FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Wirtschaftsinformatik

Reverse Engineering of organization-specific Viewpoints: Applying and Extending Building Blocks for Enterprise Architecture Management Solutions (BEAMS)

Mariana Mykhashchuk

FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Wirtschaftsinformatik

Reverse Engineering of organization-specific Viewpoints: Applying and Extending Building Blocks for Enterprise Architecture Management Solutions (BEAMS)

Reverse Engineering von organisationsspezifischen Viewpoints: Anwendung und Erweiterung der Building Blocks für Enterprise Architecture Management Solutions (BEAMS)

Author:	Mariana Mykhashchuk
Supervisor:	Prof. Dr. rer. nat. Florian Matthes
Advisor:	M.Sc. Sascha Roth
Date:	September 15, 2011

I assure the single handed composition of this master's thesis only supported by declared ressources.

Garching, September 12, 2011 Mariana Mykhashchuk

Abstract

Complexity of today's information technology (IT) together with its ubiquity and continuously growing role in enterprise arouse enormous interest in approaches meant for managing business and IT as a whole. A commonly accepted means to enhance the alignment between business and IT by means of providing the holistic view on the organization is enterprise architecture (EA) management, which guides and supports companies in transformations within a rapidly changing environment. Despite the plurality of available publications and research initiatives in this field, the topic of EA management is still faced with a challenging lack of standardization, as no common understanding of what EA management really is has yet developed. To deal with aforementioned issue, the chair of *Software Engineering for Business Information Systems* of the Technische Universität München developed configurable *Building Blocks for Enterprise Architecture Management Solutions (BEAMS)* approach, which represents a collection of best-practice solutions for recurring EA management problems and provides a practical guidance for organizations to design an organization-specific EA management function. Nevertheless, BEAMS approach is still relatively young and its thorough evaluation has not been undertaken.

In the course of this master's thesis, an applicability evaluation of the visual constituent of BEAMS approach is performed by decomposing viewpoints, collected from theory and practice, in modular components, which are further compared with viewpoint building blocks available in BEAMS collection. Gaps detected during the applicability evaluation phase are subsequently used as a basis for extending BEAMS viewpoint building blocks with new candidates.

Contents

List of Figures

List of Tables

1. Introduction and overview

The growth and rapid development of information technology (IT) in recent years entailed considerable changes in the role of IT in business. Being once considered solely as a cost center, IT is currently regarded rather as a value creator and, even more, undertakes the role of business enabler [Kr05].

1.1. Motivation and scope

Intensive use of information technology in business since the mid of 90s, together with simultaneous restructuring of the companies, gave rise to a new phenomenon of industrial society: the fully networked companies, i.e. organizations in which all essential business processes, all business functional areas, and relations to the corporate environment, in particular to customers and suppliers, are supported by IT [LLS10]. Being used along the whole value chain for both primary activities (e.g. purchasing, production, inbound and outbound logistics, operations scheduling, marketing, distribution and customer service) and support activities (e.g. administration, human resource management, accounting, technical infrastructure), and playing important role in linkages among them, information technology has become completely pervasive in the value chain and is having profound impact on competitive advantages of an enterprise [LLS10, Lu04]. Moreover, some business concepts are completely based on the usage of information technology and are absolutely unimaginable without IT [La08], e.g. e-business, e-commerce.
On the other hand, increasing support role of IT for enterprise workflow activities, together with ever-changing environment, triggered by globalized markets, specialized customer demands, shorter time to markets, emerging legal regulations, etc. [My11], which organizations permanently have to react and to adapt to, led to an explosive growth of corporate IT application landscapes, the parts of which were often locally developed to improve business-IT alignment. This resulted in huge and complex, historically grown, heterogeneous IT application landscapes, as they are found today in many organizations [Se05, De06]. Furthermore, application systems can not be considered on a stand-alone basis, but rather within a complex interaction and interdependencies of interwoven components, e.g. technologies they are developed on, organizational units they are operated by, interfaces through which they are connected to each other, hardware they are installed on, projects they are modified by, business objectives they should be aligned with, business processes they are supporting, etc. [La08, Wi07]
Complexity of today's information technology, together with its ubiquity and continuously growing role in enterprise, arouse enormous interest in approaches meant for managing business and IT as a whole. A commonly accepted means to enhance the alignment between business and IT in ever-changing environment by means of providing the holistic

view on the organization is enterprise architecture (EA) management [Bu10]. EA management enables the managed evolution of the enterprise and supports it in the transformation process by providing architectural descriptions of current, planned, and future states of enterprise. In doing so, EA management is strongly supported by an integral part of EA management function and its important instrument - visualizations, or so called viewpoints - to be able to get the picture of complex interrelationships between the components in organizations (e.g. business processes, infrastructure components, projects, IT architectures, strategies, etc.) and document them, analyze data and information in order to better manage what is going on, plan and predict what will likely happen under particular conditions in the future [Fe09], describe current, planned and future states of enterprise, as well as ensure communication and collaboration between different stakeholders by EA management function realization.
Due to aforementioned significance of EA management topic, a plurality of approaches for establishing EA management function in organizations have been developed in recent years by researchers, practitioners, standardization bodies, and tool vendors. However, despite a wide variety of available publications in this field, no commonly accepted understanding of what EA management really is has yet developed, followed by the lack of terminological clearness, which can be explained by the fact that organizational structures, contexts, cultures and requirements are very specific for each enterprise, and thus require organization-specific EA management function fulfillment [Bu10, My11]. Furthermore, many approaches in this field introduce EA management from scratch without considering actual maturity level of the organization, or are too abstract and too extensive to be implemented in the real world [Bu08].
In order to address the aforementioned issues, the chair for *Software Engineering for Business Information Systems* at the Institute for Informatics of the Technische Universität München developed a configurable, building blocks based approach, which represents a collection of practice-proven, redundancy-free and composeable building blocks for designing EA management problem-specific solutions. Having its groundings in prominent approaches from academic research and standardization bodies, proved to be successful in practice, BEAMS provides a comprehensive technique to support the enterprises seeking to establish organization-specific function. Being, however, relatively young approach, BEAMS requires a thorough evaluation and evolvement [Bu10].

1.2. Objectives of the thesis

This master's thesis addresses the visual constituent of EA management function implementation, intended to support a specific audience interested in specific concerns, and focuses on the evolvement of BEAMS viewpoint building blocks knowledge base.
The first objective of the thesis lies in Viewpoint Catalog construction performed by collecting and structuring viewpoints commonly used in literature and practice. A further purpose consists in evaluating BEAMS VBBs against the Viewpoint Catalog by decomposing its viewpoints into modular components and comparing them with viewpoint building blocks of BEAMS. And, finally, the main purpose of this thesis is to identify and document so called new viewpoint building block candidates, i.e. building blocks detected during

Viewpoint Catalog decomposition phase which are not yet a part of BEAMS collection, as well as make improvement suggestion for reorganization of existing VBBs.

1.3. Structure of the thesis

The subsequent Chapter 2 makes an introduction to the topic EA management and EA management related terminology, and discusses best-practice approaches for organization-specific EA management function design, namely the EA management pattern based and building block based approaches, with particular focus on the viewpoint perspective.
Chapter 3 reviews viewpoints, used in literature on EA management related areas, disclosing authors motivation in considering visualization techniques and providing viewpoints, identified in the corresponding research area, together with their descriptions.
Chapter 4 is concerned with Viewpoint Catalog construction revealing results of explorative analysis of specialized literature on EA management as well as outcomes of a case study by a real-life financial service provider, describes the viewpoint decomposition process and considers Viewpoint Catalog decomposition into modular building blocks on concrete examples.
Applicability evaluation of BEAMS VBBs is performed in Chapter 5. This chapter is also focused on the extension of existing VBBs with the new structural and symbol viewpoint building block candidates, deduced from performed viewpoint decomposition in Chapter 4. Additionally, some reorganization improvements for structural and symbol BEAMS VBBs are suggested in this chapter.
Finally, Chapter 6 summarizes the previous chapters, integrates the results of the thesis, and gives an outlook on further research.

2. Definitions and foundations

In order to address aforementioned lack of terminological clearness in the field of EA management and prepare the ground for further research, this chapter provides theoretical foundations and definitions of central concepts of this work. Initially, EA management concept is described in detail, afterwards the description of terminology used in this thesis is provided, and finally best-practice approaches for designing EA management function in organizations are discussed.

2.1. Enterprise architecture management

Since a plurality of definitions and approaches in the area of EA management exists, the goal of this section lies in providing definitions and explaining EA management related concepts used throughout the remainder of this master's thesis.

To explain the central constituent of the term EA management, *architecture*, definition of international standard ISO/IEC 42010 [IE00] is referred to, which specifies architecture as "the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution". Based on this general definition of architecture, [Wi07] defines *enterprise architecture* as coherent and holistic architecture of a certain enterprise, which comprises both business and IT elements in such a way that not only individual elements of a company are considered, e.g. organizational structure, business processes, application systems and infrastructure components, but also their connections and crosscutting elements such as strategies and goals, demands and projects, principles and patterns as well as key performance indicators and metrics. Thus, the EA allows to recognize the interplay between the different elements of enterprise and identify the interactions and interrelations between them, which are not recognizable within the scope of individual management processes of this elements.

Being committed to representation of all relevant elements with all interrelationships between them, EA information models achieve in some cases considerable complexity of more than 300 classes [Wi07] and are very specific for each organization. Nevertheless certain patterns, similar for different information models, can be found. Figure 2.1 provides a conceptual framework for the EA and gives an overview of the general aspects, same for each EA information model, called layers and cross functions. Architectural layers, namely *Business&Organization*, *Application&Information* and *Infrastructure&Data*, represent the overall EA structure of a company, describing logic concepts abstracting from their technical realization (e.g. products, business processes, organizational units, etc.), IT application systems supporting the organization's business execution, and the technical components, e.g. middleware and hardware systems, storage capacity, etc., used by business applications, respectively. Architecture layers are complemented by abstraction lay-

ers named *Business capabilities*, *Business services* and *Infrastructure services*, which describe EA concepts with a focus on provided functionalities of the corresponding architectural layer, instead of concretizing technical details of its realization. The crosscutting functions *Visions&Goals*, *Strategies&Projects*, *Principles&Standards* and *Questions&KPIs* may affect the elements of any layer. Thus, *Visions&Goals* addresses long range vision of an enterprise and its operational goals, deduced from enterprise vision and generating calls for action. These calls for action are implemented in a form of demands and projects within the crosscutting function *Strategies&Projects*. Thereby, success has to be measured (*Questions&KPIs*) and standards have to be followed (*Principles&Standards*). Individual components of layers and crosscutting functions diagram are described in [Se11].
To present and communicate data, stored according to information model, support of visualizations is needed. By means of visual representations complex interrelationships between elements of one or several layers and crosscutting functions, interactions of the elements, the values of their key measures, and their behavior in time can be reflected. For example, software maps, which are the graphical models of application landscapes, i.e. constellation of business applications of a certain enterprise, including communication relationships between them as well as their characteristics [Wi07], are very useful for detailed visualizing of the interplay between the elements of different layers and crosscutting functions. If, however, large information volumes have to be presented in compact form, dashboards visualizations may be used.

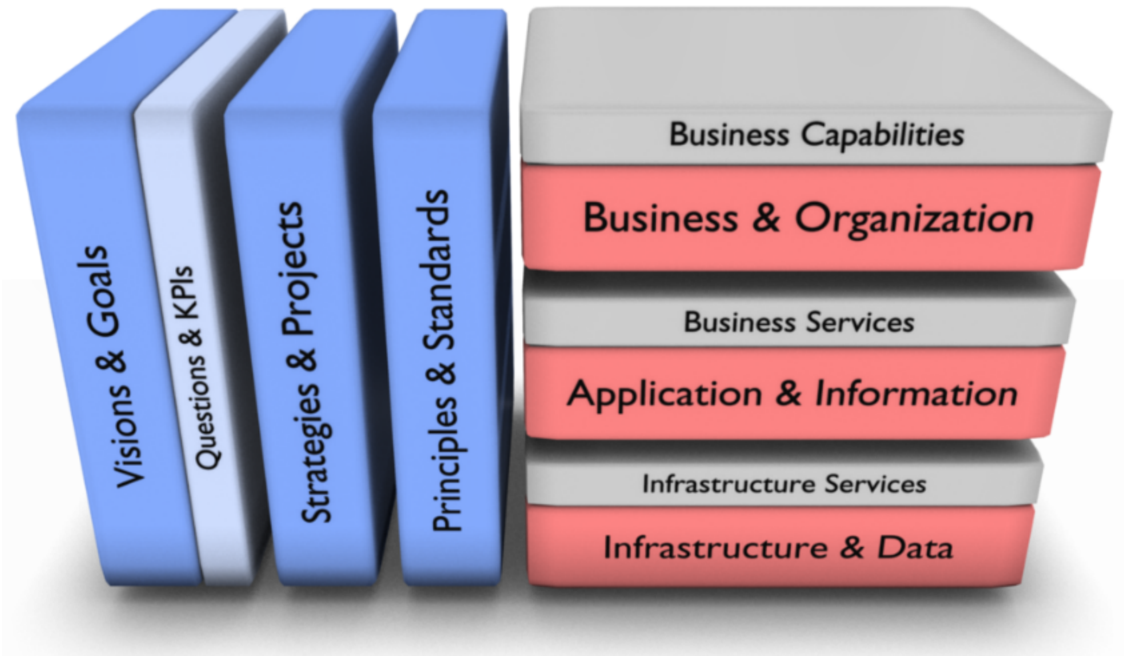


Figure 2.1.: Layers and crosscutting functions diagram. Source: [Se11]

Finally, the concept of *EA management* is used in this thesis in the meaning defined in [Ma08] as "a continuous and iterative process controlling and improving the existing and planned IT support for an organization", which not only considers enterprise IT, but also

business processes, business goals, strategies, etc. to create a holistic view on the enterprise. Thus, the task of EA management consists in documenting, managing and controlling the interplay between different interrelated elements of a certain enterprise on an architectural level. One of the main goals of EA management lies in continuous alignment of business and IT, achieved by identifying opportunities and problems arising from present state of the EA.
EA management process does not take place on a stand-alone basis and should be necessarily linked to related management processes. Figure 2.2 depicts the integration of different management processes, related to EA management, into one continuous process. Involved management processes are described in detail in [Ma08].
Certain correlations between individual management processes, shown in Figure 2.2, and cross functions of *Layers and crosscutting functions* diagram, depicted in Figure 2.1, can be noticed. *Demand management* process starts with identification of calls for actions, which are generated according to company's vision and goals of crosscutting function of the same name, and in coordination with *Strategies & goals management* process it prepares the corresponding demand initiatives and project proposals, discussed by *Strategies&Projects* crosscutting function. This crosscutting function is also correlates with *Project portfolio management* process, which proceeds with the execution of approved project portfolio, and *Syncronization management process* coordinating and synchronizing executed projects according to their interdependencies and interrelations. *IT architecture management* process, concerned with standardization of application landscapes development, resembles *Principles&Standards* cross function.

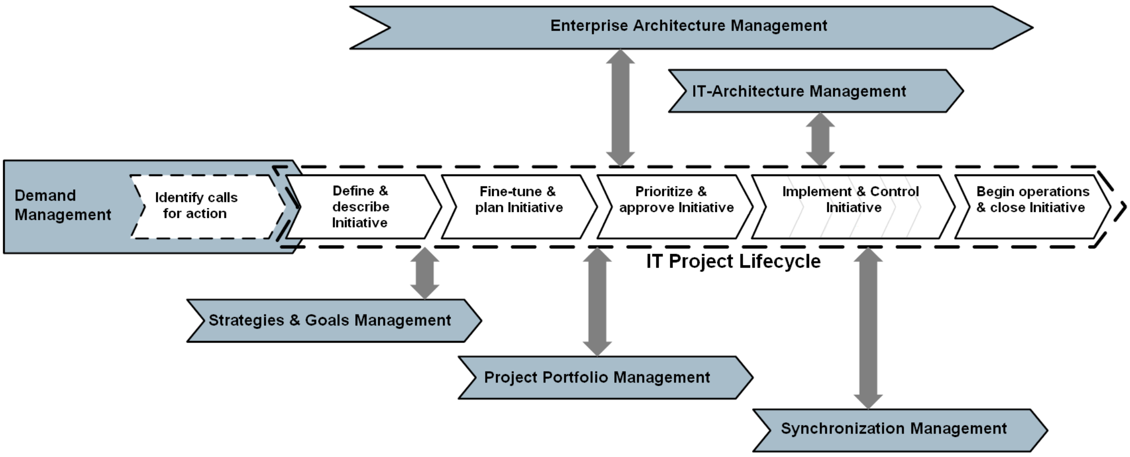


Figure 2.2.: Integration of management areas within EA management. Source: [Ma08]

2.2. ISO standard

In order to facilitate EA management function performance, a common terminology and understanding are necessary to provide the communication basis for the various stakeholders involved in EA management process. Although there is still a lack of standardization in the field of EA management, the first important step has been taken in this direction by the Institute of Electrical and Electronics Engineers Standards Association (IEEE-SA) by developing a standard for system architecture description, which was subsequently adopted by the International Organization for Standardization (ISO) and is currently known as the international standard ISO/IEC 42010. This standard introduces a framework of key concepts for architectural description, shown in Figure 2.3 together with their interrelations and defined in [IE00] as follows:

- The term *system* is defined as "a collection of components organized to accomplish a specific function or set of functions" [IE00], and can be understood as individual application, system, subsystem, system of systems or even the whole enterprise
- A certain system is always considered within a certain *environment*, represented by developmental, operational, political, etc. factors, which have an influence on the system under consideration
- Each system has *stakeholders*, i.e. a party (e.g. individuals, such as users, architects, developers, evaluators, etc., teams or even organizations) which has certain concerns regarding to the system under consideration
- *Concerns* represent the interests referring to any aspects of the system critical for one or more stakeholders
- Each system is created to realize a certain *mission*, i.e. to perform tasks for which a system is intended by one or more stakeholders according to their goals
- Each system has an *architecture* (see Section 2.1)
- A certain architecture is documented by *architectural description*, which consists of views and models according to Figure 2.3
- Architectural descriptions are represented by one or more *views*, which are defined as "a representation of the whole system from the perspective of a related sets of concerns" [IE00]. Certain views are selected by architectural descriptions based on stakeholders, the architectural description under consideration is addressed to, and respective stakeholder's concerns. Thus, architectural descriptions, which consist of views, are meaningless, if they are not based on and aligned with concerns of stakeholders. Views are created, depicted and analyzed based on rules, determined by viewpoins, and may consist of one or more architectural models
- A certain *architectural model* is created according to the methods of respective viewpoint and may participate in multiple views
- *Viewpoints* provide conventions for constructing and using the view by specifying the languages, inclusive of notations and models, intended for views description, as well as modeling methods and analysis techniques meant for these representations of the view. Viewpoints provide templates, based on which individual views can be developed. According to [Wi07], a viewpoint relates to a certain view as a class to one of its objects

- Viewpoints originating from outside of architectural description are referred as *library viewpoints*. Library viewpoints may originate from the literature or be predefined as standards within an enterprise [Wi07]

- *Technical rationale* describes the reasons for selection of a certain architecture [Wi07]

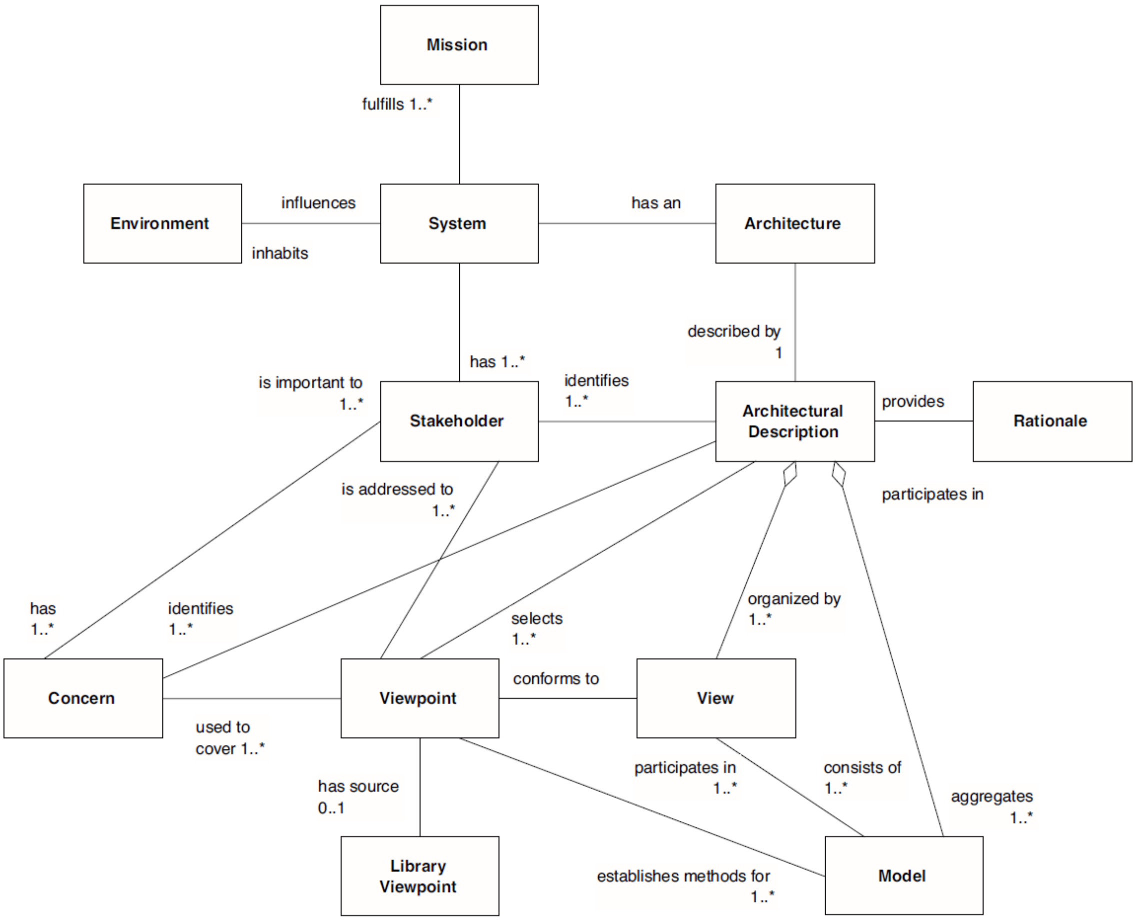


Figure 2.3.: Conceptual model of architectural description. Source: [IE00]

The concepts of viewpoint, architecture model, stakeholder, concern and architecture, used in this thesis, are understood according to the term definition specified in ISO/IEC 42010, as described above.

2.3. Best practices for visualizations in EA management

ISO/IEC 42010, considered in detail in previous section, brings together the concepts of view, concern and stakeholder by stating that only visual representations of a system that address certain concerns of certain stakeholders have the right to exist. Another important contribution of this standard consists in defining the concept of viewpoint, which determines the languages, architectural models and modeling methods to be used to describe the view. This ideas are followed by EAMPC and BEAMS best-practice approaches in the field of EA management considered in subsequent subsections.

2.3.1. Pattern Catalog - V-Patterns

Research in the field of EA management has a distinguishing tendency to develop new models and approaches instead of improving the existing ones. According to [Bu07], there are scarcely any prominent findings in the area that are further verified and developed by another researchers. Also practitioners prefer to use self-defined guidelines, neglecting results made available by academic research. That may be one of the reasons why despite of a plurality of existing research in this field still no common understanding of EA management was developed.
In order to complement existing EA management frameworks, which usually introduce EA management from scratch without considering actual maturity level of the organization, or are too abstract and too extensive to be implemented in the real world [Bu08], the chair for *Software Engineering for Business Information Systems* at the Institute for Informatics of the Technische Universität München in the course of the Enterprise Architecture Management Pattern Catalog (EAMPC) research project has developed the EA management pattern approach. Enterprise Architecture Management Pattern Catalog is an extensible collection of best-practice EA management patterns originated from academia and practice, which describe re-usable solutions for recurring problems and can be adapted to a specific enterprise context [Ma08]. Pattern catalog demonstrates the importance of viewpoints, highlighting their ability to provide conspicuous and often intuitive basis for communication, and links them with underlying information models storing the data visualized in one or more viewpoints, concerns, i.e. pain points of the company expressed in the goals of certain stakeholders used as a starting point for patterns selection, and management methodologies, meant for describing the steps to be taken to address the respective concerns. Thereby, viewpoints, information models and concern-specific management methodologies are presented in pattern catalog as three kinds of patterns, closely interlinked with one another as shown in Figure 2.4, and defined as follows:

- *Methodology patterns (M-patterns)* offer solutions for solving the pain-points of a company, defining concrete activities to be performed and providing statements about the intended usage context
- *Viewpoint patterns (V-patterns)* describe conventions for data representation, providing the languages used by M-patterns
- *Information model patterns (I-patterns)* supply the underlying information models for data required to generate a particular viewpoint

Each pattern is structured according to a particular template, which is splitted in three sections: overview section, solution section and consequence section. Overview section includes such information as unique alphanumerical pattern identifier, a short and expressive pattern name, a short summary of the pattern, a number of its version, and optionally

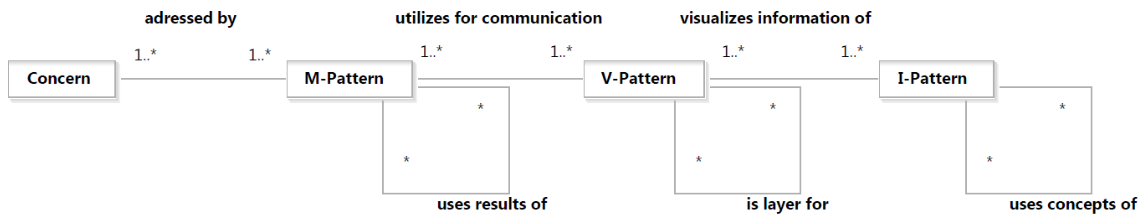


Figure 2.4.: UML class diagram describing the structure of the EA management pattern catalog. Source: [Bu08]

alternative names for this pattern, whereas solution section provides a detailed description of the pattern. Consequence section is optional and shows consequences from the usage of the corresponding EA management pattern.
Pattern catalog may be used in the following way. In the first step, concerns of the company have to be identified. This activity is supported by identically named chapter of pattern catalog, which provides the list of all 43 documented concerns, grouped by the EA management topics for easier access. Each concern has references on all related methodology patterns, which in their turn reference respective viewpoints and information models. Of particular interest for this thesis are viewpoint patterns. Altogether, 53 V-patterns are included in pattern catalog, which role is to support certain stakeholders by solving one or more of their concerns. In this sense, viewpoint patterns follow the terminology of ISO/IEC 42010. On the other hand, viewpoints borrowed the idea of layering from system cartography. According to [Wi07], layering principle allows to bring together certain characteristics of elements or relationships between them in different groups, and then assign a particular layer to each group (see Figure 2.5). Software maps use as a basis so called base maps, which determine the type of the map (e.g. cluster map, cartesian map, etc.) and on which further layers can be placed. Arranging of multiple layers on a certain base map enables a compact representation of different characteristics with high recognition value. Thus, each viewpoint pattern contains not only references on underlying information models and supported methodologies, but it also cites another viewpoint patterns in two possible ways:

- for V-patterns, which represent a certain *base map*, such viewpoints patterns are referenced, which also use the same type of base map
- if a certain V-pattern represents a certain *layer*, then it can not be used independently, without any base map. That is why such viewpoint patterns reference another viewpoint patterns (namely, base maps), where they can be alternatively used on

2.3.2. BEAMS - VBBs

Although Enterprise Architecture Management Pattern Catalog provides concern-based means for appropriate patterns selection, it lacks the integration mechanisms and artifacts, which would enable patterns integration into EA management function, defined by [Se11] as "a continuous management function seeking to improve the alignment of business and

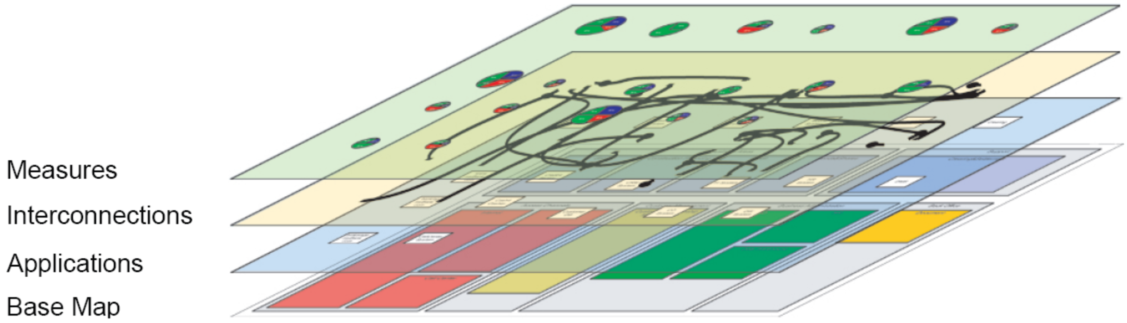


Figure 2.5.: Layering principle. Source: [Bu08]

IT and to guide the managed evolution of an organization". According to [Bu10] a design method for EA management function additionally would have to provide a guidance for:

- patterns selection, especially in case of applicability of alternative patterns
- methodology patterns integration into EA management process, preventing redundancies
- integration of V- and I-patterns into EA modeling language

In order to address this issue and to refine the EA management pattern approach a building block based approach was developed by the *sebis* chair. Building Blocks for Enterprise Architecture Management Solutions (BEAMS) provide a configurable approach to designing problem-specific EA management function by presenting method and language building blocks that can be selected and configured based on the peculiarities of the organization [Bu10]. Patterns from the EA management pattern approach are reorganized by BEAMS into practice-proven, redundancy free and composeable building blocks for EA management solutions, providing in this way a comprehensive EA modeling language. Similarly to patterns, building blocks of BEAMS are neither invented, nor developed, but observed in practice, and included into BEAMS collection according the rule of three, i.e. before a certain building block becomes a part of BEAMS at least three uses of it in EA management practice must have been detected [BMS10]. Being observed for the first time, a certain building block is registered as building block candidate and is further observed. BEAMS approach is based on the idea of interrelating competing solutions, or so-called competing theories, used to design an organization-specific EA management function. Figure 2.6 demonstrates the components of BEAMS, which are described in [Bu10, Se11] as follows:

- *Competing theories*: are the knowledge base of BEAMS represented by solutions valuable in practice, e.g. best practices, case studies, patterns, etc.
- *Problem*: is an issue to be solved, which in the area of EA management is usually expressed by certain goals to be achieved (e.g. reduce operation costs or ensure compliance, etc.) and certain concerns to be addressed (e.g. service usage or hierarchy of business processes, etc.)

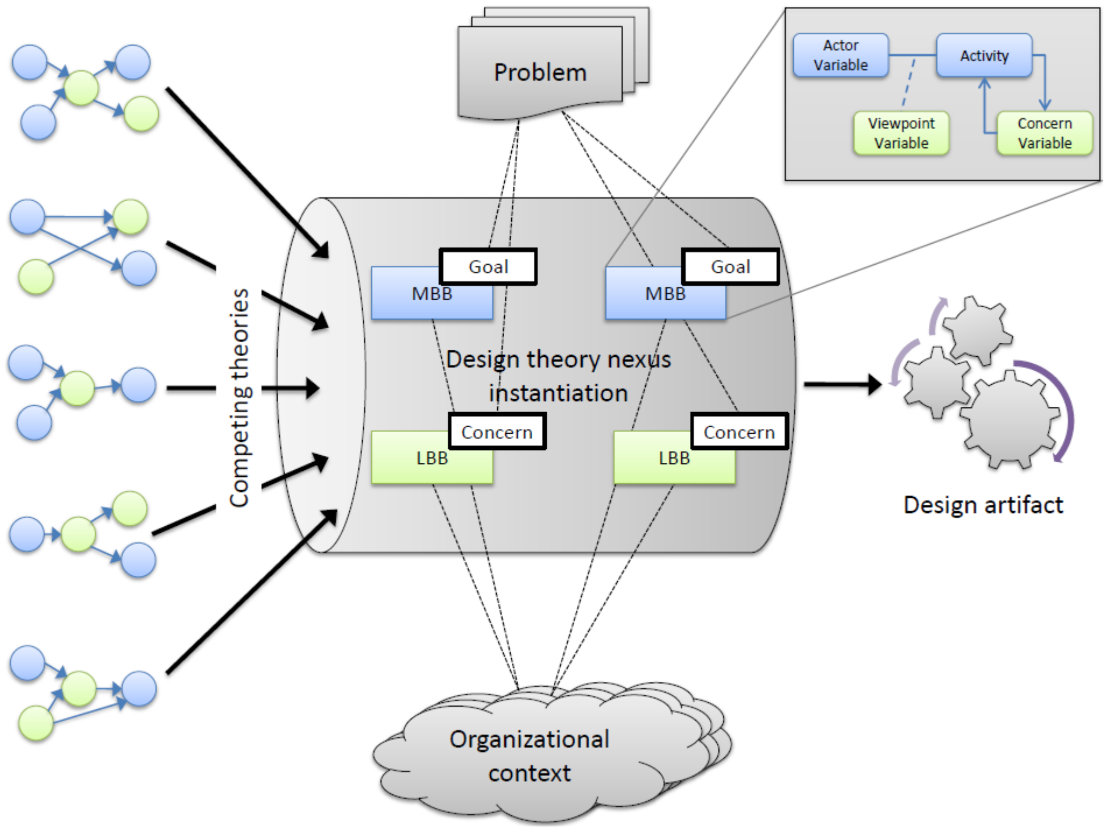


Figure 2.6.: Components of BEAMS. Source: [Bu10]

- *Organizational context*: is an environment, in which the EA management function operates (e.g. organizational culture, background of EA management initiative, etc.)
- *Building blocks*: are solution models, represented by method building blocks (MBBs) and language building blocks (LBBs), from combination of which an organization-specific EA management function is constructed
- *Method building blocks*: define actions to be performed to solve a certain problem within a certain organizational context. MBBs are splitted in four groups according to EA management function activities: *develop&describe* (i.e. MBBs for describing, developing and maintaining descriptions of current, planned, target states of the EA, and architectural principles), *communicate&enact* (i.e. MBBs for communicating EA artifacts and enacting EA plans), *analyze&evaluate* (i.e. MBBs for different kinds of analysis of EA states, e.g. whether a certain planned state helps to achieve the target state) and *configure&adapt* (i.e. MBB for assessment and improvement of EA management function itself)
- *Language building blocks*: provide MBBs with necessary information and supporting viewpoints, and are represented by information model building blocks (IBBs) and viewpoint building blocks (VBB)

The central concept of this thesis is the one of viewpoint building block (VBB), defined on a higher abstraction level than concern-specific viewpoint. While viewpoint is understood by [Se11] in line with viewpoints definition of ISO/IEC 42010 (see Section 2.2), and thus considered in the context of specific concerns of certain stakeholders, which "define the EA model elements that the viewpoint uses in constructing a corresponding view", VBBs are considered from concern-independent perspective. According to [Se11], "viewpoint building block does not build on any specific concern but may be applied to many different concerns in order to develop corresponding viewpoints". By use of BEAMS, any specific viewpoint may be produced by combination of viewpoint building blocks of different types. Three types of VBBs are considered in [Se11]:

- *Structural VBBs* describe the basic organization of a certain view
- *Symbol VBBs* provide symbols, which should be exploited in a certain view
- *Decorator VBBs* facilitate the encoding of additional information on a certain view

In terms of the layering principle discussed in Subsection 2.3.1 structural VBBs take on the role of base map, whereas symbol VBBs represent elements, which may be employed on different layers. Each viewpoint have to be necessarily composed from at least one structural VBB and at least one symbol VBB, and may optionally include one or more decorator VBBs.
Currently 8 structural, 4 symbol and 3 decorator VBBs are representing the collection of

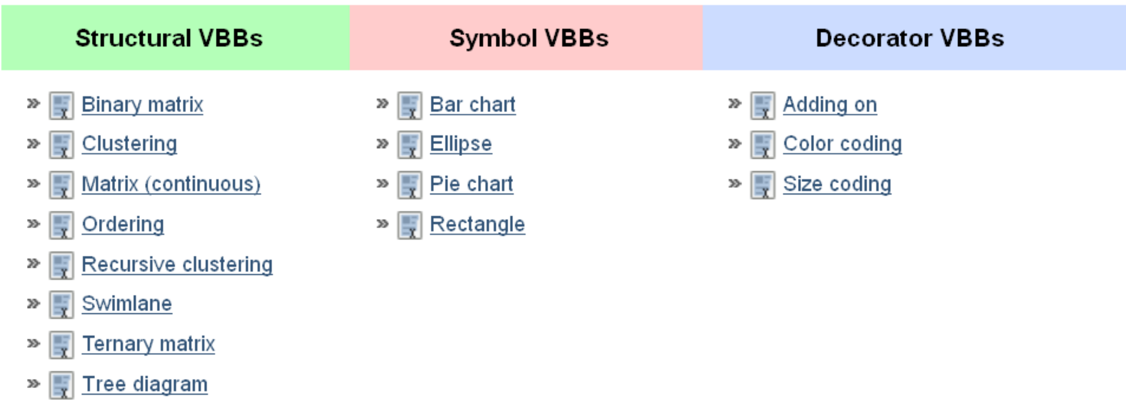


Figure 2.7.: Structural, symbol and decorator BEAMS viewpoint building blocks. Source: [Se11]

BEAMS viewpoint building blocks (see Figure 2.7). All viewpoint building blocks are described uniformly in accordance with a predefined template. Each section of this template has its name except the only one, which follows the "Tags" section and is marked in bold. To be able to address this section in remainder of this thesis, pseudonym "Usage" is assigned to this section. Some sections of this template are appropriate for certain types of VBBs and are not applicable to another. Nevertheless, such fields as "Name", "Tags", "Usage", "Explanation", "Examples" and "Type" are specified for every VBB type.

- *Name:* represents a short and expressive name of viewpoint building block

- *Tags:* contains all tags, i.e. keywords, relevant to this VBB, which help to describe it briefly and reference all related topics and issues
- *Usage:* gives a short, but meaningful summary about VBBs usage. Additionally, it specifies types of relationships or scale types of attributes appropriate for representation by this VBB
- *Explanation:* provides more concrete description of the corresponding building block, e.g. concretization of the type of relationship between the instances represented by this VBB, or description of different usage scenarios of a certain VBB
- *Examples:* illustrate possible applications of the corresponding VBB with concrete examples
- *Building block:* represents the corresponding viewpoint building block using transformation framework notation developed in [Ac10]
- *Abstract viewmodel:* displays an information model in UML notation, which supplies underlying models of data structure represented by a certain VBB. Abstract models are only provided for structural VBBs
- *Type:* depicts to which type belongs a certain VBB, e.g. structural VBB, symbol VBB or decorator VBB
- *M2 concept:* specifies VBB in terms of MOF meta model layer (M2) as an attribute or a relationship VBB. This field is only defined for structural and decorator VBBs
- *Visual variables:* represent which visual components of a certain VBB can be adjusted to graphically encode additional information, e.g. length, height, radius, border color, fill color, etc. Visual variables are only specified for symbol VBBs

3. Viewpoints overview in EA management related areas

This chapter reviews viewpoints used in literature on EA management related areas. To get an idea of visual representations commonly used in different areas, only the sources that have a focus on the subject of visualizations were researched for each application area in the course of literature review. Initially, general literature on visualizations was considered, then sources on information visualization in the field of business intelligence, which goes into practice areas of EA management, and data mining, that provides technical solutions for EA management related scope, were reviewed. Furthermore, viewpoints in the field of urban planning, which with its view on the city as on a ”dynamic, multi-layered, interconnected structure” [Be08] is often used as a metaphor for EA management, are regarded within this section. For each application area works of one author were analyzed. Subsequent sections consider thematic blocks of reviewed literature in detail, providing viewpoints identified in the corresponding block as well as motivation of authors in using them. In the course of analysis performed in this chapter, two interesting outcomes were ascertained:

- a number of viewpoints is used in all regarded application areas, e.g. bar chart, scatter plot, etc.
- same viewpoints are defined and sometimes called differently in different application domains, e.g. time series of [Tu01], line graph of [Fe09], frequency polygon of [MJ09], and line chart of [Be08]

3.1. Classic on data visualization

Information visualizations are treated generally in books of Edward Tufte [Tu01, Tu90], which have already become a classic in the area of data graphics. The main focus of these books lies on formulating principles for designing and sequential improvement of data graphics in order to explain design excellence, i.e. why some displays are better than others, and creating non-distorted visualizations. Tufte is also concerned with enhancing dimensionality and density of graphical representations to be able to use the enormous information accumulation power of visualizations at full capacity. According to Tufte, three basic structures for showing data exist, namely the sentence, the table, and the graphic, which may be used in combination with one another. In Tufte’s opinion, graphics is not only an instrument for information representation, but also a mechanism for reasoning about quantitative information.
In order to demonstrate design excellence on examples, in his books Tufte reviews numerous graphical representations from graphical practice of last two centuries, some of which

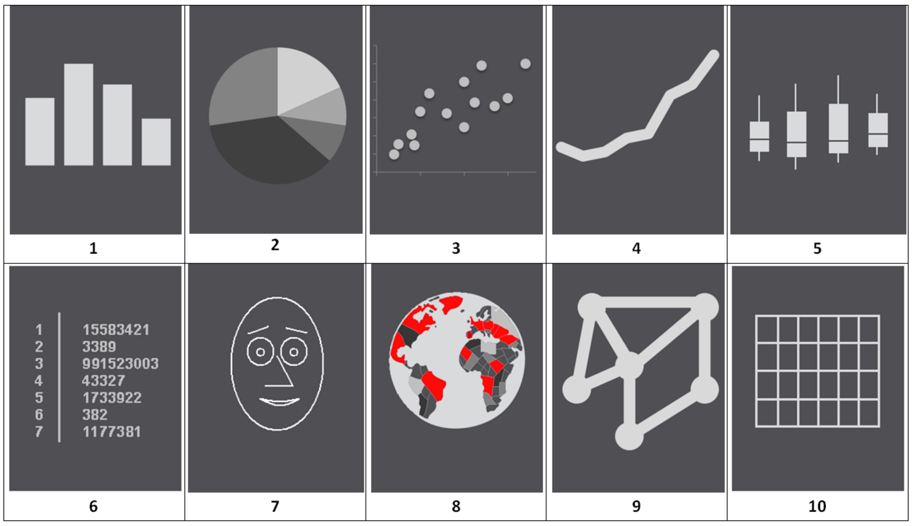


Figure 3.1.: Viewpoints from literature on classic data visualization

are unique in their art, e.g. Charles Joseph Minards drawing of disastrous Napoleon's campaign in Russia (see Figure 3.2), that brilliantly represents the loss of Napoleon's troops in time and within geographical context. On the other hand, a number of statistical graphical representations, abstracting from geographical analogy, are introduced and discussed in books of Edward Tufte. Schematical representations of viewpoints, reviewed by Tufte, are shown in Figure 3.1. Moreover, design manipulation techniques, which lead to data distortion, are exemplified by Tufte for different viewpoints.
Thus, *bar chart*, schematically shown in Figure 3.1 (case 1), invented by William Playfair,

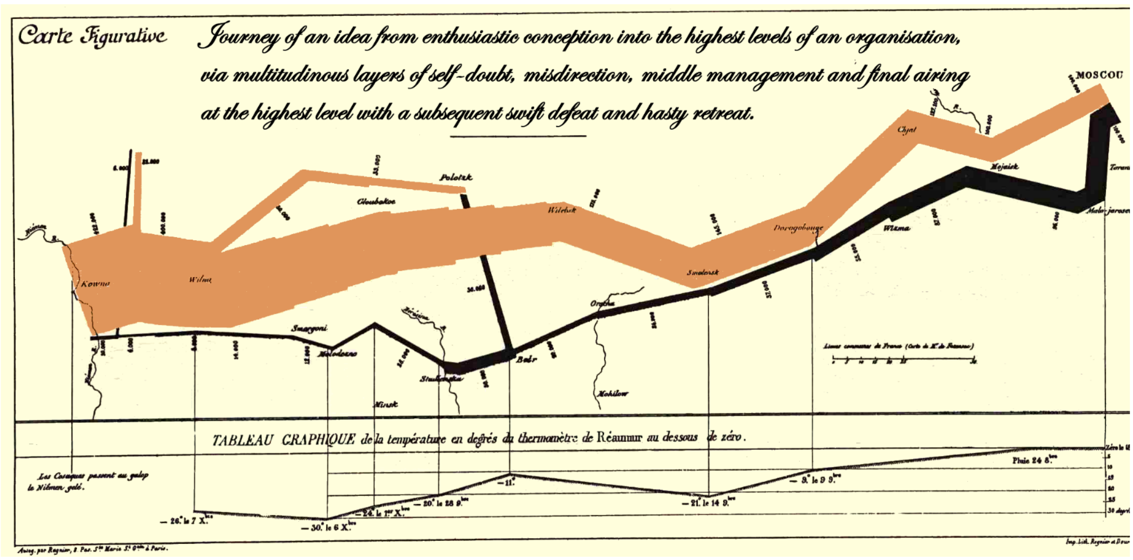


Figure 3.2.: Drawing of Charles Joseph Minard from 1869 of Napoleon's campaign of 1812 in Russia [Tu01]

a Scottish political economist, in order to compare different branches of trade by a certain characteristic and already used in 1781 (e.g. import volumes to Scotland from different countries), may be misused causing data deception by manipulating a baseline, and thus making an impression of attribute values being greater/less than they really are.
Another invention of W. Playfair, *pie chart*, appeared for the first time in his book "The Statistical Breviary" in 1801. An extract from the first representation of pie chart is shown in Figure 3.3 and visualizes proportions of the Turkish Empire located in Asia, Europe and Africa. Nevertheless, Tufte recommends not to use pie charts at all saying, "the only worse design than a pie chart is several of them", because of their low data-density, inability to order numbers along a visual dimension as well as difficulties by comparing the data from different pie charts.
Scatter plot, made for the first time by Johann Heinrich Lambert, a Swiss-German scientist and mathematician, in 1765 and shown schematically in Figure 3.1 (case 3), is intended to collate two quantitative variables and ascertain causal relationship between them. The first representation visualized evaporation rate of water as a function of temperature.
Time series, schematically depicted in Figure 3.1 (case 4) and not used until the late 1700s in scientific writings, are meant for representing behaviour of a certain quantitative variable in time. According to Tufte, the *time-series* plot is the most frequently used form of graphic design. The oldest known example of time-series usage originates from around tenth or eleventh century and represents inclinations of planetary orbits as a function of time.
Another viewpoint discussed by Tufte is *box plot*, initially introduced by Mary Eleanor

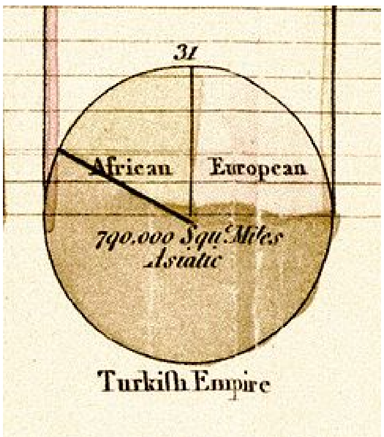


Figure 3.3.: The first representation of pie chart, used by William Playfair. Source: [Tu01]

Spear as a range bar in "Charting statistics" in 1952, and subsequently coined in its present meaning by John W. Tukey, a Princeton statistician, in 1977 as "box-and-whisker-plot" (see case 5 in Figure 3.1). Box plot is a convenient way to depict a group of data by displaying essential five-number summary, which can be easily compared for different groups: minimal value, lower quartile, median, upper quartile and maximal value.
One more invention of J.W. Tukey, so called *stem-and-leaf plot*, provides quick means to examine a distribution of values of a certain group of data and "combines summary and detail information in a single display" [Fe09]. Stem-and-leaf digram consists of two parts (see case 6 in Figure 3.1): stem, which is positioned on the left side from a vertical dividing

line, and leaves on the right side from it. Each digit from the leaf part in combination with the corresponding stem-value represents individual value of the group of data.
Case 7 in Figure 3.1 represents another viewpoint called *chernoff faces* invented by Herman

Figure 3.4.: John Snow's map of location of deaths from cholera in central London for September 1854. Source: [Tu01]

Chernoff in 1972 and used for representation of multivariate data, i.e. simultaneous representation of several variables, in a form of human face. Each individual part of face, e.g. nose, mouth, eyes, etc., defines by its size, shape, placement and orientation the value of the corresponding variable. The idea behind this viewpoint is that human perception is trained from early childhood to recognize faces and facial expressions, and thus multivariate data encoded in a form of human faces should be easy to perceive and analyze. Chernoff faces as well as scatter plots, time series and stem-and-leaf plots are considered by Tufte as practices of graphical excellence, which are able to communicate complex quantitative ideas efficiently and thereby be economical in presenting the data.
Another example of graphical excellence are *data maps*, schematically represented in Figure 3.1 (case 8). Being considered in geographical context, data maps display values of a certain variable for different regions by using different color codings or symbols for the corresponding regions. This type of data visualization is very powerful and is able to portray thousands of numbers. One of the earliest examples of data map usage is a dot map of Dr. John Snow, who in year 1854 visualized the location of deaths from cholera (with dots) in central part of London and additionally marked locations of water pumps (with crosses), and thus detected correlations between number of deaths and their distance to the Broad Street water pump (see Figure 3.4).

