comparative analysis provided an import input and an architectural design patterns survey supported the use of a Model-View-Whatever (MVW) architectural approach.

This work addresses new emergent health insurance models focused on health outcomes as an answer to new customer needs. This includes current challenges concerning data exploration and capability to enable people and organizations to make informed decisions. The proposed solution was designed to be used in any context, but for demonstration purposes, the system was configured to comply with requirements from the health insurance sector, in particular to empower the user to build interactive visualizations, configuring and customizing information artifacts to promote situational-awareness (e.g. health status and monitor unhealthy behaviors) and streamline the way events and data are presented to the user, contributing to improve patient health status and well-being.

## 5.1. Contributions

**Health Insurance Case Study.** This research work contributes with two operational scenarios (using storytelling techniques and a BPMN representation of the business process on FH's health insurance program) to model the decision-making processes and awareness challenges regarding high-risk clients (e.g. chronic patients) and health risk behaviors.

**Conceptual Metadata Model.** Following the information visualization guidelines, this thesis presents a metadata model, capable of characterizing the informational entities to address the identified requirements of an information visualization engine. This is a flexible model that supports the creation of informational artifacts addressing spatio-temporal data sensemaking (e.g. patient's biometric indicators), keeping all intervenient actors aware about events requiring their immediate attention (e.g. address patient's unhealthy behavior contributing to improve patient health status and well-being).

**System Architecture.** In order to realize the conceptual metadata model, this thesis presents a client-server architecture supported by a metadata info-structure. The proposed solution was designed to assure scalability and configurability, in an agile and lightweight implementation capable of providing a multi-user environment based on the user model, enabling multichannel awareness mechanisms as a way to assure situational-awareness. The client-side follows a Model-View-Whatever (MVW) architectural pattern providing a clear separation of responsibilities and concerns between application layers, modularity, flexibility, and testability to the implementation. The server-side exposes a RESTful API realizing a set of services that manages the metadata info-structure, supports notifications dispatch and heterogeneous data sources connectivity. The design kept the implementation lightweight, avoiding server-side overload by distributing processing power to client-side. The designed architecture simplifies the management of event driven programming, an important factor in a platform that aims to provide an interactive data exploration in order to improve self-awareness.

**Prototype.** This thesis presents an implementation of the of a prototype instantiating a health insurance case-study. The prototyped engine provides a flexible user interface that enables the creation of personalized interactive dashboards. MVW architectural pattern was implemented using AngularJS framework that provides a way to simplify management of event driven programming. Also, AngularJS two way data binding allowed to speed up the construction of informational entities that represents data model state, such as properties related with concepts like relevance or severity, and user interactions that generate model changes. On the other hand, server-side was implemented in Java, using Spring framework and the RESTful API implemented to support the main operations that allows the management of the designed metadata model.

## 5.2. Future Work

**Formal Evaluation.** It is intended to conduct a formal evaluation study of the proposed information visualization engine within the innovative health program project sponsored by FH. This formal evaluation would complement results from the designed case study. Such study will integrate the company strategy and business context.

**Metadata Model Limitations.** It is intend to address the model limitations regarding embedding and dynamically publishing the dashboard layout into external systems (e.g. webportals), as well as, capabilities of visualization type recommendation based on different types of heuristics (e.g. raking heuristics, data patterns).

**Business Intelligence Capabilities.** Finally, although this work assumes that raw data are pre-processed and are ready to be visualized, it would be interesting to address a new data mining module focused in providing the ability to address new health insurance products (more effective for customers and sustainable for a health insurance providers). Also, Smart Data Discovery[12] capabilities and Business Process Analytic (BPA) would improve data sensemaking and furthermore improve awareness mechanisms.

[12] Smart data discovery focus on analyze different data combinations and patterns with machine learning algorithms in order to provide quick insights by simply asking questions and getting answers.

# References


[1] World Health Organization, *World health statistics 2017: monitoring health for the SDGs, sustainable development goals*. 2017.

[2] R. V. Milani and C. J. Lavie, "Health care 2020: Reengineering health care delivery to combat chronic disease," *Am. J. Med.*, vol. 128, no. 4, pp. 337–343, 2015.

[3] S. Singhal *et al.*, "Global private payors : A trillion-euro growth industry," *McKinsey*, 2016.

[4] D. Johnson, S. Deterding, K.-A. Kuhn, A. Staneva, S. Stoyanov, and L. Hides, "Gamification for health and wellbeing: A systematic review of the literature," *Internet Interv.*, vol. 6, pp. 89–106, 2016.

[5] S. Few, *Information Dashboard Design The Effective Visual Communication of Data*. O'Reilly Sebastopol, CA, 2006.

[6] M. Smuts, B. Scholtz, and A. Calitz, "Design Guidelines for Business Intelligence Tools for Novice Users," *Proc. 2015 Annu. Res. Conf. South African Inst. Comput. Sci. Inf. Technol. - SAICSIT '15*, pp. 1–15, 2015.

[7] C. Ware, *Information Visualization: Perception for Design*. Elsevier, 2012.

[8] H. M. Chen, "An Overview of Information Visualization," *Libr. Technol. Rep.*, vol. 53, no. 3, p. 5, 2017.

[9] S. Card, J. Mackinlay, and B. Shneiderman, "Information visualization," *Human-computer Interact. Des. issues, Solut. Appl.*, vol. 181, 2009.

[10] J. Parenteau, R. L. Sallam, C. Howson, J. Tapadinhas, K. Schlegel, and T. W. Oestreich, "Magic Quadrant for Business Intelligence and Analytics Platforms," *Gart. Res.*, no. February, p. 82, 2016.

[11] J. Lapa, J. Bernardino, and A. Figueiredo, "A Comparative Analysis of Open Source Business Intelligence Platforms," *Proc. Int. Conf. Inf. Syst. Des. Commun. - ISDOC '14*, pp. 86–92, 2014.

[12] D. Perry, B. Howe, A. M. F. Key, and C. Aragon, "VizDeck: Streamlining Exploratory Visual Analytics of Scientific Data," *iConference 2013 Proc.*, pp. 338–350, 2013.

[13] M. Elias and A. Bezerianos, "Exploration views: Understanding dashboard creation and customization for visualization novices," *Human-Computer Interact. 2011*, vol. 6949 LNCS, no. PART 4, pp. 274–291, 2011.

[14] C. North and B. Shneiderman, "Snap-Together Visualization : A User Interface for Coordinating Visualizations via Relational Schemata Counties States," *Proc. Work. Conf. Adv. Vis. interfaces*, pp. 128–135, 2000.

[15] A. Franklin *et al.*, "Dashboard visualizations: Supporting real-time throughput decision-making," *J. Biomed. Inform.*, vol. 71, pp. 211–221, 2017.

[16] G. E. Krasner and S. T. Pope, "A Cookbook for Using the Model- View-Controller User Interface Paradigm in Smalltalk-80," *Joop J. Object Oriented Program.*, vol. 1, pp. 26–49, 1988.

[17] M. Potel, "MVP: Model-View-Presenter The Taligent Programming Model for C++ and Java," *Taligent Inc*, no. C, pp. 1–14, 1996.

[18] E. Sørensen and M. I. Mihailesc, "Model-View-ViewModel (MVVM) Design Pattern using Windows Presentation Foundation (WPF) Technology," *MegaByte J.*, vol. 9, no. 4, pp. 1–19, 2010.

[19] R. Branas, *AngularJS Essentials*. Packt Publishing, 2014.

[20] J. M. K. Kumar and M. Herger, *Gamification at Work: Designing Engaging Business Software*. The Interaction Design Foundation, 2013.

[21] S. Deterding, "Gamification: Designing for Motivation," *Interactions*, vol. 19, no. 4, p. 14, 2012.

[22] L. E. Nacke and S. Deterding, "The maturing of gamification research," *Comput. Human*

*Behav.*, vol. 71, pp. 450–454, 2017.

[23] M. R. Endsley, “Toward a theory of situation awareness in dynamic systems,” *Hum. Factors J. Hum. Factors Ergon. Soc.*, vol. 37, no. 1, pp. 32–64, 1995.

[24] D. Filonik, “Developing a Dashboard for Real-Time Data Stream Composition and Visualization,” Ludwig-Maximilians-Universität München, 2012.

[25] M. Bostock, V. Ogievetsky, and J. Heer, “D3 : Data-Driven Documents,” *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2301–2309, 2011.

[26] J. Stickdorn, Marc and Schneider, *This is sevice design thinking:basics,tools,cases*. BIS Publishers, 2011.

[27] D. Iseminger, “Power BI Security,” Microsoft, 2016.

[28] M. Rueter and E. Fields, “Tableau for the Enterprise : An Overview for IT,” Tableau Software, 2012.

[29] A. Gioia, G. Cazzin, and E. Damiani, “SpagoBI: A distinctive approach in open source business intelligence,” *2008 2nd IEEE Int. Conf. Digit. Ecosyst. Technol. IEEE-DEST 2008*, no. March, pp. 592–595, 2008.

[30] C. Howson, J. Parenteau, R. L. Sallam, T. W. Oestreich, J. Tapadinhas, and K. Schlegel, “2016, Gartner, Critical Capabilities for Business Intelligence and Analytics Platforms.pdf,” *Gart. Res.*, no. March, 2016.

[31] S. Gregor and A. R. Hevner, “Positioning and Presenting Design Science Research for Maximum Impact,” *MIS Q.*, vol. 37, no. 2, pp. 337–355, 2013.

[32] K. Conboy, R. Gleasure, and E. Cullina, “Agile Design Science Research,” *Springer International Publishing Switzerland*, vol. 9073, pp. 168–180, 2015.

[33] A. Hevner and S. Chatterjee, “Design Research in Information Systems,” *Integr. Ser. Inf. Syst.*, vol. 22, pp. 9–23, 2010.

[34] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A Design Science Research Methodology for Information Systems Research,” *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, 2007.

[35] S. T. March and G. F. Smith, “Design and natural science research on information technology,” *Decis. Support Syst.*, vol. 15, pp. 251–266, 1995.

[36] S. Madsen and L. Nielsen, “Exploring Persona-Scenarios - Using Storytelling to Create Design Ideas,” in *Human Work Interaction Design: Usability in Social, Cultural and Organizational Contexts: Second IFIP WG 13.6 Conference, HWID 2009, Pune, India, October 7-8, 2009, Revised Selected Papers*, D. Katre, R. Orngreen, P. Yammiyavar, and T. Clemmensen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 57–66.

# Appendix

## Appendix 1 - Comparative Analysis Details

### Comparison Criteria Details

This section presents a detailed view of the criteria used in the comparative analysis and solutions survey. The selection of the comparison criteria (and sub-criteria) followed Gartner's [30] approach, using ten of Gartner's critical capabilities criteria (aggregated into eight used criteria), complemented by [6] and two new criteria:

***General Criteria***

- **C1: Licensing** represents platform licensing options and availability. This criterion is relevant in current comparison context due to the direct relation to business restrictions and resources. Possible values are: Open Source (OS), Commercial (C) and Academic/Project (A).
- **C2: Platform Administration, Security and Architecture** represent platform administration and security, user access levels management and architecture definition. The sub-criteria are:
    - **Architecture and software strength**, measure platform architecture quality and software robustness;
    - **Performance and Scalability**, measure platform's performance and scalability;
    - **User administration and profiling,** measure user administration and access levels parameterization and management;
    - **Data Encryption**, measure if data encryption exists.

***Presentation Criteria***

Presentation criteria are related to data presentation and interaction through interactive dashboards. Intends to measure a wide variety of possible interactions, filters, data presentation formats, searches and other related presentation requirements:

- **C7: Interactive Visual Exploration** represents exploration and interaction possibilities when exploring and interacting directly with visual representations of data, like charts, tables, indicators or other data visualizations, using, for example, filters, color, display formats, size or shape. The sub-criteria are:
    - **Interactivity and exploration**, measure user interactivity and exploration possibilities;
    - **Global filters**, measure one filter can be used in a global context;

- **Chart types supported for analysis**, measure the variety of visualization types that can be used within analysis;
- **Display as percentages**, measure the possibility of viewing values as percentages;
- **Visualizations linking pre-definitions**, measure initial linking customization effort, based on pre-definition (automatic linking) capabilities;
- **Visualizations linking customization**, measure linking possibility between visualizations and how can they be customized;
- **Conditional formatting**, measure the possibility of formatting based on conditional parameterization;
- **Semantic query/search**, measure natural-language query or other semantic searches/queries existence.