Tufte also considers visual representation called *narratives of space and time*, which is schematically depicted in Figure 3.1 (case 9). One of such graphs, considered in [Tu90], represents a comprehensive map/schedule for Czechoslovakia Air Transport Company from year 1933, and combines a complex spatial network of routes, including information about distances between cities, together with arrival and departure times and identification numbers of planes (see Figure 3.5).
Although Tufte does not consider *tables* as graphical representations, he admits that they are preferable to graphics in case small data sets to be visualized or exact numerical values to be represented. Tables are also useful if numerous localized comparisons have to be performed on visualization. Furthermore, Tufte introduces so called *supertables*, which are divided into topical paragraphs by horizontal rules and arranged in a way that "attracts and intrigues readers through its organized, sequential detail and reference-like quality" [Tu01].

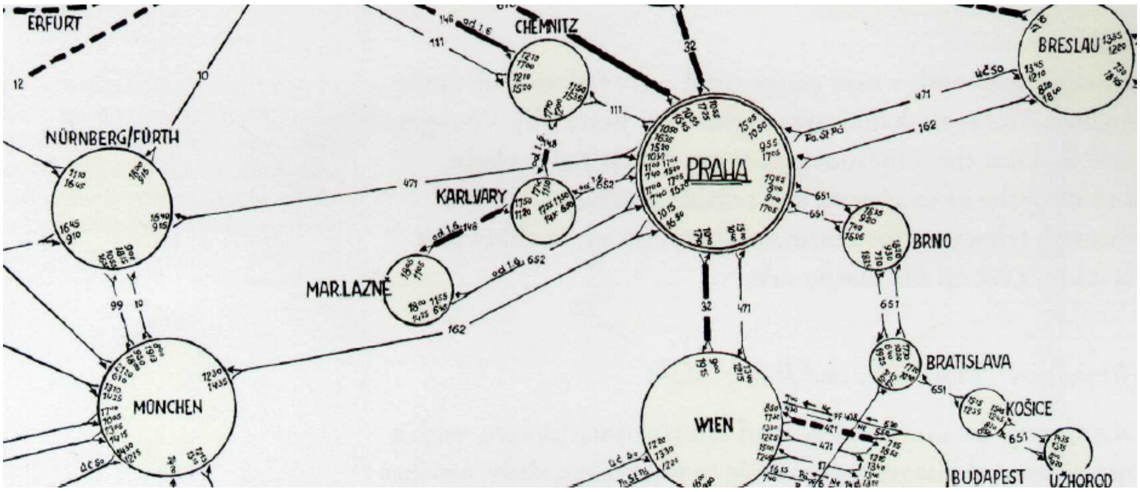


Figure 3.5.: Map/schedule for Czechoslovakia Air Transport Company in 1933. Extracted from: [Tu90]

3.2. Business intelligence

In the books of Stephen Few [Fe04, Fe09] relevance of visual representations of quantitative information is shown for business intelligence application domain. The author affirms that true business intelligence, which is supposed to provide methods and technologies to support data analysis and reporting, is impossible without a human component able to make sense of data. In his opinion, it is immensely important to understand by means of data what is really going on in a certain organization to be able to better manage company's current state, as well as make better predictions, so that opportunities can be created and problems can be avoided. The greatest concern of Few is the lack of training in visual analysis, therefore, in his monographs he teaches the fundamental skills needed to present quantitative information visually to others, on the one hand, and understand the message

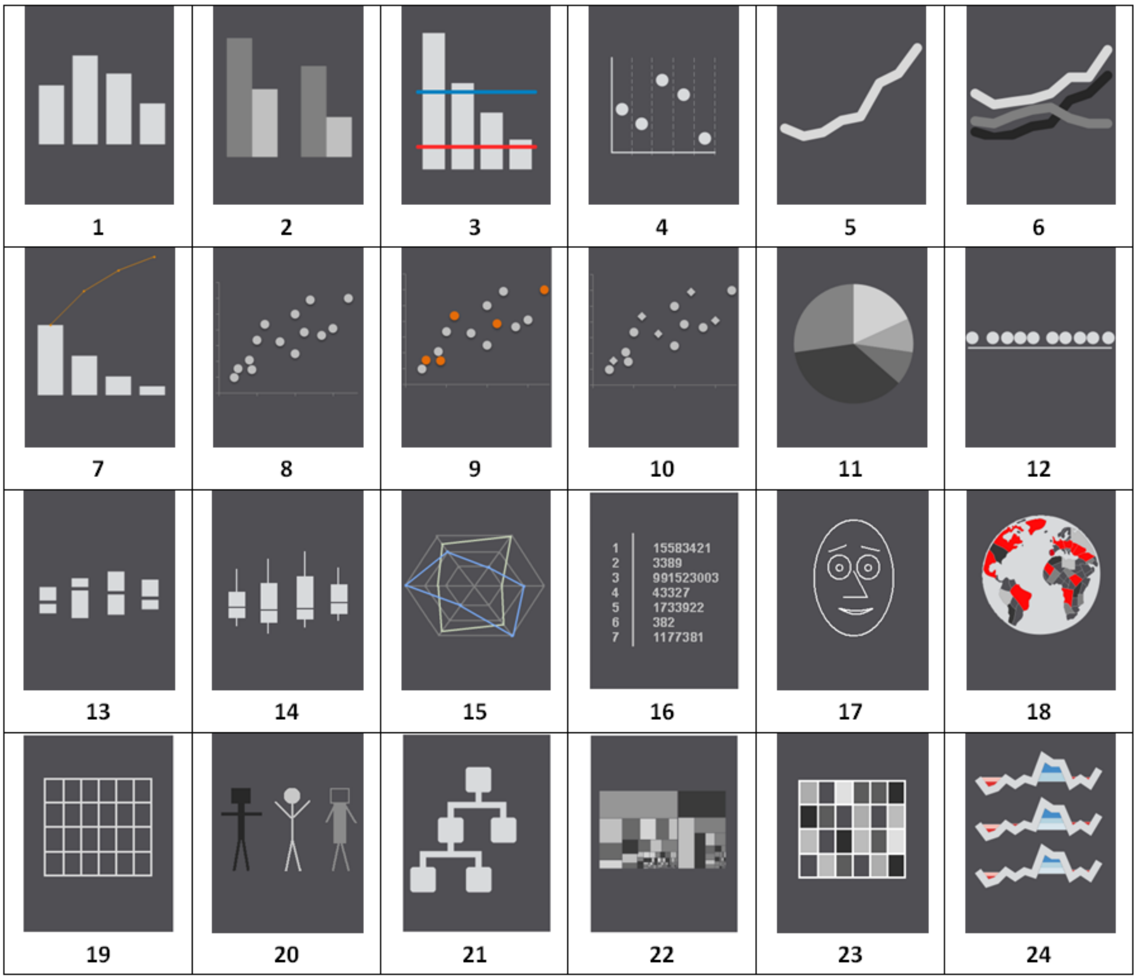


Figure 3.6.: Viewpoints in the area of business intelligence

contained in the data, on the other hand. The author does not see information visualization as a goal, but rather as a means to gain an understanding of data, which is an integral basis for good decisions.

In [Fe09] the author provides a guidance on how to learn to understand visual representations and recognize patterns within them. In doing so, he considers how human visual perception and thinking work, introduces perceptual building blocks that can visually represent quantitative data, analyzes interaction techniques, which contribute to understanding of data, and overviews general methods and practices that improve effectiveness of visual analysis. Few exemplifies his findings on a number of viewpoints, shown in Figure 3.6. These viewpoints together with recommendations for their usage are considered in detail in the remainder of this section.

According to [Fe09], *bar graphs* are useful when it is necessary to emphasize individuality of values and compare their magnitudes. Separate bars share a common baseline and accurately encode a certain quantitative characteristic of categorical items, i.e. items that have no particular order or close connection between one another, visually reinforcing by

their nature the independent nature of the items (see case 1 in Figure 3.6). Bar graphs may also be used to display a distribution of values of a certain set of data. In this case bar graphs are called *histograms*, x-axis displays numeric intervals of values being measured, whereas y-axis is represented by quantitative scale that counts the number of observations in each interval. Additionally, bar graphs may be useful for time-series analysis, displaying change of a certain value over time. Further, bar graph allows to represent and compare multiple different magnitudes for each value by placing them next to one another for the corresponding item (see case 2 in Figure 3.6). It is also possible to display explicitly boundaries, shown on a graph as reference lines or as reference areas of fill color, that define the range of normal magnitudes of values (see case 3 in Figure 3.6).
Dot plots are very similar to bar charts and are also used to visualize magnitudes of individual values. Nevertheless, the magnitudes of values are not represented by rectangle heights, but vertical positions of dots related to the quantitative scale on y-axis (see case 4 in Figure 3.6). Additional property, inherent in dot plots, is that the scale placed on y-axis can be narrowed in case all values fall within a relatively narrow range and are difficult to compare. This is impossible by bar charts because of bars binding to the baseline.
The next viewpoint, called *line graph*, is very useful if the shape of change from one value to the next one to be shown, e.g. increase, decrease or no change (see case 9 in Figure 3.6). Line graph is able to visualize an overall trend and specific patterns of change in a set of values, and is most commonly used to convey change of a certain quantitative value during a continuous period of time. Whereas bar graph is applied in case values of categorical items to be shown, line graph should be used to display values along an interval scale, i.e. "continuous range of quantitative values, divided into equal intervals" [Fe09]. Another distinction of line graph from bar graph is that the first one represent an overall change, while the second one concentrates on individual values and their comparisons. Moreover, line graphs are often used to compare several same-scaled variables by representing each of them with a separate line (see case 6 in Figure 3.6). Same as bar chart, line graph may also be used to display a distribution. In this case line chart is called *frequency polygon* and can be obtained from histogram by using instead of bars line segments connecting central points of bar top sides.
Viewpoint *Pareto chart*, shown in Figure 3.6 (case 7), combines two viewpoints considered above, namely bar graph and line graph, and is used if cumulative contribution of parts to the whole to be examined. Thereby, values represented in a form of bars are ranked by size from the largest to the smallest, and in the same order their contributions are depicted with line graph starting from top side of the first bar.
The greatest strength of the next graphical representation shown in Figure 3.6 (case 8), *scatter plot*, lies in its ability to ascertain whether, how, and to what extent two quantitative variables are correlated. Thereby, each point of scatter plot encodes magnitudes of values with its horizontal and vertical positions. Scatter plots are also able to display different groups of values, differentiating between them by assigning different colors to data points (case 9 in Figure 3.6) or using different shapes instead of dots (case 10 Figure 3.6).
Pie chart is considered and explained in [Fe09] as a means to represent part-to-whole relationships, even though Few recommends not to use them for two reasons. Firstly, circle sectors, representing magnitudes of values, are difficult to estimate and compare without having labels with exact values on them, and secondly, interpretation of pie charts may appear to be rather a time consuming task in case legends are used to label the slices, forc-

ing the eyes to glance constantly back and forth between the legend and the pie chart to make sense of it. Pie chart is shown schematically by case 11 in Figure 3.6.
Another viewpoint considered in [Fe09] and schematically shown in Figure 3.6 (case 12) is called *strip plot*. Same as histogram and frequency polygon it is used to display a distribution of values with a distinction that strip plot visualizes thereby each individual value along the interval scale. However, by providing these details strip plots are missing a clear overview of distribution shape available in first two cases.
The further viewpoint regarded in [Fe09] is *box plot*, which boxes to a certain extent resemble bars of bar chart, but still are rather different from them as they do not share a common baseline as bars do. Another peculiarity of box plot is expressed by the horizontal line that divides each box in two parts (see case 13 in Figure 3.6). Also functionality of this graphical representation differs from the one of bar chart, as each box is supposed to display a distribution of a certain set of values. Thereby, box bottom represents the lowest value in the set, the top visualizes its highest value, box length reflects the full spread of values from the lowest one to the highest one, and the line in the middle stands for median or mean value. Another manifestation of box plot is shown by case 14 of Figure 3.6 and gives five-values distribution summary, as already discussed in Section 3.1. Box plots are not very appropriate for representation of a single distribution, but are very effective in examination and comparison of several distributions.
The next viewpoint considered in [Fe09] and schematically represented in Figure 3.6 (case 15), namely *radar graph*, can be used for time-series analysis. The circular shape of radar graph is then used to represent the cyclical nature of time, e.g. its axes display hours of a day, whereas different lines of radar graph display magnitudes of a certain value measured during different days. Data visualized by radar graph may be also represented using by means of line graph.
Viewpoints *stem-and-leaf plot*, *chernoff faces*, *data map* and *table*, schematically represented in Figure 3.6 by cases 16, 17, 18 and 19, respectively, are considered in detail in Section 3.1.
Glyphs, similar to chernoff faces, are used for multivariate data visualization, i.e. they convey multiple data values. Glyphs are composed from several graphical elements, each of which represents the value of a certain variable. Glyphs exemplified by Few are illustrated schematically in Figure 3.6 (case 20). In this example glyphs take a form of people in order to encode multiple variables by using such visual variables as color, shape of head, thickness of the torso, position of the arms, and position of the legs.
The next viewpoint considered in [Fe09] is *node-link diagram* (or *tree diagram*), which is used to display hierarchically structured data. For example, node-link diagram may be applied in case it is necessary to represent sale volumes per region along given geographical hierarchy, according to which continents are placed on the highest hierarchical level followed by countries, provinces and cities. This viewpoint is schematically shown in Figure 3.6 (case 21).
Another viewpoint supposed to visualize hierarchical structures is *treemap*. Treemap was invented by Ben Schneiderman and is supposed to display hierarchically structured data using containment to represent hierarchical relationship, i.e. placing child objects inside of parent object. Additionally, treemaps are able to display two quantitative variables by applying color and size coding to rectangles. Although treemaps do not support precise quantitative comparisons, which can not be done based on relative sizes of rectangles and their color hues, these graphs are efficient when it is necessary to view simultaneously two

quantitative variables for a large number of hierarchically structured entities. An example of treemap, shown in Figure 3.7, displays companies per region, sub-region or country, which are depicted as rectangles within larger rectangles. Firm value in US $ is represented by size of inner rectangle, whereas another indicator for measuring the value of a company - EV/EBITDA (Enterprise Value, divided by EBITDA, i.e. Earnings Before Interest, Taxes, Depreciation, and Amortization) - is visualized by rectangle's color. Inner rectangles are sorted according to their size, i.e. the companies with the greatest value are placed at the very top of outer rectangles.
One more viewpoint that uses variations in color to encode quantitative values is *heatmap* (see case 23 in Figure 3.6). Heatmap may encode values on an actual geographical map representing, for example, variations in rainfall for different regions, but also be visualized by a matrix of columns and rows with values displayed by color and not by digits or text.
The last viewpoint, considered in this section, is called *horizon graph* and supposed to support analysis and comparison of a large number of time-series data sets. Horizon graph uses color coding to highlight areas of concern that should be managed (see case 24 in Figure 3.6).

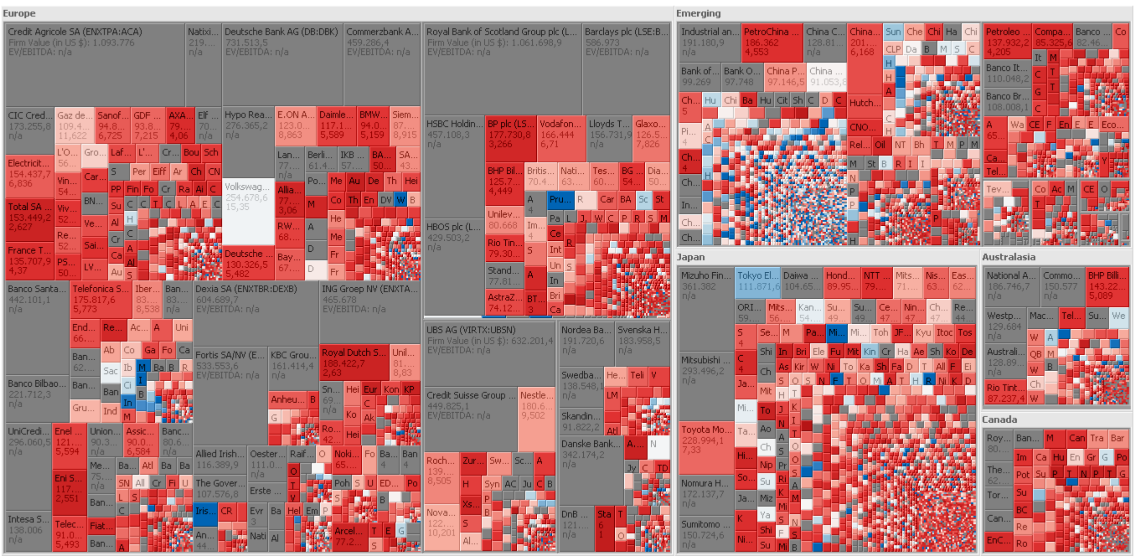


Figure 3.7.: Example of treemap (created using Panopticon explorer. URL: http://www.panopticon.com/demo_gallery/d_risk_perform_attrib_ex_demo.htm)

3.3. Data mining

In this section data visualization is considered in the context of data mining based on [MJ09]. According to Myatt and Johnson, data graphics plays an important role for understanding the context and detail of large data sets by bringing into play visual thinking. Within a data mining project data visualization may be used by data analysts while examining, scrutinizing, and validating analysis results, or decision makers by exploring

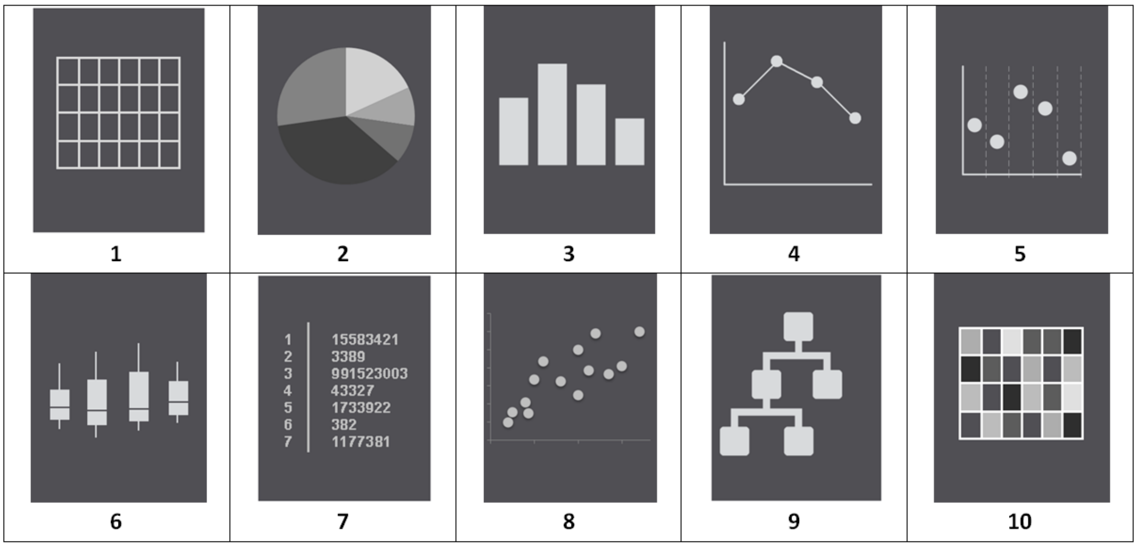


Figure 3.8.: Viewpoints from data mining block

the findings before to take any decision. The authors provide general principles for data graphics design, review and explain different types of tables, and consider a number of specific viewpoints classified according to a number of variables they display in univariate (for one variable), bivariate (for two variables) and multivariate (for many variables). Viewpoints, considered in [MJ09], are displayed schematically in Figure 3.8.
Although [MJ09] does not regard tables as graphics, same as Tufte it acknowledges that tables are effective in many situations (see Section 3.1), and reviews different table types, namely simple tables, summary tables, two-way contingency tables and supertables. Under *simple table* authors understand a usual table that contains values organized into rows and columns, and recommend using simple tables in case simple data to be represented. *Summary tables* visualize summary statistics for observations grouped by a single variable. The first column of summary table contains grouping variables, the second column displays a number of observations grouped by the corresponding grouping variable, and each remaining column represents values of a certain statistic applied to the corresponding group of observations grouped by the corresponding variable, e.g. mean, median, sum, minimum, maximum, etc. *Two-way contingency table* represents how frequencies of observations are distributed among categories or ranges of values in two variables. Header row and header column of two-way contingency table represent categories or ranges of values of two different variables, respectively, and its cells contain numbers of observations with the corresponding values of the first and the second variables. Finally, *supertables*, owing their name to Edward Tufte, are used to summarize and at the same time show in detail information, and are "as engaging as a well-written news article" [MJ09]. The example of supertable displayed in Figure 3.9 shows a profile of voters in presidential elections of 1976 and 1980.
Same as Tufte, [MJ09] asserts that *pie charts* (see case 2 in Figure 3.8) have better to be avoided, as they "force the viewer to find unaligned categorical labels inside or outside of irregularly spaced sectors within a circle, or decode colors of sectors using a legend",

How different groups voted for president	Carter	Reagan	Anderson	Carter–Ford in 1976
Democrats (47%)	66	26	6	77–22
Independents (23%)	30	54	12	43–54
Republicans (11%)	11	84	4	9–90
Liberals (17%)	57	27	11	70–26
Moderates (46%)	42	48	8	51–48
Conservatives (28%)	23	71	4	29–70
Family income				
Less than $10,000 (13%)	50	41	6	58–40
$10,000–$14,999 (14%)	47	42	8	55–43
$15,000–$24,999 (30%)	38	53	7	48–50
$25,000–$50,000 (32%)	32	58	8	36–62
Over $50,000 (5%)	25	65	8	—
Professional or manager (40%)	33	56	9	41–57
Clerical, sales or other white–collar (11%)	42	48	8	46–53
Blue–collar worker (17%)	48	47	5	57–41
Agriculture (3%)	29	66	3	—
Looking for work (3%)	55	35	7	65–34

Figure 3.9.: Portion of supertable showing voter profiles for the 1976 and 1980 U.S. elections. Source: [MJ09]

communicating in this way limited amounts of information in a rather slow way.
As already mentioned above, remaining visual representations are categorized in three groups. Within the first group, called *univariate data visualization*, [MJ09] review bar charts, frequency histograms, frequency polygrams, box plots, dot plots, stem-and-leaf plots, quantile plots and quantile-quantile plots. *Bar chart*, schematically shown by case 3 in Figure 3.8, is understood by the authors as a graphical way to quickly compare values of a set of instances by representing each value as a rectangle of proportional length with a fixed width. *Frequency histograms*, visualized in the same way as bar charts, have another interpretation and display a number of observations falling in different classes, in which all data values are grouped. The height of the bar corresponds to the size of each class, i.e. represents the number of observations in the respective class. A *frequency polygram* has an identical meaning to frequency histogram, but instead of bars it is represented by dots connected with one another by line segments (see case 4 in Figure 3.8). *Dot plots* have the same task of displaying measurements of labeled quantitative variables for small data sets as bar charts or pie charts, but perform it more effectively by representing values with dots, which position is defined by the values. Dot plots may also be used to compare two

sets of values by using different plotting symbols for the values from each set. *Box plots* (case 6 in Figure 3.8) and *stem-and-leaf plots* (case 7 in Figure 3.8) are used by Myatt and Johnson in the same meaning as by Tufte and are explained in Section 3.1. *Quantile plots* are supposed to represent a distribution of a variable by means of scatter plot, shown in 3.8 (case 8). Scatter plot is also used by *quantile-quantile plots*, which task is to compare distributions of one or two different variables.
As for *scatterplot*, it is considered within the second group, called *bivariate data visualization*, as a means for comparing one variable with another one and a graphical way to identify correlations between values of two continuous variables.
The third group, called *multivariate data visualization*, discusses how several variables may be visualized simultaneously in an understandable manner, so that the graphic that represents this data is not overlaid by the complexity of many dimensions. Additional dimensions may be added, for example, by using color coding or texture coding for graphical representations considered above. Another way to increase dimensionality of a certain graphic is to organize aforementioned viewpoints in a matrix of plots.
Furthermore, [MJ09] considers graphical displays for hierarchies (dendograms), observa-

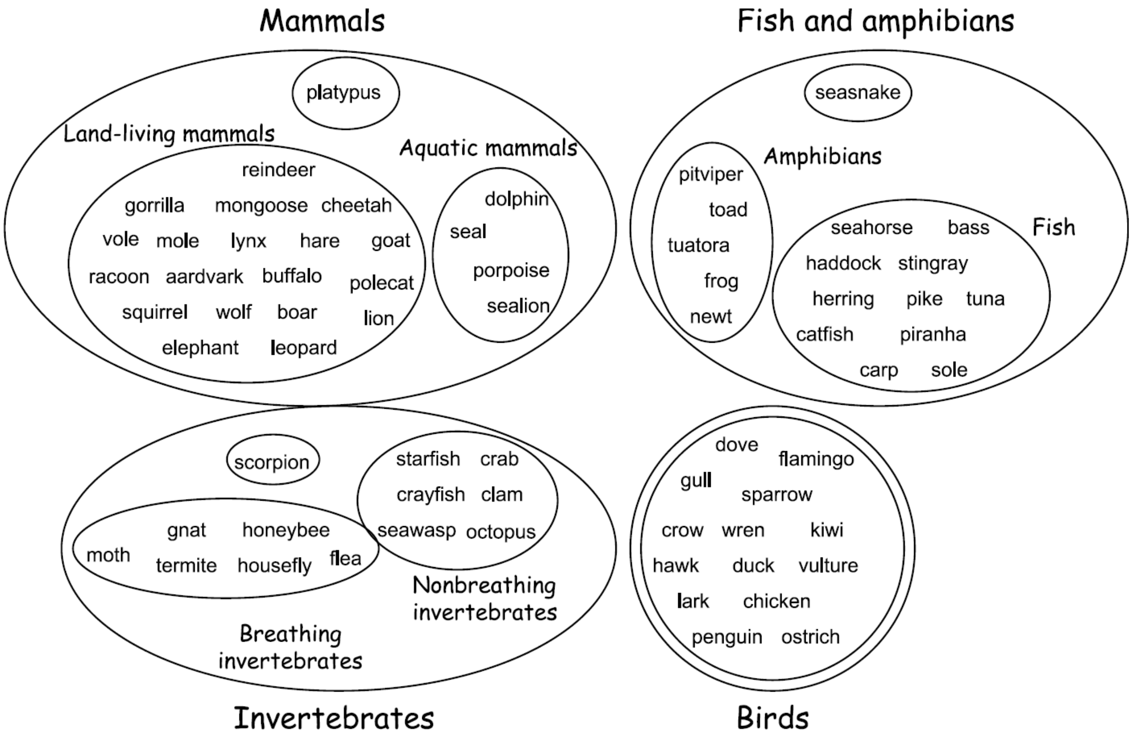


Figure 3.10.: Clustering exemplified on a data set of animals. Source: [MJ09]

tion groupings (decision trees) and complex, high density data (image maps). Under a *dendogram* a graphical representation is understood, which displays a hierarchy in the form of a tree (see case 9 in Figure 3.8). *Decision tree*, which may be also schematically represented as case 9 of Figure 3.8, visualizes a series of decision alternatives and conditions used to split a group of certain observations into smaller groups corresponding to certain criteria. *Image map* (or else *heat map*) represents a table of colored cells, where the values taken by a

variable are represented with colors (see case 10 in Figure 3.8). A special case of the heat map, *cluster image map*, combines in itself qualities of heat map and dendogram.
Myatt and Johnson are also concerned with breaking down data into smaller groups of observations that share something in common by means of *clustering*, so that observations within a particular group have more in common with one another than with representatives of any other group. Clustering is illustrated on a data set of animals shown in Figure 3.10.

3.4. Urban planning

The last section of this chapter considers viewpoints in the area of urban planning on an example of [Be08], which goal was to research application of information design for visualizing urban structures, flows and connections. As a result of this work, a taxonomy of design patterns was presented. All patterns are separated into three categories, namely display, behavior and interaction patterns. *Display patterns* describe how information is represented in the context of its usage purposes, i.e. stand for viewpoints. *Behavior patterns* reflect dynamic characteristics of interactive information visualization, e.g. reveals functionalities provided to the user relating to a certain infographics. *Interaction patterns* represent interface elements, which allow the user to interact with a certain information visualization. Within the context of this thesis display patterns are of particular interest. A rough overview of display design patterns, considered in [Be08], is provided in Figure 3.11.
Display patterns are categorized in their turn in further eight groups: correlations, contin-

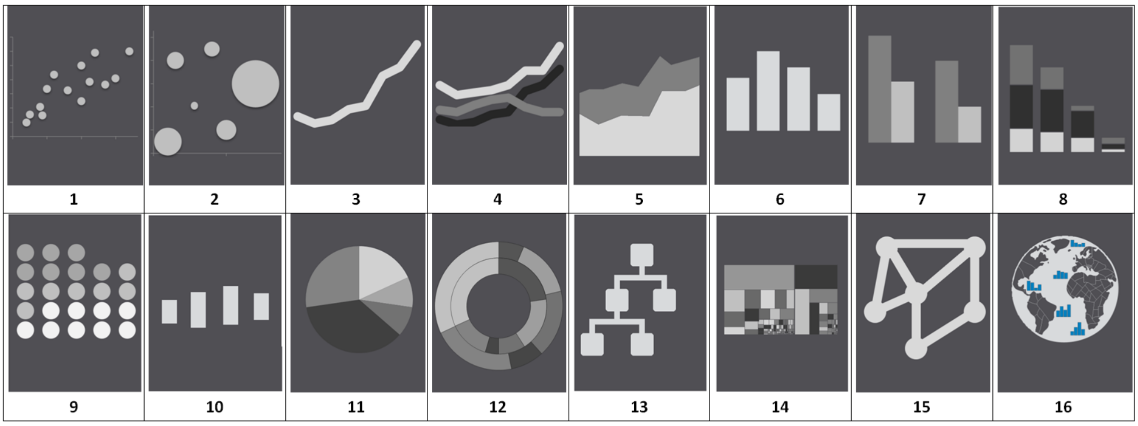


Figure 3.11.: Viewpoints used in the area of urban planning

uous quantities, discrete quantities, proportions, flows, hierarchies, networks and space.
The first group of display design patterns, *correlations*, is used, if correlations within data to be identified, and contains two representatives, namely scatter plot and bubble chart. *Scatter plot* is schematically shown by case 1 in Figure 3.11 and used in the meaning described in previous sections. *Bubble chart* being similar, to a certain extent, to scatter plot by using Cartesian coordinate system and providing a graphical way to identify correlations between two quantitative attributes, possesses two distinguishing peculiarities. Firstly, each

value represented by bubble chart has a unique label and is not anonymous as by scatter plot. Secondly, bubble chart allows to display more than two characteristics of items (quantitative or qualitative) by encoding additional attributes with color and size of bubbles, used instead of dots (see case 2 in Figure 3.11).
The second group, *continuous quantities*, provides means to display continuous datasets and contains four members: simple line chart, multiple line chart, stacked area chart and sparkline. *Simple line chart* (case 3 in Figure 3.11) as well as *multiple line chart* (case 4 in Figure 3.11) are already considered in previous sections. *Area chart* is based on a line chart and also displays a quantitative value over a continuous interval with a specific characteristic that the area below the line is filled with a color or texture. In case multiple data sets to be visualized, each new line is added on the top of the previous one, forming altogether a *stacked area chart* (see case 5 in Figure 3.11). The last representative of this group is *sparkline*, which is a minituarized form of a time-series chart. Sparkline is not used on a stand-alone basis, but rather as a redundant feature embedded in textual descriptions or tables.
The third group, *discrete quantities*, facilitates visualization of sets of discrete quantitative

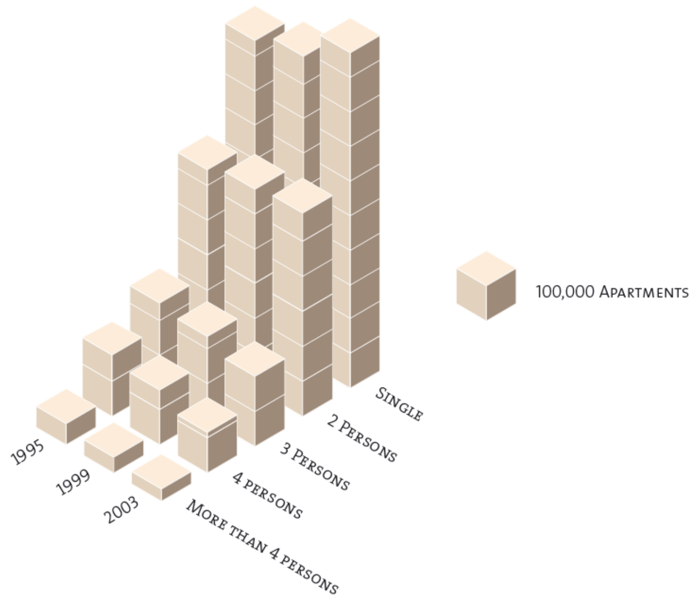


Figure 3.12.: An example of isometric bar chart showing a number of apartments in Berlin by number of occupants. Source: [Be08]

data and is represented by simple bar chart, multiple bar chart, stacked bar chart, dot matrix, isometric bar chart and span chart. *Bar chart* (see case 6 in Figure 3.11) allows to compare absolute magnitudes of nominal data items of a certain data set with one another, whereas *multiple bar chart* (see case 7 in Figure 3.11) is used to display and compare several sets of quantitative data. *Stacked bar chart* represents with its bars whole sets of quantitative data. The bars are, however, splitted into segments, each of which displays a single item of the corresponding set (see case 8 in Figure 3.11). *Dot matrix* (see case 9 in Figure 3.11) is a one-dimensional representation for discrete quantitative data, in which each value is represented by proportional quantity of dots that are filled with different colors to differ between the magnitudes of different entities. *Isometric bar chart* use three-dimensional

Cartesian coordinate system to display several sets of quantitative data, being in this way an alternative for multiple bar chart design pattern. An example of isometric bar chart is shown in Figure 3.12. And, finally, *span chart* allows direct comparison between data sets and is a special case of box plot, considered in detail in Section 3.1. Each rectangle of span chart displays only two statistics for each data set, namely its minimum and maximum (see case 10 in Figure 3.11).

The fourth group, *proportions*, displays share sizes of data set members within the total amount and is represented by two design patterns: simple pie chart and ring chart. *Simple pie chart*, schematically shown in Figure 3.11 (case 11), represents with its sectors a distribution of a number of values from a certain dataset. The full circle displays a value of a certain characteristic of the whole dataset, equal to 100%, whereas sectors of simple pie chart visualize magnitudes of the same characteristic for its subsets. *Ring chart* (see case 12 in Figure 3.11) may be considered as an extension of simple pie chart. It provides means to represent and compare distributions of several values from multiple data sets of the same configuration, i.e. such data sets that share the same number and type of elements, or those with the data related to each other.

The fifth group, named *flows*, enables representation of sequential nature of a certain data

Figure 3.13.: An example of sankey diagram showing average household income and spendings per month. Source: [Be08]

structure and consists of two members: sankey diagram and thread arcs. *Sankey diagram* describes a certain system by means of representing its input and output flows, and is represented by a main flow, usually running from left to right, with incoming arrows, designating input components, and outcoming arrows, displaying outputs (see Figure 3.13). Width of each arrow is proportional to the amount of contribution of respective component. *Thread arcs* facilitate representation of data that consist of ordered elements, which are associated with one another. This visual representation was introduced in 2004 within the scope of the project on developing experimental prototype for email client called Remail, which organized all messages in sequential thread according to the time of their creation, and thereby grouped messages related to one another in respect of a common topic. Moreover, Behrens exemplifies application of thread arcs to the area of urban planning, using them to depict an extension of underground network in Berlin (see Figure 3.14).

The sixth group, considered in [Be08], is called *hierarchies* and is supposed to display hierarchical structures by revealing parental relationships between elements, on the one hand, and representing characteristics of individual elements, on the other hand. The group contains two design patterns, namely tree diagram and tree map, which are schematically shown in Figure 3.11 by cases 13 and 14, respectively. *Tree diagram* is useful when it is necessary to represent data that consist of hierarchically structured items. All elements stand in some kind of inheritance relationship to each other, i.e. either having only descendants (root element of a tree diagram), or having only ancestors (leaf elements of a tree diagram), or having both descendants and ancestors (remaining elements of tree diagram). Each tree diagram may have only one root element, and each item may have only one ancestor, except for a root element, which does not have any parent element. *Tree maps* is an alternative representation for hierarchically structured data, which additionally provides means to encode quantitative properties of hierarchically structured elements. This viewpoints is used by Behrens in the same meaning as in [Fe09] and is considered in detail in Section 3.2. The seventh group, considered in [Be08] and called *networks*, focuses on visualization of

Figure 3.14.: An example of thread arcs representing openings and reopenings of Berlin subway stations since the reunification by line. Source: [Be08]

interconnections between different elements and is represented by three design patterns: diagram map, pearl necklet, and relation circle. The first design pattern of this group, *diagram map* (schematically shown in Figure 3.11 as case 15), is used to represent a system of considerable number of interconnected elements. The second member of the group, design pattern called *pearl necklet*, allows to represent a number of elements that are sequentially interrelated to each other. Stretched layout of this design pattern does not take into account geographical location of items, but focuses on topological structure of data (see Figure 3.15). The last representative of the seventh group, called *relation circle*, represents a system of interconnected items in a radial form. In this representation nodes of network are arranged on a circle in equal distances. Relation circle may display two types of connections, namely between several nodes placed on the circle as well as between nodes on the circle and items within the area enclosed by the circle, and its main strength lies in its ability to display amount and distribution of connections between elements. According to [Be08], relation circles may be used in combination with thematic maps when a large number of items from a thematic map should be assigned to one or multiple descriptive elements (see Figure 3.16).
The last, eighth group, called *spatial configurations*, considers space-related representations and contains two members, namely topographic and thematic maps. According to Behrens, *topographic map* allows to ”describe the features of space or an area as accurate as possible, but without confusing the user by overloading the available display with too much information”. *Thematic maps*, schematically shown as case 16 in Figure 3.11, are use-

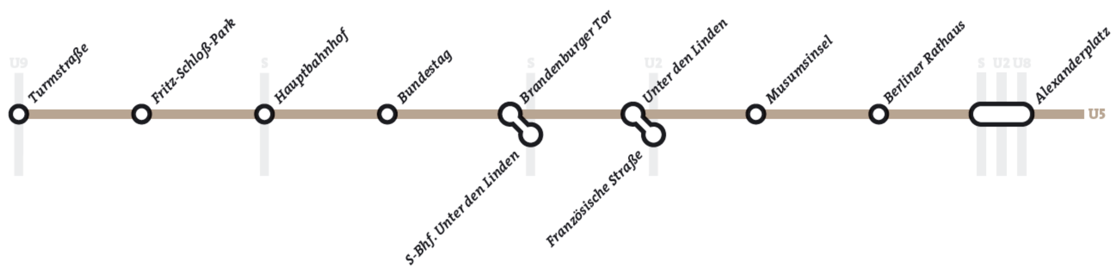


Figure 3.15.: An example of pearl necklet showing projected extension of the U5 subway with rail connections. Source: [Be08]

ful to display data that have a spatial reference and thereby represent some characteristics of these data, which may be numerical, categorical or qualitative. Data items are usually represented by a certain spatial entity, e.g. continent, region, country, state, etc., whereas attributes of these items are displayed by color coding of spatial regions or additional usage of another graphs.

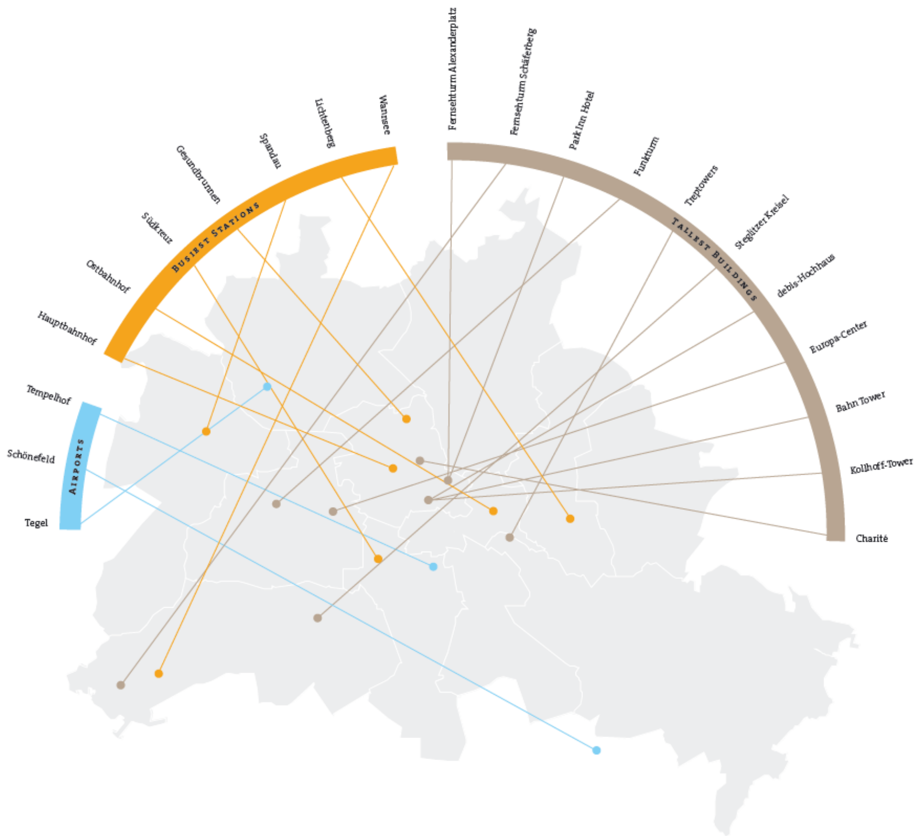


Figure 3.16.: An example of relation circle displaying an overview of superlative structures in Berlin. Source: [Be08]

4. Viewpoint Catalog from theory and practice

This chapter is concerned with Viewpoint Catalog construction used subsequently for BEAMS VBBs evaluation and extension. The Viewpoint Catalog collects and structures viewpoints from theory and practice in the area of EA management. Therefor, an explorative analysis of specialized literature as well as a case study by a real-life financial service provider have been performed.
As referred in Chapter 2, only findings in the field of EA management can be subsequently included into BEAMS collection according to the rule of three. Hence, only viewpoints originating from this field are relevant for subsequent BEAMS VBBs reorganization and extension process, and thus were included in the Viewpoint Catalog.

4.1. Viewpoints from theory

At the first step of Viewpoint Catalog construction, literature sources in the area of EA management were considered. Altogether, six sources were thoroughly analyzed in order to identify viewpoints for the catalog.
Selection of sources can be explained by two arguments. First, EA management background of the sources is critical for the further BEAMS VBBs extension and reorganization process, and second, all regarded sources focus on visualization aspect for EA management, providing in this way a comprehensive overview of viewpoints used in the field. Thus, [Se05] analyzes and evaluates various visualization capabilities for IT landscapes, whereas [Wi07] develops graphical modeling language for application landscapes as well as provides models and methods for their systematical visualization. [La08] and [Be04] are concerned with visualizations of metrics and key performance indicators relating to application landscapes. Finally, [Bu08] represents the collection of best-practice visualizations used in organizations, whereas [Ma08] researches visualization facilities of modern EA management tools.

4.2. Viewpoints from practice

In the second step viewpoints from practice were collected in the course of case study, which was conducted in cooperation with IT-subsidiary of a global enterprise active in financial services sector and affiliated in Munich, Germany. The participating organization operates in 22 countries of Western and Eastern Europe and is represented on approximately 50 markets all over the world. With more than 160.000 employees and almost

10.000 branches it belongs to one of the largest players on the Central and Eastern Europe banking market. Most of present activities of IT-subsidiary are aimed at consolidation of IT-systems and technologies in consequence of recent mergers and acquisitions of the financial institution, as well as at innovation targets.
Viewpoints were gathered in this phase by means of personal interviews, which were conducted between April and July 2011 with various stakeholders, as well as in course of documentation analysis. Altogether viewpoints from such areas as Finance, Project controlling, Project management, Contract management, Infrastructure architecture and Performance management were collected to be included into the Viewpoint Catalog.

4.3. Decomposing viewpoints to reusable building blocks

In the course of Viewpoint Catalog construction phase it was noticed, that certain visual elements occur repeatedly in different viewpoints. The focus of this section lies on breaking down the viewpoints from Visualization Catalog into reusable, nonintersecting components, which may be regarded as a certain decomposition basis. Altogether 141 viewpoints - 79 from the theoretical part of the Viewpoint Catalog and 62 from its practical block - were decomposed during viewpoint decomposition phase. Subsequently, this section considers several principles used in the course of decomposition process and exemplifies it on a number of concrete examples. Complete Viewpoint Catalog can be found in Appendixes A and B, which represent viewpoints from its theoretical and practical parts, respectively, together with the corresponding modular building blocks, ascertained in the course of decomposition phase. Thereby, each viewpoint from the Viewpoint Catalog is represented as a sum of its elementary components.
Decomposition process was performed in accordance with layering principle, discussed in Chapter 2. Viewpoints, represented as a sum of all modular building blocks they consist of, were decomposed "from bottom", i.e. at first, building blocks used as a base map were ascertained, in case they were available by the viewpoint under consideration, then modular components of the first layer were detached, after that the ones of the second layer, etc. Every time all elementary blocks of a certain layer were identified and added to the sum, BEAMS VBB *Add on* was used to move to the next layer.
In order to make decomposition process more transparent and indicate all essential components, participating in a certain viewpoint, some building blocks, as for example BEAMS VBB *Color coding*, were explicitly added to decomposition sum, even if visual variables of "neighbour" blocks already contained the same information implicitly. In cases when VBB *Color coding* added additional information to the viewpoint, it was used by decomposing in combination with VBB *Add on*. Otherwise, as for instance in cases when VBB *Color coding* was just used to increase the expressiveness of the viewpoint without adding any new sense to it, this building block was added to decomposition on a stand-alone basis. Another remark of this character concerns new viewpoint building block candidates, namely VBB *Graph* and VBB *Arrow* (see Chapter 5). As VBB *Graph* can be directed and undirected, it is not necessary to use VBB *Arrow* together with VBB *Graph* to represent its directed relations. Nevertheless, if a certain viewpoint represents relationships of different types with various arrow types, VBB *Arrow* is used explicitly within viewpoint decomposition

process together with VBB *Add on* to be able to differentiate between different types of relationships.
Viewpoint, shown in Figure 4.1 and originating from [Bu08] and [Ma08], represents a certain two-level cluster map that visualizes which business applications use which database systems together with the information, which of these database systems are running out of support. Business applications are depicted by inner elements of viewpoint cluster map, whereas its outer elements represent organizational units, in which the corresponding applications are hosted. Database systems are displayed with the corresponding icon, which color encodes its attribute ”is running of support”.
According to the principle of starting with decomposition from the bottom of the viewpoint, building block *Clustering*, that represents which business applications are located in which organizational units, is ascertained in the first place as the base map of this viewpoint. In order to move to the next layer, building block *Add on* is added to the building block *Clustering*, already included in decomposition sum. In the first layer database icons are used to represent which database systems are used by which business applications. Therefore, building block *Icon* in a form of database is added to the sum. As color coding of database systems provides additional information to the viewpoint (namely, whether certain system is running out of support or not), firstly, one more layer via building block *Add on* should be built up, and secondly, building block *Color coding* should be applied. Thus, the viewpoint illustrated in Figure 4.1 is decomposed into a sum of the following elementary components: Clustering, Add on, Database icon, Add on, Color coding.
Another example, considered in this chapter, is used by the real-life financial service provider

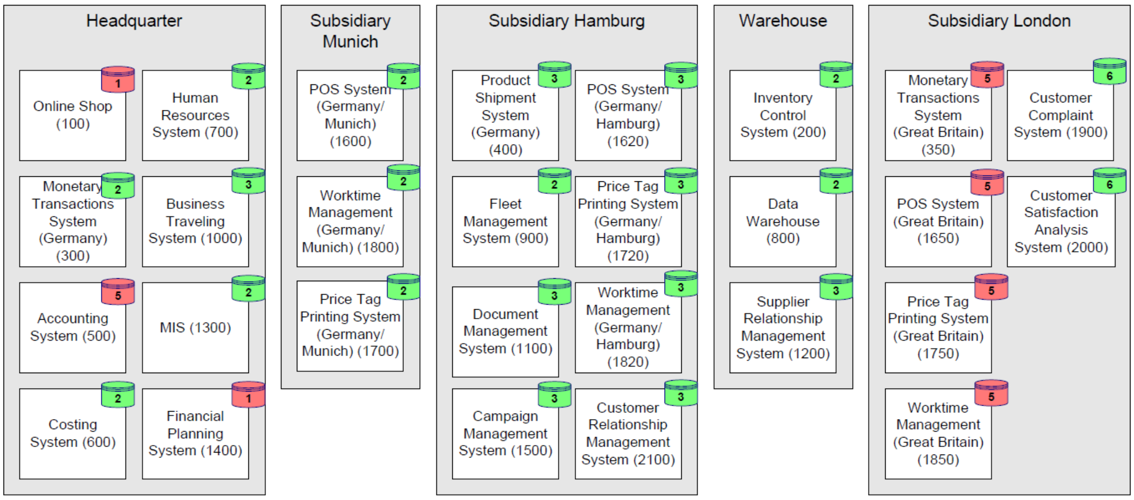


Figure 4.1.: Example of a viewpoint from the theoretical block of the Viewpoint Catalog. Source: [Bu08]

in the field of performance management and represents values of key performance indicators *KPI1*, *KPI2* and *KPI3* specified by a certain country, domain and timeframe (see Table 4.1). Values of *KPI1* and *KPI2* are shown as numbers, whereas *KPI3* values are depicted by symbols with three possible manifestations: red circle, yellow circle and green circle.
This viewpoint combines in itself several simpler patterns, i.e. modular building blocks.

		Date 1			Date 2		
		Country 1	Country 2	Country 3	Country 1	Country 2	Country 3
Domain 1	KPI1	22	22	22	22	22	22
	KPI2	100.615	35.017	23.712	2.585	2.975	5.515
	KPI3	🟢	🟢	🟢	🟢	🟢	🟡
Domain 2	KPI1	25	25	24	25	25	25
	KPI2	98.087	55.714	28.005	15.503	1.837	7.438
	KPI3	🔴	🟡	🟢	🟢	🟢	🟢
Domain 3	KPI1	738	114	256	1.110	102	876
	KPI2	3.254	1.303	1.147	16	2	1.110
	KPI3	🟢	🔴	🟢	🟢	🟡	🟢
Domain 3	KPI1	965	542	380	247	1.089	482
	KPI2	12.473	5.004	2.684	5.345	903	1.845
	KPI3	🔴	🔴	🟢	🟢	🟡	🟢

Table 4.1.: Example of a viewpoint used by the real-life financial service provider (anonymized)

Fundamental component, this viewpoint is based on, is a table, or else a matrix, which visualizes attribute values of different instances of a certain type, i.e. values of key performance indicators splitted by different countries. However, this multidimensional table contains another two types of instances. Instances of type "Timeframe" *Date1* and *Date2* serve as a grouping factor for countries, e.g. all instances of type "Country" are, firstly, grouped around *Date1*, and then *Date2*, recurring in this way several time in the table. Instances of type "Domain" take on the same role in respect of KPIs. Thus, the second building block, which represents a grouping of instances of one type by instances of another type, is identified. Further building blocks stand for colored circles, which visualize values of some key metrics. These symbols can be regarded as combination of two elementary components: shape and color.
Starting decomposition from the lowest layer of the viewpoint, building block *Matrix* is added as the first component to the sum of decomposition building blocks. However, building block *Matrix* is used in this case in combination with building block *Grouping* to form the table of the structure shown in Table 4.1. Thus, building block *Grouping* is added to the the building block *Matrix* in the next step. Elements of the table are displayed by numbers and colored symbols. The latter ones represent combination of building blocks *Ellipse* and *Color coding*. On the contrary to the previous example, in which building block *Add on* was used within decomposition process in order, first, to place database icons on the base map and, second, to lay on the color coding on icon symbols, *Add on* building block is not used for decomposition of this viewpoint at all. Ellipses, representing the values of *KPI3*, illustrate matrix elements, and thus, do not add additional layer to the base map, formed by composition of *Matrix* and *Grouping* building blocks. Color coding of ellipses in this case also takes place without adding on new layer, because ellipses do not represent anything on their own and make sense only in combination with color coding,

opposite to database icons from previous example.
In this way, the viewpoint exemplified in Table 4.1 can be regarded as a sum of the following modular building blocks: Matrix, Grouping, Ellipse and Color coding.
Results of viewpoint decomposition for examples, considered in this chapter, are summarized in Table 4.2, which demonstrates similarities and distinctions of viewpoints on the basis of building blocks and layers used for their decomposition. The table is organized according to the layering concept, and shows which building blocks of which type are used in which layer.

	Base map	Layer1	Layer2
Example1	clustering	database icons	color coding
Example2	matrix, ellipses, color coding		

Table 4.2.: Similarities and distinctions between exemplified viewpoints

5. Extending BEAMS with new viewpoint building blocks candidates

This chapter of the thesis is concerned with providing the evaluation and extension of BEAMS viewpoint building blocks based on the outcomes from Viewpoint Catalog decomposition performed in Chapter 4. The goal of this chapter is to check, whether any viewpoint from the Viewpoint Catalog can be reproduced by means of existing VBBs, and if not, which elementary viewpoint components are still missing in BEAMS. According to the rules for inclusion of new building blocks into BEAMS, considered in Subsection 2.3.2, such findings in the area of EA management may be further treated as BEAMS viewpoint building block candidates, and after detection of their three practical uses become a part of BEAMS collection.

Subsequent Section 5.1 analyzes BEAMS viewpoint building blocks utilization within the scope of decomposition process, and the gaps ascertained in the course of this analysis are taken as a basis for BEAMS VBBs extension and reorganization, suggested in Section 5.2 and Section 5.3.

5.1. BEAMS VBBs utilization analysis

After Chapter 4 demonstrated in which way different viewpoints were broken down into nonintersecting modular components, this section inspects the set of this components from the perspective of its correspondence to the BEAMS collection of viewpoint building blocks. Therefor, two sets of elements are compared, namely the set of elementary components ascertained in the course of Viewpoint Catalog decomposition and further called viewpoint catalog building blocks (VCBBs), and the collection of BEAMS VBBs.

In order to check whether any viewpoint from the Viewpoint Catalog can be reproduced by means of existing viewpoint building blocks, the inclusion of VCBBs within the set of BEAMS VBBs is analyzed in the first step, i.e. $VCBBs \subset BEAMS\ VBBs$. In the course of Viewpoint Catalog decomposition a number of viewpoint catalog building blocks were identified that are not a part of BEAMS VBBs collection. Such VCBB was considered in Section 4.3 on the example of modular building block *Grouping* that was used for decomposing the multidimensional table used by the real-life financial service provider. Hence, the conclusion can be made that not every viewpoint from the Viewpoint Catalog can be decomposed using solely viewpoint building blocks of BEAMS, i.e $VCBBs \notin BEAMS\ VBBs$.

In the second step, the inclusion of BEAMS VBBs within the set of VCBBs is analyzed, i.e. $VCBBs \supset BEAMS\ VBBs$. Table 5.1 demonstrates the results of this analysis shown for the viewpoints collected from EA management literature as well as for the outcomes from the case study by global financial service provider. Theoretical and practical blocks

BEAMS VBBs	[Be04]	[Bu08]	[La08]	[Ma08]	[Se05]	[Wi07]	Case study
Ordering		+	+	+	+	+	+
Binary matrix		+	+	+		+	+
Matrix (continuous)		+		+			
Ternary matrix		+	+	+	+	+	
Swimlane				+			+
Clustering	+	+	+	+	+	+	+
Recursive clustering						+	
Tree diagram				+			+
Bar chart		+		+			+
Pie chart	+			+		+	+
Rectangle		+	+	+		+	+
Ellipse		+	+	+			+
Adding on	+	+	+	+	+	+	+
Color coding	+	+	+	+	+	+	+
Size coding		+	+	+	+	+	+

Table 5.1.: BEAMS VBBs used for viewpoints decomposition of Viewpoint Catalog theoretical and practical parts

of the Viewpoint Catalog are depicted in the header row of the table, whereas its header column represents all viewpoint building blocks currently available in BEAMS collection. As shown in this table, in the theoretical block of the Viewpoint Catalog, represented by six sources, namely [Be04], [Bu08], [La08], [Ma08], [Se05], and [Wi07], BEAMS VBBs utilization is absolutely. Also by viewpoints used by the global financial service provider, absolute majority of VBBs found their application. These results give an evidence, that BEAMS viewpoint building blocks are highly applied in the field of EA management in both theory and practice. Another conclusion, which can be made from this comparative analysis of two sets, is that BEAMS VBBs collection is a subset of VCBBs set, as all BEAMS viewpoint building blocks were utilized during Viewpoint Catalog decomposition phase (see Figure 5.1).

After considering BEAMS viewpoint building blocks participation withing the VCBBs collection, remaining viewpoint catalog building blocks, different from BEAMS VBBs, are inspected, i.e. *VCBBs \BEAMS VBBs* (see Figure 5.1). The following sections document viewpoint catalog building blocks used for decomposition of viewpoints in the field of EA management, i.e. viewpoints originating from EA management literature and used by the real-life enterprise that are still not a part of BEAMS, and inscribe them as viewpoint building block candidates into existing BEAMS VBBs structure, providing solutions for its optimization and reorganization.

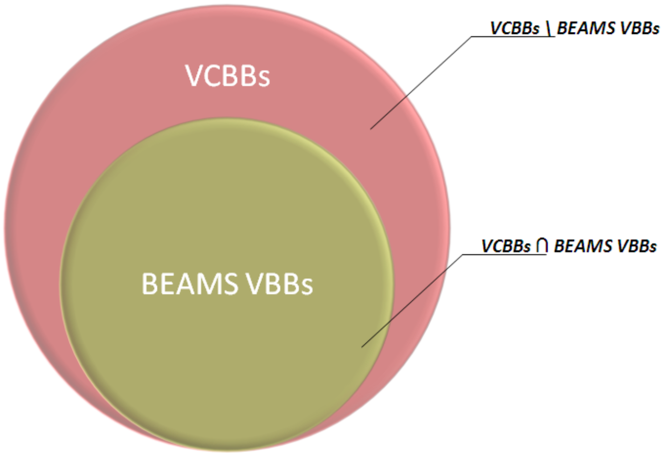


Figure 5.1.: Relationship between sets of BEAMS VBBs and VCBBs

5.2. Structural VBBs candidates

This subsection is concerned with BEAMS structural viewpoint building blocks and starts with introduction of 14 new structural VBBs candidates and 2 special cases of VBB Clustering (see Figure 5.2) sorted in alphabetical order.
Each of VBB candidates is subsequently described in detail based on layout used in [Se11] for BEAMS viewpoint building blocks description and explicated in Subsection 2.3.2. Layout sections "Name", "Explanation", "Examples" and "Type" are incorporated into viewpoint building block candidates layout one-to-one, with the only one exception for VBB Matrix, for which section "Example" is specified for each of its special cases, but not within building block candidate description itself. Section "Usage" for structural VBBs description is used in the same meaning as in [Se11], although for representation of attribute scale types the terminology in the meaning of [Fe09] is used. In doing so, it is differentiated between nominal, ordinal and interval scales. Additionally, such issues as "Formatting" and "Connections to other VBBs" are described for each structural candidate. Section "Formatting" lists format settings for VBB candidate under consideration that do not add any additional information to the base map layer playing solely the role of decorator, whereas "Connection to other VBBs" section shows which building blocks may be used in combination with considered candidate to encode additional information on additional layers or extend considered builing block. On the contrary, BEAMS VBBs layout sections "Tags", "Building block" and "M2 concept" were not included in VBBs candidates descriptions.
Finally, this subsection makes optimization recommendations for existing structural VBBs and gives an overview on extended BEAMS collection of structural viewpoint building blocks.

5.2.1. VBB Clustering: special cases

VBB Clustering is used to display a hierarchic relationship between instances of different types and is already described in [Se11]. This subsection communicates additional two layout sections for VBB Clustering, namely "Formatting" and "Connections to other VBBs"

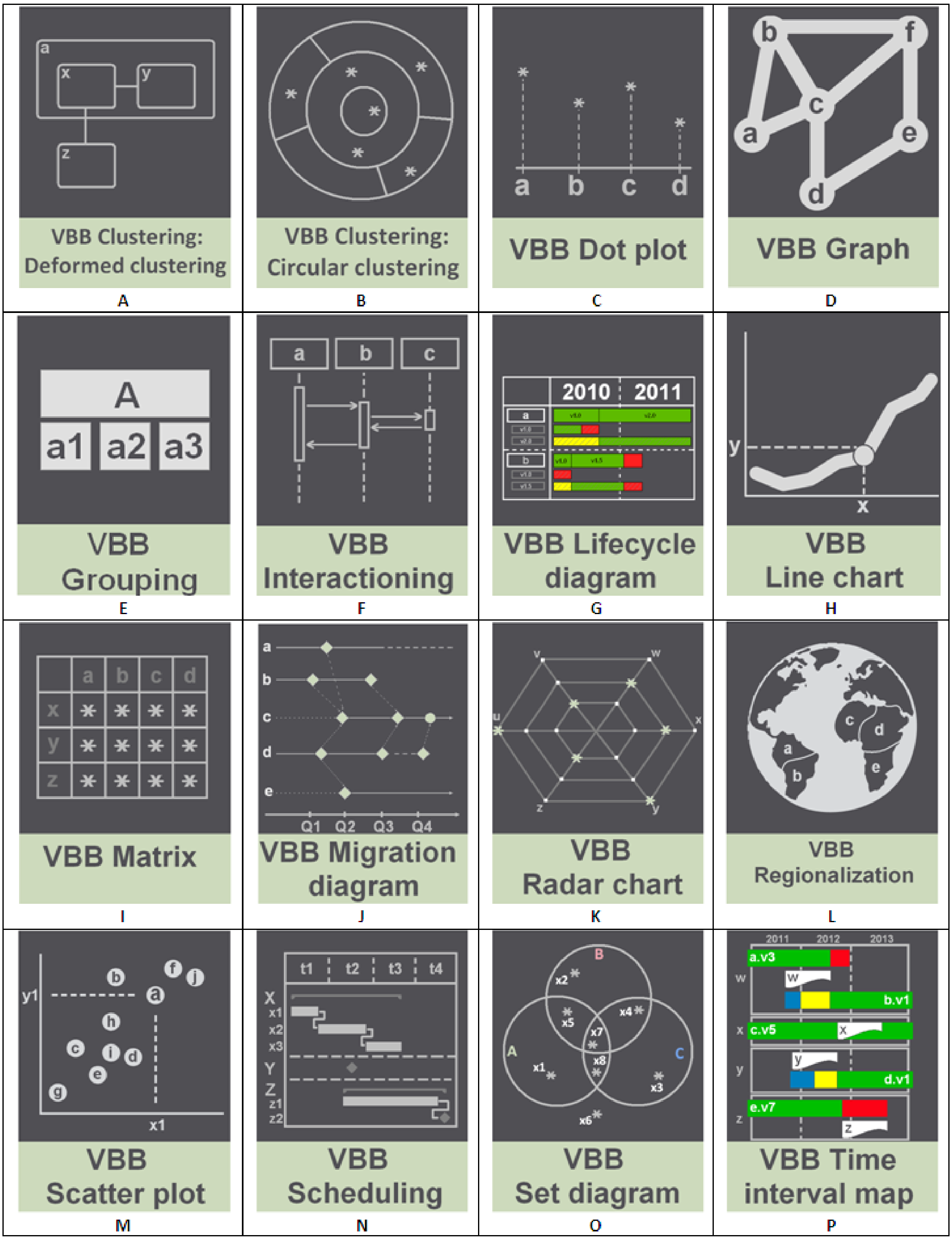


Figure 5.2.: Structural VBBs candidates

and then represents two special cases of VBB Clustering, identified among viewpoint catalog building blocks during the decomposition phase.
Formatting: Following format settings can be applied to VBB *Clustering* and all its special cases:

- set background color for clusters
- set shape of clusters
- set frame color of clusters
- set frame style of clusters

Connections to other VBBs: VBB *Clustering* may be used in combination with VBB Smiley, VBB Traffic light, VBB Battery, VBB Harvey balls, VBB Speedometer, VBB Trend arrow, VBB Progress bar, VBB Bar chart, VBB Pie chart, VBB Color coding and VBB Size coding to depict additional attributes of its elements. VBB Icon and VBB Shape may be used to differentiate between instances of different types. Furthermore, VBB *Clustering* may be used together with VBB Graph, if relationships between its elements to be shown.

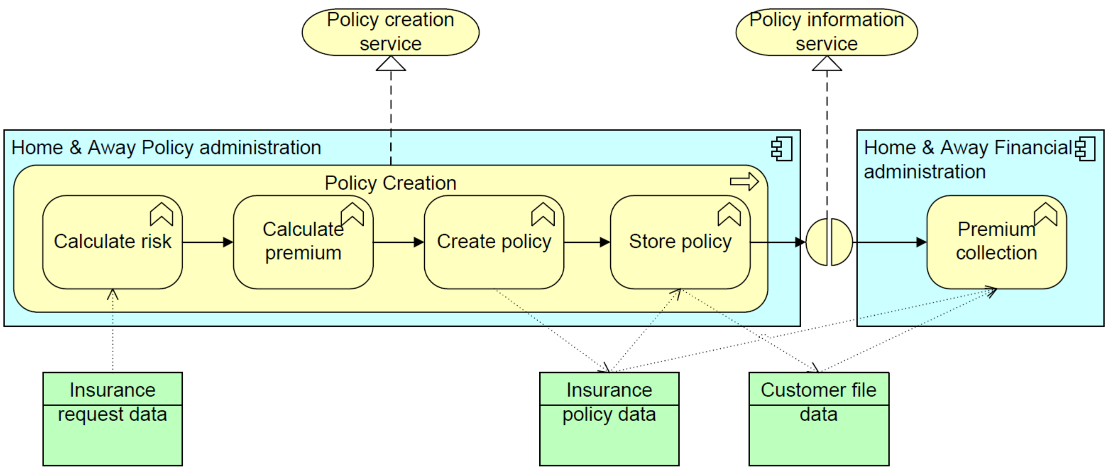


Figure 5.3.: Example of VBB Clustering: Deformed clustering. Source: [Bu08]

VBB Clustering: Deformed clustering

Deformed clustering represents a special case of VBB Clustering, as some inner elements of cluster map are not assigned to any outer ones.
Explanation: An shown in Figure 5.2 (case A) symbols labeled x, y and z represent the instances of type Z ($x{:}Z$, $y{:}Z$ *and* $z{:}Z$), whereas symbol a displays the instance of type A ($a{:}A$). While instances x and y of type Z are related to instance a of type A, instance $z{:}Z$ has no relation to any instance of type A.
Example: Shown in Figure 5.4 special case of VBB Clustering represents how business objects are used by business processes in business interactions. Here three-level hierarchy

of instances is displayed, which visualizes application components on the highest level of hierarchy (outer symbols of cluster map), business processes in between (symbols of the middle layer of cluster map) and business functions on the lowest hierarchical level (inner symbols). However, instances of type "Business object" (green, outside the cluster map) and "Business service" (yellow, outside the cluster map) are not assigned to any level of hierarchy and are out of it.
Abstract viewmodel: An abstract viewmodel for this special case of VBB Clustering is shown in Figure 5.4.

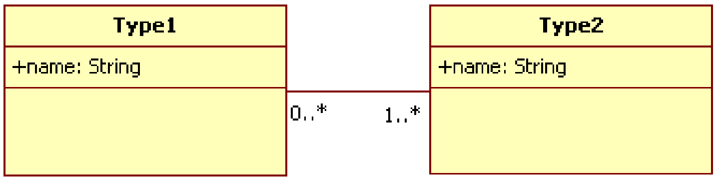


Figure 5.4.: Information model of VBB Clustering: Deformed clustering

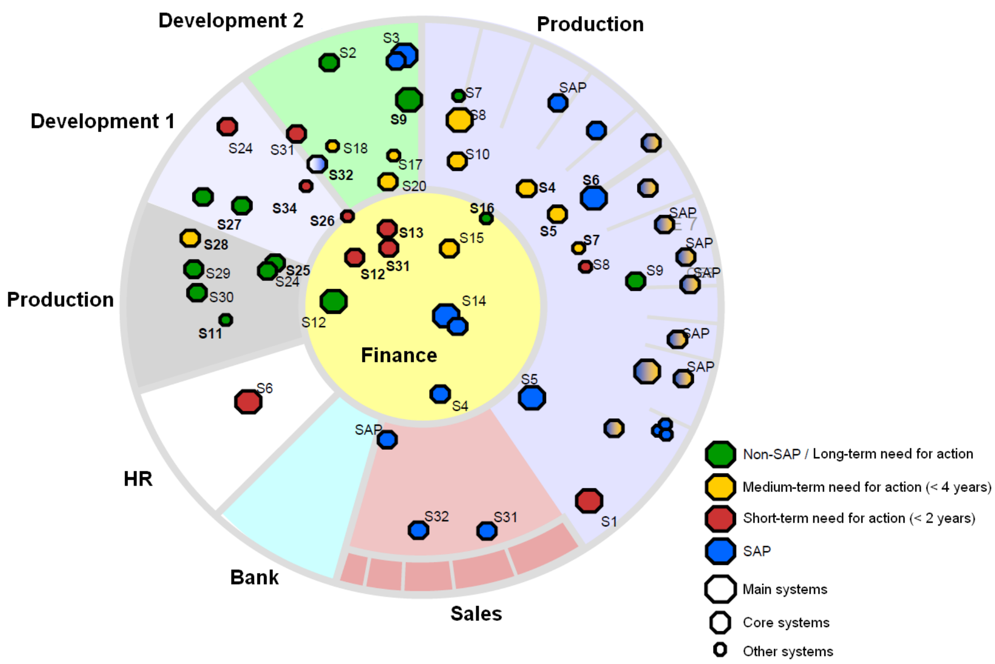


Figure 5.5.: Example of VBB Clustering: special case 2. Source: [Se05]

VBB Clustering: Circular clustering

Circular clustering displays a special case of VBB Clustering, as it is not represented in its standard form of contextless nested rectangles, but in a form that has a resemblance to the real-life structure of elements to be visualized.

Explanation: Circular representation of VBB Clustering, shown schematically in Figure 5.2 (case B), is suitable if it is necessary to show many-to-many relationships between inner and outer elements in an expressive and intuitive recognizable manner, thereby considering logical relatedness between the elements. This special case places side by side such outer components, which have the majority of common inner elements, they are related to. Inner components related to both neighbouring outer elements are placed then on their common border. Circular representation also takes into account the context of elements it visualizes. Thus, elements with the central role can be placed in the middle of the circle.
Example: Example of circular representation of VBB Clustering is shown in Figure 5.5, which considers cross-departamental use of same application systems. Thereby, departments with the highest cooperation by application systems usage placed next to one another. Moreover, organizational context is also incorporated in this visualization by locating finance department in the middle of the viewpoint emphasizing in this way its central role for organization. Color and size coding of inner elements of this viewpoint encode additional information about application systems, namely their importance (size) and urgency of need for action (color).
Abstract viewmodel: An abstract viewmodel for this special case of VBB Clustering may be also represented by Figure 5.4 with relationship multiplicity "many-to-many".

5.2.2. VBB Dot plot

Usage: You want to display the values taken by one or more ordinal or interval attributes. y-coordinate of symbols represents an attribute value.
Explanation: VBB *Dot plot* is schematically depicted in Figure 5.2 (case C). Dot plots can be used to visualize one of the following scenarios:

- values of a single attribute for instances of one particular type
- values of different attributes for a single instance of one particular type
- values of multiple attributes as taken by a collection of instances of one particular type

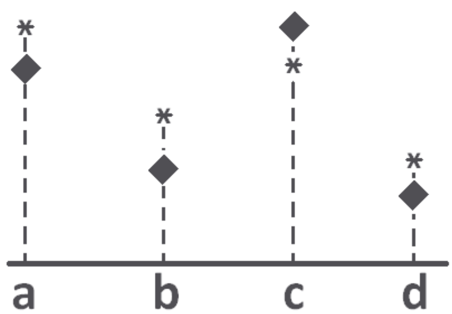


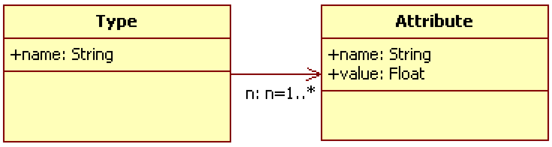
Figure 5.6.: Multiple dot plot

Example: The example in Figure 5.2 (case C) indicates four different values *a, b, c* and *d*. Here, two from three aforementioned scenarios are possible.
Scenario 1:
a, b, c and *d* represent different instances of one particular type *Z*: *a:Z, b:Z, c:Z, d:Z*. Type *Z* has one attribute *z* of interval scale, which takes values:
a.z = 5
b.z = 4
c.z = 4,5
d.z = 3
This scenario was detected by financial service provider. In the real-life example instances represents different organization domains, and dot plot symbols depicts target KPI values for each domain, respectively.
Scenario 2:
a, b, c and *d* represent different attributes of one particular instance *z* of type *Z*: *z:Z*. These attributes take values:
z.a = 5
z.b = 4
z.c = 4,5
z.d = 3
Scenario 3:
The example in Figure 5.6 shows the third scenario, in which two different values as taken by each of two attributes of interval scale are indicated.
Each group of symbols represents instances *a, b, c* or *d* of type *Z*: *a:Z, b:Z, c:Z, d:Z*. Type *Z* has two attributes of interval scale: *y* and *z*, taking values as follows:
a.y = 5
a.z = 4,4
b.y = 4
b.z = 2,8
c.y = 4,5
c.z = 5,1
d.y = 3
d.z = 2
Type: Structural VBB candidate
Formatting: Following format settings may be applied to VBB *Dot plot*:

- *y*-axis can be switched on and off
- gridlines can be switched on and off

Connections to other VBBs: VBB *Dot plot* may be used in combination with VBB Shape, VBB Icon and VBB Bar chart to depict multiple attributes of instances in *Scenario 3*. VBB Color coding may be used to depict additional attribute of a certain instance in *Scenario 1* and *Scenario 3* (e.g. attribute "Status" can be coded in the following way: green - in production; yellow - in development; red - to be retired)
Abstract viewmodel: Information models of VBB *Dot plot* for each Scenario are shown in Table 5.2.

Scenarios 1 and 3



Scenario 2

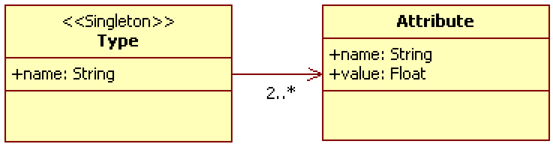


Table 5.2.: VBB Dot plot information models

5.2.3. VBB Graph

Usage: You want to visualize a set of instances of types and relations between them.
Explanation: VBB *Graph* is schematically depicted in Figure 5.2 (case D). Vertices of VBB *Graph* visualize instances of types, whereas its edges represent relationships between these instances. Graphs may be used to represent one of the following scenarios:

- instances to be visualized are of one certain type. In this case all instances are depicted by uniform representatives of VBB Shape, usually by circles - special cases of VBB Ellipse
- instances to be visualized are of different types. In this situation instances of different types are visualized by by different representatives of VBB Shape or VBB Icon
- relations between instances are symmetric, i.e. if *a* is related to *b* then *b* is related to *a*. Here graph is undirected, and relations between the instances are represented by VBB Line
- relations between instances are asymmetric, i.e. if *a* is related to *b* then *b* is not necessarily related to *a*. In this case graph is directed, and relations between instances are visualized by VBB Arrow

Example: Example shown in Figure 5.2 (case D) represents five instances of same type *A*: *a:A*, *b:A*, *c:A*, *d:A*, *e:A* and *f:A* with following symmetric relations between them:
(a,c), (c,a)
(a,b), (b,a)
(b,c), (c,b)
(c,d), (d,c)
(c,f), (f,c)
(b,f), (f,b)
(d,e), (e,d)
(f,e), (e,f)
Type: Structural VBB candidate
Formatting: Show/hide grid format may be applied to VBB *Graph*.
Connections to other VBBs: VBB *Graph* may be used in combination with such symbol VBBs as VBB Smiley, VBB Traffic light, VBB Battery, VBB Harvey balls, VBB Speedometer, VBB Trend arrow and VBB Progress bar to depict additional attributes of instances. For

decoration of vertices and edges of VBB *Graph* visual variables of VBB Shape, VBB Icon, VBB Line and VBB Arrow may be used.
Abstract viewmodel: Information model of VBB *Graph* is shown in Figure 5.7.

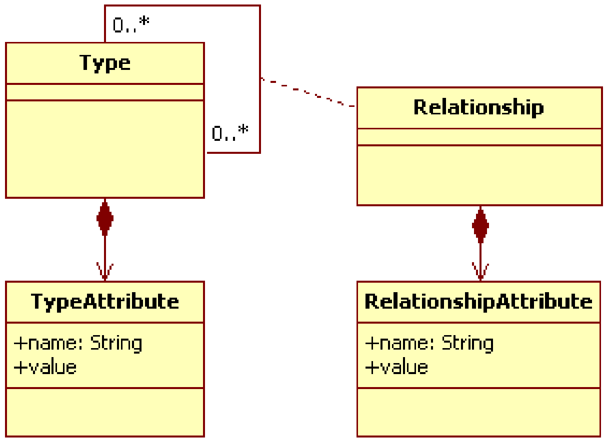


Figure 5.7.: VBB Graph information model

5.2.4. VBB Grouping

Usage: You want to show the grouping of instances of one type by instances of another type.
Explanation: VBB *Grouping* is schematically depicted in Figure 5.2 (case E). Two scenarios of its usage are thinkable:

- *Scenario 1:* VBB *Grouping* displays a hierarchical relationship between instances of one certain type
- *Scenario 2:* VBB *Grouping* displays a grouping relationship between instances of different types

Example:
Scenario 1: Symbols labeled *A*, *a1*, *a2* and *a3*, shown in Figure 5.2 (case E), represent instances of class "Process", and thus visualize a certain process *A*, which consists of smaller subprocesses *a1*, *a2*, and *a3*.
Scenario 2: Symbol labeled *A*, shown in Figure 5.2 (case E), represent a certain instance of class "Year", whereas symbols labeled *a1*, *a2* and *a3* visualize instances of type "Country". In this case VBB *Grouping* displays the grouping of instances of type "Country" by instances of type "Year" (see Table 4.1).
Type: Structural VBB candidate
Formatting: VBB *Grouping* may be established in two graphical layouts:

- a grouping instance above and grouped instances below
- a grouping instance below and grouped instances above

VBB *Grouping* may be also established in two graphical directions, namely horizontally and vertically.
Additionally, following format settings are applicable to VBB *Grouping*:

- set fill color of elements
- set shape of elements
- set frame color of elements
- set frame style of elements
- set text color of elements

Connections to other VBBs: VBB *Grouping* is not an independent VBB and is usually used complementary to VBB Matrix, VBB Ordering and VBB 2D-Swimlane.
Abstract viewmodel: Information models of VBB *Grouping* for both possible scenarios are shown in Table 5.3.

Scenario 1	Scenario 2

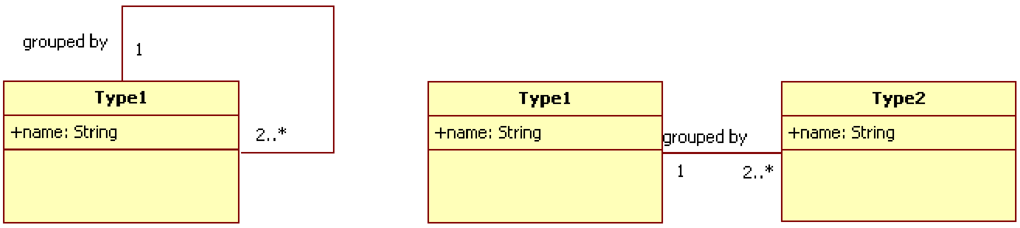


Table 5.3.: VBB Dot plot information models

5.2.5. VBB Interactioning

Usage: You want to show how instances operate with each other and in what order. This VBB is also known as UML sequence diagram.
Explanation: VBB *Interactioning* is schematically depicted in Figure 5.2 (case F). Parallel vertical lines of this VBB represent lifelines of different instances. Horizontal arrows between the lifelines of the instances represent exchange of messages between these instances in the order, in which they occur.
Example: Symbols labeled a, b, and c represent instances of the type Z (a:Z, b:Z and c:Z). Instance a sends a message to instance b and waits for reply from b. Instance b in its turn sends another message to instance c, and waits for reply from c. Instance c answers to b. Then, b answers to a.
Type: Structural VBB candidate
Formatting: Following format settings are applicable to VBB *Interactioning*:

- set fill color of elements

- set shape of elements
- set frame color of elements
- set frame style of elements
- set text color of elements

Connections to other VBBs: Differentiation between request and response messages as well as between synchronous and asynchronous requests is done by the usage of different arrows styles of VBB Arrow.

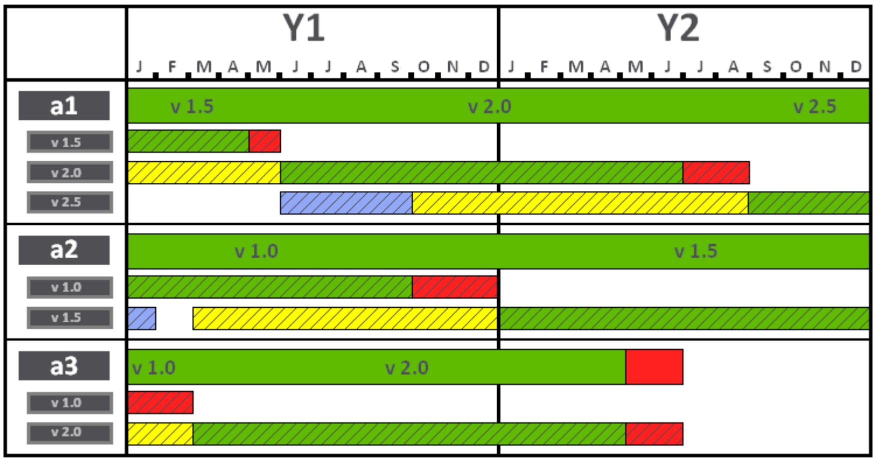


Figure 5.8.: Lifecycle diagram

5.2.6. VBB Lifecycle diagram

Usage: You want to visualize lifecycles of instances of a certain type, including lifecycles of respective instances of subtype.
Explanation: VBB *Lifecycle diagram* is schematically depicted in Figure 5.2 (case G). Header row of VBB *Lifecycle diagram* represents time scale, whereas its header column contains instances of a certain type and their constituents represented by instances of subtype. Bars placed in central part of this VBB represent different lifecycle phases of instances of the type and its subtype with different colors. Lifecycle phase of a certain instance of the type depends on lifecycle phases of the instances of its subtype.
Example: Shown in Figure 5.8 lifecycle diagram, considered in [Bu08], visualizes lifecycles of business applications and their versions. Header row represents time scale with years *Y1* and *Y2* splitted by months. Header column of this lifecycle diagram contains business applications *a1*, *a2* and *a3* together with their versions:
v 1.5, *v 2.0* and *v 2.5* for application *a1*
v 1.0 and *v 1.5* for application *a2*
v 1.0 and *v 2.0* for application *a3*

Each application as well as each application version can be in each moment of time in one of four lifecycle phases: planned, in development, in production and to be retired. Bars represent lifecycles of the corresponding instances: the ones without texture show application lifecycles, whereas the ones with texture visualize lifecycles of business application versions.
Business application lifecycle phases depend on lifecycle phases of their versions, e.g. a certain application stays in phase "In production" as long as one of its versions is in the same status. As soon as the latest version should be retired and no new version is planned, the status of the application also switches to "To be retired" (see *a3* in Figure 5.8).
Type: Structural VBB candidate
Formatting: Following decorators can be applied to VBB *Lifecycle diagram*:

- show/hide grid (show grid completely or partially in any combination: show only horizontal gridlines, only vertical gridlines, only gridlines for header row and column)
- place column headers above/beneath the lifecycle diagram
- place row headers on the left/ on the right side of the lifecycle diagram
- set cells text color
- set cells frame color
- set cells frame style

Connections to other VBBs: VBB *Lifecycle diagram* may be used in combination with such symbol VBBs as VBB Smiley, VBB Traffic light, VBB Battery, VBB Harvey balls, VBB Speedometer, VBB Trend arrow and VBB Progress bar to depict additional attributes of instances. VBB Color coding and texture coding are used to differentiate between different lifecycle phases. VBB *Lifecycle diagram* may be also used with structural VBB Grouping, e.g. to group months by years.
Abstract viewmodel: Information model of VBB *Lifecycle diagram* is shown in Figure 5.9.

5.2.7. VBB Line chart

Usage: You want to visualize a trend in ordinally or interval-scaled attribute values of a certain instance of a certain type over time.
Explanation: VBB *Line chart* is schematically depicted in Figure 5.2 (case H). x-axis depicts a time scale, and y-axis represents the scale of an attribute of a certain instance of a certain type. A line represents dependency of the attribute from time, i.e. each point of the line represents a certain value taken by the attribute in the corresponding moment of time. Line chart is formed by connecting the points, which represent individual measurements, with line segments.
VBB *Line chart* can be represented by only one or multiple lines. In case of multiple lines following two scenarios may be visualized:

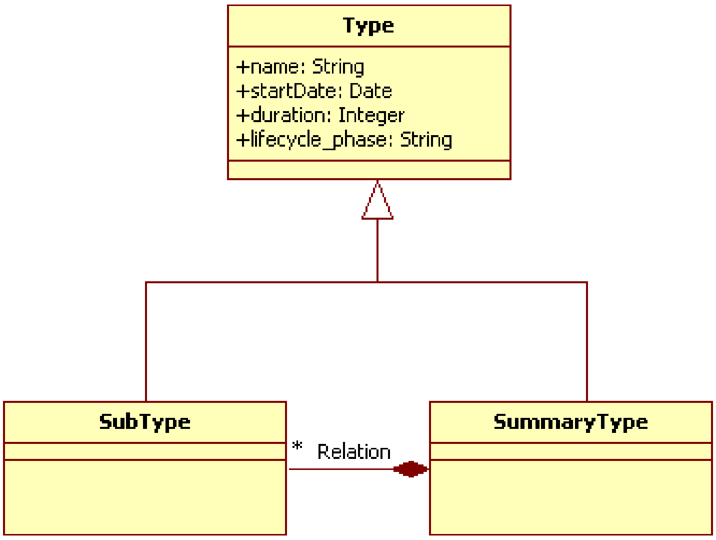


Figure 5.9.: VBB Lifecycle diagram information model

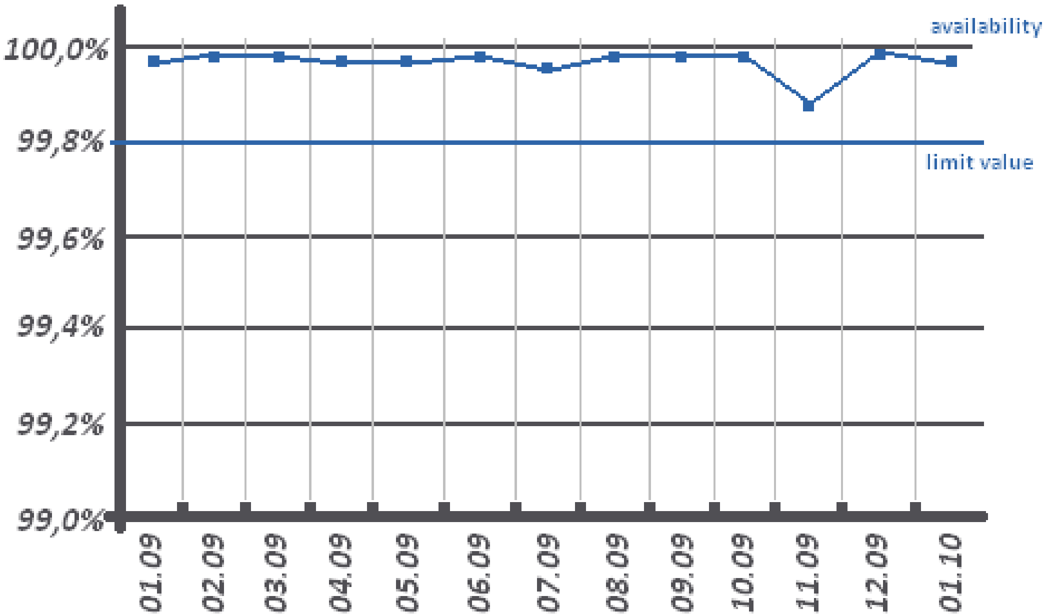


Figure 5.10.: Line chart of the real-life financial service provider (anonymized)

- *Scenario 1:* Different lines show changes in values of same attribute for different instances of a certain type. If u and v are some instances of type Z, which have an attribute a, then VBB *Line chart* may visualize changes of this attribute over time for both instances with two lines: $u.a$ and $v.a$.

- *Scenario 2:* Different lines show changes in values of different attributes of a certain instance of a certain type. In this case the attributes should be comparable, i.e. have the same scale. If z is an instance of type Z, which has two comparable identically scaled attributes a and b, then VBB *Line chart* may visualize changes of these attributes over time with two lines: $z.a$ and $z.b$

Example: Example of VBB *Line chart* shown in Figure 5.10, used in the area of contract management of the real-life financial service provider, represents two lines. The upper line depicts changes in values of availability of a certain infrastructural element. x-axis depicts the months, and y-axis shows the scale for availability. According to the figure, attribute ”Availability” is measured once a month, because there is only one measurement point per month visualized.
The lower line of this viewpoint represents the limit value for attribute ”Availability” equal to *98,0%*. It is shown as a horizontal line, because the limit value is equal in each month, and thus its y-coordinate is equal to *98,0%* for each value of time x.
In this example the second scenario is represented, as different lines depict different attributes of a certain instance of type ”Infrastructural element”, namely actual availability and limit availability.
Type: Structural VBB candidate
Formatting: Following format settings can be applied to VBB *Line chart*:

- Switch on/off the gridlines
- Switch on/off the values captions of the points
- Fade in/down the axes
- Fade in/down the axes scales
- Axes rotation
- One/multiple lines shown

Connections to other VBBs: VBB *Line chart* may be used in combination with VBB Shape and VBB Color coding to differentiate between attribute values for different instances (scenario 1) or values of different attributes (scenario 2).
Abstract viewmodel: Information models of VBB *Line chart* for each of aforementioned scenarios are shown in Figures 5.11 and 5.12.

Figure 5.11.: VBB Line chart information model for Scenario 1

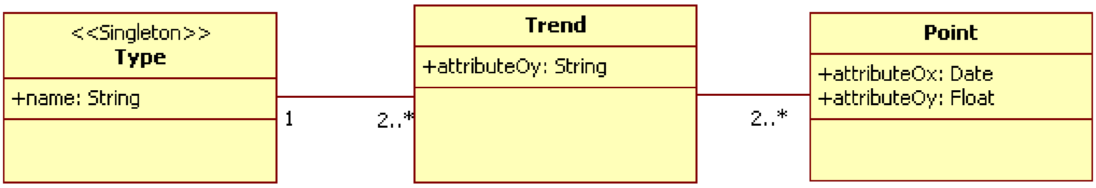


Figure 5.12.: VBB Line chart information model for Scenario 1

5.2.8. VBB Matrix

Usage: You want to visualize one of the following scenarios:

- which values various attributes of instances of particular type take
- how instances/attributes of two types relate
- how instances/attributes of three types relate

Explanation: *VBB Matrix* is schematically depicted in Figure 5.2 (case I). Columns *a*, *b*, *c*, *d* and rows *x*, *y*, *z* of the matrix can represent instances of particular type, different attribute labels or different values of particular attribute. Cells of matrix may contain:

- numerals
- binary values
- categorical data
- text
- instances of particular type
- symbol VBBs

Type: Structural VBB candidate
Formatting: Following format settings may be applied to VBB *Matrix* and all its special cases:

- show/hide grid (show grid completely or partially in any combination: show only horizontal gridlines, only vertical gridlines, only gridlines for column and row headers)
- add/remove axes
- swap column and row headers of the matrix
- place column headers above/beneath the matrix
- place row headers on the left/ on the right side of the matrix
- set cells background color

- set cells text color
- set cell frame color
- set cell frame style

Three special cases of *VBB Matrix* are considered in detail in subsequent subsections.

Department	KPI1	KPI2	KPI3	...
Dpt. 1	28,6	0,8	14,9	...
Dpt. 2	82,1	5,4	14,3	...
Dpt. 3	79,5	3,6	15,9	...
...	...	...	...	...

Table 5.4.: Unary matrix of the real-life financial service provider (anonymized) - example 1

VBB Unary matrix

Usage: You want to visualize attribute values grouped by different instances of a certain type or by different values of another attribute.
Explanation: Header column of VBB *Unary matrix* represents some values (either instances of particular type or attribute values of particular attribute), and its header row contains attribute labels. Cells of the matrix visualize values of the attributes specified in header row grouped by the values stated in header column. These values can be visualized by numerals, categorical data, text or symbol VBBs.
Example:
Example 1: As shown in Table 5.4 header column of this unary matrix, used in the area of finance by the real-life financial service provider, represents different departments of the organization (values of type ”Department”), its header row depicts different KPI names for finance area, and the numerals at the intersections of columns and rows display the values of the corresponding KPIs in the corresponding departments, i.e. values of the corresponding KPI grouped by the corresponding department.
Example 2: Another example of VBB *Unary matrix* found in project controlling area of the real-life financial service provider is shown in Table 5.5. Header column of this unary matrix designates two possible values of the attribute ”Resource type”, namely external and internal, whereas its header row contains labels for different KPIs used in project controlling area. The numerals at the intersections of columns and rows indicate the values of the corresponding KPI grouped by the corresponding value of the attribute ”Resource type”.
Not only numerals, but also textual descriptions, symbol VBBs (e.g. traffic lights, trend arrows and smileys, etc.), and categorical data (e.g. *in development/in production/to be retired* or *red/yellow/green*, etc.) as well as their combination are used by the industry partner to depict attribute values in unary matrix.
Connections to other VBBs: VBB *Unary matrix* may be used in combination with such symbol VBBs as VBB Smiley, VBB Traffic light, VBB Battery, VBB Harvey balls, VBB Speedometer, VBB Trend arrow, VBB Progress bar, VBB bar chart and VBB Pie chart, to depict the

Resource type	KPI1	KPI2	KPI3
External	70	70	81
Internal	30	30	18

Table 5.5.: Unary matrix of the real-life financial service provider (anonymized) - example 2

contents of its cells. VBB *Unary matrix* may be also used with structural VBB Grouping to increase its dimensionality by grouping its instances by instances of another type.
Abstract viewmodel: Information model of VBB *Unary matrix* is shown in Figure 5.13.

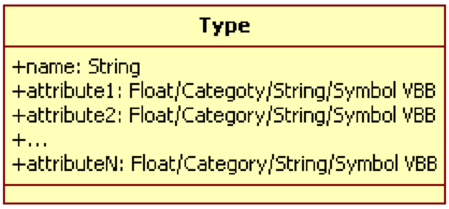


Figure 5.13.: VBB Unary matrix information model

Stakeholder Document	Stakeholder 1	Stakeholder 2	Stakeholder 3	Stakeholder 4
Document 1	M	OD	M	M
Document 2			OD	M
Document 3		OD	M	M
Document 4			M	M

Table 5.6.: Binary matrix of the real-life financial service provider (anonymized) - example 1: Review matrix: M - review is mandatory, OD - review on demand

Labor category Role	A	B	C	D
Role 1	660	0	N/A	N/A
Role 2	660	N/A	N/A	N/A
Role 3	N/A	N/A	1300	0
Role 4	260	0	N/A	N/A

Table 5.7.: Extracted from binary matrix of the real-life financial service provider (anonymized) - example 2: Labor category matrix

Binary matrix

Usage: You want to visualize how instances of two types relate. Thereby, you want either just to show relation existence, or you want also to specify existing relations.
Explanation: VBB *Binary matrix* visualizes relationships between instances of two different types and therefor uses a many-to-many relationship. Header column and header row of VBB *Binary matrix* represent instances of two different types, respectively, and matrix cells either show the existence of relation between certain instances of these two types, if they take only two values, thus representing binary data (e.g. empty value, if no relation between the corresponding instances exists, and cross otherwise [Se11]), or they may specify the existing relation by taking different values in non-empty cells, usually represented by numerals, categorical data or symbol VBBs.
Example: In this section three possible scenarios of VBB *Binary matrix* use are considered.
Scenario 1: Header row and column contain instances of two different types correspondingly.
As shown in Table 5.6 header row of this binary matrix, used in the project management area of the real-life service provider, represents different stakeholders of review process, while its header column depicts different documents to be reviewed. Cells of this matrix take two different values: M - review is mandatory and OD - review on demand, or are empty in case no review of particular document should be performed by particular stakeholder. Thereby, the first row of the matrix means that Document 1 should be mandatory reviewed by Stakeholders 1, 3 and 4, and it should be considered by Stakeholder 2 in case of necessity.
Another example of usage of *Scenario 1* is CRUD-matrix considered in [Se05], which represents how different information objects can be used by different processes. Four possible types of use are: create, read, update and delete.
Scenario 2: Header row and column contain instances of a certain type and values of a certain attribute, respectively.
As shown in Table 5.7 header row of this binary matrix contains values of attribute ”Labor category” (*A, B, C* and *D*), and its header column represents instances of type ”Role”. Cells of this binary matrix represent values of attribute ”Salary” of type ”Role”. The values of cells take numeral values which correspond to salary rates of the corresponding role in the corresponding labor category. The cells which take the value ”N/A” show that the corresponding role can not be attributed to the corresponding category, in other words no relationship between them exists.
Scenario 3: Header row and column contain values of two different attributes, respectively.
Table 5.8 depicts project costs grouped by two attributes: project priority and project program, which designates the group of related projects managed in a coordinated way. Header row contains the values, which attribute ”program” takes (*p1, p2, p3* and *p4*), header column represents the values taken by attribute ”priority” (*a, b* and *c*). Cells of this binary matrix depict the values of attribute ”project costs” grouped by the corresponding values of ”priority” and ”program” attributes.
Connections to other VBBs: VBB *Binary matrix* may be used in combination with such symbol VBBs as VBB Smiley, VBB Traffic light, VBB Battery, VBB Harvey balls, VBB Speedometer, VBB Trend arrow, VBB Progress bar, VBB bar chart and VBB Pie chart, which may be used to depict the contents of its cells. VBB *Binary matrix* may be also used with structural

VBB Grouping to increase its dimensionality by grouping its instances by instances of another types.
Abstract viewmodel: Information model of VBB *Binary matrix* is shown in Figure 5.14.