- **C8: Interactive Dashboards** represents the construction and customization of highly interactive dashboards with visual exploration and embedded analytics. The sub-criteria are:
  - **Chart formatting options**, measure visual formatting options in dashboard design;
  - **Chart types supported for design**, measure the variety of visualization types that can be used in dashboard design;
  - **Immediate visualization type changes**, measure the possibility of changing visualization type directly in dashboard design.
  - **Formatting and layout**, measure dashboard possibilities for layout and formatting organization, such as tabs, sections, or others;
  - **Load scheduling**, measure load schedule functionality for dashboard visualization contents.

***Business Logic Criteria***

Business logic criteria are related to visual data preparation and usage. Intends to measure visual data analytics preparation and publishing through interactive dashboards, alerts or embedded platforms:

- **C5: Self-Service Visual Preparation** represents user-driven data combination capabilities of creating graphical data visualizations such as charts, maps, tables or KPI's, with fast feedback editable workspace and real time data visualization. The sub-criteria are:
  - **Drag-and-drop data content preparation**, measure the possibility and flexibility of prepare data contents with drag and drop, user-driven, operations;
  - **Immediate visual feedback**, measure if data content preparation changes can be viewed in real time while editing, allowing "fast feedback" changes;

    - **KPI support**, measure if the platform supports KPI’s;
    - **KPI construction**, measure flexibility and simplicity of KPI’s construction and parameterization. This sub-criteria is separated from “KPI support” due to KPI’s importance and relevance, separating the measuring of support and “quality” construction;
    - **Thresholds**, measure is the platform supports thresholds.
- **C9: Publish and Embed Analytic Content** represents the possibility of creating and modifying analytic content, embedding them into a business process, application or portal and the possibility to publish data content through several output formats. The sub-criteria are:
    - **Export output format**, measures the diversity of data output formats;
    - **Portal integration**, measure possibilities of integrations in external portals, and flexibility of supported operations;
    - **Embeddability**, measure the possibility of embedding content elements in other applications or business processes.
- **C11: Event-Based Awareness** represents the ability to configure, visualize and notify events and alert situations that require user attention. The sub-criteria are:
    - **Event-Based alerts**, measure conditional alerts support (based on data events), diversity of communication means and parameterization flexibility;
    - **Event visual display**, measure the system ability to display (helping user’s to be aware) occurred events.
    - **Event level of concern**, measure the possibility of managing different levels of concern in events.

***Data Criteria***

Data criteria are related to data preparation and connectivity. Measures data models preparation and changeability, and the capability to connect to different data sources.

- **C3: Data Source Connectivity** represents the capability to connect to different source types and storage platforms. The sub-criteria are:
    - **Personal and Web data**, measure the capability to access data of personal data and Web, such as rest connection;
    - **OLAP connectivity**, measure the possibility to access data from OLAP;
    - **Unstructured and semi-structured data**, measure the capability to access data that is not completely structure, such as CSV, JSON or website content.

  - **Relational query access**, measure the possibility to access data from relational databases;

- **C4: Self-Service Data Preparation** represents user-driven data combination capabilities of creating analytic models such as measures, sets, groups or hierarchies creation and generation. The sub-criteria are:

  - **Drag-and-drop data preparation**, measure the possibility and flexibility of prepare data with drag and drop, user-driven, operations;

  - **Flexibility of Analytic Models**, measure flexibility of the analytic model in terms of data categorization and usage when building data contents;

  - **User Defined Analytic Models**, measure the possibility and flexibility of users create analytic model attributes, such as categories or measures.

***Metadata Criteria***

Metadata criteria are related to metadata structure and management. Intends to measure the centralized and structured management of the metadata model quality:

- **C6: Metadata Management** represents the quality of the metadata model in terms of expressiveness, completeness and standardization. That is, the ability to create and customize, in a centralized way, semantic model objects such as dimensions/categories, metrics, hierarchies, performance metrics/key performance indicators (KPIs), layout objects, parameters and so on. Also takes in account the ability to extend this model with new metadata fields. The sub-criteria are:

  - **Reuse**, measure reuse possibilities of created informational entities objects and metadata structure;

  - **User metadata**, measure the ability to manage user defined metadata fields;

  - **Data modeling**, measure data modeling functionalities;

  - **Expressiveness and completeness**, measure if the metadata model has a clear expressive definition and is complete.

***Collaboration Criteria***

Collaboration criteria are related to collaborative work functionalities. Measures real-time information and sharing capabilities as a way to create discussion threads, comments or ratings.

- **C10: Collaborative work** represents the ability to share information in form of comments, annotations, ratings/recommendations or discussion threads/chats. The sub-criteria are:

  - **Discussion threads**, measure information sharing ability throw discussion threads;

  - **Real-time collaboration**, measure functionalities for real-time collaborative work;

- **Rating and recommendations**, measure the possibility of rate or recommend informational entities objects;
- **Comments**, measure the ability to add comments in different levels of the application.

## Solutions Survey Details

Due to their ability to build online interactive dashboards and visualizations, without using programming environments, hard in-depth analysis platforms with focus on complex BI analysis or other tools that allow data analysis, but without interactive visualization functionalities, a set of solutions/projects were selected and used in the comparison analysis (See Table 0.1). For each solution, and selected criteria, a score was given and the following detailed description aims to support chosen scores in used criteria. Also, this detailed description aims to clarify the reason for solution selection and gives high level description of solution architectural aspects.

### *Snap-Together Visualization*

Snap-Together Visualization (Snap) [14] is a flexible tool that enables a quick and dynamic creation of data visualizations and relations (coordination) in order to build customizable exploration interfaces for rapid exploration and navigation of data and relationships without any need to use programming skills. In system evaluation, the authors concluded that customizable information visualization tools are in reach of novice users and enhanced in 30% to 80% performance in several browsing tasks.

Accordingly with North and Shneiderman’s [14] flexibility classification, Snap has top flexibility level. It achieves this level of flexibility due to the proposed conceptual model. This conceptual model is based on a relational database, where created visualizations correspond to relations and coordinations to joins in database model. Snap tool allows users to load data from files, memory or relational databases that can then be loaded into visualizations, using a menu that displays tables, queries and available visualizations tools (These visualizations tools can be extended using a standard API). Then, provides the ability to coordinate the created visualizations (“snap them together”), defining the relational schemata and enables the exploration of the resulting data related visualizations. These coordinations are bi-directional and can be chain across several views supporting operations such as Drill-down or Brushing-and-linking.