Priority \ Program	p1	p2	p3	p4
a	216	6.426	1.584	0
b	2.282	6.500	477	7.961
c	0	6.312	5.573	56

Table 5.8.: Extracted from binary matrix of the real-life financial service provider (anonymized) - example 3: Project costs matrix

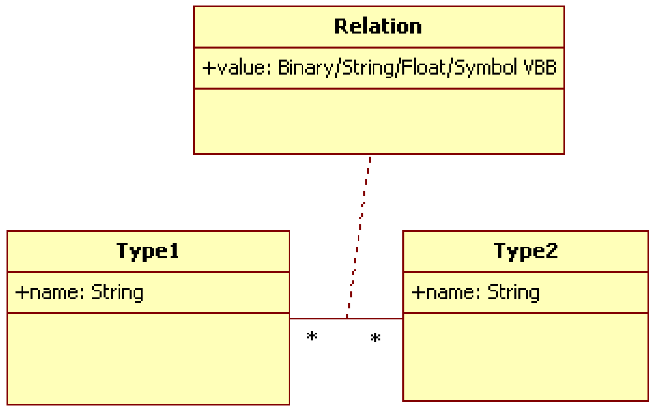


Figure 5.14.: VBB Binary matrix information model

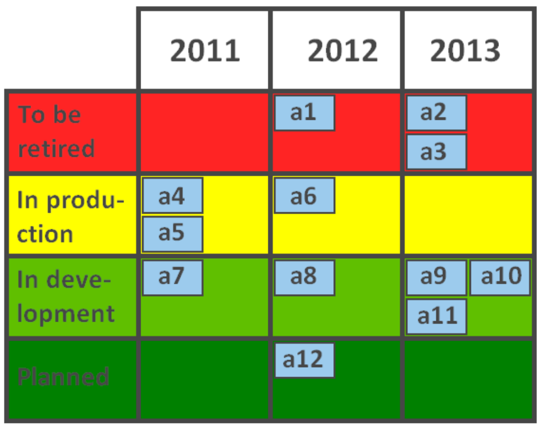


Figure 5.15.: Ternary matrix - example 1

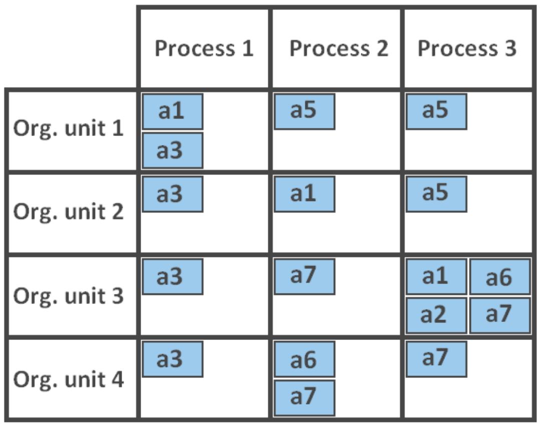


Figure 5.16.: Ternary matrix - example 2

Ternary matrix

Usage: You want to visualize how instances of three types relate.
Explanation: VBB *Ternary matrix* visualizes relationships between instances of three different types and therefor uses a many-to-many relationship. Header row, header column and cells of ternary matrix represent instances of three different types, respectively. Instances placed in ternary matrix cells are aligned with the instances of two other types that are represented in its header column and row.
Example:
Example 1: Example shown in Figure 5.15 and considered in [Bu08] depicts which applications are in which lifecycle phase in different years. In this example instances of two types, namely "Application" and "Time", are represented together with an attribute "Lifecycle phase". Header column of this ternary matrix depicts values of attribute "Lifecycle phase" and its header row displays years. Cells contain instances of type "Application" that is in the corresponding lifecycle phase in the corresponding year. In this example, each instance of type "Application" is represented only once in the cells of ternary matrix.
Example 2: Figure 5.16 schematically shows which business applications are participating in which processes of which organizational units. In this example instances of three types are involved, namely "Process", "Organizational unit" and "Business application". Header column depicts instances of type "Organizational unit", whilst header row contains instances of type "Process". Cells of this matrix represent instances of type "Business application" which are aligned with instances of two other types. In this example, some instances are placed in multiple cells.
Connections to other VBBs: VBB *Ternary matrix* may be used together with VBB Smiley, VBB Traffic light, VBB Battery, VBB Harvey balls, VBB Speedometer, VBB Trend arrow, VBB Progress bar, VBB bar chart and VBB Pie chart to encode additional information about the instances. Also it may be used in combination with VBB Icon, VBB Color coding and VBB Shape to depict instance status or differentiate between the instances of different types.
Abstract viewmodel: Information model of VBB *Ternary matrix* can be represented in two

ways depending on the specific situation.
Each of the instances of type "Application", represented in the first example (see Figure 5.15), appear only once in matrix cells. For this example information model shown in Figure 5.17 is suitable. In the ternary matrix considered in the second example (see Figure 5.16) some instances of type "Business application" recur in different cells several times, thereby some identical instances are positioned diagonally relative to each other. For this example the information model shown in the Figure 5.16 is not appropriate considering its inconsistencies related to diagonal location of identical instances. In such situation it is better to use the information model represented in Figure 5.18.

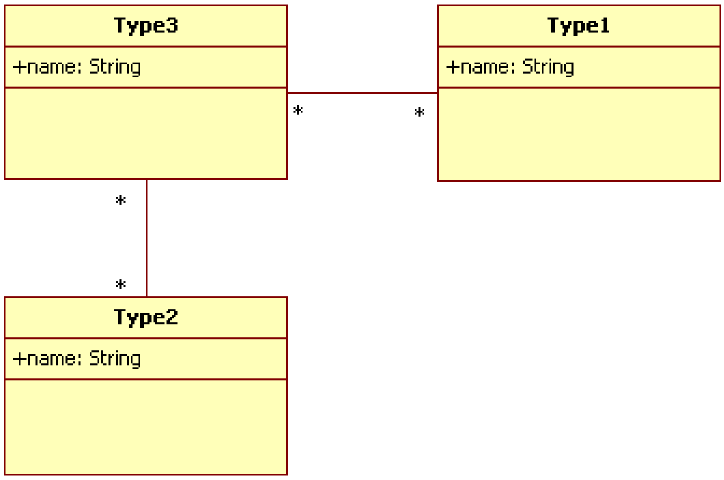


Figure 5.17.: VBB Ternary matrix information model (with intermediate class)

5.2.9. VBB Migration diagram

Usage: You want to visualize migration of functionality from one instance of a certain type to another.
Explanation: VBB *Migration diagram* is schematically depicted in Figure 5.2 (case J). Axis below visualizes time scale with time moments *Q1*, *Q2*, *Q3* and *Q4*. Symbols labeled with *a*, *b*, *c*, *d* and *e* represent instances of a certain type *Z* (*a*:*Z*, *b*:*Z*, *c*:*Z*, *d*:*Z* and *e*:*Z*), which lifecycles are depicted by the corresponding lines. Diamond symbols designate milestones, i.e. the moments of migration relevant decisions and events, and circles symbolize goals achievement. Different line styles represent different phases of lifecycle, and arrows which connect milestones symbolize migration of functionality from one instance of a certain type to another one with the corresponding milestones.
Example: Example shown in Figure 5.19 represents migration of functionality from one business application to another [Bu08].
Type: Structural VBB candidate
Formatting: Following format settings may be applied to VBB *Migration diagram*:

- show/hide vertical gridlines

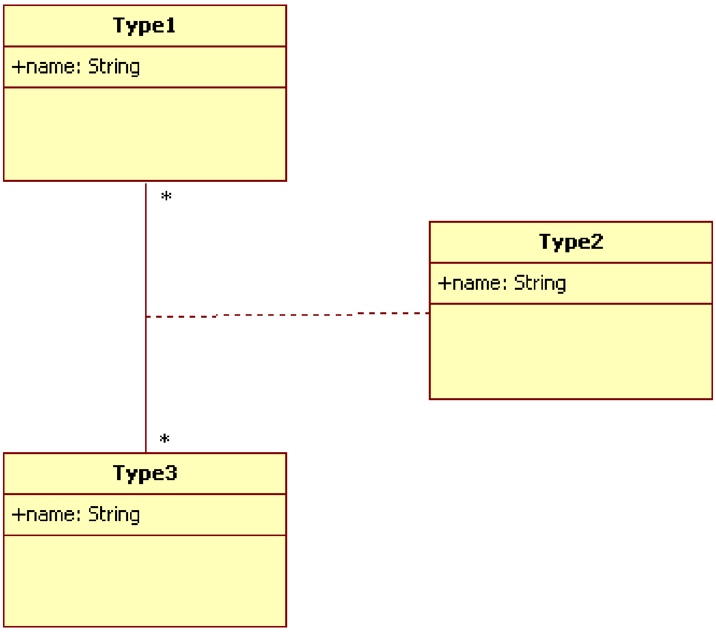


Figure 5.18.: VBB Ternary matrix information model (directly related classes)

- tabular/axial time scale
- timescale above/beneath the graphic area
- set background color
- set gridlines/axis color
- show/hide letterings of VBB graphical elements (e.g. diamonds, arrows)

Connections to other VBBs: VBB *Migration diagram* may be used with structural VBB Grouping, e.g. to group quarters by years.
Abstract viewmodel: Information model of VBB *Migration diagram* is shown in Figure 5.20.

5.2.10. VBB Radar chart

Usage: You want to display the values taken by one or more ordinally or interval-scaled attributes. Attribute value is represented by the distance of symbols from center.
Explanation: VBB *Radar chart* is schematically depicted in Figure 5.2 (case K) and is represented by a number of axes, which start from the pole and radiate. All axes have the same scale. VBB *Radar chart* can be used to visualize one of the following scenarios:

- *Scenario 1:* Values of a single attribute for a collection of instances of one particular type.
  Axes in this scenario represent different instances of a certain type. Attribute values

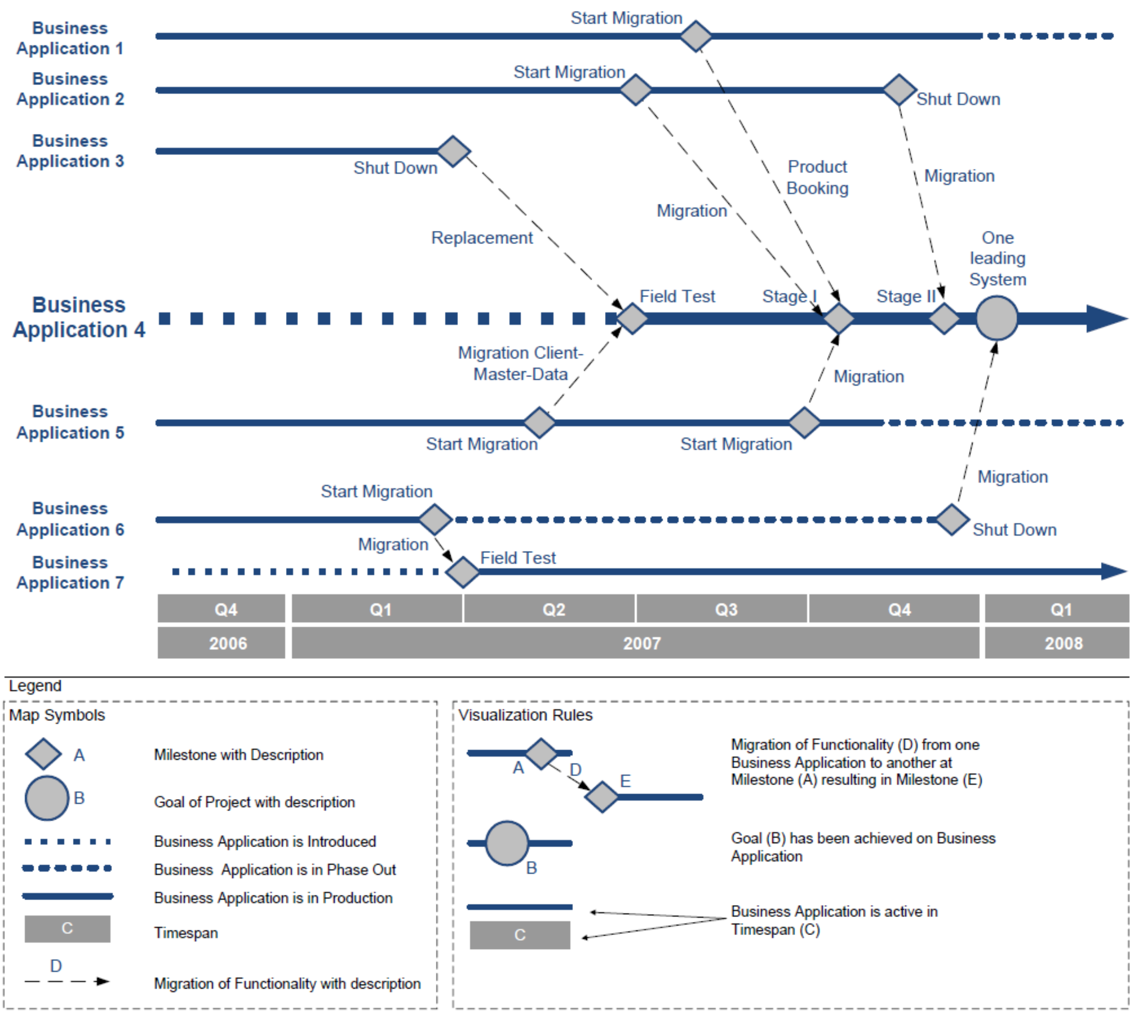


Figure 5.19.: Example of VBB Migration diagram [Bu08])

of a certain instance are represented by symbols, which are positioned on the corresponding axis. The farther away from the pole the symbol is depicted, the bigger the attribute value of the corresponding instance is. If scenario 1 is used to interpret VBB *Radar chart* in Figure 5.2, then u, v, w, x, y and z are instances of a certain type Z: u:Z, v:Z, w:Z, x:Z, y:Z and z:Z, and a is an attribute of type Z, which takes following values by different instances:
$u.a = y.a = 3$
$v.a = z.a = 1$
$w.a = x.a = 2$

- *Scenario 2:* Values of different attributes for a single instance of one particular type. In this case attributes have to be comparable, i.e. equally scaled. Axes in this scenario represent different attributes of a certain instance. Attribute values of these

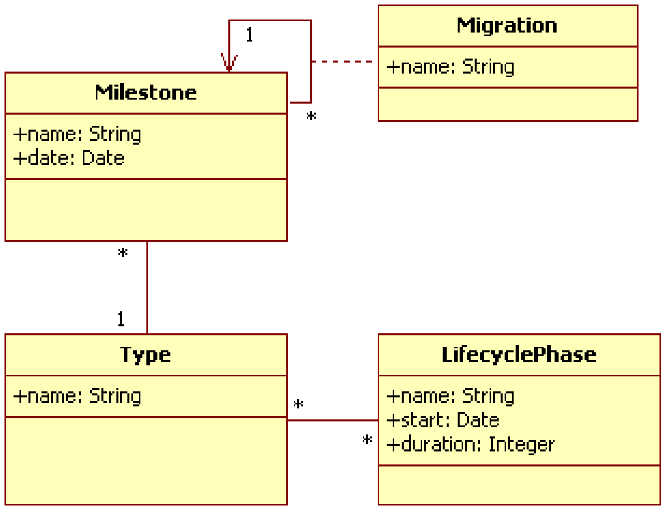


Figure 5.20.: VBB Migration diagram information model

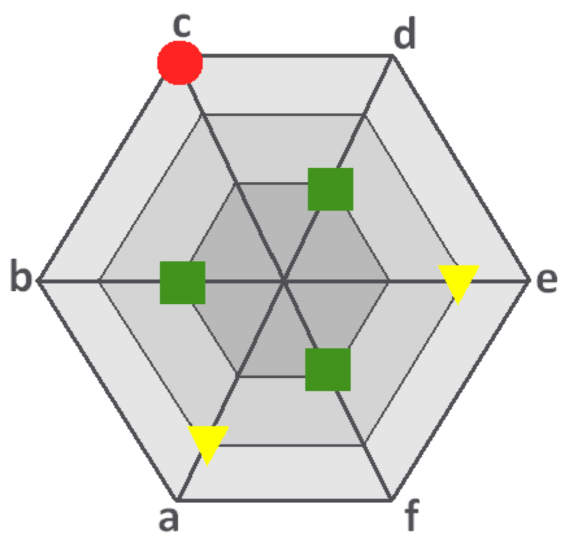


Figure 5.21.: VBB Radar chart of the real-life financial service provider (anonymized) - example 1

attributes are represented by symbols, which are positioned on the corresponding axis. The farther away from the pole the symbol is depicted, the bigger the value of the corresponding attribute is. If scenario 2 is used to interpret VBB *Radar chart* in Figure 5.2, then u, v, w, x, y and z are the attributes of a certain instance a of a certain type A: a:A, which take following values:
$a.u = a.y = 3$
$a.v = a.z = 1$
$a.w = a.x = 2$

- *Scenario 3:* Values of multiple attributes as taken by a collection of instances of one particular type. In this case attributes have also to be comparable. Axes in this scenario represent different instances of a certain type. Attribute values of a certain instance are represented by symbols, which are positioned on the corresponding axis. To differentiate between different attributes different symbols for their visualization are used

Example: Example shown in Figure 5.21 and used in finance area of the real-life financial service provider, visualizes different ordinally scaled status attributes, and thus belongs to the second scenario. Attributes in this example take only three values:
watch (status green)
manage (status yellow)
act (status red)
Each attribute value is represented by a certain symbol for the purpose of expressiveness reinforcement: "watch" is represented by green rectangle, "manage" corresponds to yellow triangle, and "act" is visualized by red circle.
Another example of VBB *Radar chart* is shown in Figure 5.22 and also represents the second scenario. Axes of this radar chart designate eight ordinally scaled attributes characterizing a certain tool, which take eight integer values on a scale of 0-7. The difference between this radar chart and the one considered before is that attribute values are connected with one another via line segments, which reinforce the expressiveness of the viewpoint.
Type: Structural VBB candidate
Formatting: Following format settings may be applied to VBB *Radar chart*:

- grid switch on/off
- axes switch on/off
- symbols (dots) connected / not connected with line segments
- cells fill color set
- grid color set

Connections to other VBBs: VBB *Radar chart* is close to the VBB Dot plot from the functionality perspective.
It may be used in combination with VBB Shape, VBB Color coding and VBB Icon to depict multiple attributes of the instances.
Abstract viewmodel: Information models of VBB *Radar chart* are identical to the ones of VBB Dot plot and are shown for each Scenario in Table 5.2.

5.2.11. VBB Regionalization

Usage: You want to visualize some instances of types, relations between them or their attributes (also aggregated per region) in geographical context.

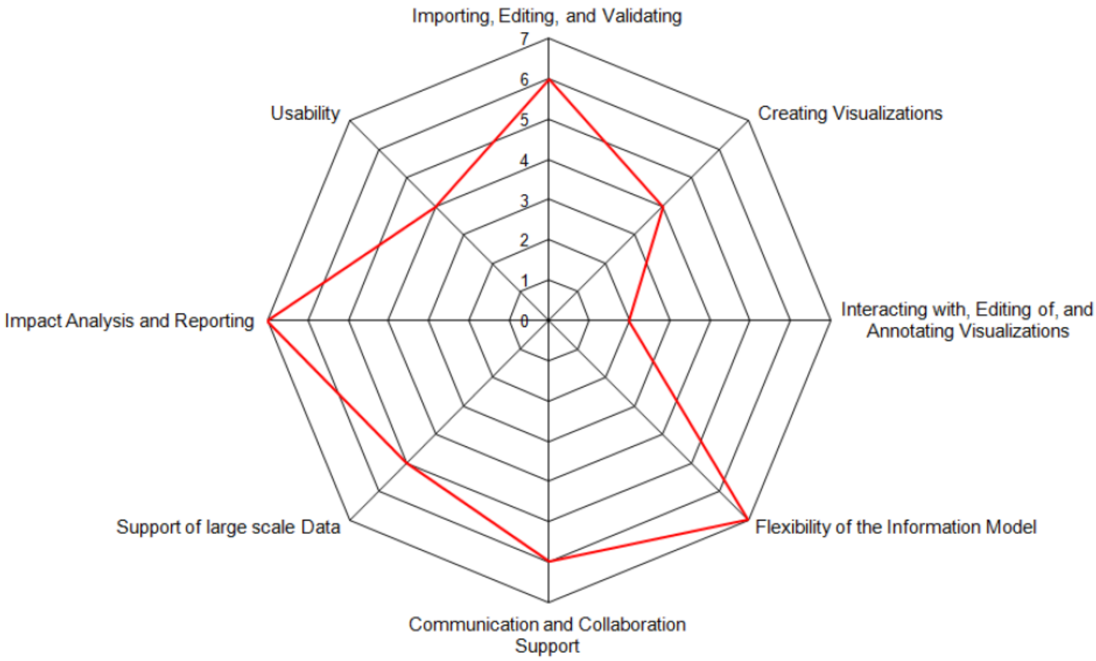


Figure 5.22.: VBB Radar chart - example 2: Source [Ma08]

Explanation: VBB *Regionalization* is schematically depicted in Figure 5.2 (case L) and represents geographical map splitted in regions, which do not necessarily mark state borders, but areas relevant to viewpoint concerns (designated by *a*, *b*, *c*, *d* and *e*).
Example: VBB *Regionalization* is used by the real-life financial service provider to visualize location of its branches, agencies and core markets, and is represented by the world political map. Thus, each region visualizes a certain country. The countries, in which the branches, agencies or core markets of financial service provider are located, differ by color. Concrete position is marked with a certain symbol, different for branches and agencies.
Type: Structural VBB candidate
Formatting: Zooming +/-.
Connections to other VBBs: VBB *Regionalization* may be used in combination with:

- VBB Color coding, VBB Shape and VBB Size to encode additional attributes
- VBB Icon to depict different instances within regions
- Compound chart symbol and compound dashboard symbol VBBs to depict a certain characteristic of a certain region or instances within this region
- VBB Graph do represent relations between instances of certain regions

5.2.12. VBB Scatter plot

Usage: You want to represent a relationship between two interval-scaled attributes of instances of a certain type.

Explanation: VBB *Scatter plot* is schematically depicted in Figure 5.2 (case M). *x*-coordinate depicts values of the first attribute, and *y*-coordinate represents values of the second attribute. Dots in central part of scatter plot represent values combinations of both attributes as taken by the different instances in the collection. All these instances are of one particular type.
Example: As shown in Figure 5.2 (case M), instances *a, b, c, d, e, f, g, h, i* and *j* of type Z (*a*:Z; *b*:Z; *c*:Z; *d*:Z; *e*:Z; *f*:Z; *g*:Z; *g*:Z; *h*:Z; *i*:Z; *j*:Z) have two attributes *x* and *y*. Attributes *x* and *y* of instance *a* take values *x1* and *y1*, respectively, and the corresponding dot is placed on the intersection of these two coordinate values.
A concrete use of this VBB is demonstrated on the following example: type Z represents class of projects, whereas instances *a, b, c, d, e, f, g, h, i* and *j* express concrete projects. Class of projects has two attributes: *x* - development effort in man-days, and *y* - total project costs. Figure 5.2 (case M) shows then that development effort for project *a* is equal to *x1* man-days, while its total costs are *y1*.
Type: Structural VBB candidate
Formatting: Following format settings may be applied to VBB *Scatter plot*:

- Switch on/off the gridlines
- Fade in/down the axes
- Axes rotation

Connections to other VBBs: VBB *Scatter plot* may be used in combination with VBB Size coding and VBB Color coding to encode additional attributes. VBB Shape coding may be applied to the dots of scatter plot to encode additionally the instances of other types.
Abstract viewmodel: Information model of VBB *Scatter plot* is shown in Figure 5.23.

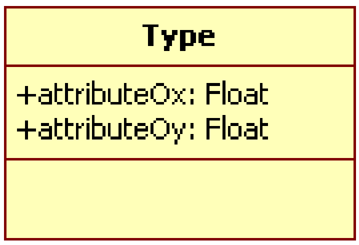


Figure 5.23.: VBB Scatter plot information model

5.2.13. VBB Scheduling

Usage: You want to visualize scheduling of activities, dependencies between them and important milestones. This VBB is also known as Gantt chart.
Explanation: VBB *Scheduling* is schematically depicted in Figure 5.2 (case N). Header row of this VBB represents time scale, and its header column shows activities and phases, which combine several subordinate activities under one umbrella activity. Bars placed

in the central part of VBB *Scheduling* represent duration of each individual activity: the longer the bar is, the longer the corresponding activity lasts. Multiple activities may be performed simultaneously. Dependencies between activities are represented by arrows, black diamonds visualize milestones, and thick black bars with arrows at the beginning and at the end depict duration of the corresponding phase.
Example: Shown in Figure 5.2 (case N) VBB *Scheduling* represents scheduling of three phases: *X*, *Y* and *Z*, and their subordinated activities $x1$, $x2$, $x3$, $z1$, $z2$. *t1*, *t2*, *t3* and *t4*, placed in header row, represent time units.
Type: Structural VBB candidate
Formatting: Following format settings are applicable to VBB *Scheduling*:

- show/hide grid
- place column headers above/beneath lifecycle diagram
- set cells text color
- set cells frame color

Connections to other VBBs: VBB *Scheduling* may be used in combination with VBB Shape and VBB Color coding to differentiate between such graphic elements as milestone (black diamond), activity (blue bar) and phase (black bar with arrows on its sides), and VBB Arrow to depict dependencies between activities.
Abstract viewmodel: Information model of VBB *Scheduling* is shown in Figure 5.24.

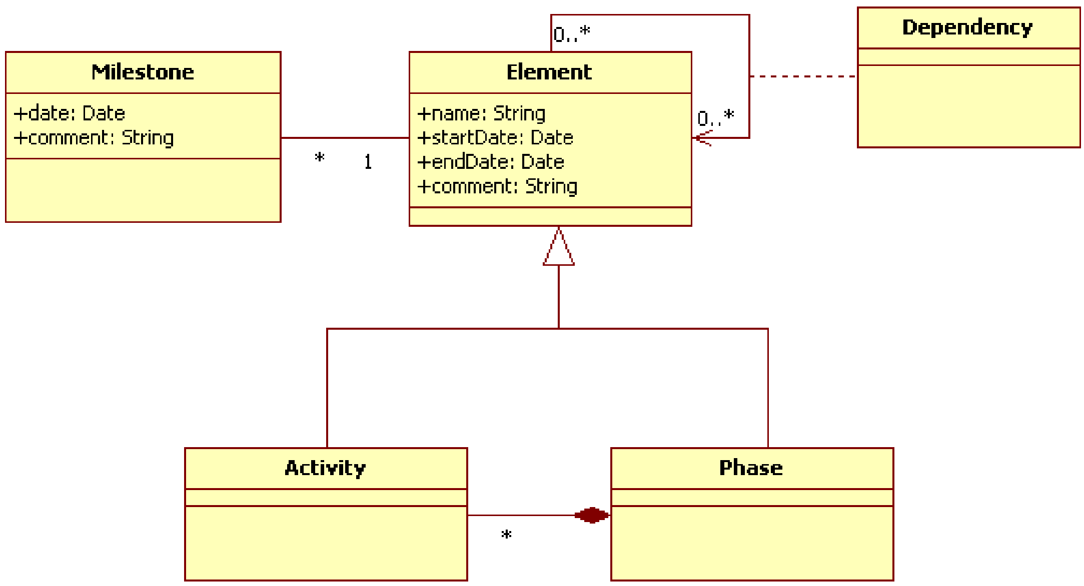


Figure 5.24.: VBB Scheduling information model

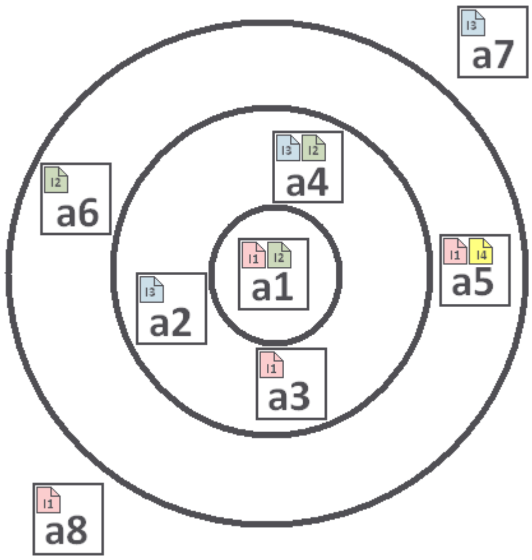


Figure 5.25.: VBB Set diagram

5.2.14. VBB Set diagram

Usage: You want to visualize membership of instances of a certain type in collection of sets. This VBB is also known as Venn diagram.
Explanation: VBB *Set diagram* is schematically depicted in Figure 5.2 (case O). *A, B,* and *C* represent sets, and *x1, x2, x3, x4, x5, x6, x7* and *x8* visualize instances of a certain type *Z*. Four scenarios of instances membership are possible in case of three sets:

- a certain instance belong to only one set, and do not belong to the other two: $x1 \in A$, $x2 \in B$, $x3 \in C$
- a certain instance belong to two sets, and do not belong to remaining third set: $x5 \in (A \cap B)$, $x4 \in (B \cap C)$, $x8 \in (A \cap C)$
- a certain instance belong to all three sets: $x7 \in (A \cap B \cap C)$
- a certain instance do not belong to any of sets: $x6 \notin (A \cup B \cup C)$

Example: VBB *Set diagram* shown in Figure 5.25 visualizes conformity to a programming language specific corporate standard by distance to center [La08]. *a1, a2, a3, a4, a5, a6* and *a8* represent business applications, icons *l1, l2,l3* and *l4* encode programming languages, different business applications are written in. Concentric circles represent sets of instances with different values of attribute ”degree of conformity with a specific corporate standard” from 100% (completely conform) to 0% (absolutely non-conform):
A1: 76%-100% conform: *a1*
A2: 51%-75% conform: *a2, a3, a4*
A3: 26%-50% conform: *a5, a6*

A4: 0%-25% conform: *a8*
Circles, which represent sets, are concentric according to their nature: instances which are 80% conform to a certain standard, are also 60% conform to this standard, and thus belong to both sets *A1* and *A2*.
Type: Structural VBB candidate
Formatting: Following decorators can be applied to the VBB *Set diagram*:

- set/remove sets headers
- set fill color of circles
- set circles color

Connections to other VBBs: VBB *Set diagram* may be used in combination with VBB Shape and VBB Icon to depict instances of types. This VBB can be used alternatively to non-exclusive VBB Clustering.
Abstract viewmodel: Information model of VBB *Set diagram* is shown in Figure 5.26.

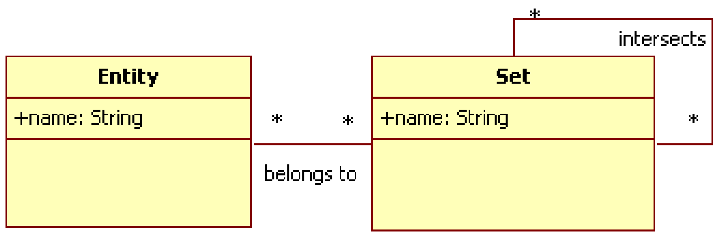


Figure 5.26.: VBB Set diagram information model

5.2.15. VBB Time interval map

Usage: You want to show how instances of a certain type are affected by instances of another type in time.
Explanation: VBB *Time interval map* is schematically depicted in Figure 5.2 (case P). Header row displays time scale, whereas header column represents instances *w*, *x*, *y*, *z* of a certain type *Z*: *w:Z*, *x:Z*, *y:Z*, *z:Z*. Instances of type *Z* affect versions of instances of type *A* (e.g. *a.v3* is a version of instance *a* and *b.v1* is a version of instance *b* of type *A*, *c.v5* displays a version of instance *c* of type *A*, etc). Here four scenarios are possible:

- *Scenario 1:* A certain instance of type *Z* initiates the replacement of a certain version of a certain instance of type *A* by another one (e.g. version *v3* of instance *a* of type *A* is replaced by version *v1* of instance *b* of type *A* due to instance *w* of type *Z* (see Figure 5.2))
- *Scenario 2:* A certain instance of type *Z* modifies a certain version of instance of type *A* (e.g. version *v5* of instance *c* of type *A* is modified by instance *x* of type *Z* (see Figure 5.2))

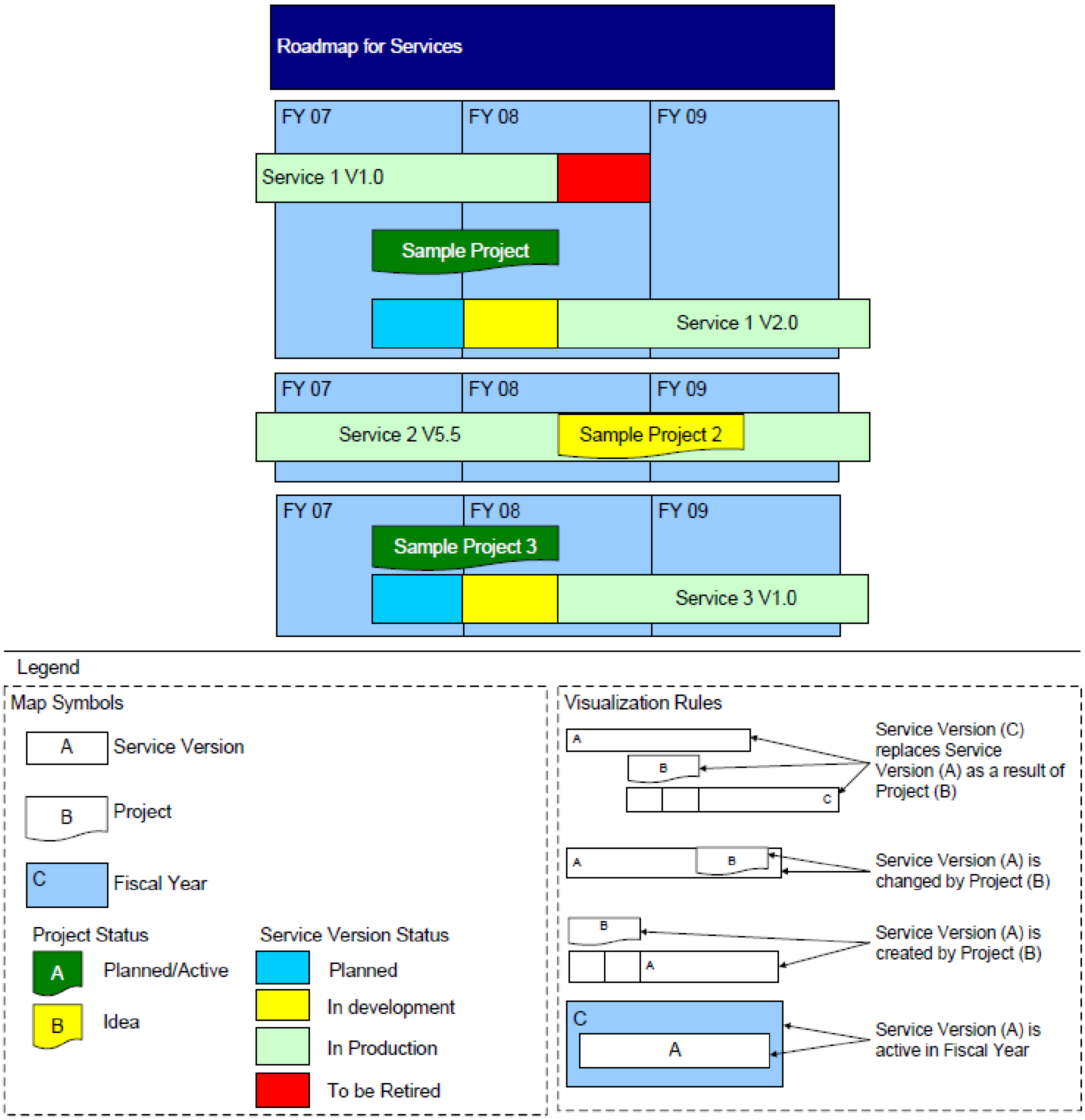


Figure 5.27.: Example of time interval map (Source: [Bu08])

- *Scenario 3:* A certain instance of type *Z* initiates creation of a certain version of instance of type *A* (e.g. version *v1* of instance *d* of type *A* is created by instance *y* of type *Z* as shown Figure 5.2)
- *Scenario 4:* A certain instance of type *Z* initiates deletion of a certain version of instance of type *A* (e.g. version *v7* of instance *e* of type *A* is deleted by instance *z* of type *Z*)

VBB *Time interval map* also differentiates between different lifecycle phases of versions of instances, namely: planned, in development, in production and to be retired.
Example: Example of VBB Time interval map shown in Figure 5.27 displays how project

proposals intend to affect services with time [Bu08]. In this example type *Z* represents project proposals, whereas type *A* displays services.
Type: Structural VBB candidate
Formatting: Following decorators can be applied to VBB *Time interval map*:

- show/hide vertical gridlines
- put/remove breaks between horizontal gridlines
- tabular/axial/superscribed time scale
- timescale above/beneath the graphic area
- set background color
- set gridlines/axis color
- show instances names in header column / depict instances names direct on the graphical elements

Connections to other VBBs: VBB *Time interval map* is used in combination with VBB Color coding to depict different statuses of instances, as well as VBB Shape and VBB Icon to differentiate between the instances of different types.
Abstract viewmodel: Information model of VBB *Time interval map* is shown in Figure 5.28.

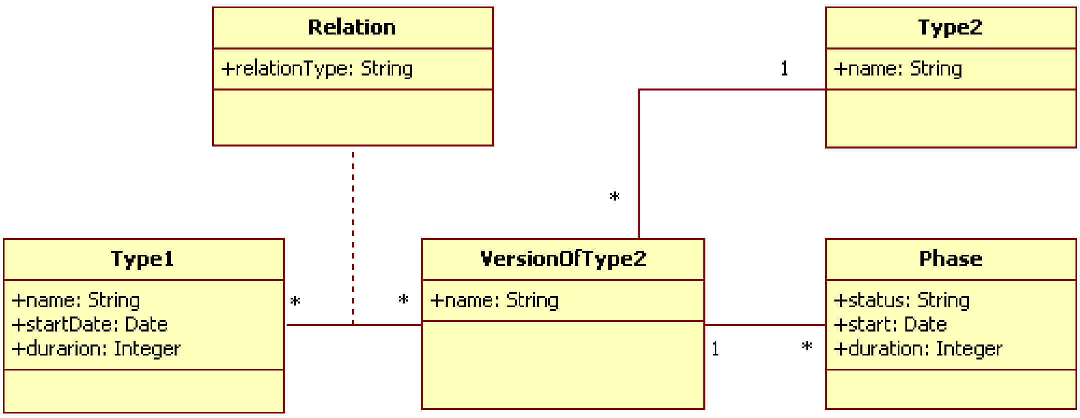


Figure 5.28.: VBB Time interval map information model

5.2.16. Structural VBBs optimization recommendations

After introducing new structural viewpoint building block candidates in previous subsections, this subsection briefly reviews existing structural VBBs and makes optimization proposal on their reorganization.
As already mentioned in Subsection 2.3.2, currently 8 structural VBBs are available in BEAMS collection, namely:

- VBB Binary matrix
- VBB Clustering
- VBB Matrix (continuous)
- VBB Ordering
- VBB Recursive clustering
- VBB Swimlane
- VBB Ternary matrix
- VBB Tree diagram

VBBs Binary matrix and VBB Matrix (continuous) should not be considered any more as separate viewpoint building blocks, as they are included in VBB Matrix as special cases. The same applies to VBB Recursive clustering, as this building block can be also considered as special case of VBB Clustering along with another two special cases regarded above. VBB Ternary matrix, on the contrary, does not belong to VBB Matrix and its special cases for the reason that its elements may be stretched out among several rows and columns elements, which is not allowed by VBB Matrix. On the other hand, VBB Ternary matrix in concept is very similar to VBB Swimlane with the only difference, that its interior instances are aligned with instances of two other types, and not only one type, as it is in case of VBB Swimlane. For that reason VBB Ternary matrix is better to rename to VBB 2D-Swimlane that reflects the similarities and differences between these VBBs. Following this, reorganized BEAMS structural VBBs collection is shown in Figure 5.29.

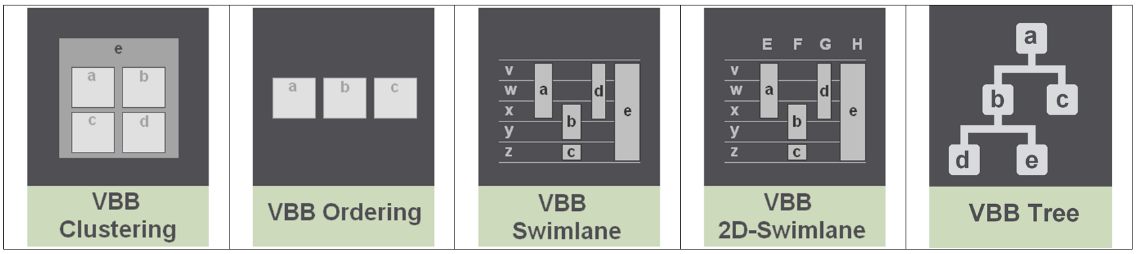


Figure 5.29.: Reorganized structural BEAMS VBBs

5.3. Symbol VBBs candidates

During the viewpoints decomposition phase also the need for symbol viewpoints building blocks arose, which are not yet available in BEAMS. Although a plurality of additional symbols is needed, they all can be categorized in four groups:

- Shapes

- Icons
- Compound chart symbols
- Compound dashboard symbols

Subsections below describe briefly each of these groups and provide detailed specification of each representative of the groups. Hence not only the introduction of new symbol viewpoints building blocks candidates, but also the incorporation of existing BEAMS VBBs in new viewpoints building blocks structure are provided.
In the same way as for structural VBB candidates symbol VBB candidates descriptions are based on layout used in [Se11] (see Subsection 2.3.2). Layout sections ″Name″, ″Explanation″, ″Examples″, ″Visual variables″ and ″Type″ are incorporated into viewpoint building blocks candidates layout one-to-one. Section ″Usage″ for structural VBBs description is used in the same meaning as by structural VBB candidates description (see Section 5.2). Furthermore, the issue named ″Connections to other VBBs″ is described for each symbol VBB candidate to show which building blocks may be used as a base map for the candidate under consideration. On the contrary, BEAMS VBBs layout section ″Tags″ was omitted in VBBs candidates descriptions.
This section concludes with subsection which integrates the existing symbol VBBs into certain groups and gives an overview on extended BEAMS collection of symbol viewpoints building blocks.

5.3.1. Shape

This group includes all shapes identified during the viewpoints decomposition phase, which can be divided into the following building blocks (see Figure 5.30):

- VBB Polygon
- VBB Ellipse
- VBB Rounded rectangle
- VBB Line
- VBB Arrow

Subsections below consider these building blocks in detail.

VBB Polygon

Usage: You want to visualize some instances of types.
Explanation: This VBB may be used in combination with structural VBBs to indicate instances of certain type, or it can be used as a part of compound symbols. VBB *Polygon* as well as its most frequent special cases, among which are triangle, rectangle, square and rhomb, are schematically depicted in Figure 5.31. All special cases are obtained from VBB Polygon by visual variables adjustments

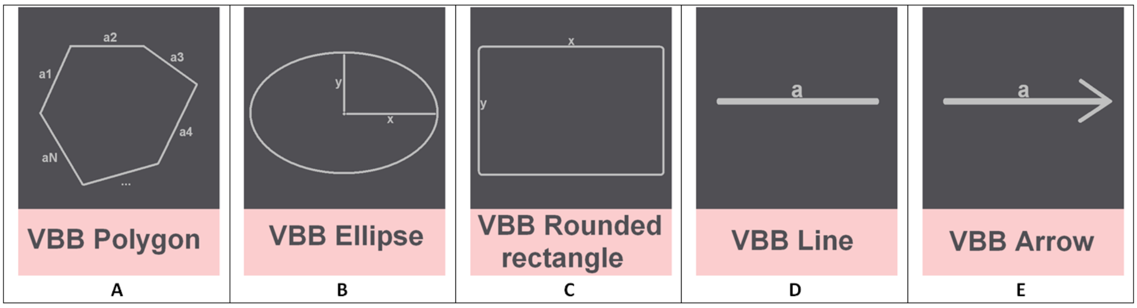


Figure 5.30.: VBBs of the group "Shape"

Type: Symbol VBB candidate
Connections to other VBBs: Different special cases of VBB *Polygon* are often used in combination with VBB Graph and VBB Tree. Its special case *Rectangle* is a part of VBB Swimlane and VBB 2D-Swimlane. Another example of use for *Rectangle* is VBB Bar chart (see Section VBB Traffic light).
Visual variables: In relation to VBB *Polygon* following visual variables may be used to encode additional information:

- border color
- border style
- border thickness
- fill color
- fill texture
- shape size (side lengths)
- angle sizes

By setting the number of vertices, defining angle sizes, side lengths and shape rotation any special case shown in Figure 5.31 can be generated from VBB Polygon. Special case of VBB *Polygon* rectangle is already a part of BEAMS collection [Se11].

VBB Ellipse

This VBB is already described in BEAMS [Se11]. It is also important to mention, in relation to VBB *Ellipse* the same visual variables as by VBB Polygon may be applied to encode additional information. This viewpoint building block as well as its special case *Circle* are very often used in combination with VBB Graph and VBB Tree in order to depict instances of particular type. VBB *Ellipse* is also a constitute part of VBB Traffic light.

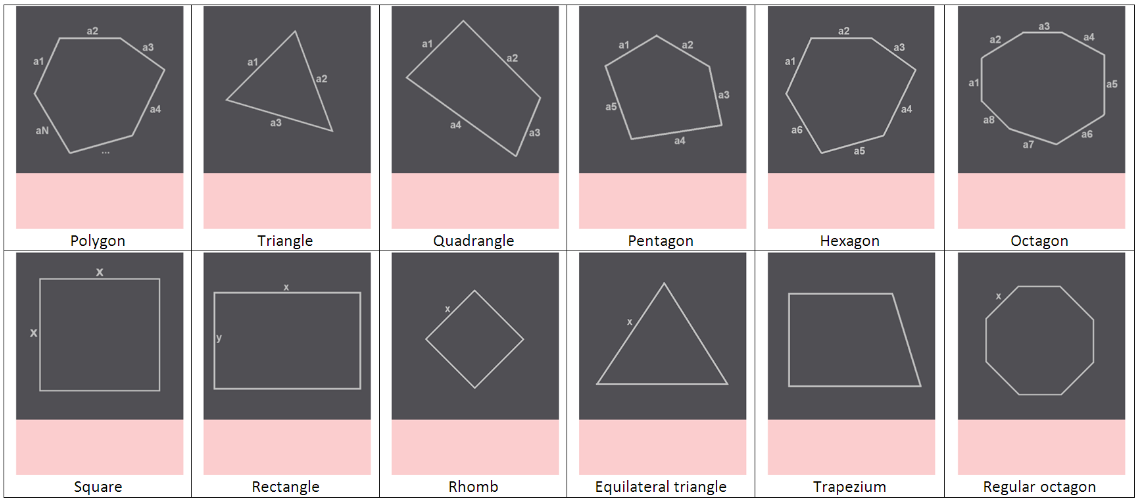


Figure 5.31.: VBB Polygon and its special cases

VBB Rounded rectangle

The characteristics of VBB *Rounded rectangle* are very close to the ones of special case of VBB Polygon - Rectangle. This VBB is also used to represent some instances of particular type from EA information model, has very similar cooperative VBBs and the same visual variables with an additional visual variable: roundness. If the roundness is equal to zero, rounded rectangle transforms into usual rectangle considered within special cases of VBB Polygon.

VBB Line

Usage: You want to visualize relations between some instances of types.
Explanation: VBB *Line* may be used in combination with structural VBBs to indicate relationships between instances of one particular type or different types.
Type: Symbol VBB candidate
Connections to other VBBs: This VBB is an embedded part of VBB Graph.
Visual variables: In relation to VBB *Line* following visual variables may be used to encode additional information:

- line color
- line style
- line thickness

By setting the number of endpoints and their position different polylines as special cases of VBB *Line* may be defined (see Figure 5.32).

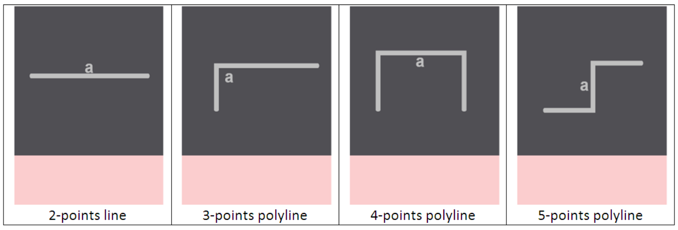


Figure 5.32.: Special cases of VBB Line

VBB Arrow

Usage: You want to visualize directional relationships between some instances of types.
Explanation: Often it is necessary not only to display relation existence between two instances, but also to show the direction. Moreover, it can be necessary if considered relationship oriented by its nature, as for example the relation of subordination or ordering, and representation of data flow. In these cases VBB *Arrow* is used. This VBB may be used in combination with structural VBBs to depict directional relationships between instances of one particular type or different types.
Type: Symbol VBB candidate
Connections to other VBBs: This VBB may be used in combination with VBB Graph and is an embedded part of VBB Tree, VBB Scheduling and VBB Interactioning.
Visual variables: Relating to VBB *Arrow* following visual variables may be used to encode additional information:

- arrow color
- line style
- line thickness
- end style

In Figure 5.33 different forms of VBB arrow identified during the viewpoints decomposition phase are represented.

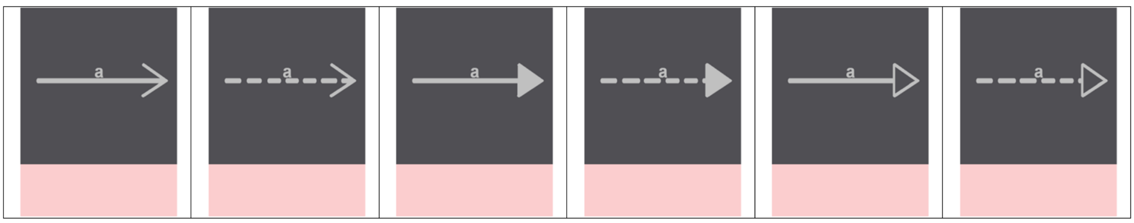


Figure 5.33.: Forms of VBB Arrow identified during the viewpoints decomposition phase

5.3.2. Icon

Usage: You want to visualize some instances of some types and thereby to use symbols, which resemble the instances they are representing.
Explanation: Whereas shapes are abstract instance representations, for which it is almost always necessary to have explanatory legends, VBB *Icons* usually have a resemblance to the instances they are representing, and are relatively self-explanatory. Since many authors and organizations often use their own icon representations, a huge variety of them exist. In Figure 5.34 the most common icons identified during the viewpoints decomposition step are represented. The last figure "Icon" signifies the multitude of remaining icons, which are not considered in detail in this thesis through their specificity.
Type: Symbol VBB candidate
Connections to other VBBs: This VBB may be used in combination with VBB Matrix, VBB Clustering, VBB Graph and VBB 2D-Swimlane.
Visual variables: Usually no visual variables are used by VBB *Icon* to encode additional information, as icons are often depicted as pictures. However, in some cases (e.g., database icon, lollipop icon, tick icon, etc.) shown in Figure 5.34 such visual variables as line color, line style, line thickness, fill color and fill texture may be applied.

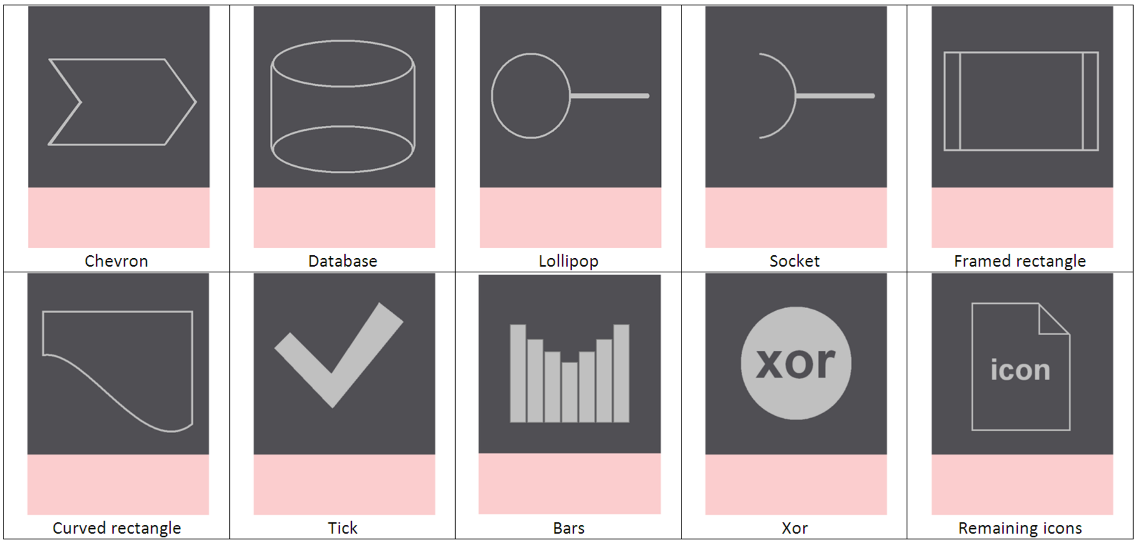


Figure 5.34.: Most common representatives of VBB Icons detected during the viewpoints decomposition step

5.3.3. Compound chart symbol

This group includes symbol VBBs composed from a number of simple shapes, which take on a new meaning being combined in particular way, namely form certain chart. The representatives of this group are VBB bar chart and VBB pie chart, which are already present in BEAMS [Se11]. However, during the viewpoints decomposition several special cases of these two building blocks have been identified. In Figure 5.35 all new candidates of this group are shown.

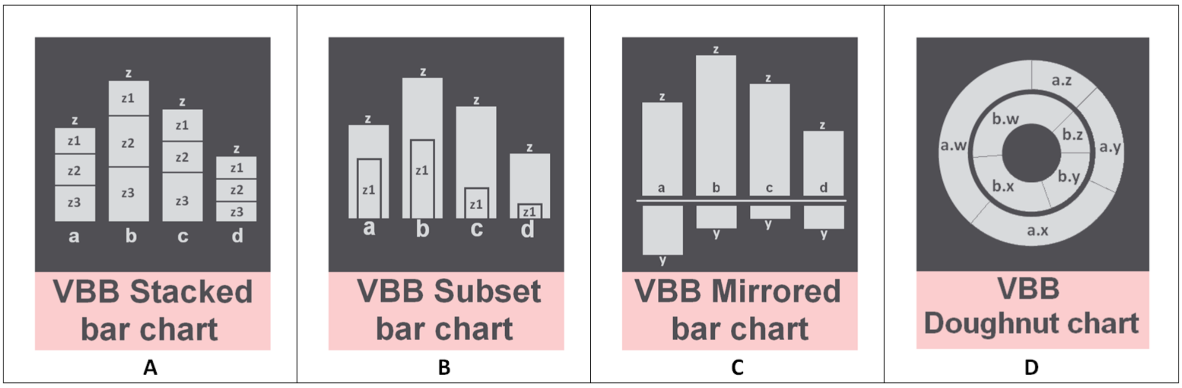


Figure 5.35.: Symbol VBBs: compound chart symbols candidates

VBB Bar chart

VBB *Bar chart* together with its two special cases, namely VBB *Multiple bar chart* and VBB *Bar chart with variable width*, are already parts of BEAMS collection and considered in detail in [Se11]. This subsection is concerned with three further special cases of VBB *Bar chart*:

- VBB Stacked bar chart
- VBB Subset bar chart
- VBB Mirrored bar chart

VBB Stacked bar chart
Usage: You want to display the values taken by a certain interval-scaled attribute, and thereby to show how different part attributes contribute to the values of this attribute. Size of a bar chart is further used to display attribute values and contributions of part attributes.
Explanation: VBB *Stacked bar chart* is schematically depicted in Figure 5.35 (case A). The Figure indicates four different values as taken by a certain interval-scaled attribute. Symbols labeled *a*, *b*, *c* and *d* represent instances of a certain type *Z* (*a:Z*, *b:Z*, *c:Z* and *d:Z*), where type *Z* has an interval-scaled attribute *z*, taking values as follows:
a.z = 4.0
b.z = 5.0
c.z = 4.5
d.z = 3.0
z1, *z2* and *z3* signify part attributes of attribute *z*, i.e. represent the same attribute as *z*, yet taken not for the whole set of elements, but for its subsets. That means that the sum of part attribute values of a certain instance is equal to the value of its attribute. Part attribute values for each instance are represented in a form of stack, which serves as a basis for the name of this VBB candidate. Part attributes shown in Figure 5.35 (case A) take values as

follows:
a.z1 = 1.6; a.z2 = 1.4; a.z3 = 1.0 → a.z1 + a.z2 + a.z3 = a.z = 4.0
b.z1 = 2.0; b.z2 = 1.8; b.z3 = 1.2 → b.z1 + b.z2 + b.z3 = b.z = 5.0
c.z1 = 1.9; c.z2 = 1.4; c.z3 = 1.2 → c.z1 + c.z2 + c.z3 = c.z = 4.5
d.z1 = 0.9; d.z2 = 1.1; d.z3 = 1.0 → d.z1 + d.z2 + d.z3 = d.z = 3.0
Example: VBB *Stacked bar chart* is used by the real-life global financial service provider to represent the following scenario:

- Instances *a, b, c, d* display organizational units of an enterprise
- Attribute *z* visualizes activities performed in a certain year in this organization and measured in million euro
- Part attributes *z1, z2* and *z3* represent expenditures for different types of activities: *z1* refers to internal projects implemented, *z2* depicts volumes of application support projects, and *z3* shows amount of product maintenance projects.

Type: Symbol VBB candidate
Connections to other VBBs: VBB *Stacked bar chart* may be used in combination with VBB Matrix, VBB Clustering, VBB Graph, VBB Regionalization, VBB Swimlane and VBB 2D-Swimlane to represent additional information of their elements.
Visual variables: VBB Color coding is used in combination with VBB *Stacked bar chart* to differentiate between its part attributes, and VBB Size coding (length of bars) visualizes attribute and part attribute values.

VBB Subset bar chart
Usage: You want to display values taken by a certain interval-scaled attribute, and thereby to highlight a value of particular part attribute of this attribute. Size of a bar chart is used to display attribute and part attribute values.
Explanation: VBB *Subset bar chart* is schematically depicted in Figure 5.35 (case B). The figure indicates four different values as taken by a certain interval-scaled attribute. Symbols labeled *a, b, c* and *d* represent instances of a certain type Z (*a*:Z, *b*:Z, *c*:Z and *d*:Z), where type Z has an interval-scaled attribute *z*, taking values as follows:
a.z = 4.0
b.z = 5.0
c.z = 4.5
d.z = 3.0
z1 signifies part attribute of attribute *z*, i.e. represent the same attribute as *z*, yet taken not for the whole set of elements, but for its certain subset, which should be highlighted. Part attribute is represented within attribute for each instance, and its value is always less than the value of attribute for the corresponding instance, as it is calculated for a part of the set.
Part attribute *z1* shown in Figure 5.35 (case B) takes values as follows:
a.z1 = 2.9 → a.z1 < a.z = 4.0
b.z1 = 3.7 → b.z1 < b.z = 5.0
c.z1 = 1.1 → c.z1 < c.z = 4.5
d.z1 = 0.5 → d.z1 < d.z = 3.0
Example: VBB *Subset bar chart* is used by the real-life global financial service provider to represent the following scenario:

- Instances *a, b, c, d* display organizational units of an enterprise
- Attribute *z* visualizes all activities planned for a certain year in this organization and measured in million euro
- Part attribute *z1* represents the part of planned activities, which is already performed

Type: Symbol VBB candidate
Connections to other VBBs: VBB *Subset bar chart* may be used in combination with VBB Matrix, VBB Clustering, VBB Graph, VBB Regionalization, VBB Swimlane and VBB 2D-Swimlane to represent additional information of their elements.
Visual variables: VBB Color coding is used in combination with VBB *Subset bar chart* to highlight its part attribute values, and VBB Size coding (length of bars) visualizes attribute and part attribute values.

VBB Mirrored bar chart
Usage: You want to visualize values taken by two interval-scaled attributes. Size of a bar chart is used to display attribute values.
Explanation: VBB *Mirrored bar chart*, schematically shown in Figure 5.35 (case C), is used analogically to another special case of VBB Bar chart, namely VBB Multiple bar chart. Symbols labeled *a, b, c* and *d* represent instances of type Z (*a:Z, b:Z, c:Z* and *d:Z*), where type Z has two interval-scaled attributes *y* and *z*, taking values as follows:
a.z = 4.0
a.y = 2.5
b.z = 5.0
b.y = 0.8
c.z = 4.5
c.y = 0.5
d.z = 3.0
d.y = 0.8
Values of attribute *z* are represented by the upper bars, whereas values of attribute *y* are displayed by the bars placed under the x-axis.
Example: VBB *Mirrored bar chart* is used by the real-life global financial service provider to track numbers of ordered personal days involved in projects and application support (*z*) and product maintenance (*y*) per month (type Z represent type "Date", whereas a, b, c, d are concrete instances of this type, i.e. concrete months of the current year).
Type: Symbol VBB candidate
Connections to other VBBs: VBB *Mirrored bar chart* may be used in combination with VBB Matrix, VBB Clustering, VBB Graph, VBB Regionalization, VBB Swimlane and VBB 2D-Swimlane to represent additional information of their elements.
Visual variables: VBB Size coding (length of bars) visualizes attribute values.

VBB Pie chart

VBB *Pie chart* is already included in BEAMS collection and considered in detail in [Se11]. This subsection is concerned with its special case VBB *Doughnut*.

VBB Doughnut

Usage: You want to visualize ratios between values taken by one or more interval-scaled attributes. Sector sizes are used to display attribute values.

Explanation: VBB *Doughnut* is schematically shown in Figure 5.35 (case D) and may be used to represent two scenarios:

- *Scenario 1:* Each ring of VBB *Doughnut* visualizes a certain instance of type, whereas ring sectors represent ratios among interval-scaled attributes of the corresponding instance
- *Scenario 2:* Each ring of VBB *Doughnut* visualizes a certain interval-scaled attribute of type, whereas ring sectors represent ratios among values of the corresponding attribute taken on different instances of type

In principle *VBB Doughnut* may contain any number of rings. In the case of one ring VBB *Doughnut* is equivalent to VBB Pie chart in its characteristics.

Example:

Scenario 1: The rings of VBB *Doughnut*, shown in Figure 5.35 (case D), represent instances *a* and *b* of type Z (a:Z, b:Z), where type Z has four interval-scaled attributes *w, x, y* and *z*, taking values that sum up to 100% as follows:

a.w = 40%
a.x = 40%
a.y = 15%
a.z = 5%
b.w = 40%
b.x = 35%
b.y = 18%
b.z = 7%

In this case each ring of VBB *Doughnut* visualizes a certain instance of type Z, whereas ring sectors represent ratios between values of different attributes for a certain instance.

Scenario 2: In the second scenario the rings of VBB *Doughnut* visualize attributes *a* and *b* of type Z, whereas ring sectors represent ratios among attribute values taken on different instances *w, x, y* and *z* of type Z (*w*:Z, *x*:Z, *y*:Z and *z*:Z) as follows: w.a = 40%
x.a = 40%
y.a = 15%
z.a = 5%
w.b = 40%
x.b = 35%
y.b = 18%
z.b = 7%

VBB *Doughnut* with only one ring is used by the real-life global financial service provider to visualize number of locations per region (in %). In this case Scenario 2 is used. Instances *w, x, y* and *z* represent the countries, and the whole ring displays values of attribute "Ratio of locations per region" taken on different values of class "Region".

Type: Symbol VBB candidate

Connections to other VBBs: VBB *Doughnut* may be used in combination with VBB Matrix,

VBB Clustering, VBB Graph, VBB Regionalization, VBB Swimlane and VBB 2D-Swimlane to represent additional information of their elements.
Visual variables: VBB Size coding (sector sizes) visualizes attribute values. Additionally, VBB Color coding may be used to differentiate between different attributes (Scenario 1) or instances of type (Scenario 2).

5.3.4. Compound dashboard symbol

This group includes symbol VBBs formed by a number of simple shapes, which take on a new meaning being combined in particular way, namely represent dashboard charts. VBBs of this group are used to visualize status, trend or degree to which a particular item meets a particular criterion. They also allow to compare two or more values. Further, it is distinguished between two types of compound dashboard symbols, namely:

- discrete scaled compound dashboard symbols
- continuously scaled compound dashboard symbols

The representatives of continuously scaled compound dashboard symbols are able to depict each value taken by some interval-scaled variable, whereas discrete scaled compound dashboard symbols possess less visual power than number of values of interval-scaled variable exist, and thus is more appropriate for visualization of ordinally scaled values. If, however, discrete scaled compound dashboard symbols should be used for interval-scaled attribute values representation, a certain transformation of interval-scaled values into ordinally scaled ones should be performed. The transformation implies that certain multitudes of interval-scaled values are set in correspondence to certain ordinally scaled values.
Compound dashboard symbol VBBs are shown in Figure 5.36 and are considered in detail in subsequent subsections.

VBB Battery

Usage: You want to visualize key measure relevant for particular objective, business object, business process or technical component.
Explanation: VBB *Battery* belongs to discrete scaled type, takes only three different values, namely full, half-full and empty (see Table 5.9), and is used to visualize ordinally scaled values. To be able to use this VBB to visualize interval-scaled values, it is necessary to perform the following transformation. Usually, two values for quantitative key measure are defined: target level and acceptance level. The usage of three different forms of VBB *Battery* is defined by the following rules:

- If the actual value of a certain KPI is greater/lower than its target value, then empty VBB *Battery* is used
- If the actual value of a certain KPI is lower/greater than its target value, but is greater/lower than its acceptance value, then half-full VBB *Battery* is used

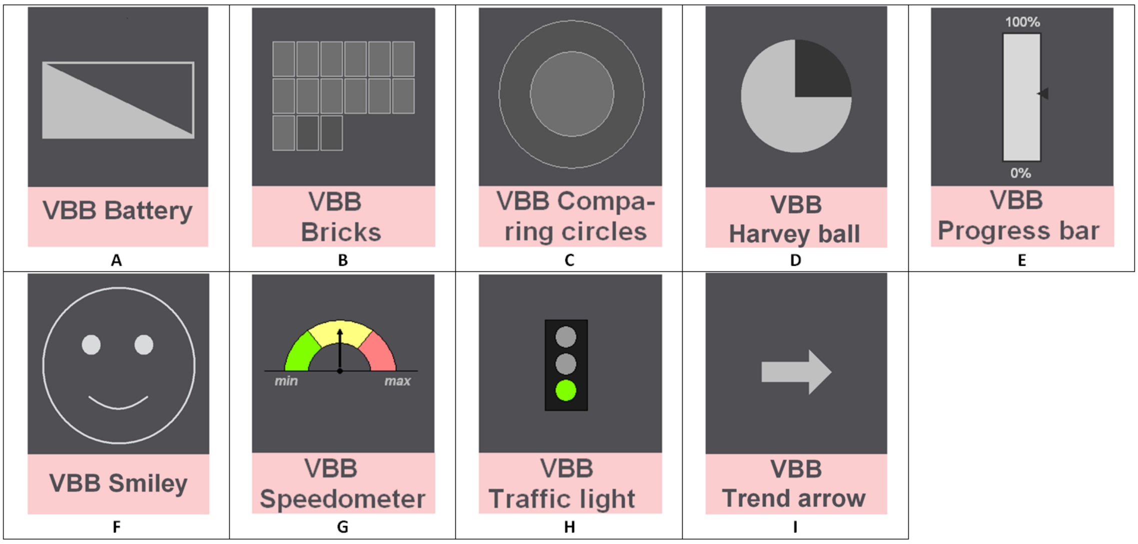


Figure 5.36.: Symbol VBBs: compound dashboard symbols

- If the actual value of a certain KPI is lower/greater than its acceptance value, then full VBB *Battery* is used

Example: Concrete example of VBB *Battery* usage is considered by [La08]. This VBB is applied in combination with VBB Graph and denotes a failure propagation probability of the interface's technical realization. Full battery corresponds to a high failure propagation probability, half-full battery represents an average failure propagation probability, and empty battery depicts a low failure propagation probability.
Type: Symbol VBB candidate
Connections to other VBBs: This VBB may be used in combination with VBB Matrix, VBB Clustering, VBB Graph, VBB Swimlane, VBB 2D-Swimlane and VBB Regionalization.
Visual variables: No additional visual variables are used.

Table 5.9.: VBB Battery

VBB Bricks

Usage: You want to visualize two or more quantitative key measures for the purpose of comparison in such a way that it can be recognized and evaluated at a glance.
Explanation: VBB *Bricks* (see Figure 5.36) visualizes two or more comparable quantitative

key measures by the use of groups of different colored bricks. Thereby, each value is represented by the whole collection of bricks of the corresponding color: the greater the number of bricks, the greater the value of the corresponding key measure. Bricks shown in Figure 5.36 are colored two different colors, and thus depict two different values.
As interval-scaled variables to be visualized by this VBB take any value whereas the number of bricks is limited, scaling should be used. Scale *s* can be calculated by the use of the following formula:

$$s = \frac{v_{max} - v_{min}}{n_{bricks}}$$

where v_{max} is the maximal value certain quantitative variable takes, v_{min} is the minimal value of the variable, and n_{bricks} is a number of bricks of particular color corresponding to this variable. Calculated in this way *s* is a number of key measure units equivalent to one brick of the corresponding color. Scale *s* should be the same for all values within the VBB, otherwise their comparison is not possible.
Example: VBB *Bricks* is used in [Be04] to compare the actual number of incidents with the target value (see Figure 5.37). Collection of green bricks depicts the as-is incidents number, whereas collection of rose-colored ones represents to-be value. In this example the scale 10 incidents per brick is used. That means, that the actual number of incidents is 70 (as 7 green bricks are depicted), whilst target incidents number is 10 (only 1 rose-colored brick). So the actual incidents number is seven times greater then its target.
Type: Symbol VBB candidate
Connections to other VBBs: This VBB may be used in combination with VBB Clustering, VBB Graph, VBB Swimlane, VBB 2D-Swimlane and VBB Regionalization.
Visual variables: Fill color of the bricks is used to differentiate between different key measures values. No additional visual variables are used.

Figure 5.37.: An example of VBB Bricks

VBB Comparing circles

Usage: You want to visualize two comparable interval-scaled key measures being in relation "one is a part of another" in such a way that it can be recognized and evaluated at a glance.
Explanation: VBB *Comparing circles* (see Table 5.10) visualizes two interval-scaled key measures, one of which is a part of another (part attribute and generalization attribute, respectively), by the use of two concentric circles of different colors. Thereby, each value is represented by a circle of the corresponding color: the greater the radius of the circle, the greater the value of the corresponding key measure. Part attribute is represented by the smaller circle, whereas generalization attribute is depicted by the bigger one.

This VBB belongs to continuously scaled type, as part attribute can take any value between zero and value of generalized attribute. Mapping of attribute values on VBB *Comparing circles* representations is defined in the following way:

- If the value of the part attribute is equal to zero, the radius of the inner circle is equal to zero as well, i.e. representation of VBB *Comparing circles* for this extreme case consists only from the outer circle
- If the value of the part attribute is equal to the value of the generalization attribute, then the radius of the inner circle is equal to the radius of the outer circle and VBB *Comparing circles* for this extreme case consists solely from the inner circle
- Representations of VBB *Comparing circles* for remaining values of the part attribute are evenly distributed between aforementioned two extremes

Example: VBB *Comparing circles* is used to compare the value of total planned budget with the value of already consumed budget [Be04], thereby the value of consumed budget forms a part of the whole planned budget, i.e the relation "one is a part of another" between considered two key measures exists.
Type: Symbol VBB candidate
Connections to other VBBs: This VBB may be used in combination with VBB Clustering, VBB Graph, VBB Swimlane, VBB 2D-Swimlane and VBB Regionalization.
Visual variables: Fill color of the circles is used to differentiate between different key measures values, and their size represents the values of key measures. No additional visual variables are used.

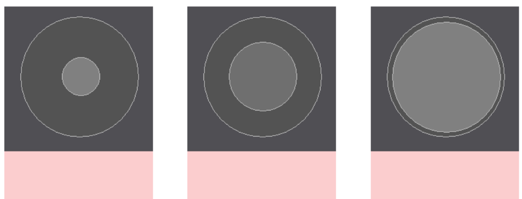


Table 5.10.: VBB Comparing circles: three exemplary values

VBB Harvey balls

Usage: You want to visualize ordinally scaled or interval-scaled key measure relevant for particular objective, business object, business process or technical component in such a way that an overall status can be recognized at a glance.
Explanation: VBBs *Harvey balls* of two types exist: the one with a discrete scale and the one with a continuous scale. Discrete scaled harvey balls take five forms (see Table 5.11) and are usually used to represent ordinally scaled values of key metrics, whereas continuously scaled ones can take any value between completely empty and completely filled circle, and thus visualize interval-scaled values. For both types the amount of filled area corresponds

to the value of key measure. Harvey balls are used to visualize a certain value of just one key measure.
If discrete scaled harvey balls are used, each of five possible harvey balls forms corresponds to a certain value of some key metric. If continuously scaled harvey balls are used, a certain kind of scaling should be performed to define which harvey balls form represents which key metric value. Thereto the minimal value of a certain variable is assigned to completely empty harvey ball, and the maximal value is associated with completely filled one. Representations of values in between are evenly distributed between these two extremes.
Example:
Example 1: Discrete scaled VBB *Harvey balls* is used to represent usability of certain business application. This key measure takes values: very low, low, ordinary, high, very high. Following mapping of the values of usability on VBB *Harvey balls* representations can be performed (see Table 5.11):

- very low usability corresponds to empty harvey ball
- low usability corresponds to a quarter-filled harvey ball
- ordinary usability is represented by half-filled harvey ball
- high usability is represented by three-quarter filled harvey ball
- very high usability is represented by completely filled harvey ball

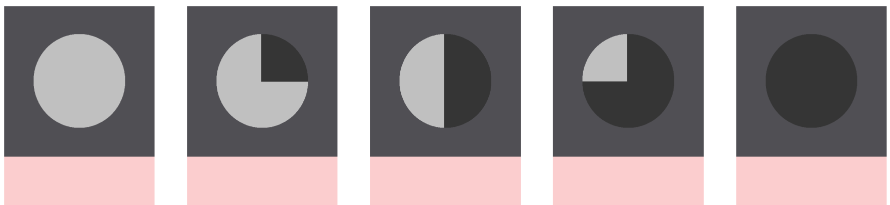


Table 5.11.: Discrete scaled VBB Harvey balls

Example 2: Continuously scaled VBB *Harvey balls* is used to represent degree of match to a standard for a certain infrastructure component. This key indicator can take any value from 0% to 100%. Following mapping of key indicator values on VBB *Harvey balls* representations can be performed (see Table 5.12):

- no match to a standard (0%) is depicted by completely empty harvey ball
- absolute match to a standard (100%) is depicted by completely filled harvey ball
- intermediate values of match to a standard (between 0% and 100%) are represented by partly filled harvey balls with the corresponding amount of filled area

Type: Symbol VBB candidate
Connections to other VBBs: This VBB may be used in combination with VBB Matrix, VBB Clustering, VBB Graph, VBB Swimlane, VBB 2D-Swimlane and VBB Regionalization.
Visual variables: Size of filled sector of a circle is used to differentiate between different key measure values. No additional visual variables are used.

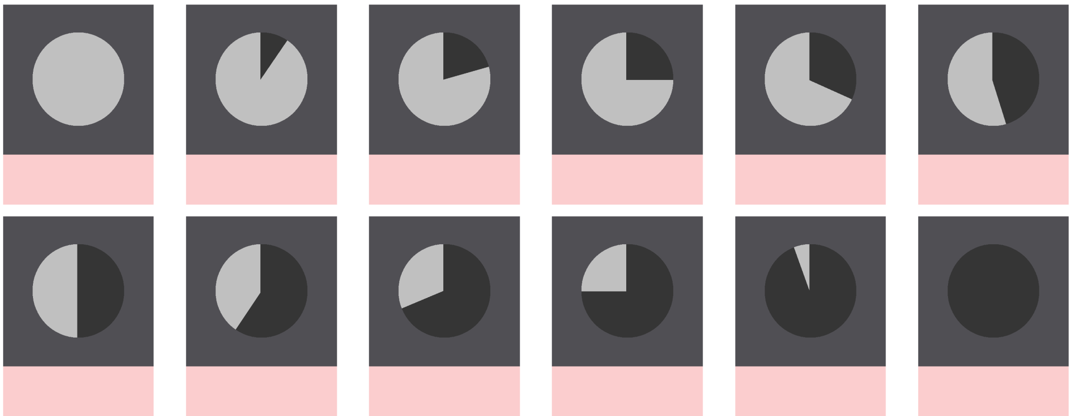


Table 5.12.: Continuously scaled VBB Harvey balls: exemplarily values

VBB Progress bar

Usage: You want to visualize degree to which a particular item meets a particular criterion in such a way that not only an overall status can be recognized at a glance, but also a concrete key measure value is represented.
Explanation: VBB *Progress bar* belongs to continuously scaled type. Visually a certain value of a certain key measure is represented by arrow position on progress bar scale between its maximum and minimum (see Table 5.13). Maximum and minimum are often defined as 100% and 0%, respectively. These are also extreme values of degree variable, which is to be depicted by progress bar.
Example: VBB *Progress bar* is used to represent the progress of particular project (degree of projects readiness).
Type: Symbol VBB candidate
Connections to other VBBs: This VBB may be used in combination with VBB Matrix, VBB Clustering, VBB Graph, VBB Swimlane, VBB 2D-Swimlane and VBB Regionalization.
Visual variables: Arrow y-coordinate represents the value of key measure. No additional visual variables are used.

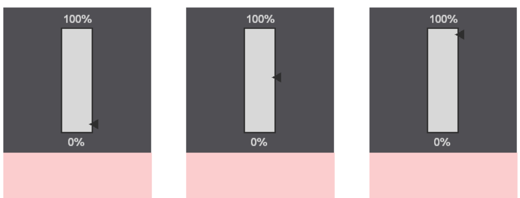


Table 5.13.: VBB Progress bar: three exemplary values

VBB Smiley

Usage: You want to visualize key measure relevant for particular objective, business object, business process or technical component in such a way that an overall status can be recognized at a glance.
Explanation: VBB *Smiley* belongs to discrete scaled type, takes only three different values (see Table 5.14), which can be interpreted as sad, indifferent and happy smileys, and is used to visualize ordinally scaled values. To be able to use this VBB to visualize interval-scaled values, it is necessary to perform the following transformation. Usually, two values for quantitative key measure are defined: target level and acceptance level. The usage of three different forms of VBB Smiley is defined by the following rules:

- If the actual value of a certain KPI is greater/lower than its target value, then happy VBB *Smiley* is used
- If the actual value of a certain KPI is lower/greater than its target value, but is greater/lower than its acceptance value, then indifferent VBB *Smiley* is used
- If the actual value of a certain KPI is lower/greater than its acceptance value, then sad VBB *Smiley* is used

Example: VBB *Smiley* is used to visualize the status of business applications availability on some cluster map. Acceptance value for availability is 95% and its target value is 98%. As availability variable can take any value, whereas VBB *Smiley* can take only three different values, the following rules are defined in a legend:

- if $availability \geq 98\%$, then happy smiley is used to show that availability of the corresponding application is above the target level and no actions for improvement should be done
- if $95\% \leq availability \leq 98\%$, then indifferent smiley is used to give notice that, however, availability value is above the acceptance level, it is still below the target level and should be managed to be improved
- if $availability \leq 95\%$, then sad smiley is used to warn that availability of an application is below the acceptance level and must be improved immediately

Thus, each application on the cluster map is additionally marked with VBB *Smiley* in one of its three manifestations depending on the values taken by availability attribute of the corresponding application.
Type: Symbol VBB candidate
Connections to other VBBs: This VBB may be used in combination with VBB Matrix, VBB Clustering, VBB Graph, VBB Swimlane, VBB 2D-Swimlane and VBB Regionalization.
Visual variables: Sometimes, such visual variables as line color and fill color are additionally applied to VBB *Smiley*. These visual variables do not add any additional information and are solely used to reinforce the visual expressiveness of this VBB and to make the recognition of the symbol even more intuitive and prompt.

Table 5.14.: VBB Smiley

VBB Speedometer

Usage: You want to visualize interval-scaled key measure relevant for particular objective, business object, business process or technical component in such a way that not only the overall status can be recognized at a glance, but also a concrete key measure value is represented.
Explanation: VBB *Speedometer* belongs to continuously scaled type. Visually a certain value of a certain key measure is represented by arrow position on speedometer scale between its maximum and minimum (see Table 5.15). The scale of VBB *Speedometer* is divided in three segments of different colors (usually green, yellow and red). All values taken by this VBB (and, of course, also the values taken by the corresponding key measure), are thus categorized in three groups depending on the color of the scale segment, in which the arrow is positioned. Categorization of the scale is performed on the basis of two values defined for the variable, namely target level and acceptance level, in the following way:

- Scale segment between the minimum/maximum and the acceptance level is colored in red
- Scale segment between the acceptance level and the target level is colored in yellow
- Scale segment between the target level and maximum/minimum is colored in green

Extreme values of the key measure are assigned to the extreme positions of speedometer arrow, and representations of the values in between are evenly distributed between these two extreme arrow positions. VBB *Speedometer* is used to visualize a certain value for just one key measure.
Example: VBB *Speedometer* is used to represent response time of a certain application. This key indicator can take any value: theoretically from 0s to infinity. However, it is not reasonable to place all these possible values on speedometer scale, even if a certain big number is taken instead of infinity for maximal value of the measure, because real values usually taken by response time variable are near to 0, and thus can not be seen on such a large scale. It would also contradict to reality, where nobody will wait forever waiting for a system to reply. In this example it is recommended to define the maximum for response time artificially and assume, that all values taken by response time variable, which are greater than this value, are also visualized with the arrow positioned on this maximum. Artificially defined maximum value should be selected in accordance to target and acceptance levels, so that sizes of the scale categories are comparable with each other. Otherwise it is difficult to differentiate between visualizations of different values. Target value and acceptance level are defined for response time as 0,1s and 5s, respectively. Thus, the maximum

for response time scale can be defined as 10s. On the basis of minimum, maximum, target and acceptance values, scale categorization is performed as follows:

- Scale segment $(0s; 0,1s)$ is colored in green, which means that the values from this interval have a low response time so that the user feel that the system is reacting instantaneously
- Scale segment $(0,1s; 5s)$ is colored in yellow, which means that although the user notices the delay, the limit for the user's flow of thought is not exceeded
- Scale segment $(5s; 10s)$ (which in reality means a response time greater than 5s) is colored in red. It means that the limit for keeping user's attention is exceeded and he will probably want to perform other tasks rather than wait for the computer to finish.

Type: Symbol VBB candidate
Connections to other VBBs: This VBB may be used in combination with VBB Matrix, VBB Clustering, VBB Graph, VBB Swimlane, VBB 2D-Swimlane and VBB Regionalization.
Visual variables: Fill color is used to perform speedometer scale categorization, and arrow orientation represents the value of the key measure. No additional visual variables are used.

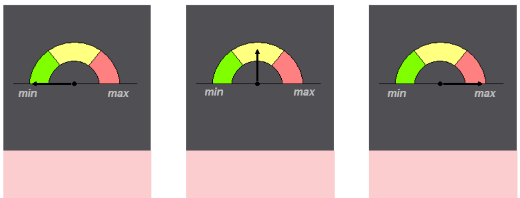


Table 5.15.: VBB Speedometer: three exemplary values

VBB Traffic light

Usage: You want to visualize key measure relevant for particular objective, business object, business process or technical component in such a way that an overall status can be recognized at a glance.
Explanation: VBB *Traffic light* belongs to discrete scaled type, takes only three different values: red, yellow and green (see Table 5.16), and is used to visualize ordinally scaled values. To be able to use this VBB to visualize interval-scaled values, it is necessary to perform the following transformation. Usually, two values for quantitative key measure are defined: target level and acceptance level. The usage of three different forms of VBB *Traffic light* is defined by the following rules:

- If the actual value of a certain KPI is greater/lower than its target value, then green VBB *Traffic light* is used
- If the actual value of a certain KPI is lower/greater than its target value, but is greater/lower than its acceptance value, then yellow VBB *Traffic light* is used

- If the actual value of a certain KPI is lower/greater than its acceptance value, then red VBB *Traffic light* is used

Example: VBB *Traffic light* is used to visualize an overall staff workload rate for a certain organization, i.e. the relation of number of actual working days of employees to the total number of working days in a year. Acceptance value for staff workload rate is 77% and its target value is 82%. As workload rate variable can take any value, whereas VBB *Traffic light* can take only three different values, the following rules are defined in a legend:

- if $workload \geq 82\%$, then green traffic light is used to show that the staff workload is above the target level and no actions for improvement should be done
- if $77\% \leq workload \leq 82\%$, then yellow traffic light is used to give notice that even though staff workload rate is above the acceptance level, it is still below the target level and should be managed to be improved
- if $workload \leq 77\%$, then red traffic light is used to warn that staff workload rate in organization is below the acceptance level and must be improved immediately

Type: Symbol VBB candidate
Connections to other VBBs: This VBB may be used in combination with VBB Matrix, VBB Clustering, VBB Graph, VBB Swimlane, VBB 2D-Swimlane and VBB Regionalization.
Visual variables: No additional visual variables are used.

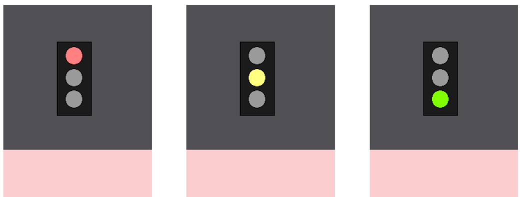


Table 5.16.: VBB Traffic light

VBB Trend arrow

Usage: You want to visualize trend for particular key measure in such a way that it can be recognized at a glance.
Explanation: VBB *Trend arrow* belongs to discrete scaled type and takes three different values: head-up, horizontal and head-down (see Table 5.17). This VBB visualizes the trend value, calculated on the basis of a key measure value, registered in previous timeframe (or several earlier timeframes) and its current value. If the trend value should be calculated only for two timeframes (the current and the previous ones), then the trend value *t* is calculated as follows:

$$t = (\frac{v_{current}}{v_{previous}} - 1) * 100\%$$

where $v_{current}$ designates the value of the key measure in the current timeframe and $v_{current}$

represents the value from the previous one.
The usage of three different forms of VBB *Trend arrow* is defined then by the following rules:

- If $t \geq 0$, then up directed arrow is used
- If $t = 0$, then horizontal arrow is used
- If $t \leq 0$, then down directed arrow is used

If the trend value should be calculated for multiple timeframes (e.g. to show an overall trend from the beginning of the year or since several years) linear regression may be used to construct a regression line. In this case representations of VBB *Trend arrow* are defined in the following way:

- If angle α between regression line and positive x-axis is acute, then up directed arrow is used
- If regression line is parallel to the x-axis, then horizontal arrow is used
- If angle α between regression line and positive x-axis is obtuse, then up directed arrow is used

VBB *Trend arrow* is used to visualize a trend for just one key measure.
Example: VBB *Trend axis* is used to visualize the trend for project costs key measure. The actual costs of projects sum up to 150 million euro, and project costs in previous month amount to 100 million euro.

$$t = (\frac{150}{100} - 1) * 100\% = 50\%$$

As $t \geq 0$, directed up arrow is used to show that the trend is positive in comparison with the previous month in a sense that project costs increased since last month. VBB *Trend* do not give any estimations of the actual status and only shows the trend of the key measure values. Project costs status in this example got rather worse in a result of costs increase.
Type: Symbol VBB candidate
Connections to other VBBs: This VBB may be used in combination with VBB Matrix, VBB Clustering, VBB Graph, VBB Swimlane, VBB 2D-Swimlane and VBB Regionalization.
Visual variables: Arrow orientation represents the trend value of a certain key measure. Fill color may be additionally used to depict the status of current key measure value, e.g. green in case the target value is achieved, yellow if current value is between the target and the acceptance levels, and red in case it is below the acceptance level.

5.3.5. Symbol VBBs optimization recommendations

After introducing the groups of symbol viewpoints building blocks and new VBB candidates in previous subsections, this subsection briefly reviews symbol VBBs already included in BEAMS collection and makes proposal on their reorganization and integration

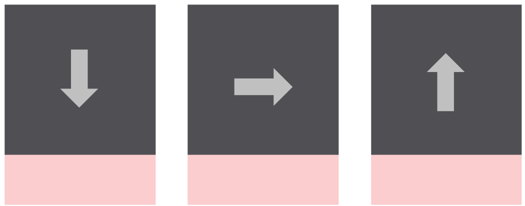


Table 5.17.: VBB Trend arrow

in symbol VBBs groups introduced above.
As already listed in Subsection 2.3.2, currently 4 symbol VBBs are available in BEAMS collection, namely:

- VBB Bar chart
- VBB Ellipse
- VBB Pie chart
- VBB Rectangle

VBB Bar chart and VBB Pie chart together with their special cases form the group of compound chart symbols, whereas VBB Ellipse is a representative of the group of shapes. VBB Rectangle is considered as a special case of VBB Polygon.

6. Summary and Outlook

This chapter summarizes the findings of this work and gives an outlook on future research possibilities that can build on the results of this thesis.

6.1. Summary

In this work reorganization and extension of BEAMS viewpoint building blocks framework was suggested based on the Viewpoint Catalog collected from theory and practice in the field of EA management.

The first chapter of this thesis provides an introduction to the topic of EA management, explains its significance in the context of contemporary rapid development and increasing role of information technology in business, and highlights importance of visual aspect within EA management function realization. On the other hand, Chapter 1 reveals the problems of plethora of existing literature and approaches in the field as well as lack of uniform terminology, the modern research in the area of EA management has to deal with, and introduces best-practice BEAMS approach, recently developed on *sebis* chair of the Technische Universität München in order to address these research problems. Finally, Chapter 1 formulates the goal of the thesis, which consists in applying and evolving BEAMS viewpoint buildings blocks framework, and outlines the structure of the thesis.

Chapter 2 introduces theoretical foundations of the work by considering the topic of EA management in Section 2.1 and providing terminological basis of this thesis according to the definitions of ISO standard in Section 2.2, and discusses two best-practice approaches for visualizations in the area of EA management in Section 2.3, namely Enterprise Architecture Management Pattern Catalog (EAMPC) and Building Blocks for Enterprise Architecture Management Solutions (BEAMS). The section briefly describes motivations of these approaches, their structure and use, with particular focus on visualization aspect, and concludes with an overview of viewpoint building blocks available in BEAMS collection as well as a detalization of layout fields used for their description.

Chapter 3 provides an overview on viewpoints collected from EA management related areas such as business intelligence, which goes into practice areas of EA management, data mining, that provides technical solutions for EA management related scope, urban planning, often used as a metaphor for it, as well as from classic literature on data visualization.

Chapter 4 is concerned with Viewpoint Catalog construction, collecting viewpoints from theory and practice. Section 4.1 is focused on the theoretical part of the Viewpoint Catalog, which was constituted based on results of explorative literature analysis in the field of EA management, and gives an overview on the literature that deals with the problem area, and thus was used for the construction of the catalog. Section 4.2 describes the conducted case study for the practical part of the Viewpoint Catalog and introduces involved enterprise. Viewpoints from both theoretical and practical parts of the Viewpoint Catalog are

decomposed in Section 4.3 into nonintersecting modular components, so called viewpoint catalog building blocks (VCBB). The section explains the principles used in the course of decomposition process and illustrates this process on concrete examples. Complete list of decomposed viewpoints, together with the corresponding VCBBs, may be found in Appendix A for EA management viewpoints from theory and Appendix B for viewpoints obtained in the course of case study by the real-life global financial service provider.
Based on the outcomes from decomposition process, Chapter 5 analyzes in Section 5.1 BEAMS VBBs participation in this process and states that BEAMS VBBs collection is included in the set of VCBBs. Sections 5.2 and 5.3 use VCBBs distinct from VBBs as a basis for extension of structural and symbol BEAMS VBBs, respectively, and describe new BEAMS VBBs candidates in detail based on layout of BEAMS viewpoint building blocks. Reorganized and extended BEAMS VBBs collection is shown in Figure 6.1.