### *Exploration Views*

Exploration Views (EV) [13], regardless of being a research project, is categorized by its authors as a fully operational BI dashboard. It supports large amounts of data from different types of data sources. EV was built in a three layered architectural approach. Firstly, consists in an Adobe Flex web-client responsible for dashboard rendering and exploration, and chart management creation and recommendation services. The server, representing the second layer, manages users, dashboard state, and data sources synchronization enabling multi-chart visualizations. Finally, the third layer is composed by a semantic layer that enables view synchronization and query relations, and an in-memory database to enhance data storage and retrieval.

EV empowers end users in dashboard’s easy creation, customization and interaction with multiple visualizations in a unified environment. Promotes fast feedback actions, encouraging user experimentation, aiming to be easily customizable and respond to user needs. EV focuses on novice users, implemented using several novice user interaction guidelines and designed to be simple, with continuous visual feedback on user actions, providing chart suggestions and templates in building process, supporting several interactive actions or manipulation mechanisms, and enabling the use of common data formats, such as excel files or OLAP data. The EV prototype focused more on the interface design and visualization exploration by novice users, then in the possible insights retrieved from the existent data.

EV has a main working area that corresponds to the dashboard itself and its composing components, a drawer with chart templates and created widget samples, global filters section and a miscellaneous function’s toolbar. Users can add to the dashboard working area, from the drawer clipboard, charts, tables and filters, and the system automatically realigns existing components to accommodate new additions. Users can still rearrange dashboard’s layout by dragging or resizing components.

To support visualization exploration, EV provides text search support, global (entire dashboard) and specific filters (particular visualization), and automatic linked coordinates providing data relations filtering and highlighting. Because the linked coordinates cannot be customized, accordingly with North and Shneiderman [14], EV has level two (scale from zero to three) of flexibility.

***VizDeck***

VizDeck [12] project is a prototype of a web-based visual analytics tool for relational data (no support for other data source types measure in criteria C6) providing interactive visual dashboards. Uses a card game metaphor and automatic visualization techniques allowing rapid visualization of data with no programming necessary in an easy to use and learn platform. Accordingly with evaluations made [12] users were able to complete tasks in less time with VizDeck then with comparative platforms. However, this “quick to view” gives a bad trade-off between automating the visualization creation process and the control/interaction possibilities in these visualizations, based on the defined criteria. VizDeck interface was designed to speed up dashboard building processes and visualization selection when compared to more complex visualization tools and doesn’t intend to support self-service data preparation (C8), metadata management (C9) or embedding and publishing analytic content (C13/C14).

VizDeck architecture is composed by a server developed in Ruby and a Javascript client. The server manages data source connections and query results, and, on the other hand, the client manages visualizations rendering and user interactions. Each query result creates several visualizations sorted by their score (further discussed) and diversification selecting the top *k* to recommend to the user, displayed in interactive *Vizlets*. Vizlet model [12] is described as a general model based on a 4-tuple(ft, fi, p, s): two render functions ft (thumbnails) and fi (interactive), one boolean predicate indication of compatibility to a given dataset and a scoring function. This general model definition aims for a simple extensibility of current vizlet types.

Displayed in a grid of thumbnails, users can see the pre-generated recommendation before adding them to the dashboard, reason why was classified above average for content preparation fast feedback (C7.2). Also, a search-box feature can help to find types of visualizations or data attributes (semantic query/search criteria C11.8).

Visualization objects, when selected, feed the scoring function, for the dataset providing the underlying data, accordingly, if the visual object was discarded ("dislike" vote) or promoted ("like" vote) for inclusion in an interactive dashboard. The recommendation model, not only looks to user input, but uses a combination of heuristics avoiding random recommendations caused by initial lack of user input. Each Dashboard has a unique and permanent URL that can be shared. Also, in that share, steps taken to create the dashboard can be reviewed with an undo/redo type of functionality, improving cross-training and communication between users. This strong real-time collaboration (C15.2) and rating/recommendations (C15.3) are classified with the top score in each criterion.

Dashboards can have multiple visualizations' grids where each visualization automatically affects the display of other visualizations, based on user interaction and event processing model. This processing model allows designers to define different behavior of each visualization type in response to user interaction events. Dashboard interactions and customizations are limited. Also, *vizlets* can't be customized and reused, since the only model entity that can be built and saved are dashboards. In short, VizDeck focus on make dashboard creation easy and fast, but lacks flexibility, reuse, self-service preparation, data source connectivity possibilities and metadata management.

***Microsoft Power BI***

Microsoft was classified leader and visionary for BI and Analytic Platforms after a focused analysis of Power BI [10]. Microsoft's vision for Power BI with natural-language query (C7.8), Cortana personal digital assistant (C7.8), strong partner network and prebuilt contents strategy, positions it as best solution regarding the Completeness of Vision, accordingly with Gartner. Power BI is one of the lowest-priced solutions on the market today, particularly from larger vendors, providing a desktop-based (Power BI Desktop) application and browser-based (Power BI Service) online software service, from now on called just as "Power BI".

Power BI is built on Microsoft's cloud computing infrastructure and platform, Azure, supporting high availability, where each of the active deployments around the world have an equal number of passive deployments that serve as backups [27]. Power BI architecture building blocks, in Figure 0.1, are a Web Front End that manages the initial connection and authentication to the Power BI service and a Back End that handles all visualizations, dashboards, reports, datasets, data connectivity, storage or refresh, and other elements of Power BI. The Back End is composed of a Presentation Role that manages data presentation and how contents are shown to the user, a Data Role that manages data (data at rest), and a Data Movement Role that manages data related events, such as updates or refreshes. For instance, when data changes, the Data Role create and loads an in-memory database with the dataset, managed then by Data Movement Role.

Power BI architecture shows deep concern in security and background manage of data, with a clear division between presentation responsibility, data management and data changes. With regard to storing of data itself, Power BI uses two repositories: Azure Blob for data uploaded from users; Azure SQL Database to all metadata, such as dashboards, reports, recent data sources, workspaces, or other metadata about the system.