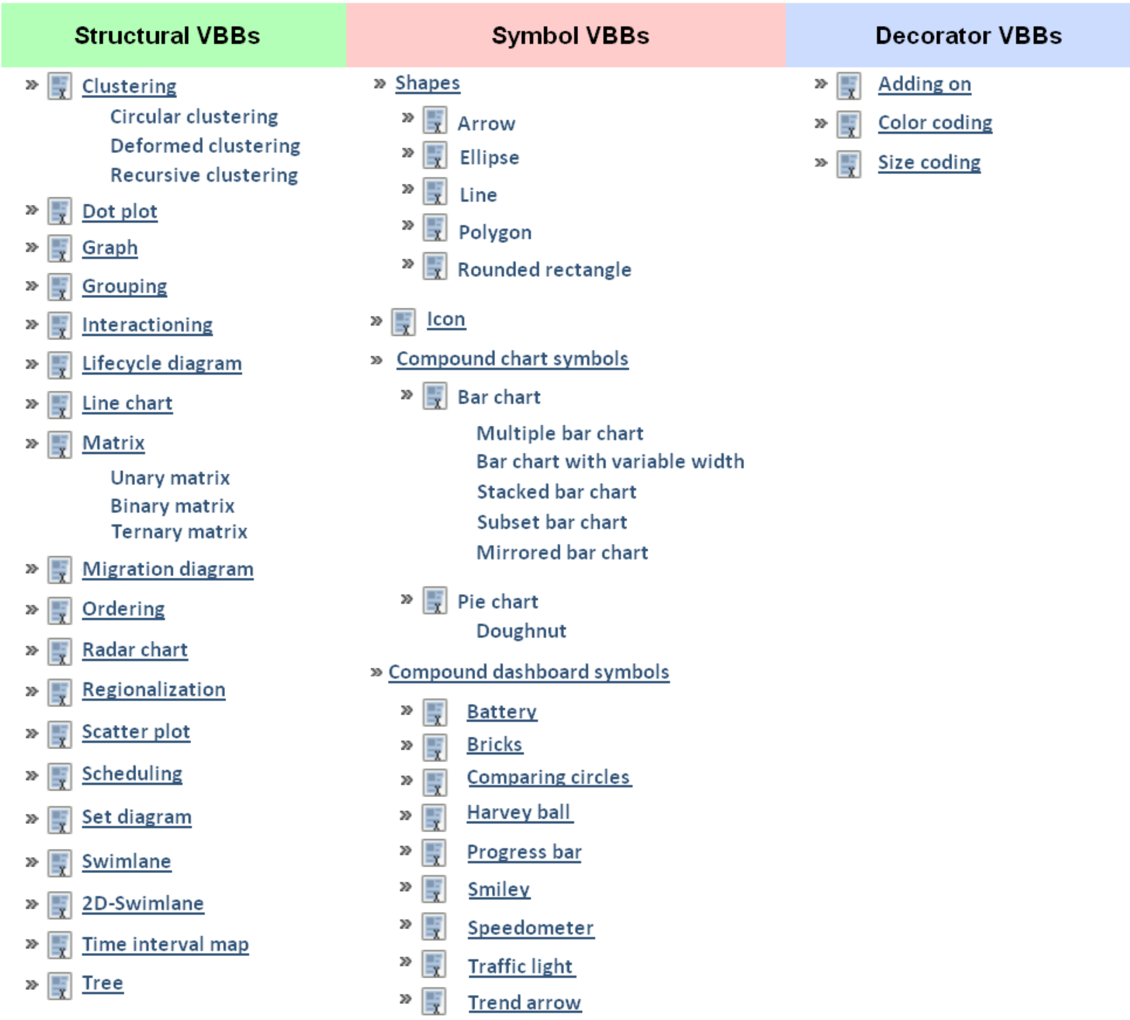


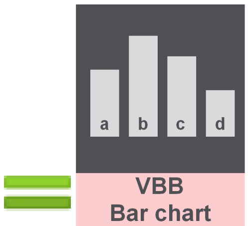
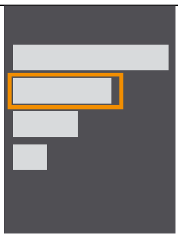
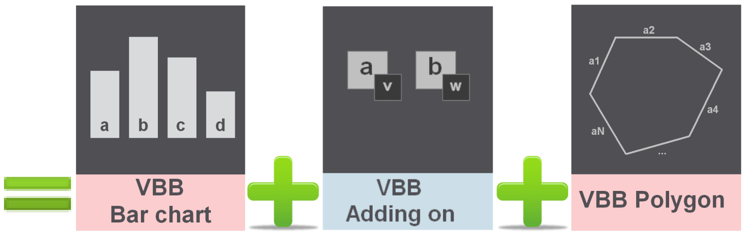
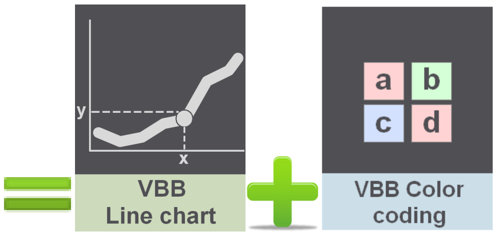
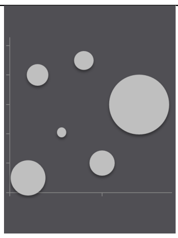
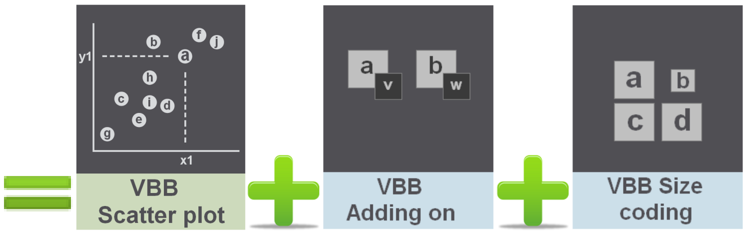
Figure 6.1.: BEAMS VBBs collection reorganized and extended with new viewpoint building block candidates

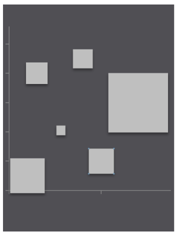
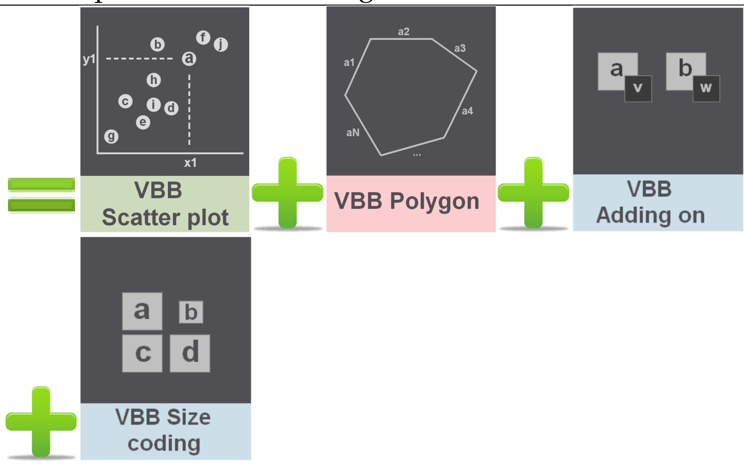
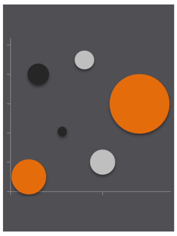
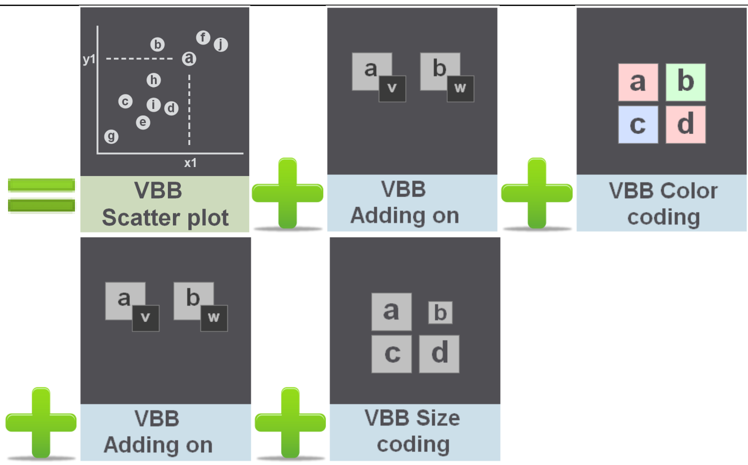
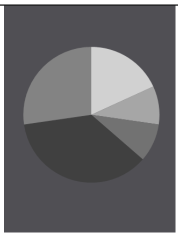
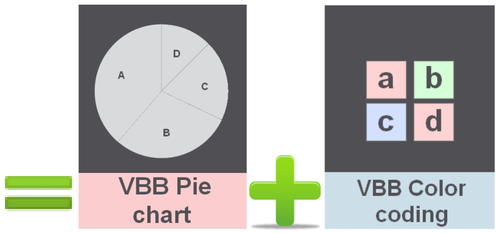
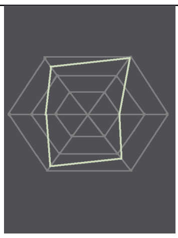
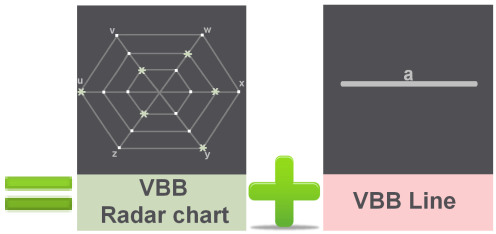
6.2. Outlook

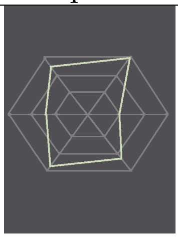
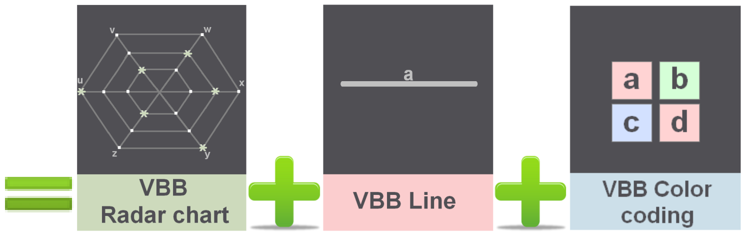
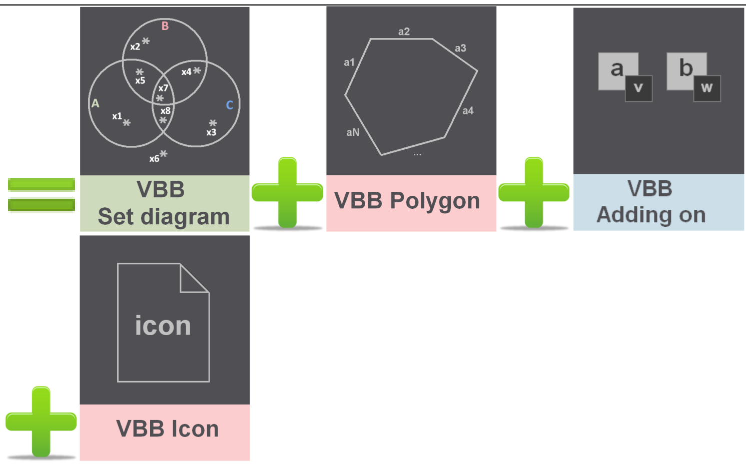
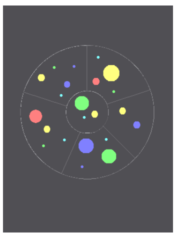
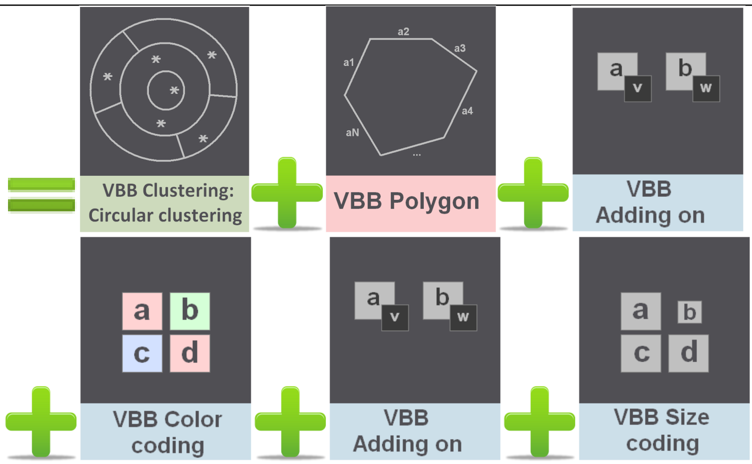
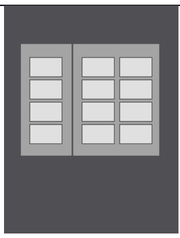
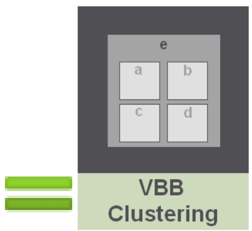
This thesis provides the first step in the direction of BEAMS VBBs evaluation and evolvement by ascertaining viewpoint building blocks candidates from theory and practice observed in the field of EA management. However, practical significance of each building block candidate should be initially proven, as it should be observed at least three times before it can be included in BEAMS best-practice collection. Thus, further observations of new candidates have to be made with the prospect to include them into BEAMS.
On the other hand, viewpoints for Viewpoint Catalog construction, further used as a basis for BEAMS VBBs reorganization and extension, were collected from a limited number of sources: the outcomes of the case study performed in just one organization, as well as solely the findings from selected literature sources on EA management, were used to develop BEAMS knowledge base. Thus, the further prospect of this thesis is to broaden search possibilities for building block candidates identification.
Additionally, evaluation and extension of BEAMS building blocks is not a one-time-only initiative, but the recurrent one according to BEAMS nature of best-practice solutions. Rapid development of information systems, as well as permanent increase of IT complexity in enterprise, foster visualization techniques development used to support EA management function realization by providing transparency and enhancing communication and collaboration between different stakeholders. Thus, it is recommended to review viewpoints from theory and practice periodically in order to complement BEAMS collection with up-to-date best-practice solutions, and in this way to maintain it at the state of the art level.

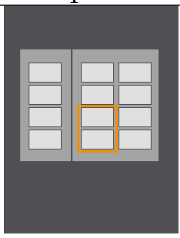
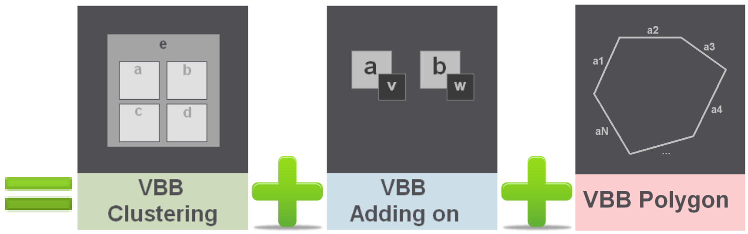
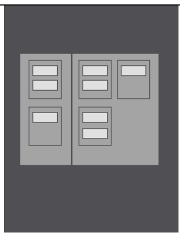
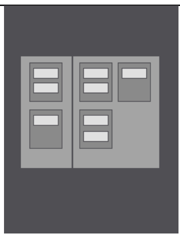
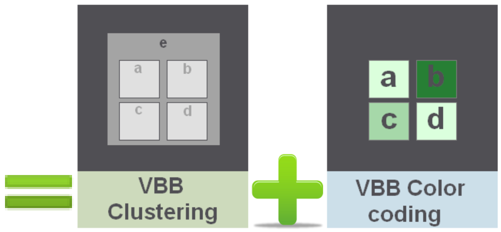
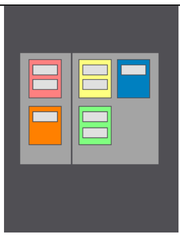
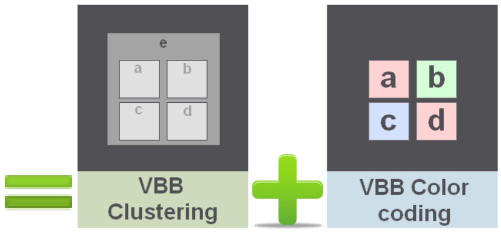
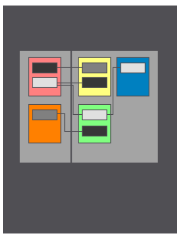
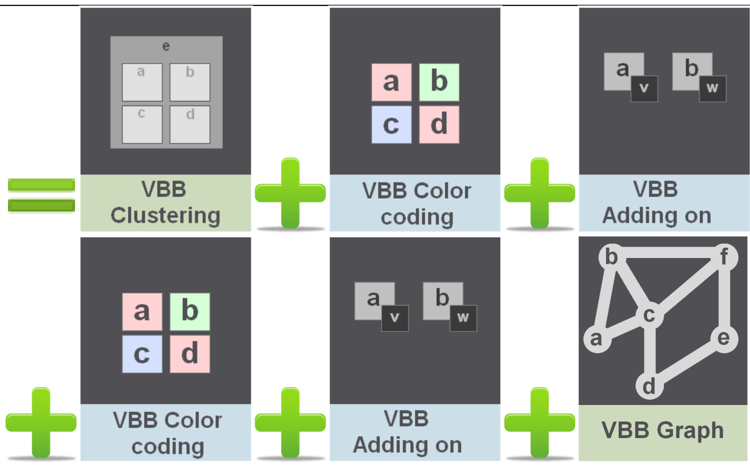
Appendix

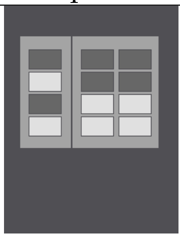
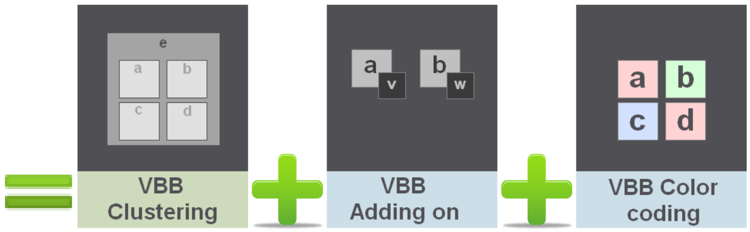
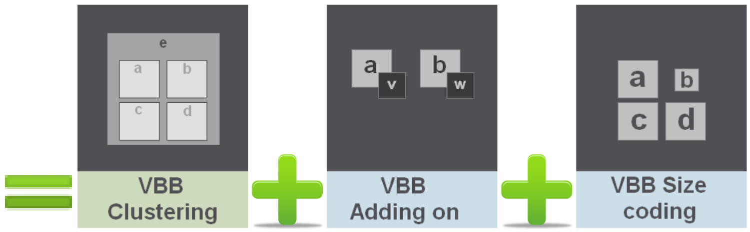
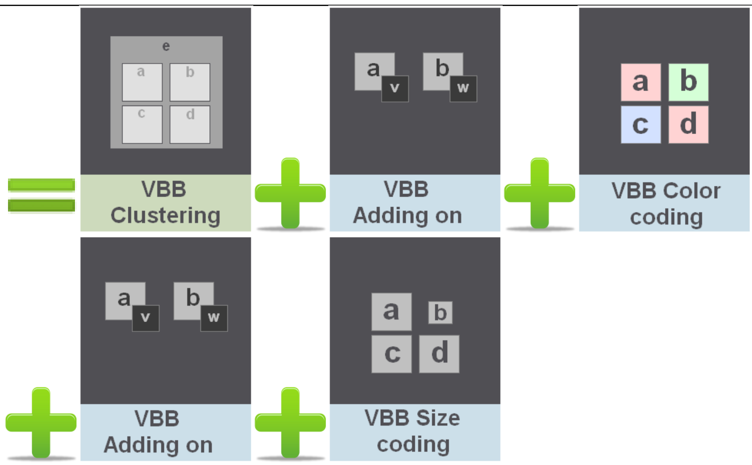
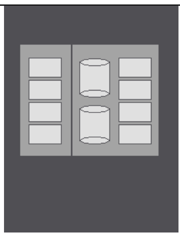
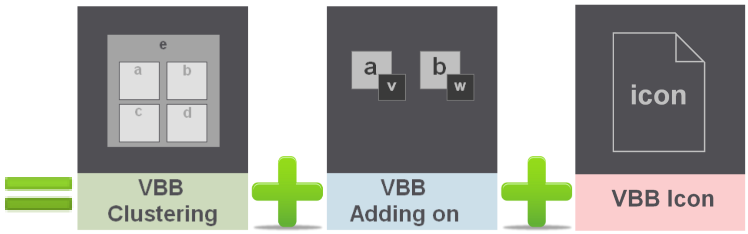
A. Decomposition of viewpoints found in EA management literature

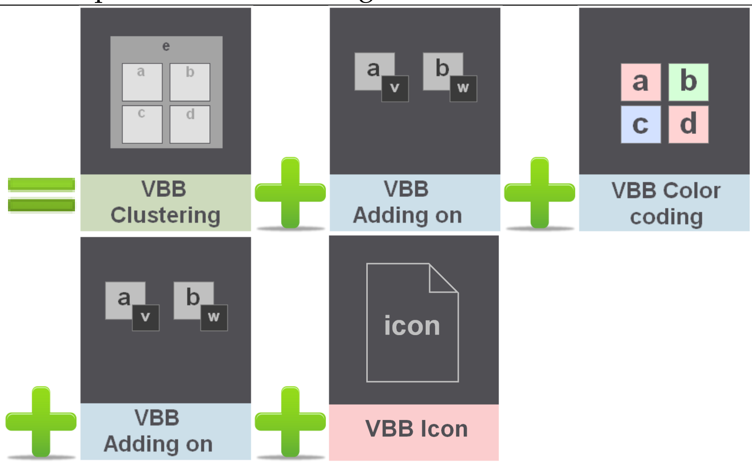
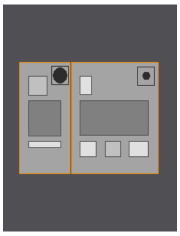
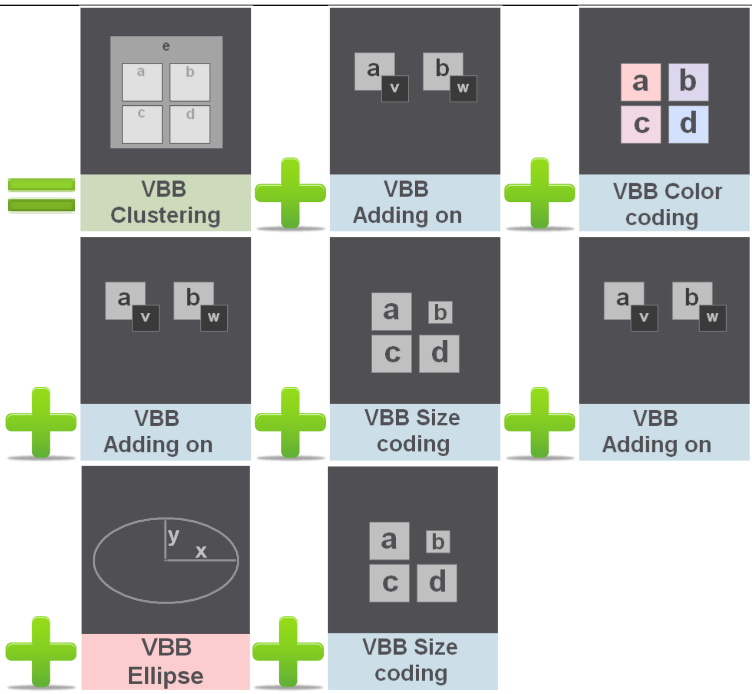
	Viewpoint	Decomposition into building blocks	VBB candidates
1	 [Bu08, Ma08]		
2	 [Bu08]		Symbol VBBs → Shapes → VBB Polygon
3	 [La08]		Structural VBBs → VBB Line chart
4	 [Ma08]		Structural VBBs → VBB Scatter plot

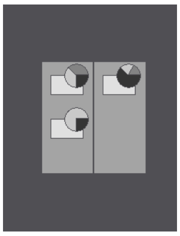
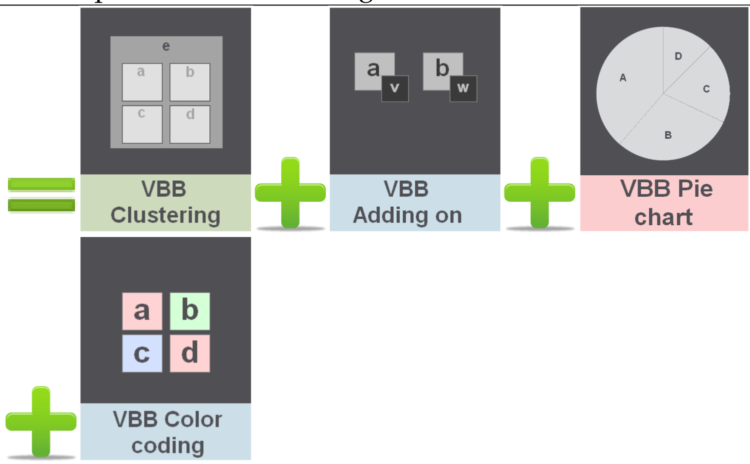
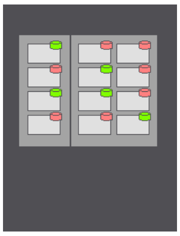
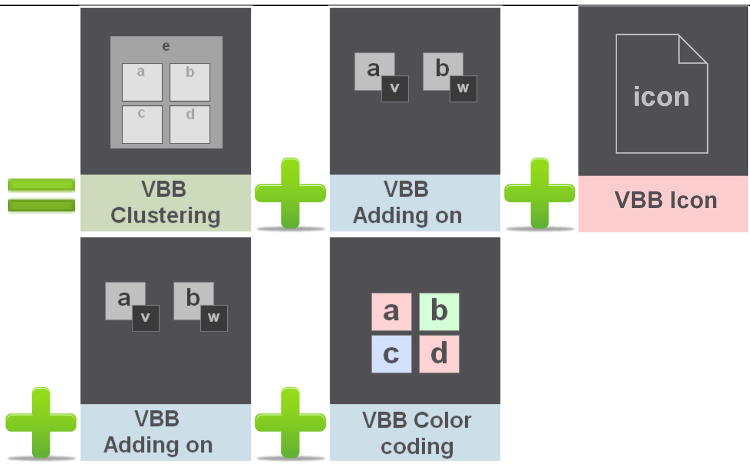
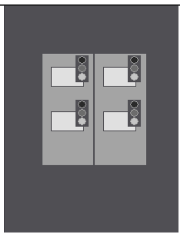
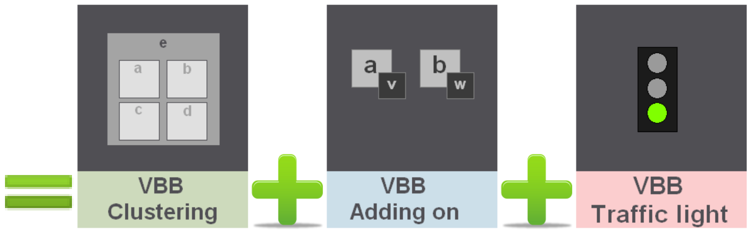
	Viewpoint	Decomposition into building blocks	VBB candidates
5	 [Ma08]		1.Structural VBBs → VBB Scatter plot 2.Symbol VBBs → Shapes → VBB Polygon
6	 [Bu08, Ma08, Wi07]		Structural VBBs → VBB Scatter plot
7	 [Be04, Ma08]		
8	 [La08, Ma08]		1.Structural VBBs → VBB Radar chart 2.Symbol VBBs → Shapes → VBB Line

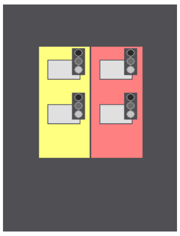
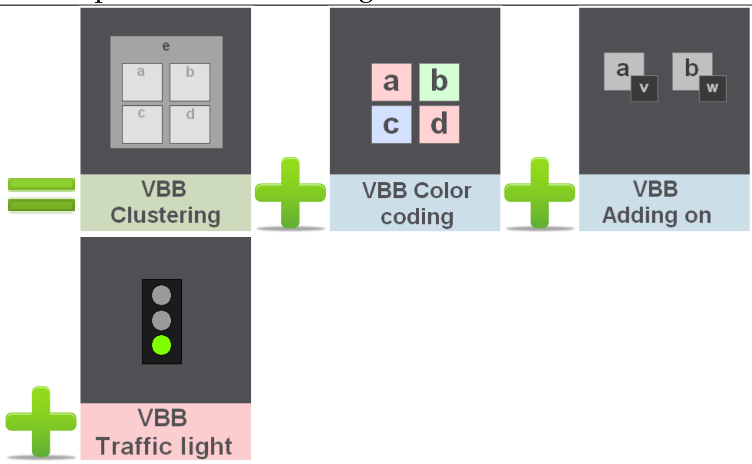
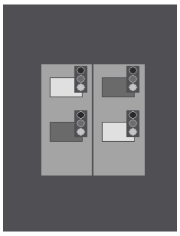
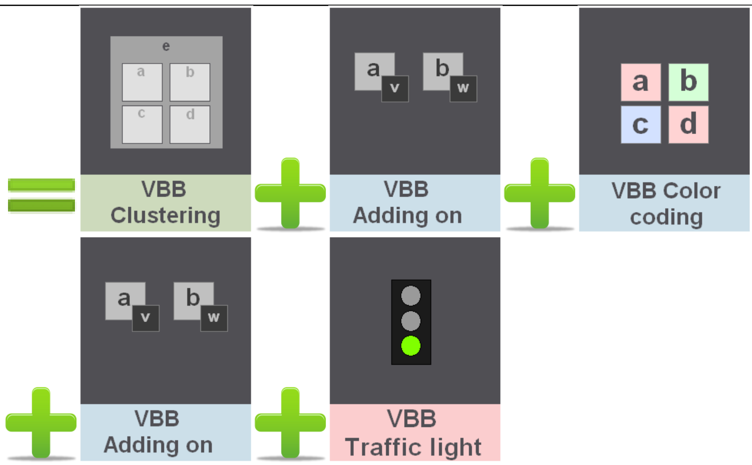
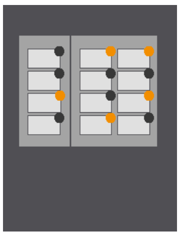
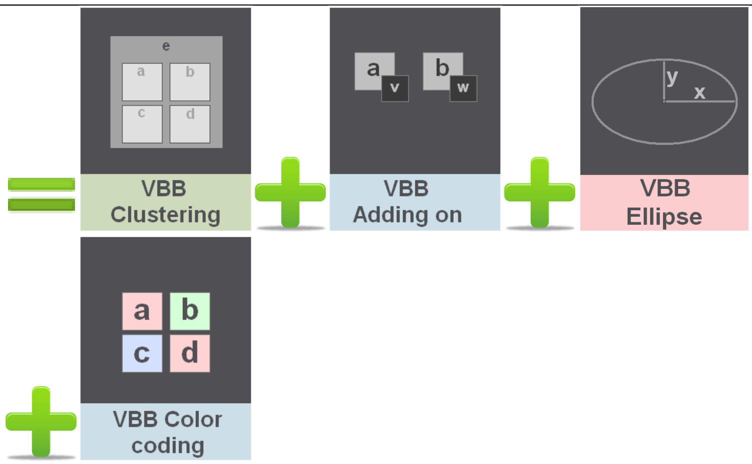
	Viewpoint	Decomposition into building blocks	VBB candidates
9	 [Wi07]		1.Structural VBBs → VBB Radar chart 2.Symbol VBBs → Shapes → VBB Line
10	 [La08]		1.Structural VBBs → VBB Set diagram 2.Symbol VBBs → Shapes → VBB Polygon 3.Symbol VBBs → VBB Icon
11	 [Se05]		1.Structural VBBs → VBB Clustering: special case 2.Symbol VBBs → Shapes → VBB Polygon
12	 [Bu08, Ma08, Wi07]		

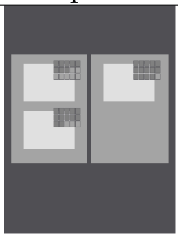
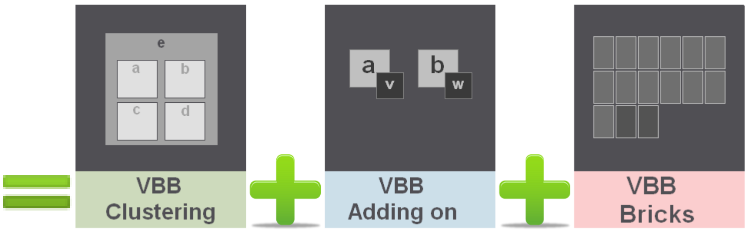
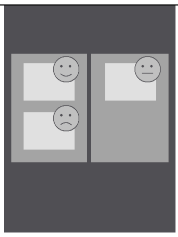
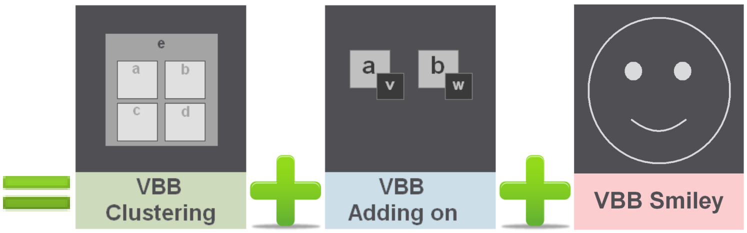
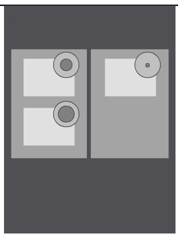
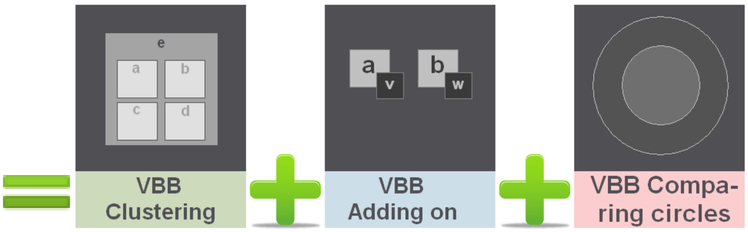
	Viewpoint	Decomposition into building blocks	VBB candidates
13	 [Wi07]		Symbol VBBs → Symbol VBBs → Shapes → VBB Polygon
14	 [Wi07]		
15	 [La08]		
16	 [Be04, Se05, Wi07]		
17	 [Wi07]		Structural VBBs → VBB Graph

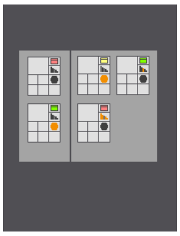
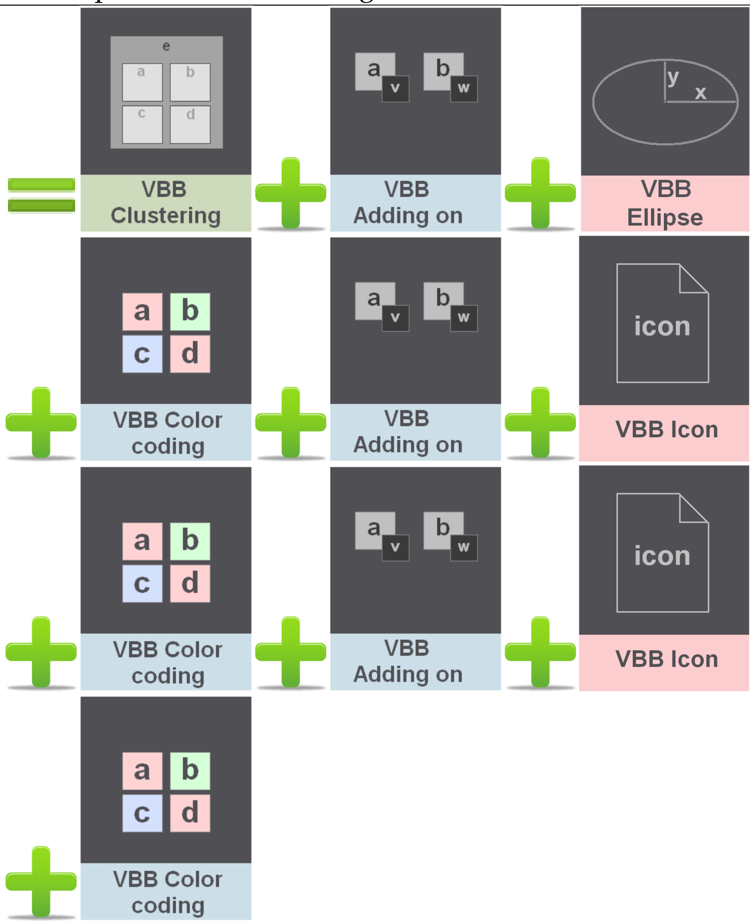
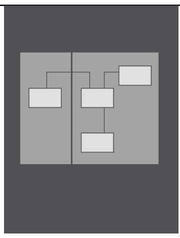
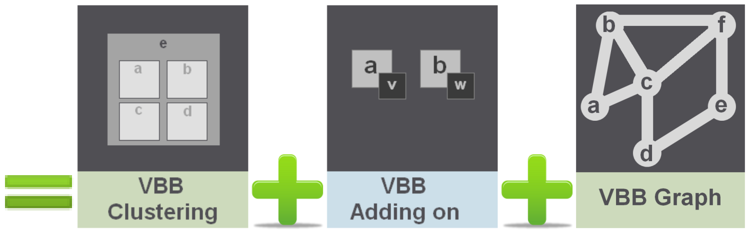
	Viewpoint	Decomposition into building blocks	VBB candidates
18	 [Bu08, La08, Ma08, Wi07]		
19	 [La08, Wi07]		
20	 [La08, Ma08]		
21	 [Se05]		Symbol VBBs → VBB Icon → Data base

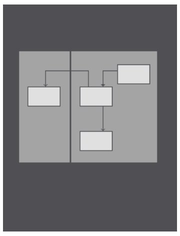
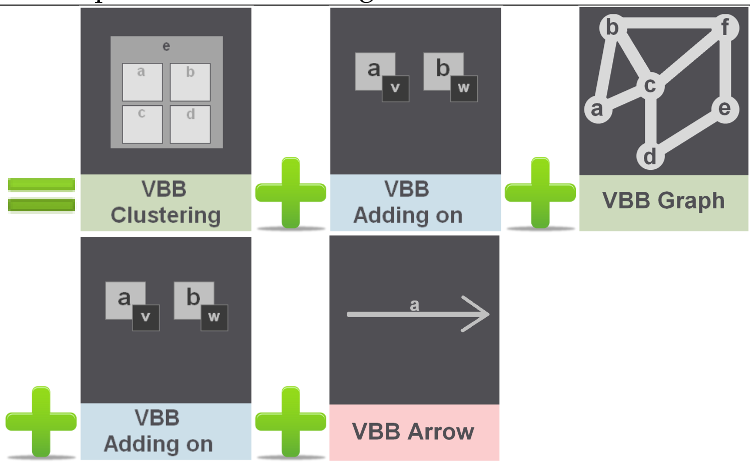
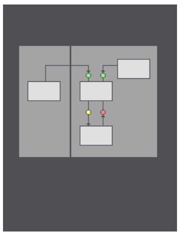
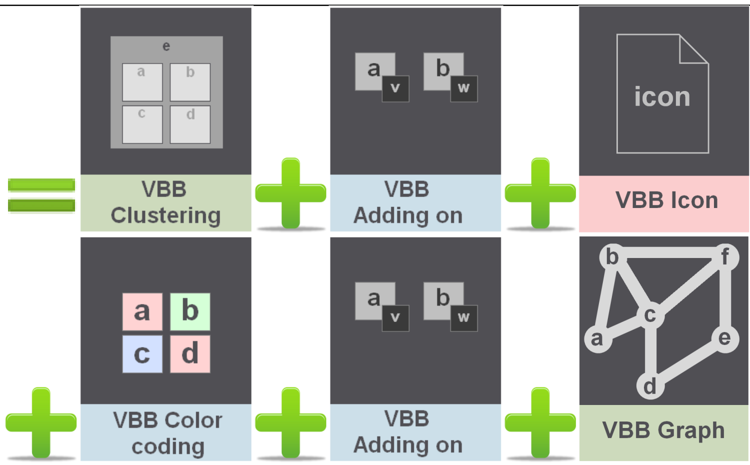
	Viewpoint	Decomposition into building blocks	VBB candidates
22	 [Bu08, La08]		Symbol VBBs → VBB Icon → Tick
23	 [La08]		

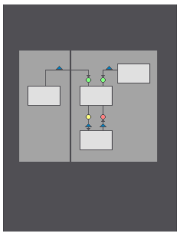
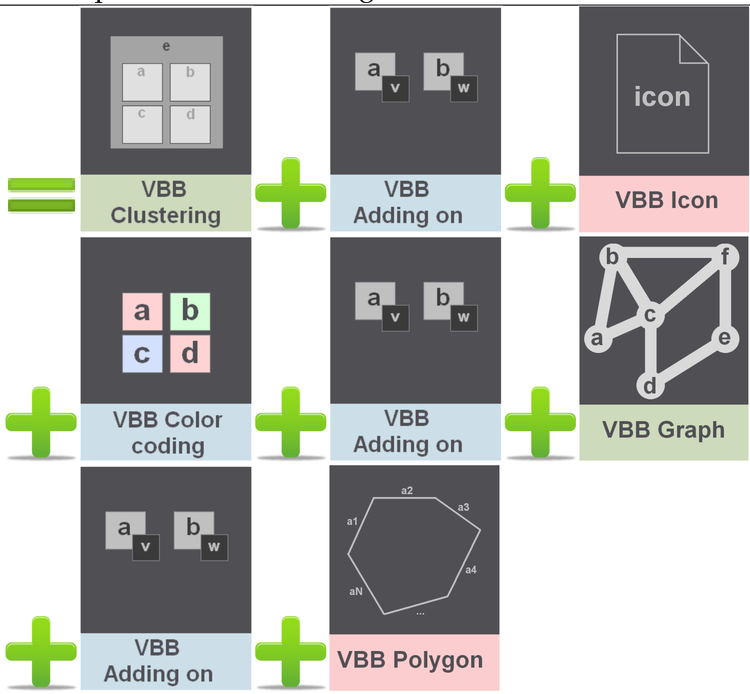
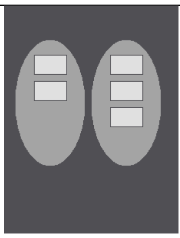
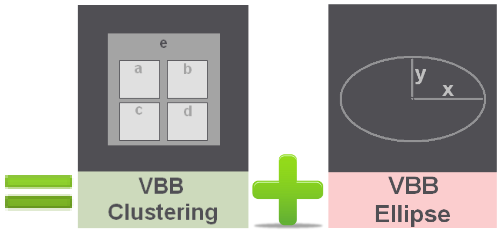
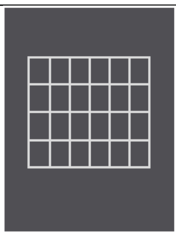
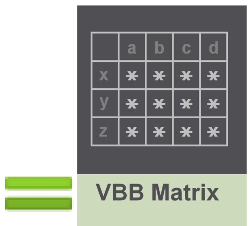
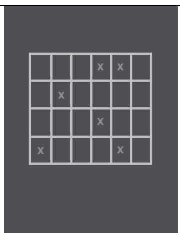
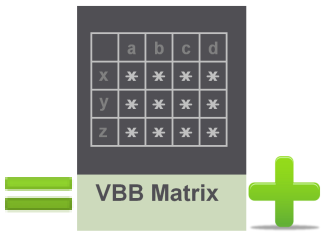
<table>
<tr><th></th><th>Viewpoint</th><th>Decomposition into building blocks</th><th>VBB candidates</th></tr>
<tr><td>24</td><td>
[Wi07]</td><td>
</td><td></td></tr>
<tr><td>25</td><td>
[Bu08, Ma08]</td><td>
</td><td>Symbol VBBs → VBB Icon → Data base</td></tr>
<tr><td>26</td><td>
[Be04, La08, Ma08, Wi07]</td><td>
</td><td>Symbol VBBs → Compound dashboard symbols → VBB Traffic light</td></tr>
</table>

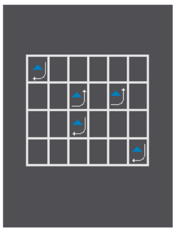
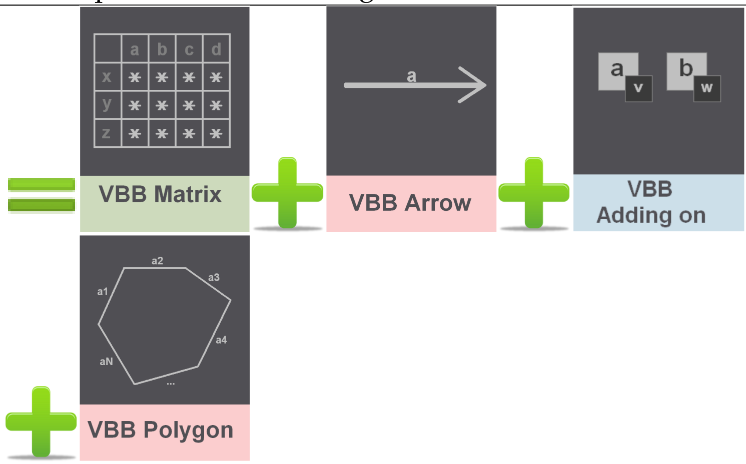
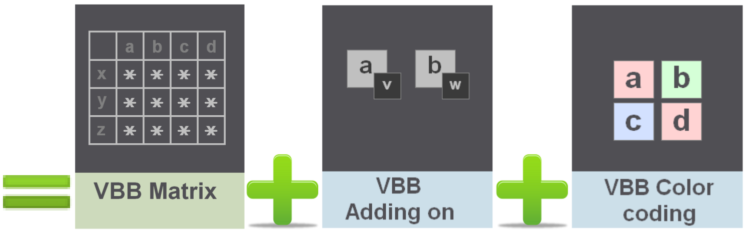
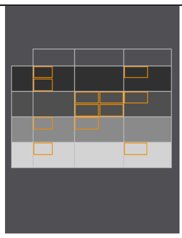
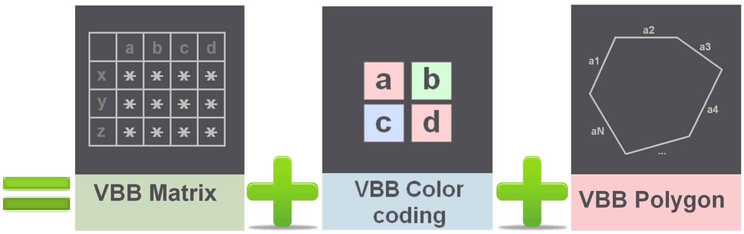
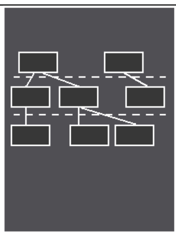
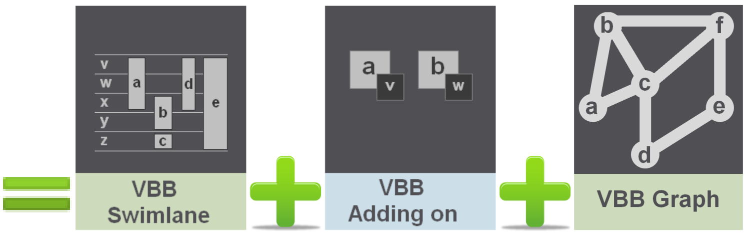
	Viewpoint	Decomposition into building blocks	VBB candidates
27	 [Se05]		Symbol VBBs → Compound dashboard symbols → VBB Traffic light
28	 [Ma08, Wi07]		Symbol VBBs → Compound dashboard symbols → VBB Traffic light
29	 [Ma08]		

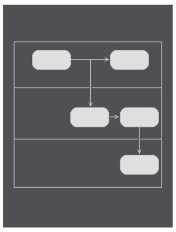
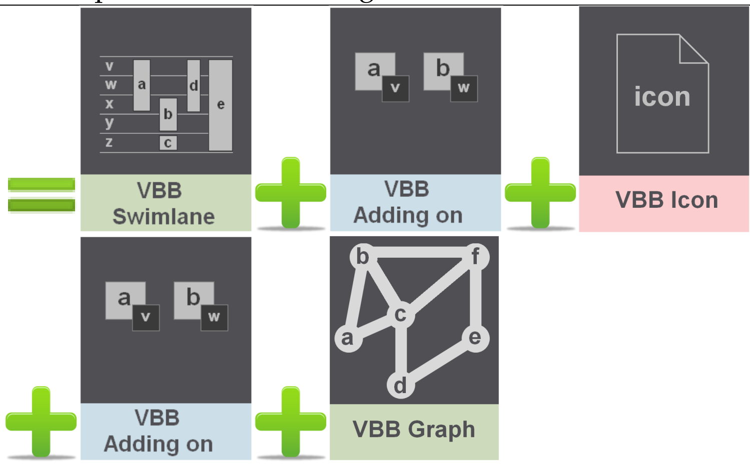
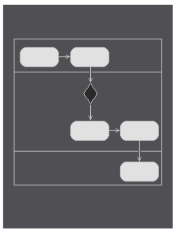
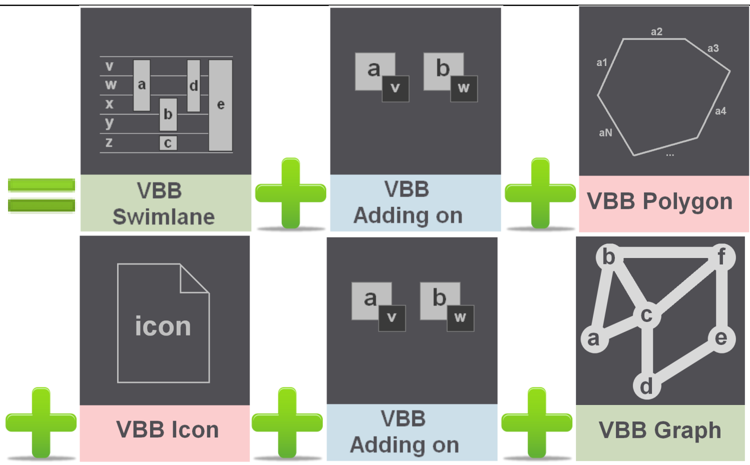
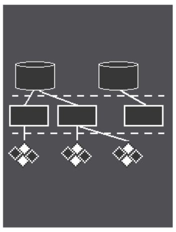
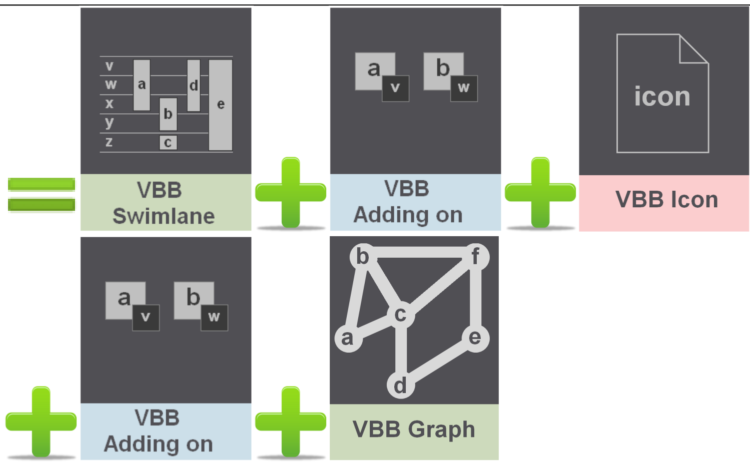
	Viewpoint	Decomposition into building blocks	VBB candidates
30	 [Be04]		Symbol VBBs → Compound dash-board symbols → VBB Bricks
31	 [Be04]		Symbol VBBs → Compound dash-board symbols → VBB Smiley
32	 [Be04]		Symbol VBBs → Compound dash-board symbols → VBB Comparing circles

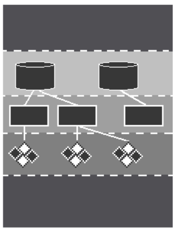
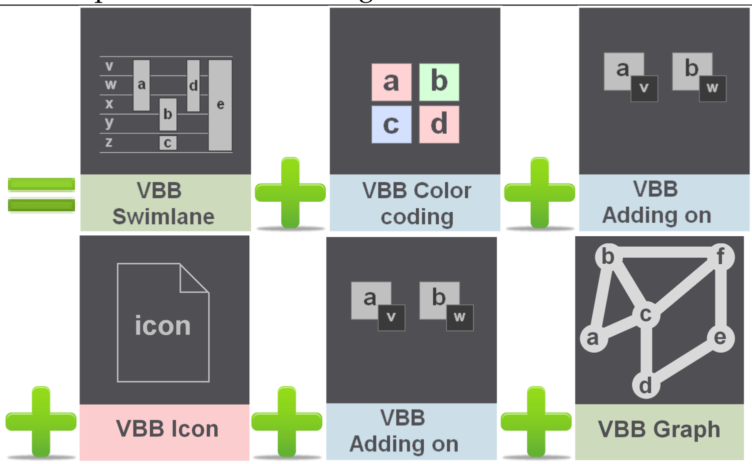
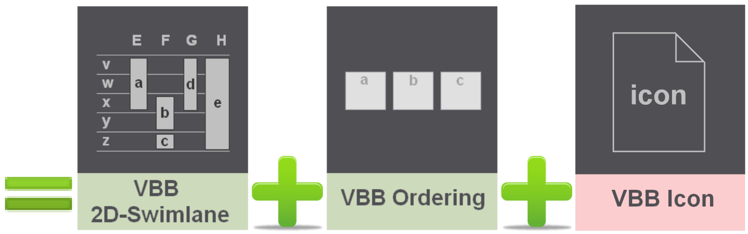
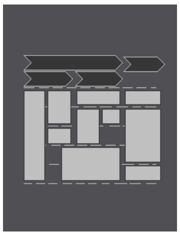
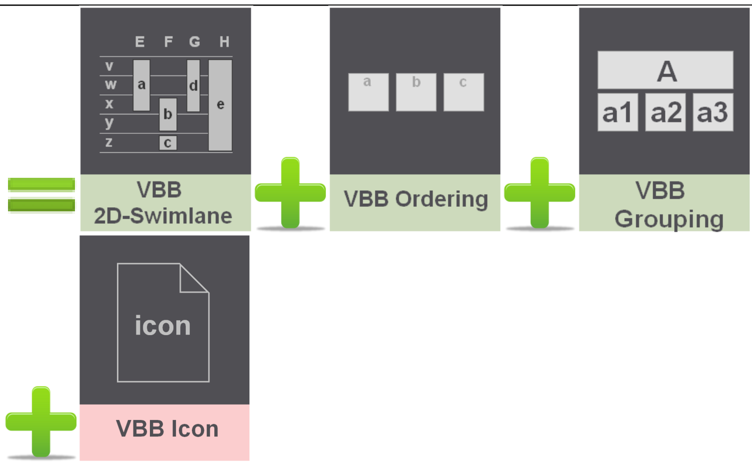
	Viewpoint	Decomposition into building blocks	VBB candidates
33	 [Ma08]		Symbol VBBs → VBB Icon → Data base, Bars
34	 [La08]		Structural VBBs → VBB Graph

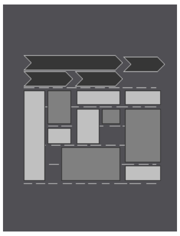
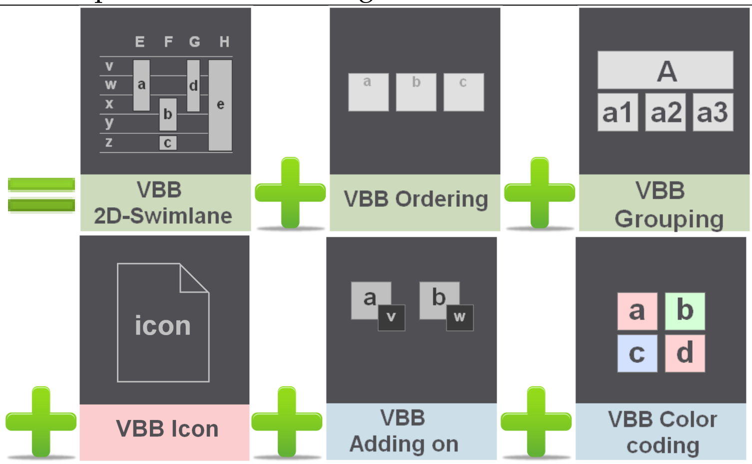
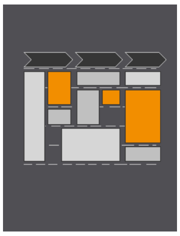
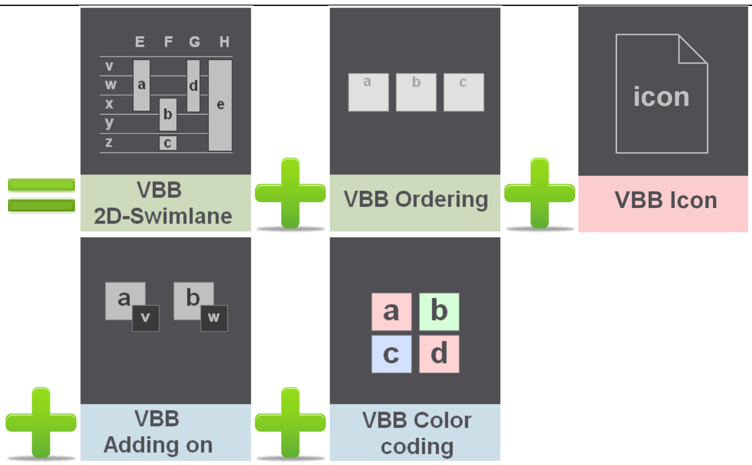
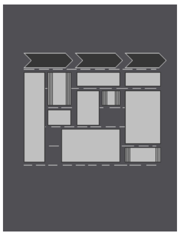
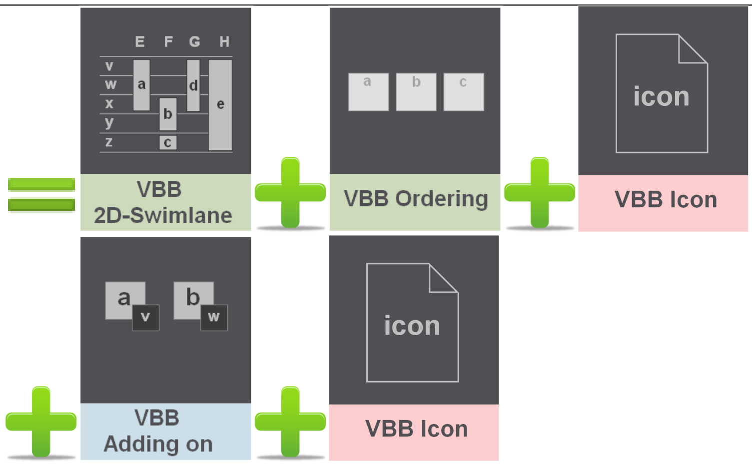
	Viewpoint	Decomposition into building blocks	VBB candidates
35	 [Bu08, Wi07]		1.Structural VBBs → VBB Graph 2.Symbol VBBs → Shapes → VBB Arrow
36	 [Bu08, Ma08]		1.Symbol VBBs → VBB Icon → Lollipop 2.Structural VBBs → VBB Graph

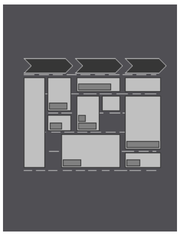
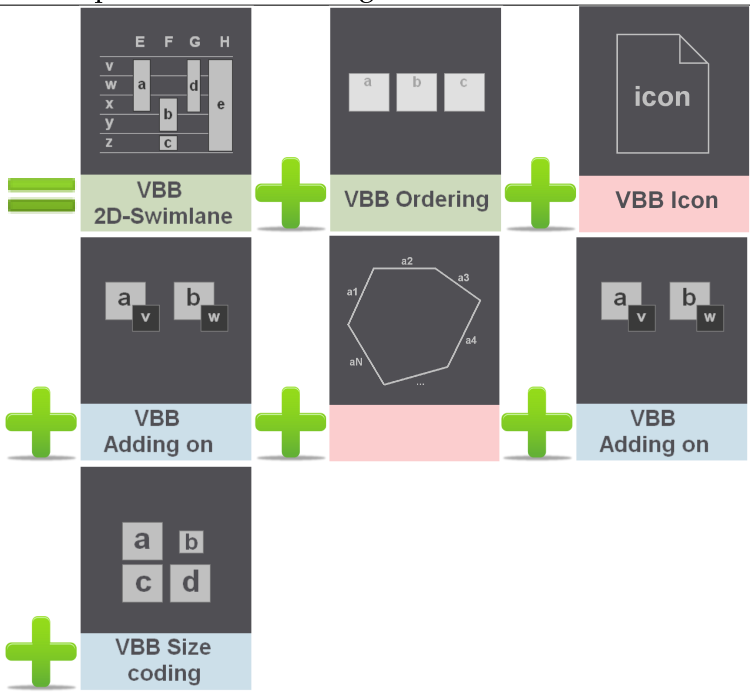
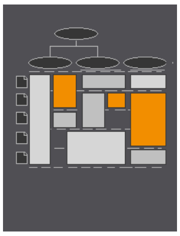
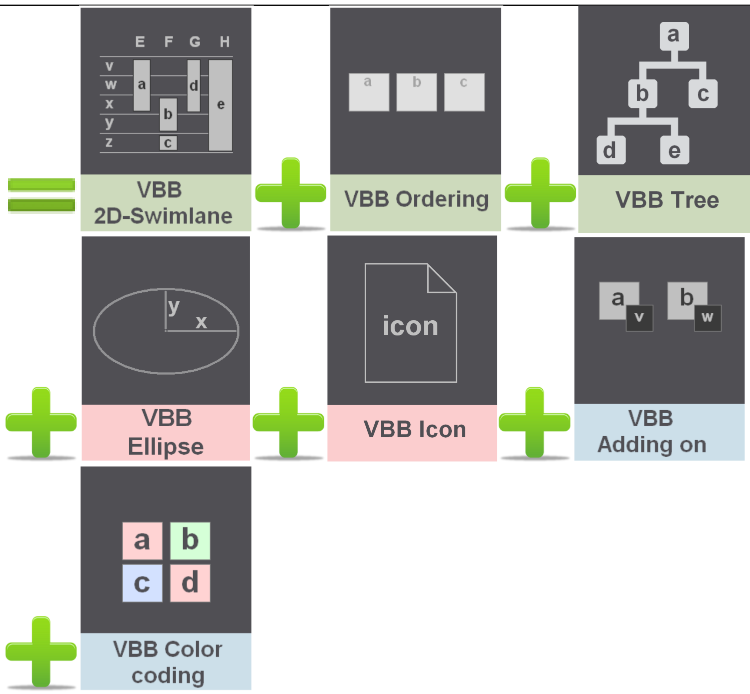
	Viewpoint	Decomposition into building blocks	VBB candidates
37	 [Bu08, Ma08]		1.Symbol VBBs → VBB Icon → Lollipop 2.Structural VBBs → VBB Graph 3.Symbol VBBs → Shapes → VBB Polygon
38	 [Ma08]		
39	 [Bu08, Ma08]		Structural VBBs → VBB Matrix
40	 [Bu08, La08, Ma08, Wi07]		Structural VBBs → VBB Matrix

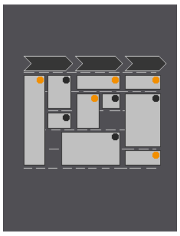
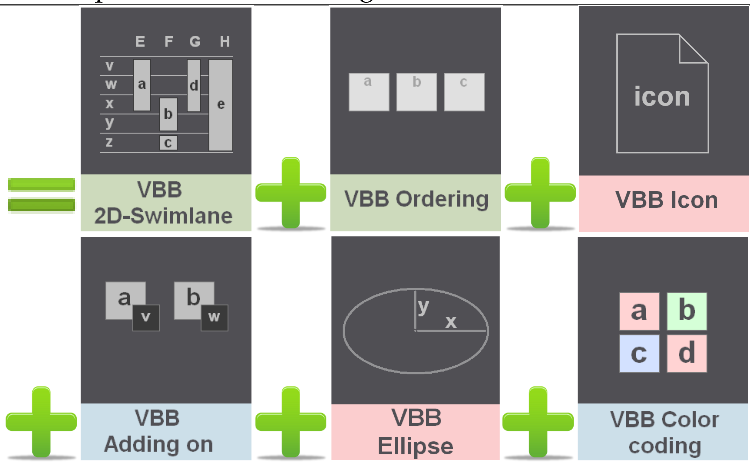
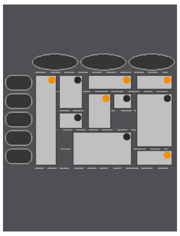
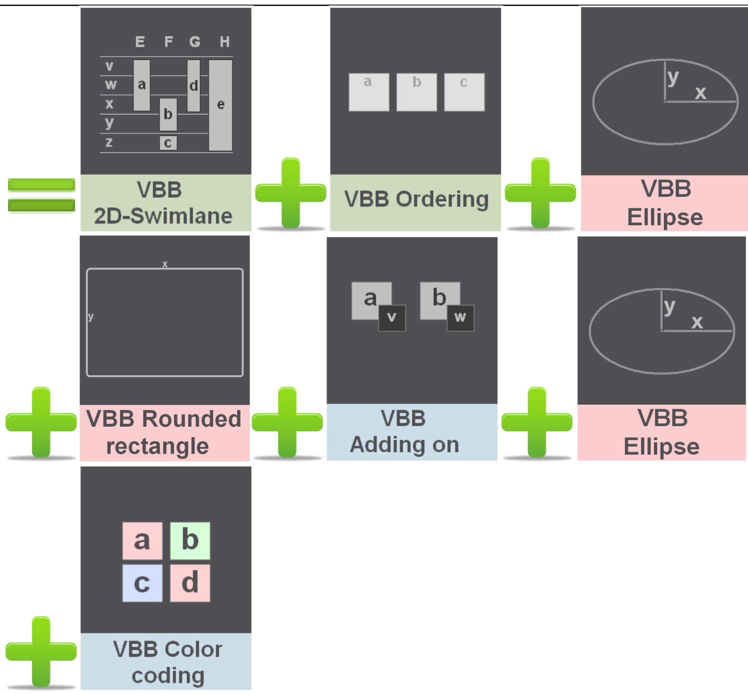
	Viewpoint	Decomposition into building blocks	VBB candidates
41	 [Bu08, Ma08]		1.Structural VBBs → VBB Matrix 2.Symbol VBBs → Shapes → VBB Arrow 3.Symbol VBBs → Shapes → VBB Polygon
42	 [Ma08, Se05, Wi07]		Structural VBBs → VBB Matrix
43	 [Bu08]		1.Structural VBBs → VBB Matrix 2.Symbol VBBs → Shapes → VBB Polygon
44	 [Ma08]		Structural VBBs → VBB Graph

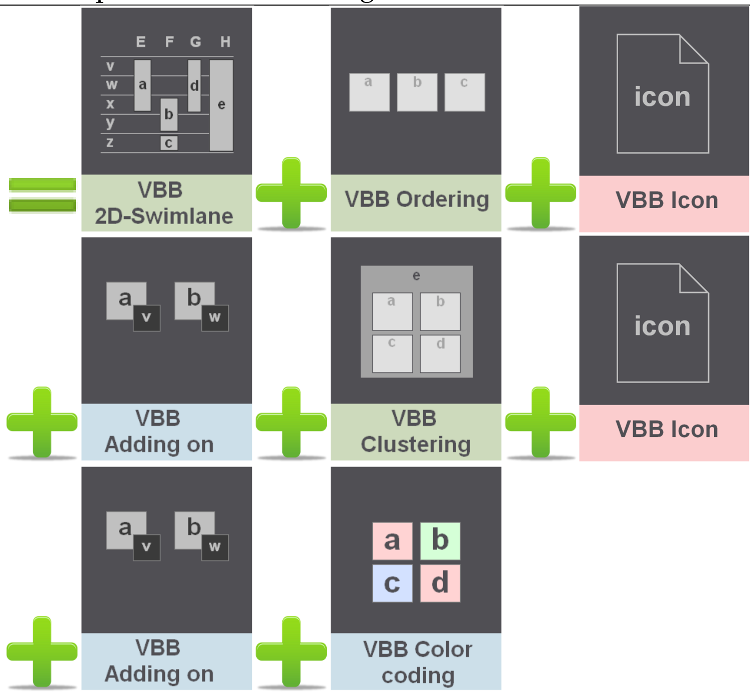
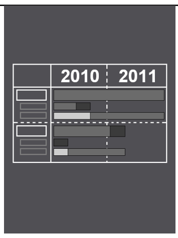
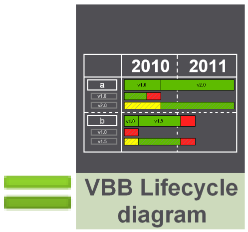
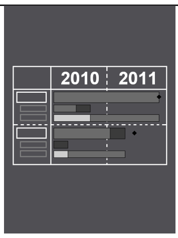
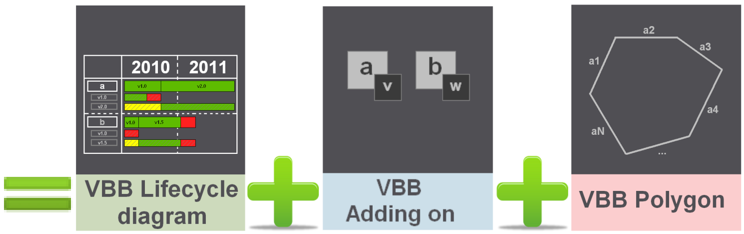
	Viewpoint	Decomposition into building blocks	VBB candidates
45	 [Ma08]		1.Symbol VBBs → VBB Icon 2.Structural VBBs → VBB Graph
46	 [Ma08]		1.Symbol VBBs → Shapes → VBB Polygon 1.Symbol VBBs → VBB Icon 2.Structural VBBs → VBB Graph
47	 [Ma08]		1.Symbol VBBs → VBB Icon 2.Structural VBBs → VBB Graph

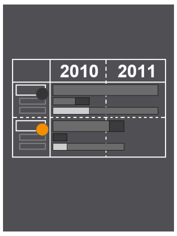
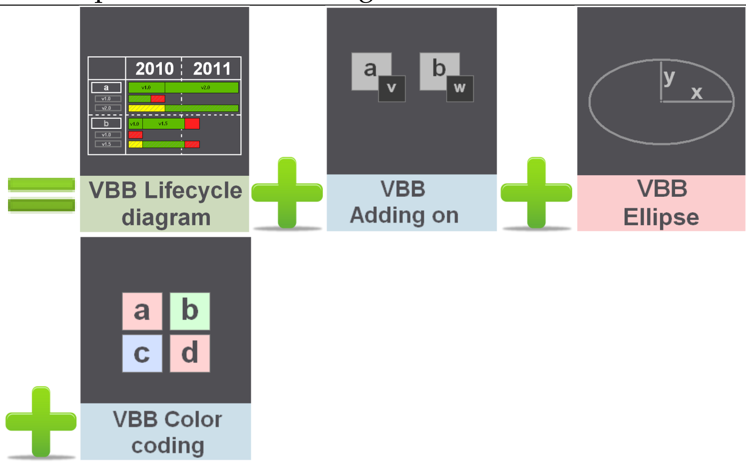
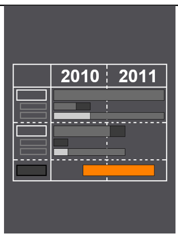
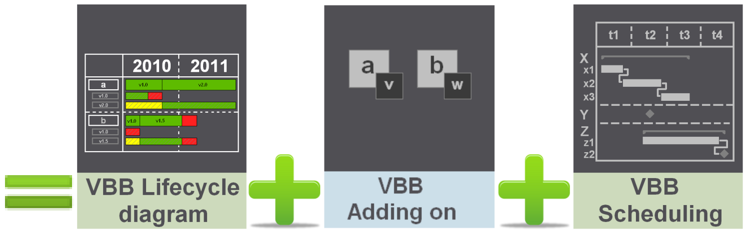
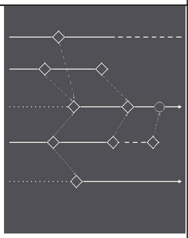
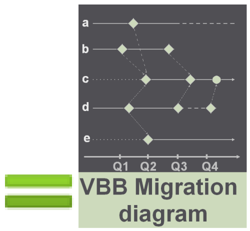
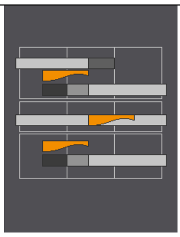
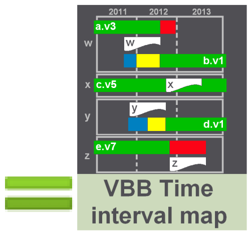
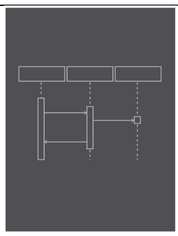
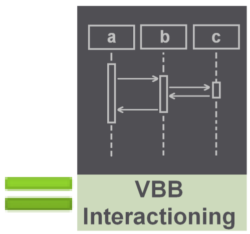
	Viewpoint	Decomposition into building blocks	VBB candidates
48	 [Ma08]		1.Symbol VBBs → VBB Icon 2.Structural VBBs → VBB Graph
49	 [Bu08, La08]		Symbol VBBs → VBB Icon → Chevron
50	 [Ma08, Se05, Wi07]		1.Structural VBBs → VBB Grouping 2.Symbol VBBs → VBB Icon → Chevron

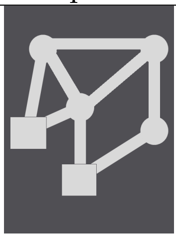
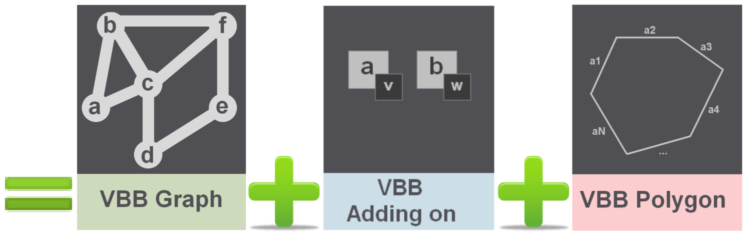
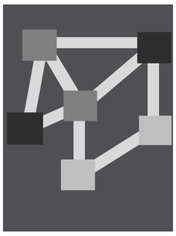
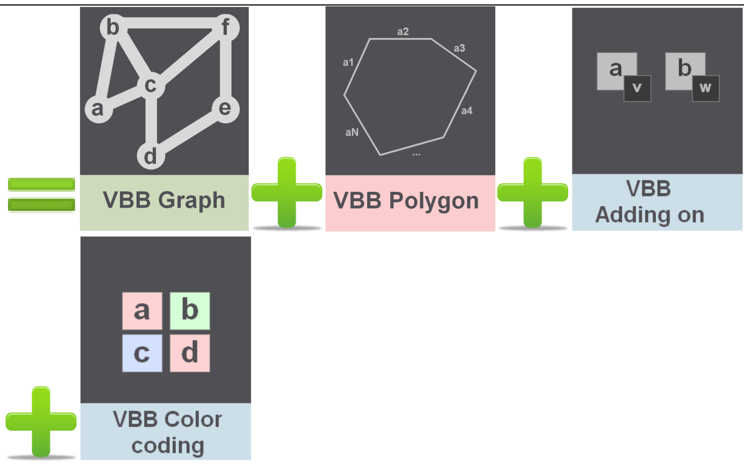
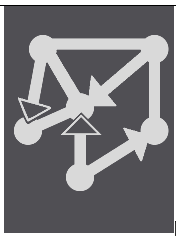
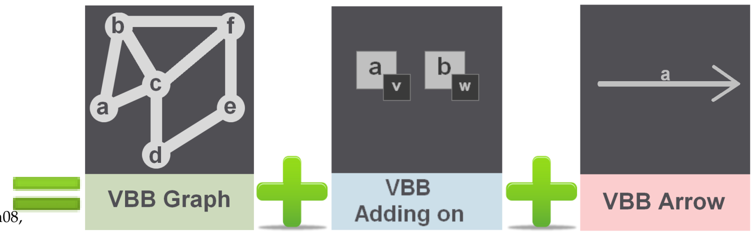
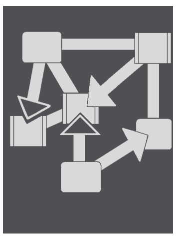
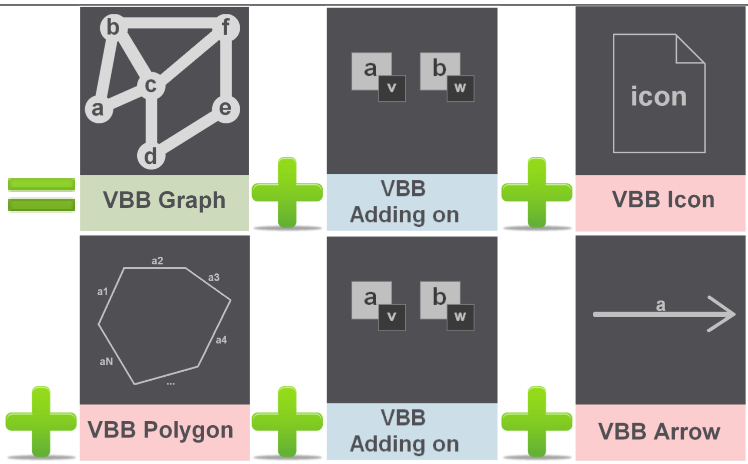
	Viewpoint	Decomposition into building blocks	VBB candidates
51	 [Ma08, Wi07]		1.Structural VBBs → VBB Grouping 2.Symbol VBBs → VBB Icon → Chevron
52	 [Bu08, Ma08, Se05, Wi07]		Symbol VBBs → VBB Icon → Chevron
53	 [Bu08, Ma08]		Symbol VBBs → VBB Icon → Chevron, Framed rectangle

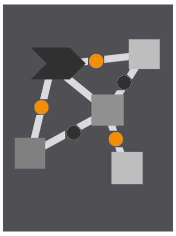
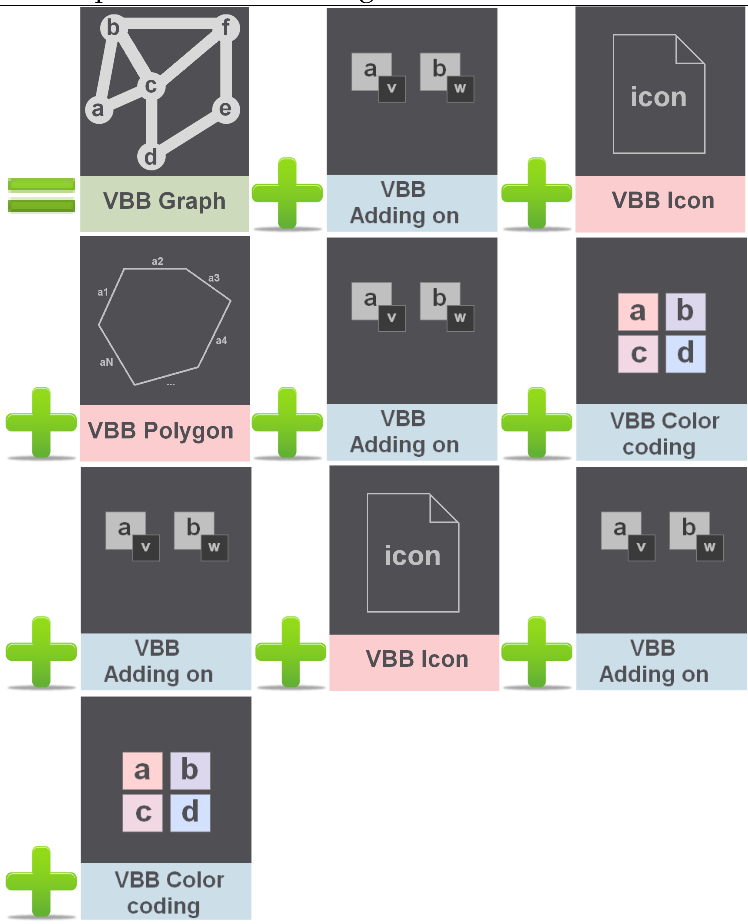
	Viewpoint	Decomposition into building blocks	VBB candidates
54	 [Bu08]		1.Symbol VBBs → VBB Icon → Chevron 2.Symbol VBBs → Shapes → VBB Polygon
55	 [Ma08]		Symbol VBBs → VBB Icon

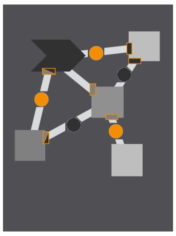
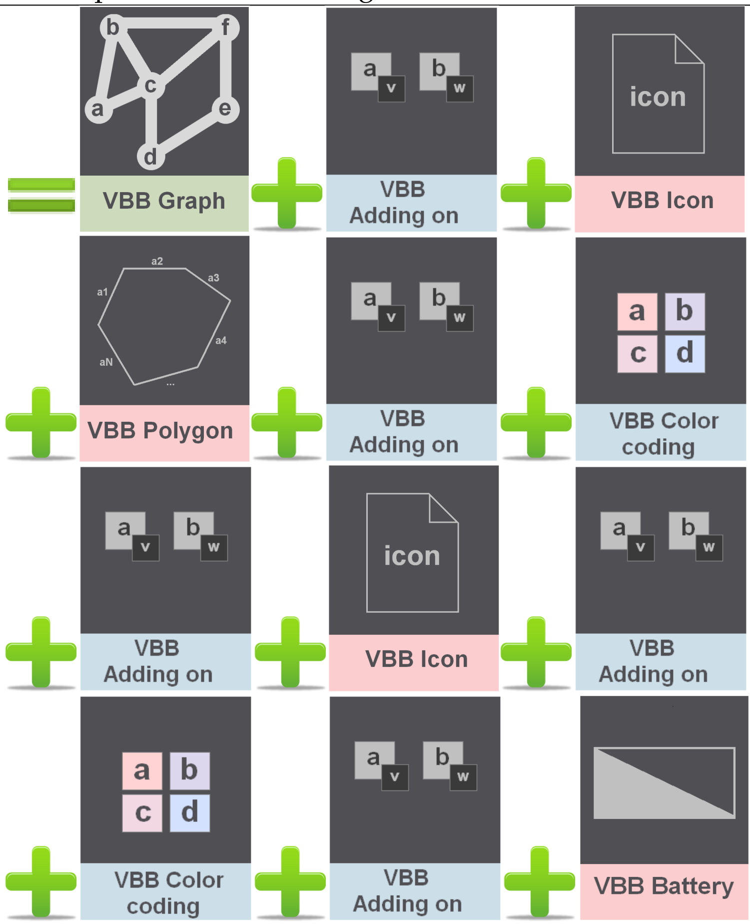
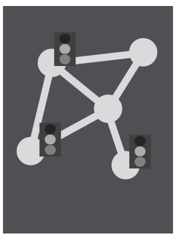
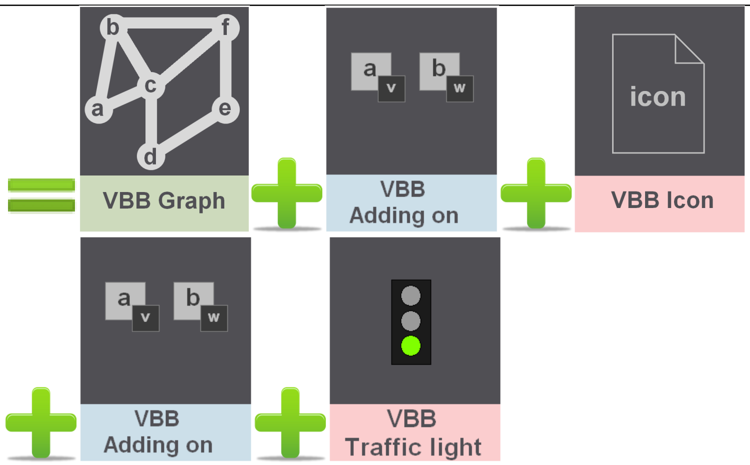
<table>
<tr><th></th><th>Viewpoint</th><th>Decomposition into building blocks</th><th>VBB candidates</th></tr>
<tr><td>56</td><td>
[Ma08]</td><td>
</td><td>Symbol VBBs → VBB Icon → Chevron</td></tr>
<tr><td>57</td><td>
[Ma08]</td><td>
</td><td>Symbol VBBs → Shapes → VBB Rounded rectangle</td></tr>
</table>

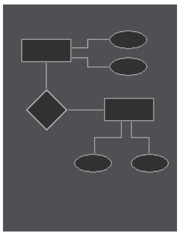
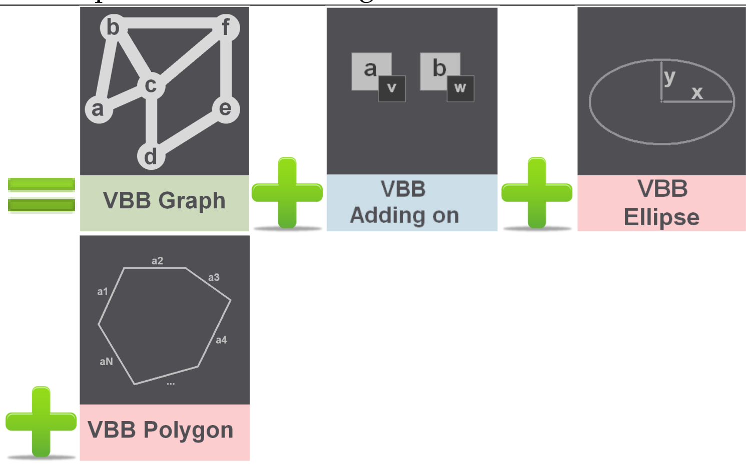
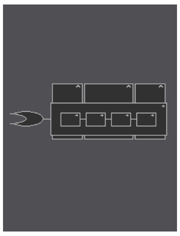
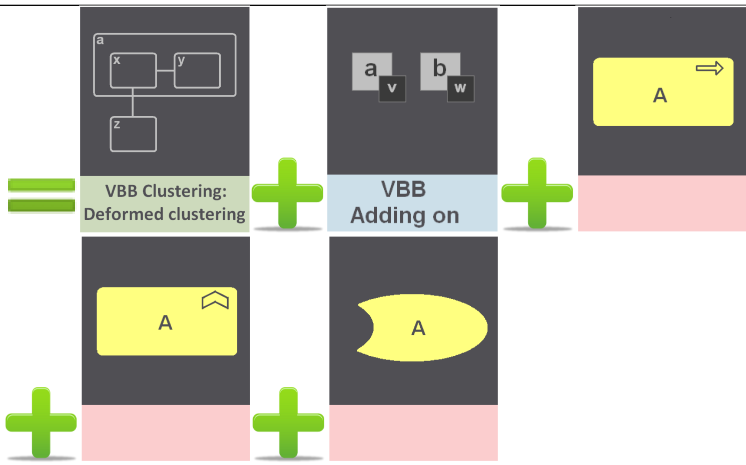
	Viewpoint	Decomposition into building blocks	VBB candidates
58	 [Wi07]		Symbol VBBs → VBB Icon → Chevron, Data base
59	 [Bu08, La08, Ma08, Se05, Wi07]		Structural VBBs → VBB Lifecycle diagram
60	 [Ma08]		1.Structural VBBs → VBB Lifecycle diagram 2.Symbol VBBs → Shapes → VBB Polygon

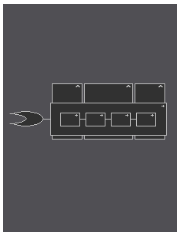
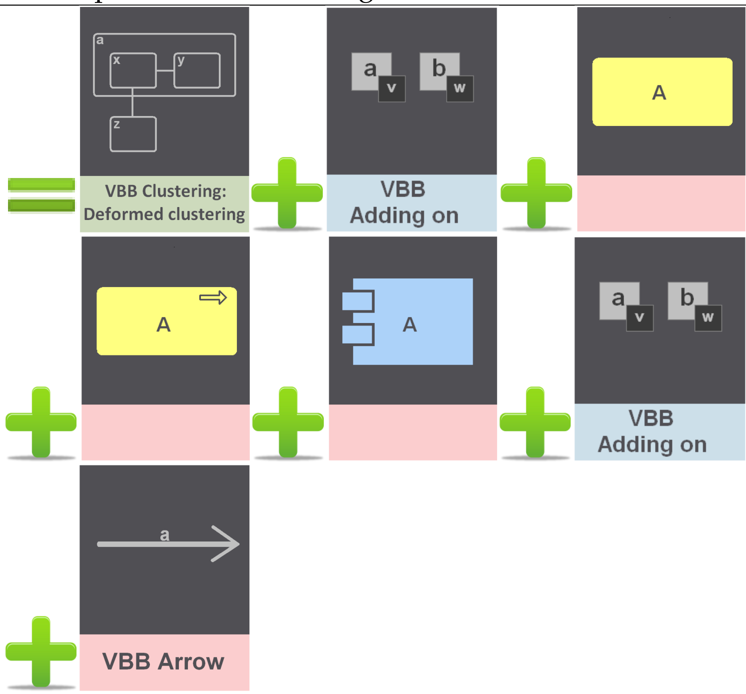
	Viewpoint	Decomposition into building blocks	VBB candidates
61	 [Ma08]		Structural VBBs → VBB Lifecycle diagram
62	 [Bu08]		Structural VBBs → VBB Lifecycle diagram
63	 [Bu08, Ma08]		Structural VBBs → VBB Migration diagram
64	 [Bu08]		Structural VBBs → VBB Time interval map
65	 [Bu08]		Structural VBBs → VBB Interactioning

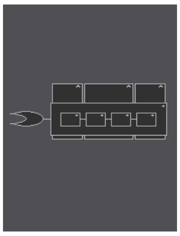
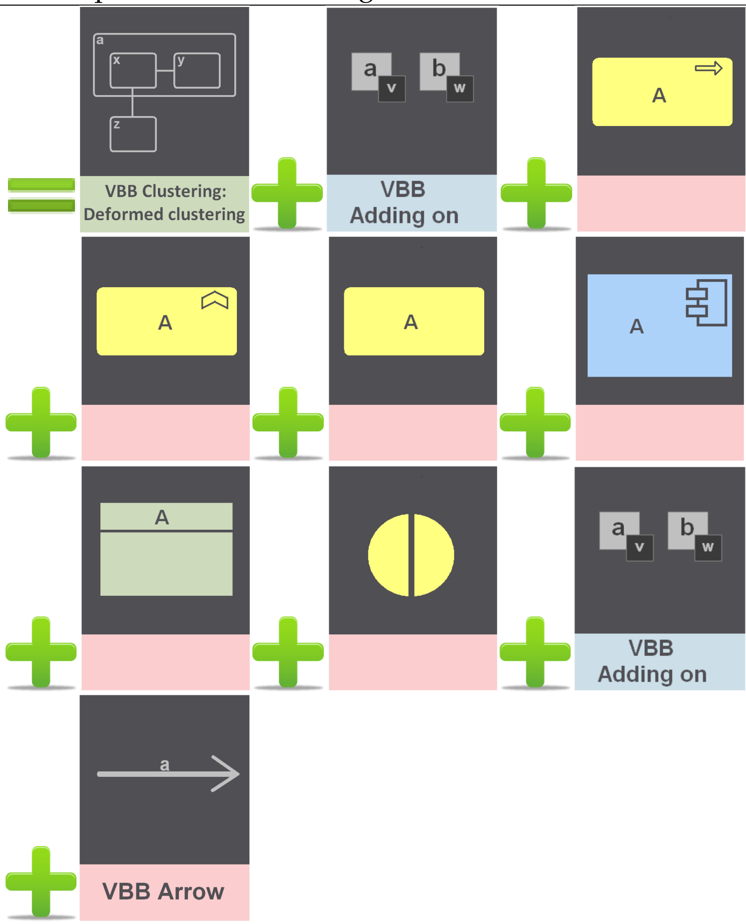
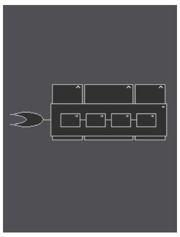
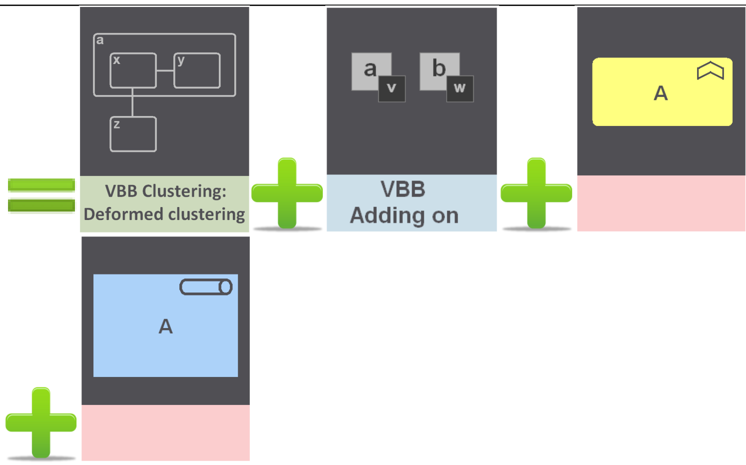
	Viewpoint	Decomposition into building blocks	VBB candidates
66	 [Ma08]		1.Structural VBBs → VBB Graph 2.Symbol VBBs → Shapes → VBB Polygon
67	 [Ma08]		1.Structural VBBs → VBB Graph 2.Symbol VBBs → Shapes → VBB Polygon
68	 [La08, Ma08, Wi07]		1.Structural VBBs → VBB Graph 2.Symbol VBBs → Shapes → VBB Arrow
69	 [Ma08]		1.Structural VBBs → VBB Graph 2.Symbol VBBs → VBB Icon → Framed rectangle 3.Symbol VBBs → Shapes → VBB Polygon 4.Symbol VBBs → Shapes → VBB Arrow

	Viewpoint	Decomposition into building blocks	VBB candidates
70	 [La08]		1.Structural VBBs → VBB Graph 2.Symbol VBBs → VBB Icon → Chevron, Lollipop, Depressed lollipop 3.Symbol VBBs → Shapes → VBB Polygon

	Viewpoint	Decomposition into building blocks	VBB candidates
71	 [La08]		1.Structural VBBs → VBB Graph 2.Symbol VBBs → VBB Icon → Chevron, Lollipop, Depressed lollipop 3.Symbol VBBs → Shapes → VBB Polygon 4.Symbol VBBs → Compound dashboard symbols → VBB Battery
72	 [Ma08]		1.Structural VBBs → VBB Graph 2.Symbol VBBs → VBB Icon 3.Symbol VBBs → Compound dashboard symbols → VBB Traffic light

	Viewpoint	Decomposition into building blocks	VBB candidates
73	 [Bu08]		1.Structural VBBs → VBB Graph 2.Symbol VBBs → Shapes → VBB Polygon
74	 [Bu08]		1.Structural VBBs → VBB Clustering: special case 2.Symbol VBBs → VBB Icon → ArchiMate

	Viewpoint	Decomposition into building blocks	VBB candidates
75	 [Bu08]		1.Structural VBBs → VBB Clustering: special case 2.Symbol VBBs → VBB Icon → ArchiMate 3.Symbol VBBs → Shapes → VBB Arrow

	Viewpoint	Decomposition into building blocks	VBB candidates
76	 [Bu08]		1.Structural VBBs → VBB Clustering: special case 2.Symbol VBBs → VBB Icon → ArchiMate 3.Symbol VBBs → Shapes → VBB Arrow
77	 [Bu08]		1.Structural VBBs → VBB Clustering: special case 2.Symbol VBBs → VBB Icon → ArchiMate

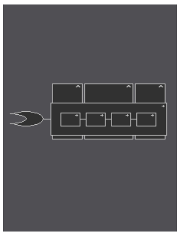
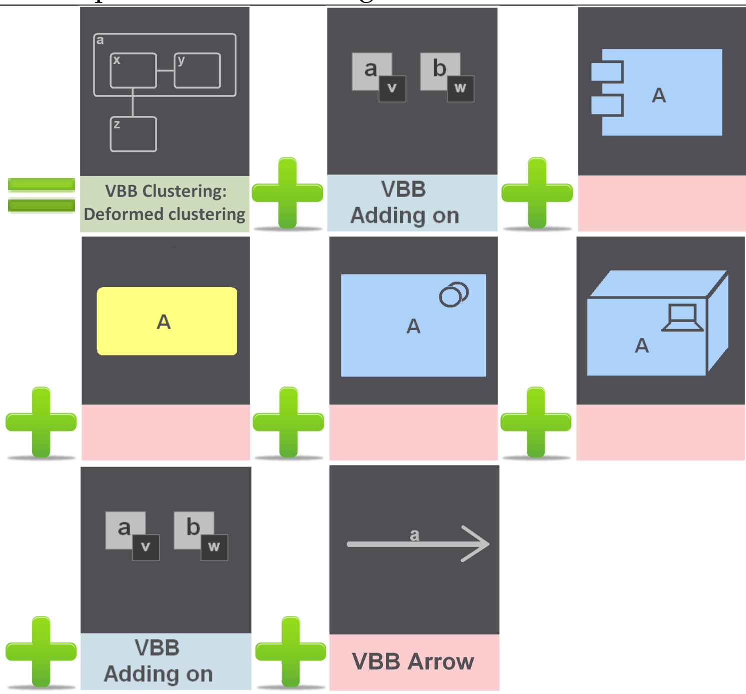
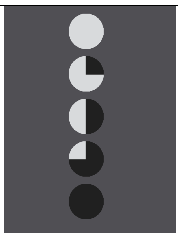
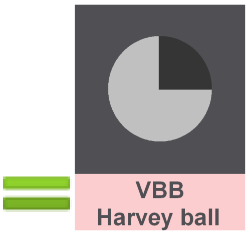
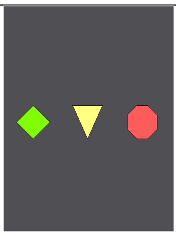
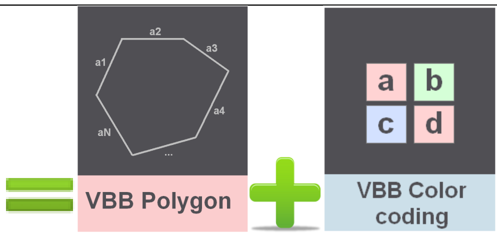
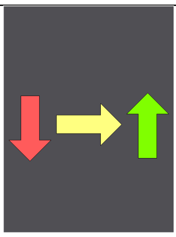
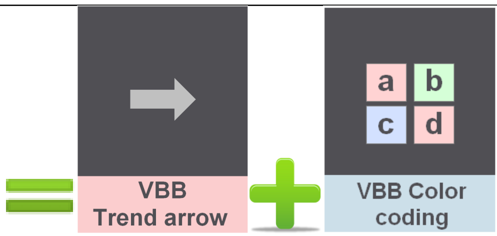
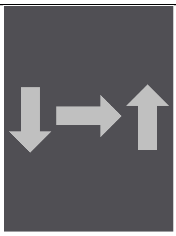
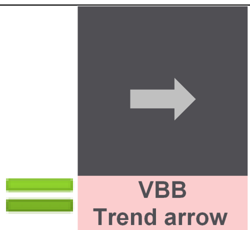
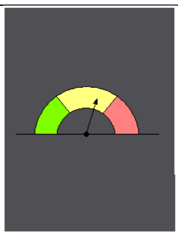
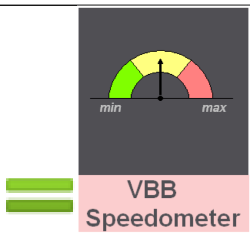
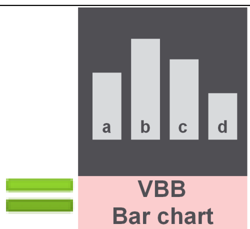
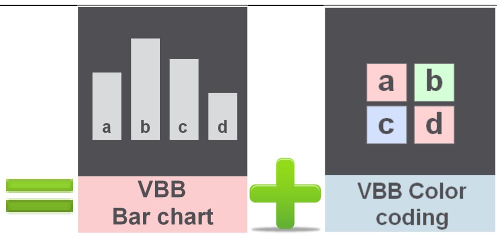
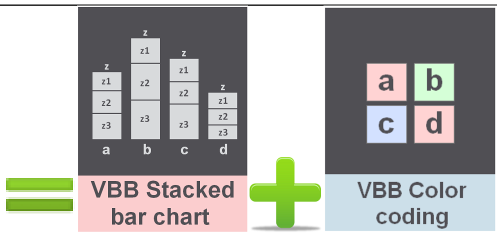
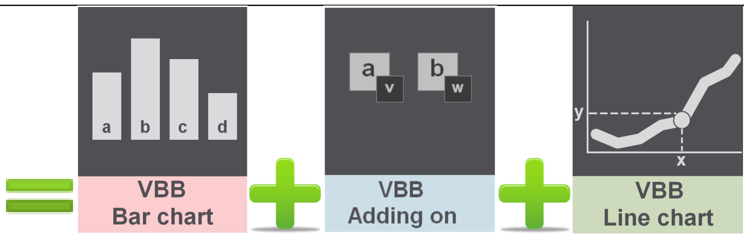
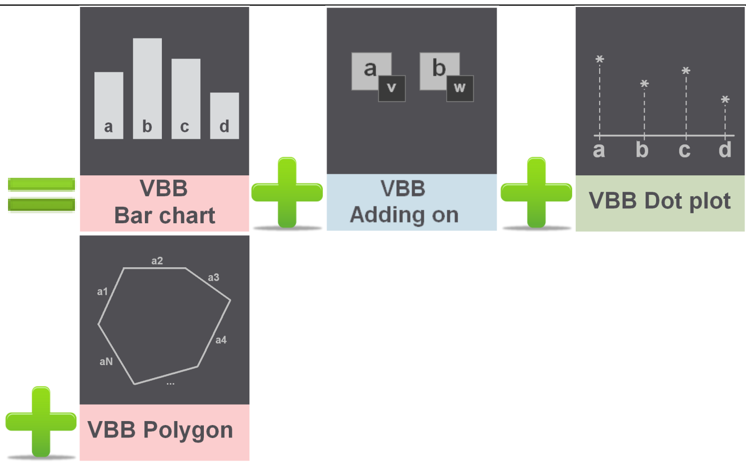
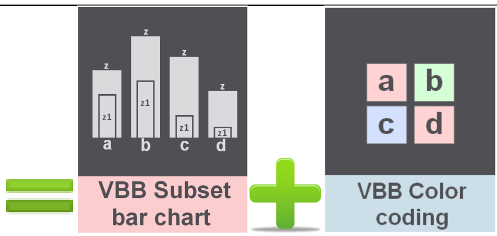
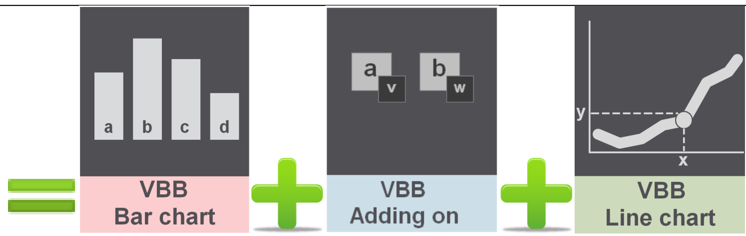
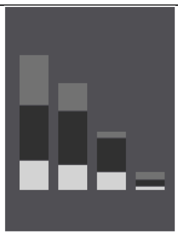
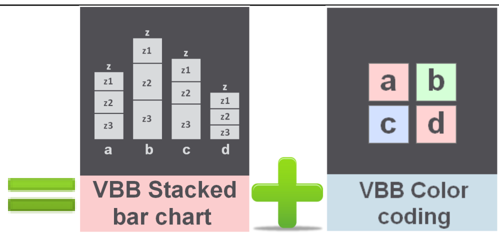
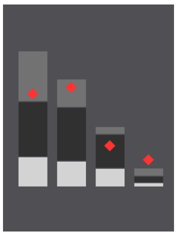
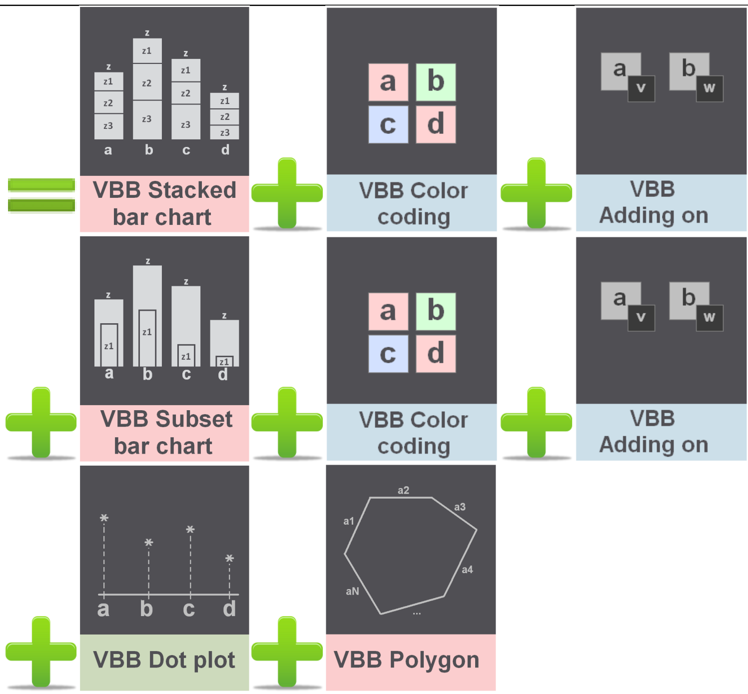
	Viewpoint	Decomposition into building blocks	VBB candidates
78	 [Bu08]		1.Structural VBBs → VBB Clustering: special case 2.Symbol VBBs → VBB Icon → ArchiMate 3.Symbol VBBs → Shapes → VBB Arrow
79	 [Se05]		Symbol VBBs → Compound dashboard symbol → VBB Harvey ball