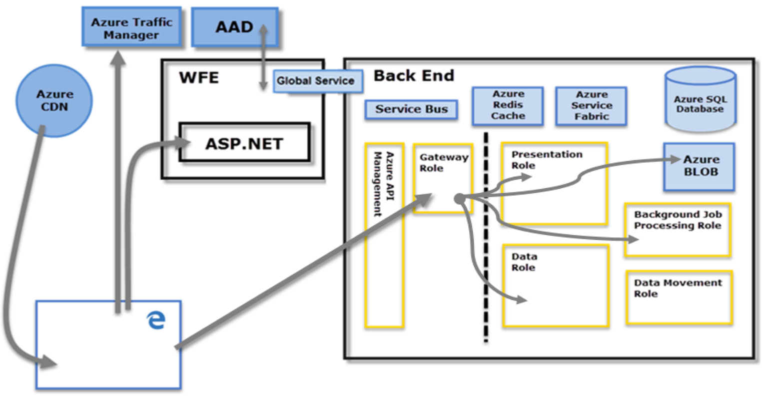


Figure 0.1: Power BI architecture, source: [27]

Microsoft PowerBI offers an easy and quick way of creating self-service dashboards, reports, datasets, and visualizations supported by the three major building blocks encapsulating informational entities architectural aspects: Dashboards, Reports and Datasets. These components express data access (Datasets), data visualization (Reports) and data monitoring (Dashboards):

- **Datasets** allow connections to one or more data sources and manages the data model fields (categories and measures) providing access to needed data feeding each Visual to provide Data Visualization and Interaction. This Dataset informational entity is limited in terms of flexibility, because only the dataset owner can change the way a field is categorized, making the analytic model to restrict when using Fields as measures or categories, damaging criteria C4.2 score;

- **Reports** are an aggregator of Visuals in a tab-based organization called Pages. Visuals can be created and customized by users, through the Q&A (Question & Answers) interface or from Quick Insights. Q&A explores data using natural language capabilities and Quick Insights generates visualizations through automatic analytic process of data. Each Report uses a single Dataset and is composed by one or more Report Pages. Furthermore, each Report Page is composed by one or more Visuals, such as charts, graphs, images, KPI's or other interactive visualizations that can be filtered. Filters can also be used in Report Pages or entire Reports;

- **Dashboards** are composed by one or more Tiles. A Tile is a snapshot of data and configuration, pinned to one dashboard. Tiles can be created from Report Pages, Visuals, Datasets, Q&A, Quick Insights or Widgets (standalone Tiles such as text boxes or images). Dashboards can be shared and data classification tags can be added to categorize data importance, type, relevance, or any other classification desired.

Power BI presents a wide variety of standard and advanced visualization types, such as charts, simple indicators or single value. Also, developer visualization types can be added, allowing bigger flexibility to create and customize visualizations (reflected in a top score at C7.3). Quick Insights and Q&A interface grants an easy way to analyze data and generate Visuals even when users struggle to know what to look for. Although a wide variety of powerful exploration and interactive functionalities, Power BI lacks in a simple support for displaying data as percentages (C7.4).

In terms of interaction within the dashboard, Power BI shows some limitations. The interaction between Tiles can't be totally customized (C5.6) and have limit user interaction, since a Tile configuration can't be changed. Interaction in the Dashboard forces a "drill down" interaction to the report, dataset, or other base configuration, depending on the Tile type. The exception is Report Tiles where an entire page of one Report is added to a "Live" Tile and a user can interact with Reports on the dashboard. Also any changes made to visualizations in the report editor are reflected in the dashboard tile as well. C8.3 (Immediate visualization type changes) received a 2.5 score (partially achieved) due to this exception.

Another import component is Data Alerts. Data Alerts can be associated to Tiles, managing one or more Alert Rules to generate notifications, based on a conditional analysis of data. This Data Alert allows notifications by email or in the platform notification center, but no other dispatch mean. This lack of flexibility when it comes to choose the notification dispatch mean affects the score of criteria C11.1. A missing requirement is collaborative work (C10) capability that is not contemplated. The metadata model is clear and expressive but doesn't allow an extension to metadata fields by the user (C6.2).

***Pyramid Analytics BI Office***

Gartner selected Pyramid Analytics as a niche player in BI and Analytic Platforms. Its inclusion was based on Pyramid Analytics BI Office Platform (from now on referred as "Pyramid Analytics") where can be found a broad range of interactive visualizations, analytic dashboards with some advanced analytics capabilities, extensive drilling capabilities and a strong metadata management. All this delivered in a easy to use interface, accordingly with customer ratings [10].

Pyramid Analytics scores below average at criteria C2.1 (Architecture and software strength) since some customers raised concerns about the use of Silverlight in the software [30]. Despite this, Pyramid Analytics presents a clear architecture where each layer can be scaled up. Figure 0.2 shows this layered architecture approach:

- **Web self-service layer** exposes the four main building blocks of the platform model and uses the service layer to provide the needed functionalities:
    - **Data Model** defines data relationships and uses one or more data sources to building categories, hierarchies and measures. These data models are then used to build reports in Data Discovery;

- Data Discovery is based on reports, organized in panels, and uses one data model to generate visual elements. These visuals, depending on the report panel type, can be shown in many different ways, such as tables, charts or gauges;

- Story Board (or Dashboard) can have one or more slides with a predefined personalized layout. This predefinition restriction affects C8.4 layout criteria since the layout can be customized, but isn't dynamic by adding report contents (report panels and sliders) to the Dashboard. A special case of this dashboard (added) objects are global slicers, where a slicer is imported in a global way and not in a specific slide. Also, some assets can be imported, such as shapes, images, URL's, and so on;

- Publication provides the creation of written reports with embedded visualizations and is not relevant for the present study, except for the creation of a special type of written report: Alerts. Each Alert represents a series of trigger configurations and lacks in flexibility (C11.1) when it comes to choose the notification dispatch since alerts can only be sent to the platform or by email.

- **Service layer** grants scalability and manages, in a centralized way, security and content management. Is completely independent from client side and is accessible through services;
- **Data layer** manages and provides data access. The Data Model uses user authorization and privileges information to access data, managing data security this way.

Pyramid Analytic presents a clear, easy to use, interface allied to a wide variety of standard/advanced visualization types and special attention to KPI's graphical visualizations. With a custom and specialized interface module for KPI construction, Pyramid Analytic, receiving a top score at KPI construction criteria C5.4, allowing the definition of thresholds and targets using one metric or the comparison between more than one metric (analyzing evolution).

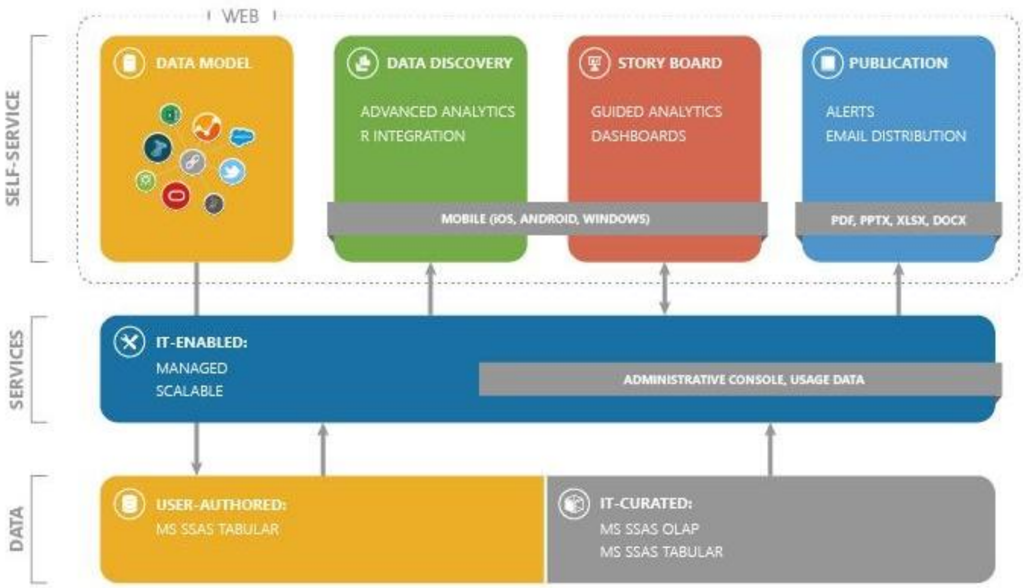


Figure 0.2: Pyramid Analytics layered architecture, source: Pyramid Analytics Site

Collaborative work capability lacks in support of ratings and recommendations (C10.3), but a highly flexible creation of conversations and comments associated to different levels of the application (C10.4), such as Dashboard Slides, Reports, Measures, or Cells, represents a strong point in this platform. Security management (where across all components, model roles are defined to match permissions to roles in each metadata entity created) is also very consistent.

In terms of interaction management between dashboard objects, Pyramid Analytics, grants the best flexibility and simplicity in the current comparison analyses (top score ate C7.6), but interaction and customization of dashboard objects by the user are limited since it depends on the pre-established type of the report page without the possibility of change it at the dashboard in “run-time” (C8.3). Also data interaction cannot be done using any semantic search (C7.8). The metadata model is clear and well defined, but doesn’t allow an extension to metadata fields by the user (C6.2).

***Tableau***

Tableau is classified by Gartner [10] as a leader for BI and Analytic Platforms. According to Gartner, Tableau's clear strength is in interactive visual exploration where they have been improving user experience with an easy to use drag and drop set of functionalities (C4.1 and C5.1). Tableau has three primary products: Tableau Desktop, Tableau Server and Tableau Online. This analysis focus on Tableau 10.1 Desktop and Server Professional Editions, from now on called just as “Tableau”. Figure 0.3 presents’ Tableau highly, scalable and stable layered architecture [28]:

- **Data connectors layer** manage data integration and data source connectivity, allowing users to create an informational entity “Connection” without any scripting skills. Over thirty native connectors are provided, but all can be defined with few connectors’ types like ODBC, MDX or flat files. A Connection stores information about data available and how to access it. The access to data can be made in two different ways. A “real-time” connection where a new query is sent to the database every time the user changes their analysis or an initial extraction where data is stored locally (except on multi-dimensional databases, such as cubes).
- **Main Components layer** is composed by three components:
    - **The Application Server** handles security and user administration;
    - **The Data Server** manages and stores data sources, calculations, definitions or other metadata model elements represented by the Data Model informational entity. Data Model defines the dimensions and measures characterizing them automatically, either reading directly from metadata (in cubes) or with intelligent heuristics (in relational data) and uses the Connection information to access data. Also manages user-generated fields, such as data sets or calculations. Data Models can use other Data Models;
    - **The VizQL Server** manages VizQL data queries (VizQL Model) and cache shared between users. VizQL is a visual query language that translates drag-and-drop actions into data queries and then expresses that data visually. This is represented in

a run-time editable VizQL Model. This model supports all data queries made by user interactions using the Data Model definitions to build sheets/visuals and Connections to request data, providing interactive data visualization. This model allows users to adjust the Data Model changing the role and aggregation of the fields at run time. For example, a user can change a measure to a dimension.

- **Gateway and Load Balancer** are used to achieve high availability, performance and scalability when managing client applications.

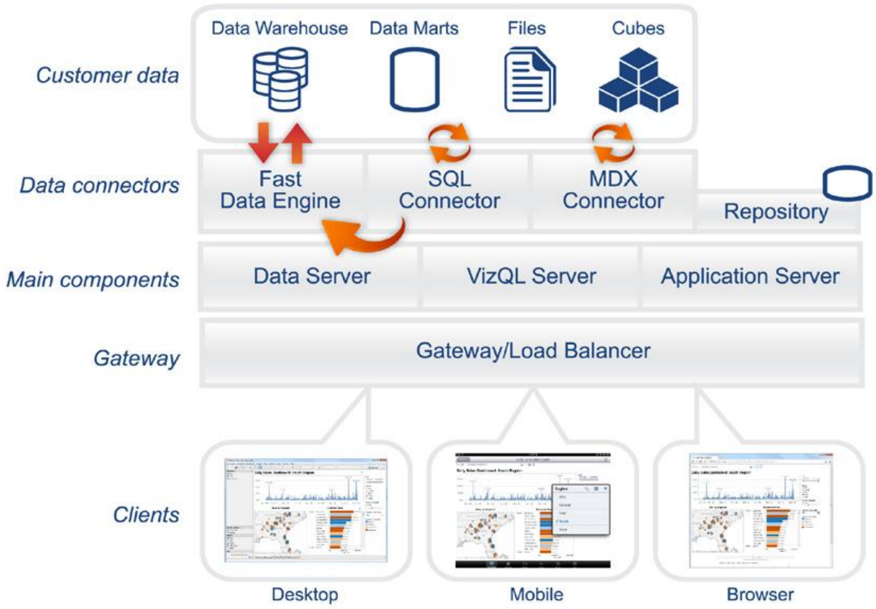


Figure 0.3: Tableau Architecture, source: [28]

Tableau delivers an easy-to-use drag and drop visualization tool where anyone (not only IT users) can analyze and visualize data, although this interaction with data cannot be done using any semantic search (C7.8). Due to its simplicity and visualization capabilities, supporting a wide variety of standard/advanced charts and simple pivot tables, the creation of interactive dashboards is easy. However, the lack of support for simple indicators, such as gouges or single values, and the over complex configuration of KPI's are a limitation, reflected in C5.4 criteria score.

In client applications, Tableau provides two kinds of sheets: worksheets and dashboard sheets. A worksheet allows the creation of a single visual interactive element, such as, a chart or a map. A dashboard sheet uses worksheet's objects and combines them together to build an interactive dashboard composed by dashboard objects. The interaction between dashboard objects can't be fully customized (C7.6), as well as the interactive customization of visual elements themselves, making this interaction within the dashboard limited. For example, in the dashboard there is no chance of changing chart type (C8.3) in one visual element. This limited interaction reflects on the lack of expressiveness and completeness in "run time" operations availability affecting the score of criteria C6.4.

Each dashboard can be published to Tableau Server, Tableau Online, Tableau Public or through a "package" containing all dependencies. It's also possible to export work in Tableau to an Excel worksheet, an Access database or make an image/pdf file from a worksheet or dashboard. Apart from publishing and exporting, Tableau allows other forms of "sharing information", such as comments or "Conditional Subscriptions". Comments can be associated with visual objects, values or other information entities objects. Conditional Subscriptions, a new feature in latest version, combines conditional sheets and subscriptions, providing conditional alerts and notifications based on some criteria and/or thresholds. This configuration is still a bit fuzzy and complicated (C11.1), opposing to the major simplicity of Tableau, but allows, for example, to configure an email notification when some salesperson achieve a sales amount target (a measure goes above a target value). The lack of support to ratings and recommendations is reflected in criteria C10.3.

When it comes to metadata management, the ability to reuse metadata objects over dashboards is quite limited, as the score to C6.1 suggests. Changes made to data models, such as created measures, can't be reused in other data source. It can be said that all modeling is "data-source-specific". Also, doesn't allow an extension to metadata fields by the user (C6.2).

***SpagoBi***

SpagoBI is a very complete, robust, secure, scalable and competitive full open source platform that can be considered an effective alternative to other commercial platforms [11], [29]. SpagoBI has several products but this analysis focus on SpagoBI Server 5, from now on called just as "SpagoBI". SpagoBI architecture is composed by four components, present in Figure 0.4:

- **Delivery Layer** provide, through the Enterprise Application Integration, the integration with other portals (C9), such as specific BI portals or generic enterprise portal services and the access to SpagoBi services for client applications;
- **Analytical Layer** represents the core of the architecture and is composed by two components: BI Applications that supports the analytical core of the platform and BI Engines that are the interfaces for the engines realizing the analytical documents. Outside of project scope, but differentiated by SpagoBi architecture, are development tools, not a predefined set of development tools by SpagoBi, but users choice and analytics models, such as a CRM or HR, outside of the project scope that can influence general models;
- **Data and Metadata Layer** manages data source and integration tools responsibilities related to source systems, achieving data and metadata integration. This integration, and the generated analytic model, lacks in flexibility (C4.2), due to the fields fixed categorization as measure or category, and in user defined analytics model limitations (C4.3). Also isn't possible for users to define new metadata properties (C6.2). All these limitations affects metadata expressiveness and completeness (C6.4);

- **Administration** provides, across all layers, support for the management functionalities, such as scheduling or user administration and profile definition, but, doesn't provide Data Encryption (C2.3).

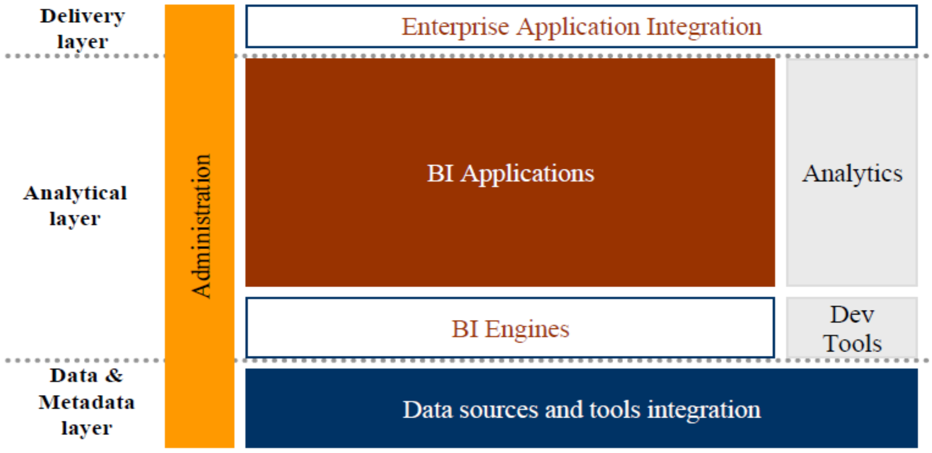


Figure 0.4: SpagoBI high level architecture overview, source: [29]

SpagoBI provides interactive dashboards, called cockpits, displayed in a single flat layout, without any type of organizational division possibility (C8.4), supporting standard/advanced interactive visualization types, such as charts, maps, tables or KPI's cards. Despite this interactive dashboard support, SpagoBI have a report-base orientation which clearly affects the overall result in the analytic dashboards related criteria (C8), where it scores the lowest score of all compared solutions. Dashboard graphical objects, comparatively with report graphical objects, support less chart types, customizations and interactions.

In terms of interactions linking customizations for interactions between dashboard objects, SpagoBI presents one of the best linking customizations flexibility (C7.6), allowing definition on how different data columns connect with each other, but it doesn´t establish any connections from the start (C7.5), imposing a configuration on every visual object before taking advantage of linking interactions.

Visualization object exploration and interaction in reports can be highly customizable, but also with some relevant limitations: No semantic search functionality is supported (C5.8), visual objects can't be formatted based on conditions (C5.7) and visual content preparation doesn't provide fast feedback (C5.2) for each user change, due to the separation between edition and "run-time". Also, some functionalities are over complex, such as KPI's, with strong support in the platform and a dedicated building view, but with a complex process for creation and management (C5.4), and conditional alerts, with a complex configuration and a restricted dispatch mean capabilities (C11.1).

Collaborative work capabilities, such as real-time collaboration (C10.2), are affected by the lacks in support of criteria C10.1, discussion threads, since each user can only add one note for each informational entity and this notes (or comments) doesn't meet the requirements for criteria C15.4, because they can only be added at high level objects, such as dashboards.

## Comparison Table Details

Table 0.1: Full detailed comparison grid for solutions survey

| | | Proposed Solution | Tableau | PowerBI | Pyramid | SpagoBI |
|---|---|---|---|---|---|---|
| **C1: Licensing** | | **A[1]** | **C[1]** | **C[1]** | **C[1]** | **OS[1]** |
| **C2: Platform Admin, Security and Architecture** | | Y[2] | **3,00** | **3,00** | **2,88** | **2,50** |
| | C2.1: Architecture and software robusticity | | 3,00 | 3,00 | 2,50 | 3,00 |
| | C2.2: Performance and scalability | | 3,00 | 3,00 | 3,00 | 3,00 |
| | C2.3: User administration and profiling | | 3,00 | 3,00 | 3,00 | 3,00 |
| | C2.4: Data Encryption | | 3,00 | 3,00 | 3,00 | 1,00 |
| **C3: Data Source Connectivity** | | Y[2] | **3,00** | **3,00** | **3,00** | **3,00** |
| | C3.1: Personal and Web data | | 3,00 | 3,00 | 3,00 | 3,00 |
| | C3.2: OLAP connectivity | | 3,00 | 3,00 | 3,00 | 3,00 |
| | C3.3: Unstructured and semistructured data | | 3,00 | 3,00 | 3,00 | 3,00 |
| | C3.4: Relational query access | | 3,00 | 3,00 | 3,00 | 3,00 |
| **C4: Self-Service Data Preparation** | | Y[2] | **3,17** | **2,83** | **3,00** | **2,50** |
| | C4.1: Drag-and-drop data preparation | | 3,50 | 3,00 | 3,00 | 3,00 |
| | C4.2: Flexibility of Analytic Models | | 3,00 | 2,50 | 3,00 | 2,00 |
| | C4.3: User Defined Analytic Models | | 3,00 | 3,00 | 3,00 | 2,50 |
| **C5: Self-Service Visual Preparation** | | Y[2] | **2,90** | **3,00** | **3,20** | **2,60** |
| | C5.1: Drag-and-drop content preparation | | 3,50 | 3,00 | 3,00 | 3,00 |
| | C5.2: Immediate visual feedback | | 3,00 | 3,00 | 3,00 | 2,00 |
| | C5.3: KPI support | | 3,00 | 3,00 | 3,00 | 3,00 |
| | C5.4: KPI construction | | 2,00 | 3,00 | 4,00 | 2,00 |
| | C6.5: Thresholds | | 3,00 | 3,00 | 3,00 | 3,00 |
| **C6: Governance and Metadata Management** | | Y[2] | **2,13** | **2,50** | **2,50** | **1,88** |
| | C6.1: Reuse | | 2,00 | 3,00 | 3,00 | 1,50 |
| | C6.2: Extension (User metadata) | | 1,00 | 1,00 | 1,00 | 1,00 |
| | C6.3: Data modeling | | 3,00 | 3,00 | 3,00 | 3,00 |
| | C6.4: Expressiveness and completeness | | 2,50 | 3,00 | 3,00 | 2,00 |
| **C7: Interactive Visual Exploration** | | Y[2] | **2,63** | **3,13** | **2,88** | **2,50** |
| | C7.1: Interactivity and exploration | | 3,00 | 3,00 | 3,00 | 3,00 |
| | C7.2: Global filters | | 3,00 | 3,00 | 3,00 | 3,00 |
| | C7.3: Chart types supported for analysis | | 3,00 | 4,00 | 3,00 | 3,00 |
| | C7.4: Display as percentages | | 3,00 | 2,50 | 3,00 | 3,00 |
| | C7.5: Visualizations linking pré-definitions | | 3,00 | 3,00 | 3,00 | 2,50 |
| | C7.6: Visualizations linking customization | | 2,00 | 2,50 | 4,00 | 3,50 |
| | C7.7: Conditional formatting | | 3,00 | 3,00 | 3,00 | 1,00 |
| | C7.8: Semantic query/search | | 1,00 | 4,00 | 1,00 | 1,00 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **C8: Interactive Dashboards** | | $Y^2$ | **2,60** | **2,80** | **2,50** | **1,90** |
| | C8.1: Chart formatting options | | 3,00 | 3,00 | 3,00 | 1,50 |
| | C8.2: Chart types supported for design | | 3,00 | 3,00 | 3,00 | 1,50 |
| | C8.3: Chart type changes | | 1,00 | 2,00 | 1,00 | 1,00 |
| | C8.4: Formatting and layout | | 3,00 | 3,00 | 2,50 | 2,50 |
| | C8.5: Load scheduling | | 3,00 | 3,00 | 3,00 | 3,00 |
| **C9: Publish and Embed Analytic Content** | | $N^2$ | **3,00** | **3,00** | **3,00** | **3,00** |
| | C9.1: Export Output format | | 3,00 | 3,00 | 3,00 | 3,00 |
| | C9.2: Portal integration | | 3,00 | 3,00 | 3,00 | 3,00 |
| | C9.3: Embeddability | | 3,00 | 3,00 | 3,00 | 3,00 |
| **C10: Collaborative Work** | | $Y^2$ | **2,50** | **1,00** | **2,75** | **2,38** |
| | C10.1: Discussion threads | | 3,00 | 1,00 | 3,00 | 1,50 |
| | C10.2: Real-time collaboration | | 3,00 | 1,00 | 3,00 | 2,50 |
| | C10.3: Rating and recommendations | | 1,00 | 1,00 | 1,50 | 3,00 |
| | C10.4: Comments | | 3,00 | 1,00 | 3,50 | 2,50 |
| **C11: Event-Based Awareness** | | $Y^2$ | **2,33** | **2,50** | **2,33** | **2,00** |
| | C11.1: Conditional alerts | | 2,50 | 2,50 | 2,50 | 2,00 |
| | C11.2: Event-Based scheduling | | 2,00 | 2,00 | 2,00 | 2,00 |
| | C11.3: Event Visual Display | | 2.50 | 3.00 | 2.50 | 2,00 |
| | | **Final Score:** | **2,73** | **2,68** | **2,80** | **2,43** |