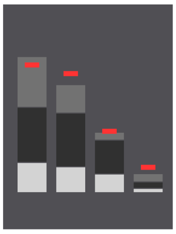
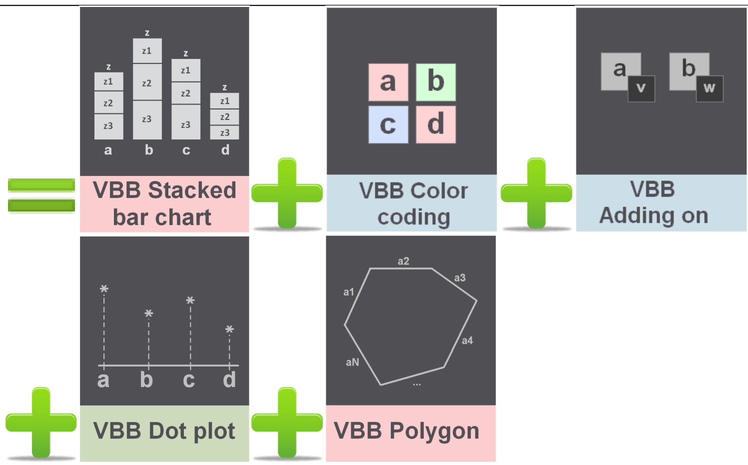
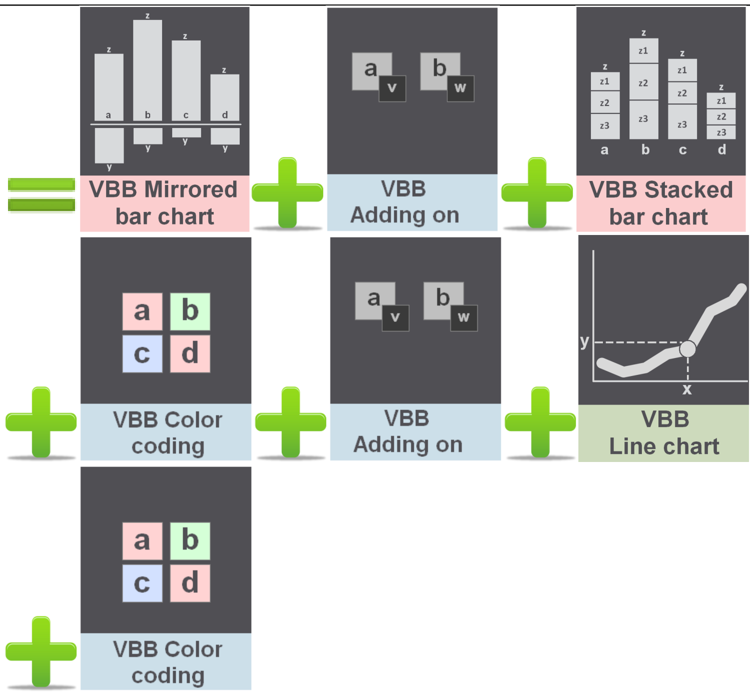
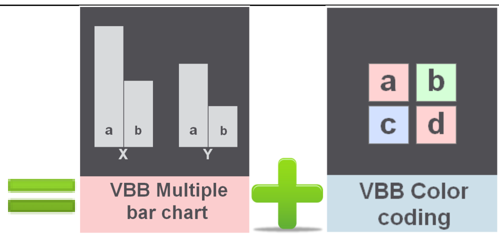
Table A.1.: Decomposition of viewpoints found in EA management literature

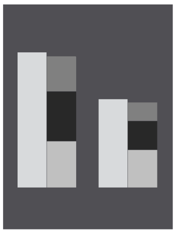
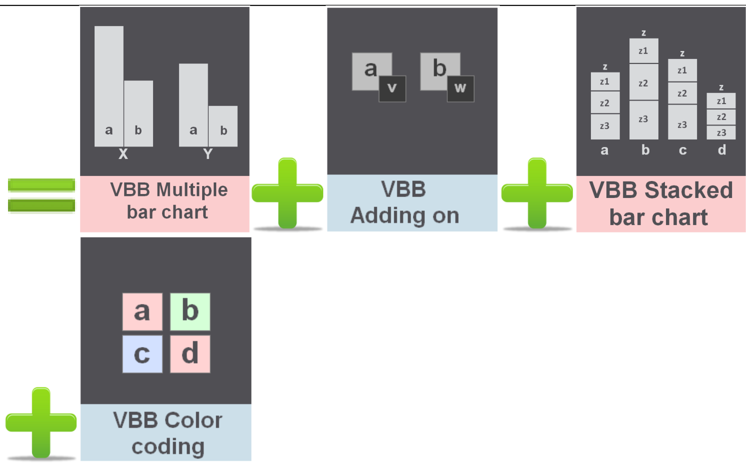
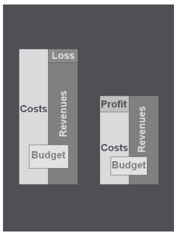
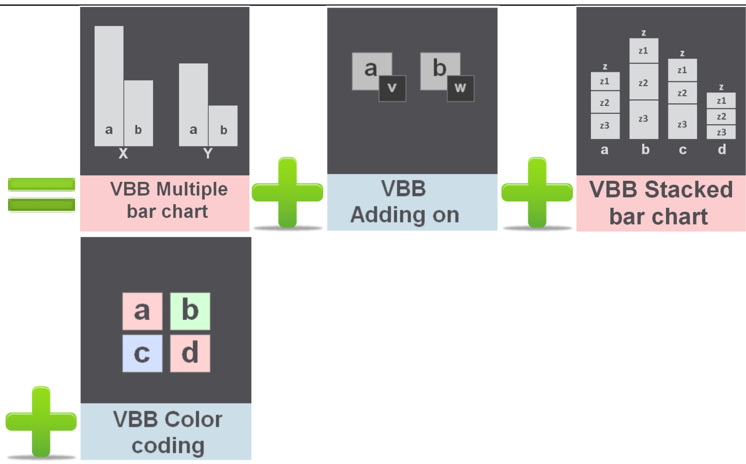
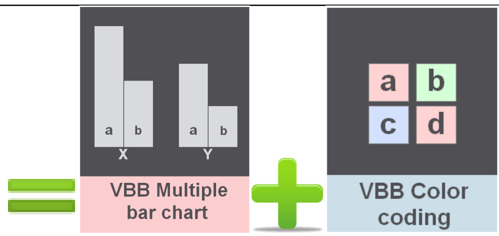
B. Decomposition of viewpoints ascertained in the course of case study

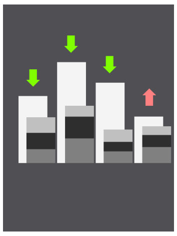
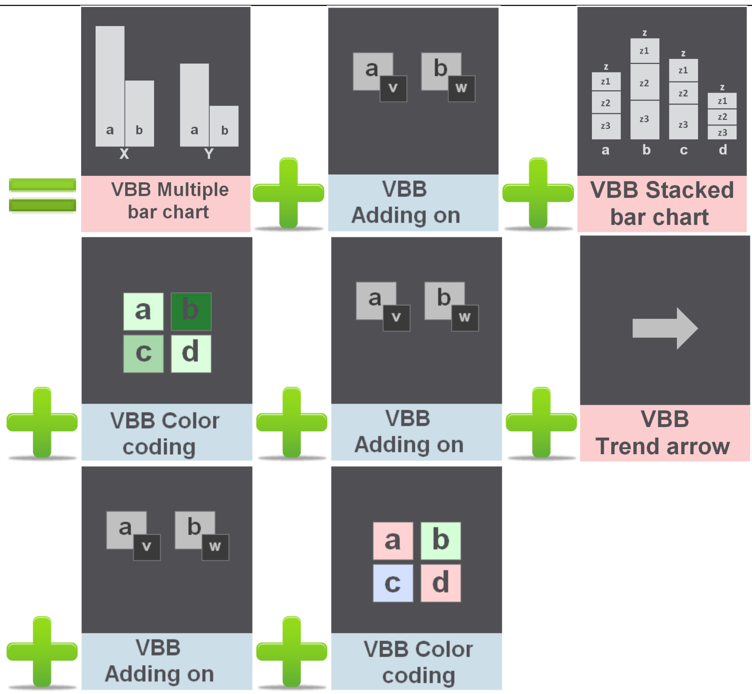
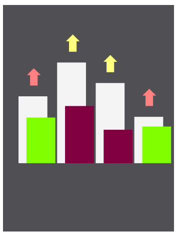
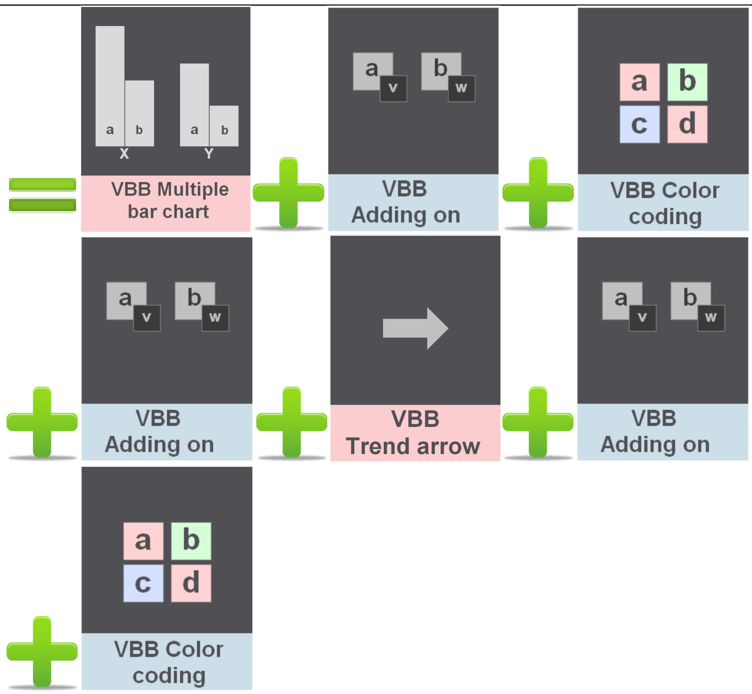
	Viewpoint	Decomposition into building blocks	New VBB candidates
1			Symbol VBBs → Shapes → VBB Polygon
2			Symbol VBBs → Compound dashboard symbols → VBB Trend arrow
3			Symbol VBBs → Compound dashboard symbols → VBB Trend arrow
4			Symbol VBBs → Compound dashboard symbols → VBB Speedometer
5			

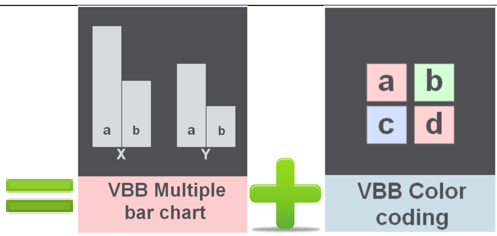
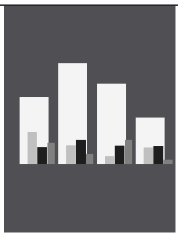
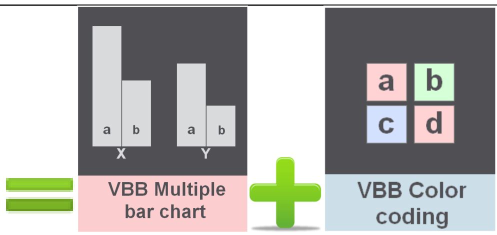
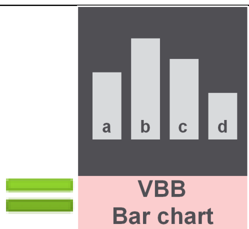
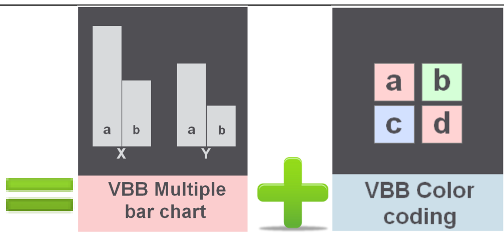
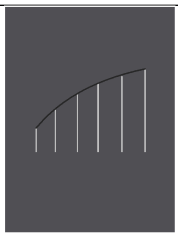
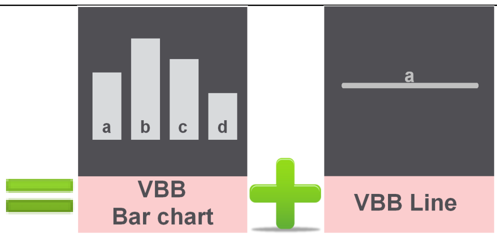
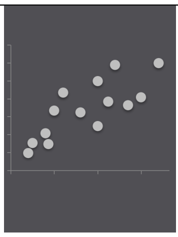
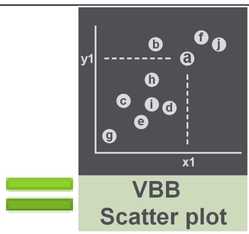
	Viewpoint	Decomposition into building blocks	New VBB candidates
6			
7			Symbol VBBs → Compound chart symbols → VBB Bar chart → VBB Stacked bar chart
8			Structural VBBs → VBB Line chart
9			1.Structural VBBs → VBB Dot plot 2.Symbol VBBs → Shapes → VBB Polygon
10			Symbol VBBs → Compound chart symbols → VBB Bar chart → VBB Subset bar chart

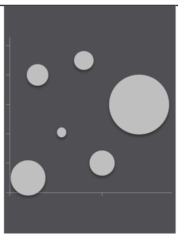
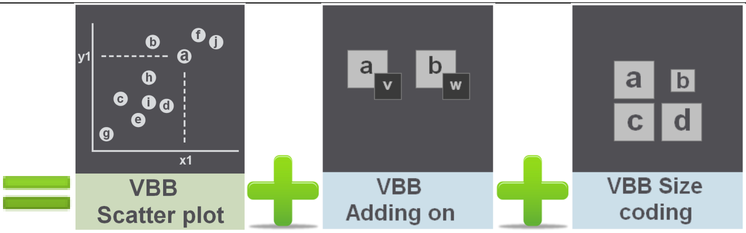
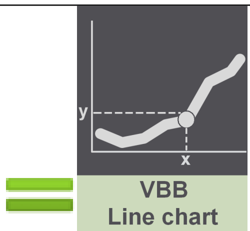
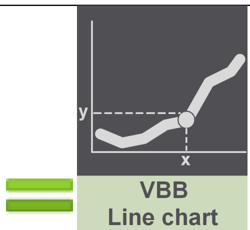
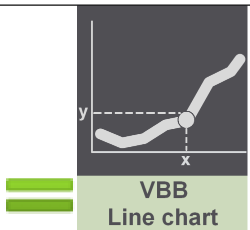
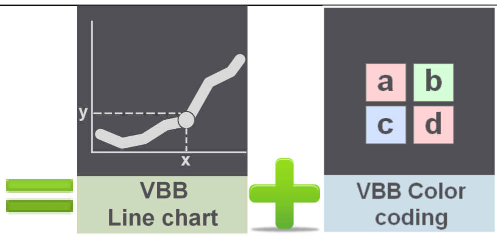
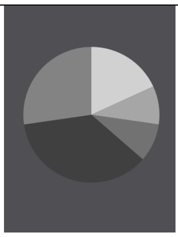
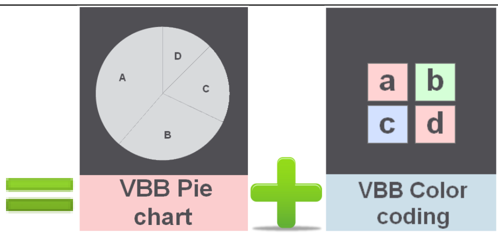
	Viewpoint	Decomposition into building blocks	New VBB candidates
11			Structural VBBs → VBB Line chart
12			Symbol VBBs → Compound chart symbols → VBB Bar chart → VBB Stacked bar chart
13			1.Symbol VBBs → Compound chart symbols → VBB Bar chart → VBB Stacked bar chart 2.Symbol VBBs → Compound chart symbols → VBB Bar chart → VBB Subset bar chart 3.Structural VBBs → VBB Dot plot 4.Symbol VBBs → Shapes → VBB Polygon

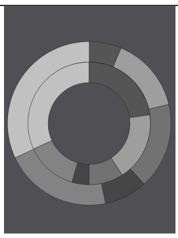
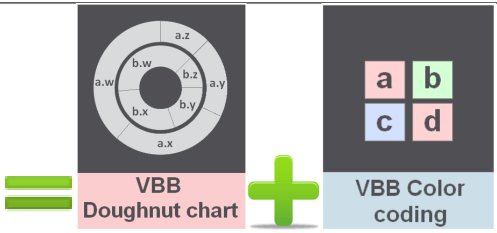
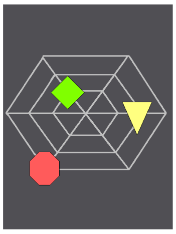
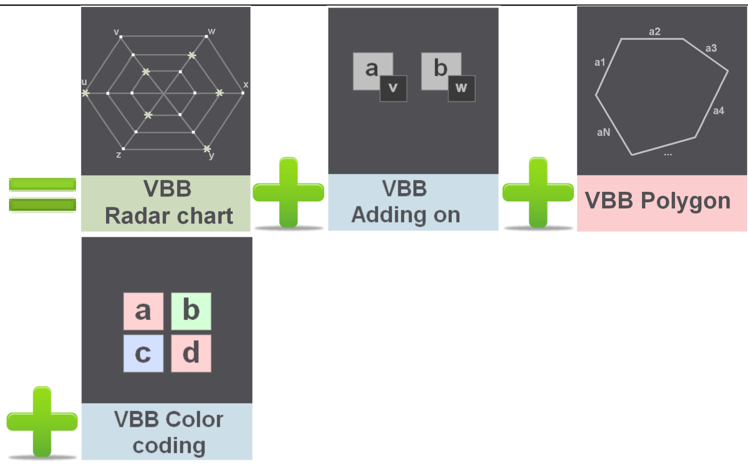
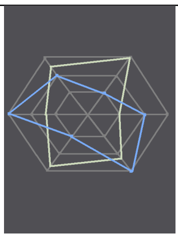
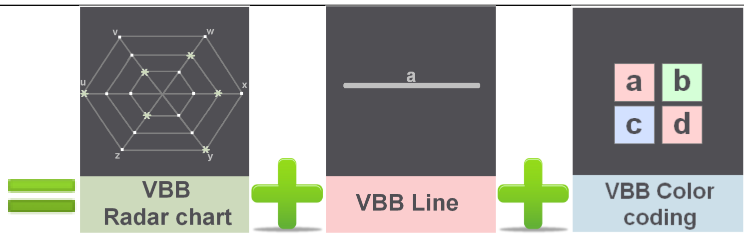
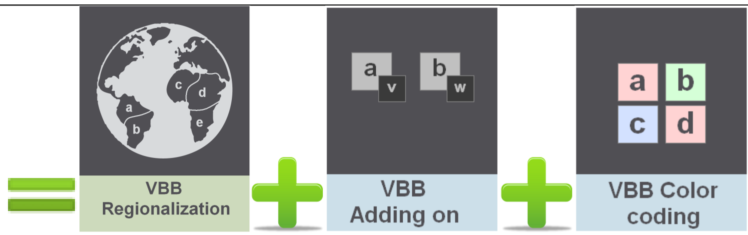
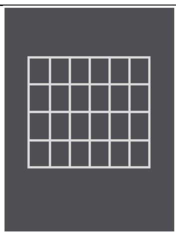
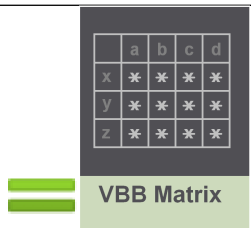
<table>
<tr><th></th><th>Viewpoint</th><th>Decomposition into building blocks</th><th>New VBB candidates</th></tr>
<tr><td>14</td><td></td><td>
</td><td>1.Symbol VBBs → Compound chart symbols → VBB Bar chart → VBB Stacked bar chart
2.Structural VBBs → VBB Dot plot
3.Symbol VBBs → Shapes → VBB Polygon</td></tr>
<tr><td>15</td><td></td><td>
</td><td>1.Symbol VBBs → Compound chart symbols → VBB Bar chart → VBB Mirrored bar chart
2.Symbol VBBs → Compound chart symbols → VBB Bar chart → VBB Stacked bar chart
3.Structural VBBs → VBB Line chart</td></tr>
<tr><td>16</td><td></td><td>
</td><td></td></tr>
</table>

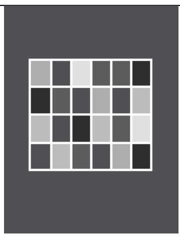
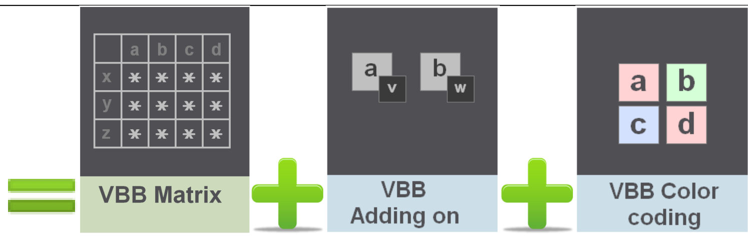
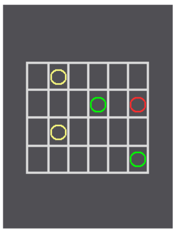
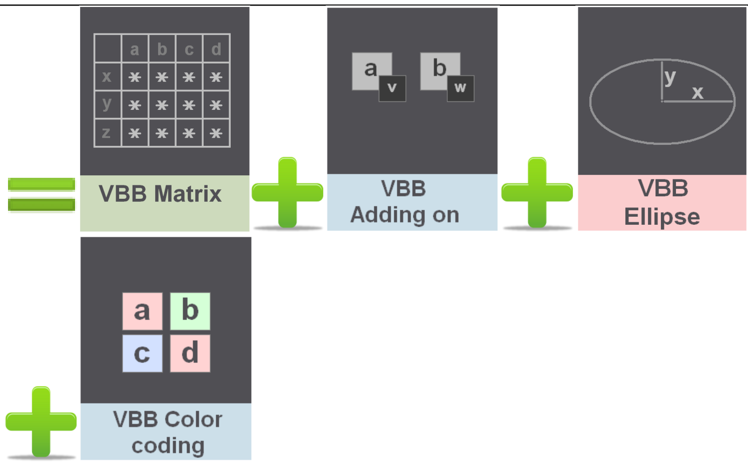
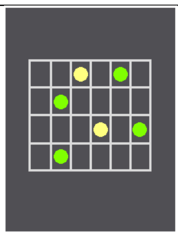
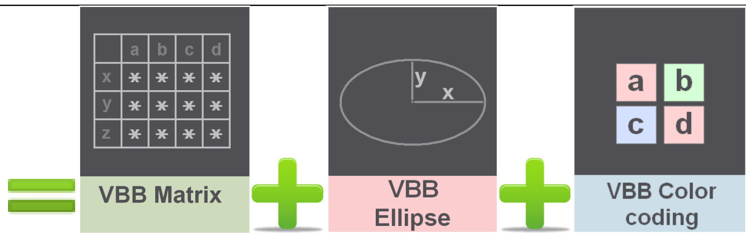
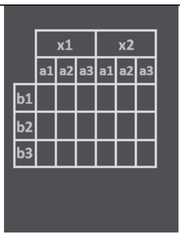
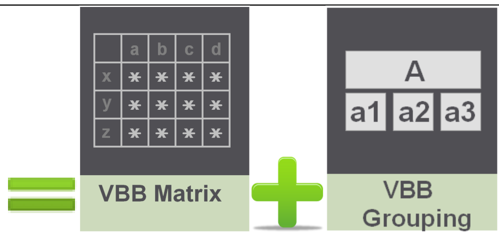
	Viewpoint	Decomposition into building blocks	New VBB candidates
17			Symbol VBBs → Compound chart symbols → VBB Bar chart → VBB Stacked bar chart
18			Symbol VBBs → Compound chart symbols → VBB Bar chart → VBB Stacked bar chart
19			

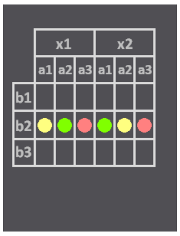
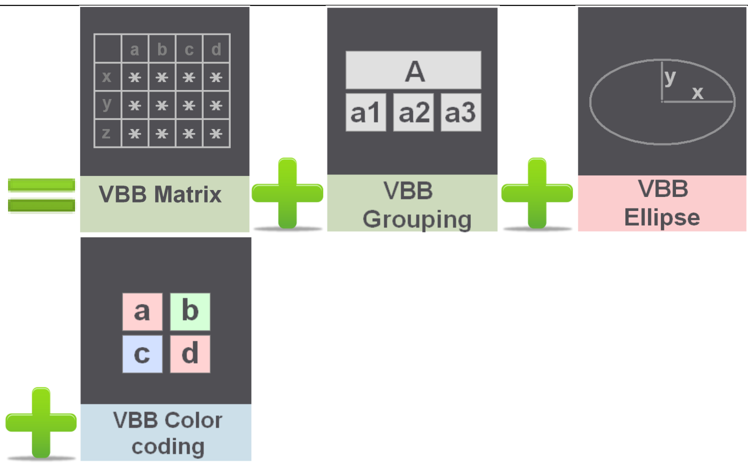
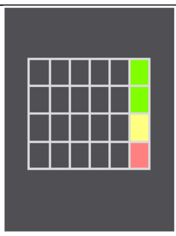
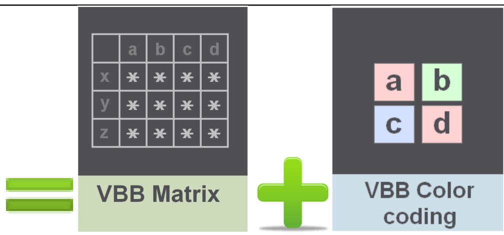
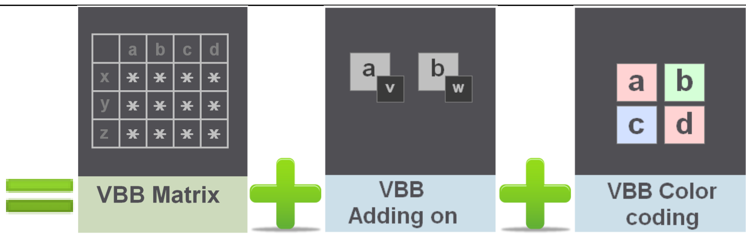
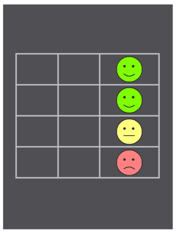
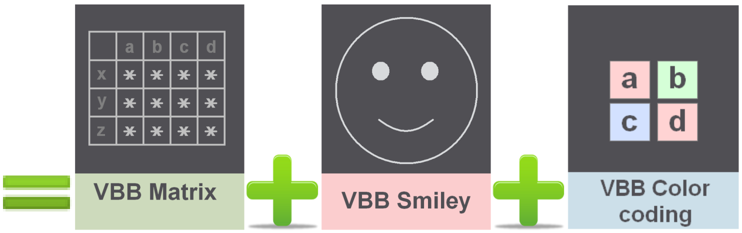
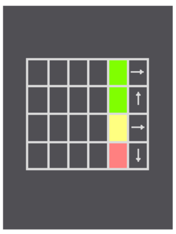
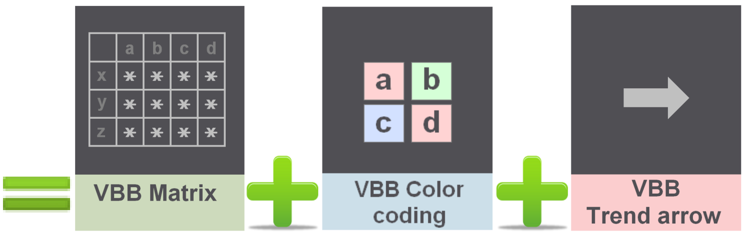
	Viewpoint	Decomposition into building blocks	New VBB candidates
20			1.Symbol VBBs → Compound chart symbols → VBB Bar chart → VBB Stacked bar chart 2.Symbol VBBs → Compound dashboard symbols → VBB Trend arrow
21			Symbol VBBs → Compound dashboard symbols → VBB Trend arrow

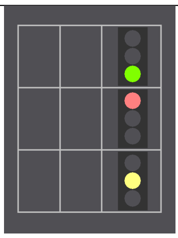
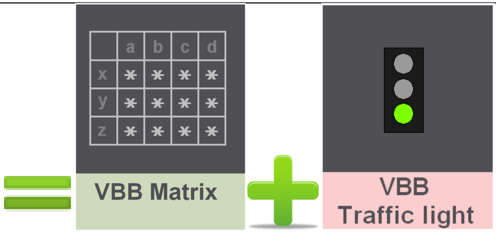
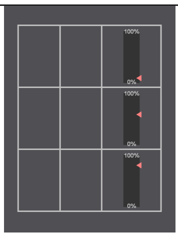
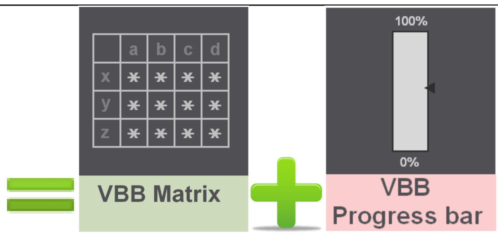
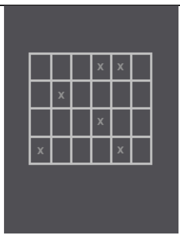
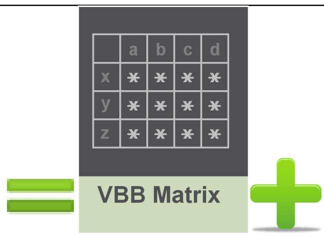
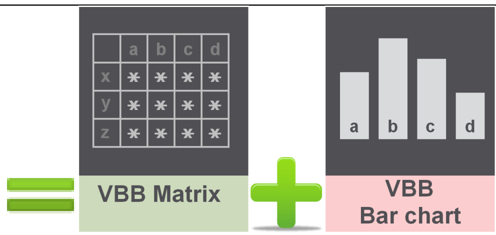
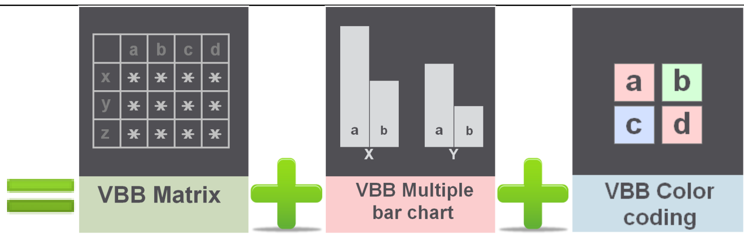
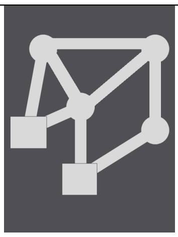
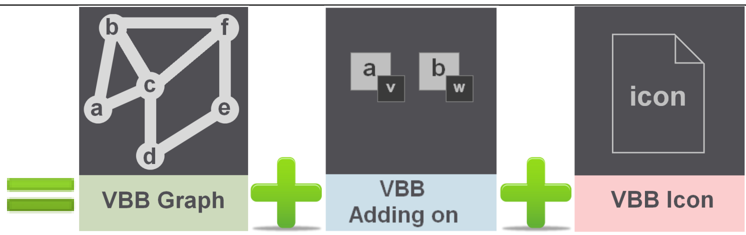
	Viewpoint	Decomposition into building blocks	New VBB candidates
22			
23			
24			
25			
26			Structural VBBs → VBB Line chart
27			Structural VBBs → VBB Scatter plot

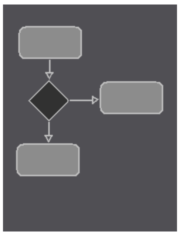
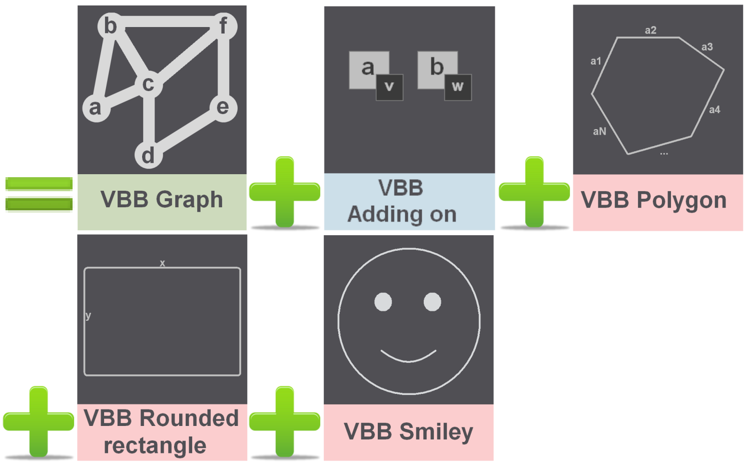
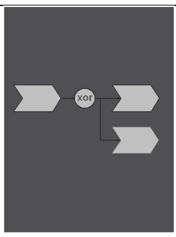
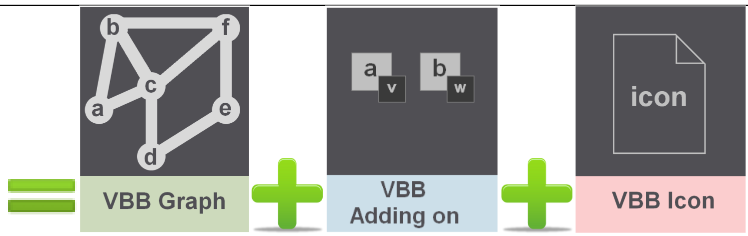
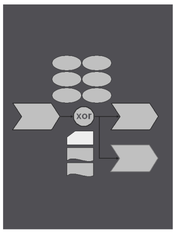
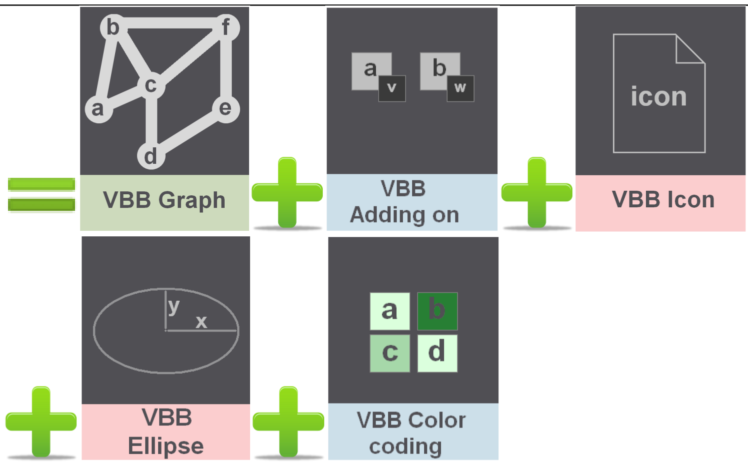
	Viewpoint	Decomposition into building blocks	New VBB candidates
28			Structural VBBs → VBB Scatter plot
29			Structural VBBs → VBB Line chart
30			Structural VBBs → VBB Line chart
31			Structural VBBs → VBB Line chart
32			Structural VBBs → VBB Line chart
33			

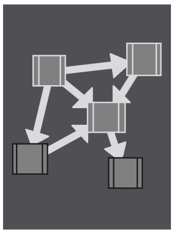
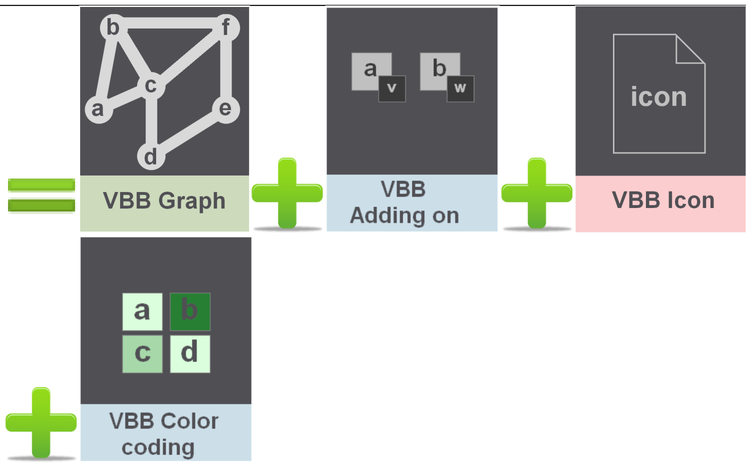
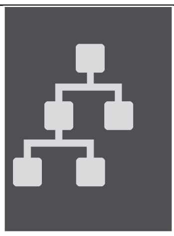
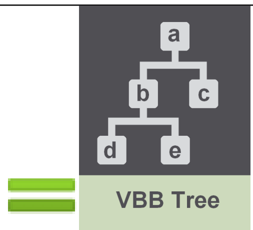
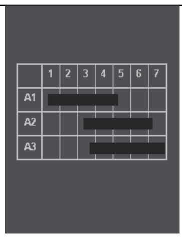
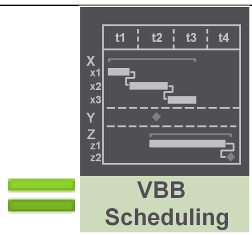
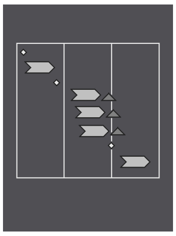
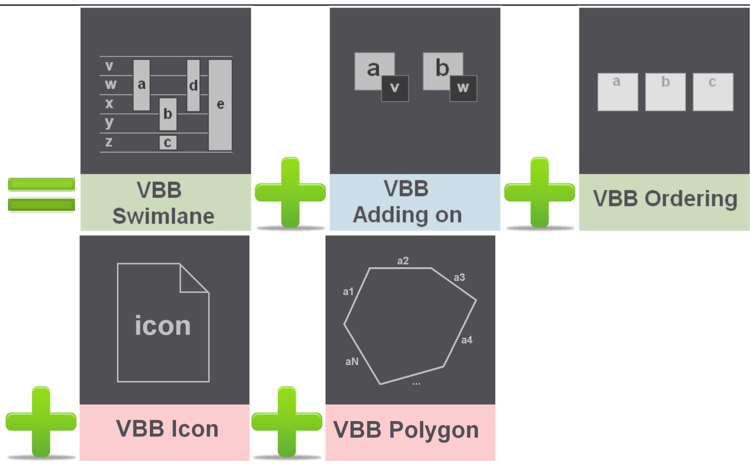
	Viewpoint	Decomposition into building blocks	New VBB candidates
34			Symbol VBBs → Compound chart symbols → VBB Pie chart → VBB Doughnut chart
35			1.Structural VBBs → VBB Radar chart 2.Symbol VBBs → Shapes → VBB Polygon
36			Structural VBBs → VBB Radar chart
37			Structural VBBs → VBB Regionalization
38			Structural VBBs → VBB Matrix

	Viewpoint	Decomposition into building blocks	New VBB candidates
39			Structural VBBs → VBB Matrix
40			Structural VBBs → VBB Matrix
41			Structural VBBs → VBB Matrix
42			1.Structural VBBs → VBB Matrix 2.Structural VBBs → VBB Grouping

	Viewpoint	Decomposition into building blocks	New VBB candidates
43			1.Structural VBBs → VBB Matrix 2.Structural VBBs → VBB Grouping
44			Structural VBBs → VBB Matrix
45			Structural VBBs → VBB Matrix
46			1.Structural VBBs → VBB Matrix 2.Symbol VBBs → Compound dashboard symbols → VBB Smiley
47			1.Structural VBBs → VBB Matrix 2.Symbol VBBs → Compound dashboard symbols → VBB Trend arrow

	Viewpoint	Decomposition into building blocks	New VBB candidates
46			Symbol VBBs → Compound dashboard symbols → VBB Traffic light
49			Symbol VBBs → Compound dashboard symbols → VBB Progress bar
50			Structural VBBs → VBB Matrix
51			Structural VBBs → VBB Matrix
52			Structural VBBs → VBB Matrix
53			1.Structural VBBs → VBB Graph 2.Symbol VBBs → VBB Icon

	Viewpoint	Decomposition into building blocks	New VBB candidates
54			1.Structural VBBs → VBB Graph 2.Symbol VBBs → Shapes → VBB Polygon 3.Symbol VBBs → Shapes → VBB Rounded rectangle 4.Symbol VBBs → Compound dashboard symbols → VBB Smiley
55			1.Structural VBBs → VBB Graph 2.Symbol VBBs → VBB Icon → Chevron, Xor
56			1.Structural VBBs → VBB Graph 2.Symbol VBBs → VBB Icon → Chevron, Xor, Curved rectagle, Cutted rectangle

	Viewpoint	Decomposition into building blocks	New VBB candidates
57			1.Structural VBBs → VBB Graph 2.Symbol VBBs → VBB Icon → Framed rectangle
58			
59			Structural VBBs → VBB Scheduling
60			1.Symbol VBBs → VBB Icon → Framed rectangle 2.Symbol VBBs → Shapes → VBB Polygon

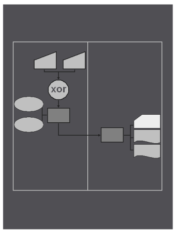
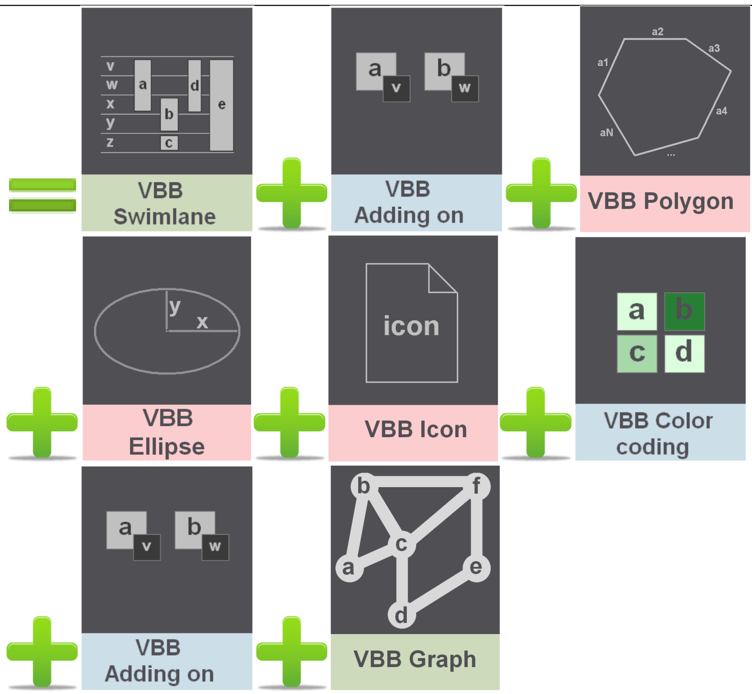
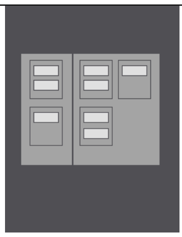
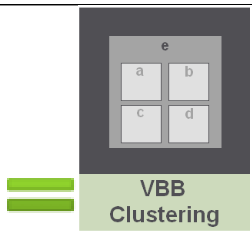
	Viewpoint	Decomposition into building blocks	New VBB candidates
61			1.Structural VBBs → VBB Graph 2.Symbol VBBs → Shapes → VBB Polygon 3.Symbol VBBs → VBB Icon → Xor, Cutted rectangle, Curved rectangle
62			

Table B.1.: Decomposition of viewpoints used by a real-life global financial service provider

C. List of abbreviations

BEAMS	Building Blocks for Enterprise Architecture Management Solutions
DFD	Data Flow Diagrams
EA	Enterprise Architecture
EAMPC	Enterprise Architecture Management Pattern Catalog
EBITDA	Earnings Before Interest, Taxes, Depreciation, and Amortization
EV	Enterprise Value
IBB	Information model building block
IEEE-SA	Institute of Electrical and Electronics Engineers Standards Association
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
IT	Information technology
KPI	Key performance indicator
LBB	Language building block
MBB	Method building block
SE	Software engineering
UML	Unified modeling language
VBB	Viewpoint building block
VCBB	Viewpoint catalog building block

Bibliography

[Ab08] Abela, A.: *Advanced Presentations by Design: Creating Communication that Drives Action*. Pfeiffer, 2008.

[Al77] Alexander, C.: *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, 1977.

[Ac10] Achenbach, P.: *Eine domänenspezifische Sprache zur Generierung von Sichten der Unternehmensarchitektur*. Bachelorarbeit, Technische Universität München, Fakultät für Informatik, 2010.

[AW10] Alexander, M.; Walkenbach, J.: *Excel Dashboards and Reports*. John Wiley & Sons, 2010.

[Be04] Beyer, N.: *Kennzahlen zur Beschreibung von Anwendungslandschaften und ihre Visualisierung auf Softwarekarten*. Bachelorarbeit, Technische Universität München, Fakultät für Informatik, 2004.

[Be08] Behrens, C.: *The Form of Facts and Figures: Design Patterns for Interactive Information Visualization*. Masterarbeit, Fachhochschule Potsdam, 2008.

[BMS10] Buckl, S.; Matthes, F.; Schweda, C.M.: *Utilizing Patterns in Developing Design Theories*. In: 2010 International Conference on Information Systems (ICIS 2010), St. Louis, MI, USA, paper 122, 2010.

[Br05] Brendebach, K.: *Integrierte Modelle und Sichten für das IT-Management: Analyse und Entwicklung in Zusammenarbeit mit der HVB Systems*. Diplomarbeit, Technische Universität München, Fakultät für Informatik, 2005.

[Bu08] Buckl, S. et al.: *Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008)*. Technical Report TB0801, Chair for Informatics 19 (sebis), Technische Universität München, 2008.

[Bu07] Buckl, S. et al.: *A Pattern based Approach for constructing Enterprise Architecture Management Information Models*. In: 8. Internationale Tagung Wirtschaftsinformatik, Karlsruhe, p. 145-162, 2007.

[Bu10] Buckl, S. et al.: *Building Blocks for Enterprise Architecture Management Solutions*. In: Harmsen, F. et al. (Edt.): Practice-Driven Research on Enterprise Tramsformation, second working conference, PRET 2010, Delft, Springer, Lecture Notes in Business Information Processing (LNBIP), Vol. 69, p. 17-46, 2010.

[Bu11] Buckl, S. et al.: *A meta-language for enterprise architecture analysis*. In: 16th International Conference on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD 2011), London, United Kingdom, 2011.

[De06] Dern, G.: *Management von IT-Architekturen: Leitlinien für die Ausrichtung, Planung und Gestaltung von Informationssystemen*. 2., verbesserte und erweiterte Auflage, Friedr. Viewweg & Sohn Verlag, 2006.

[Er08] Ernst, A.M.: *Enterprise Architecture Management Patterns*. In: Proceedings of the 15th Conference on Pattern Languages of Programs (PLoP'08), ACM New York, NY, USA, 2008.

[Fe04] Few, S.: *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. Analytics Press, 2004.

[Fe09] Few, S.: *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press, 2009.

[IE00] IEEE: *Systems and software engineering - Recommended practice for architectural description of software-intensive systems*. The Institute of Electrical and Electronics Engineers, Inc., 2000.

[Kr05] Krcmar, H.: *Informationsmanagement*. 4., überarb. und erw. Auflage, Springer, 2005.

[La08] Lankes, J.: *Metrics for Application Landscapes: Status Quo, Development, and a Case Study*. Dissertation, Technische Universität München, Lehrstuhl für Informatik 19, 2008.

[LLS10] Laudon, K.C.; Laudon, J.P.; Schoder, D.: *Wirtschaftsinformatik: Eine Einführung*. 2., aktualisierte Auflage,Pearson Studium, 2010.

[Lu04] Luftman, J. N. et al.: *Managing the Information Technology Resource: Leadership in the Information Age*. Pearson Prentice Hall, 2004.

[Ma08] Matthes, F. et al.: *Enterprise Architecture Management Tool Survey 2008*. Technische Universität München, Chair for Informatics 19, Prof. Matthes (sebis), 2008.

[Mo09] Moody, D.: *The 'Physics' of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering*. In: IEEE Transactions on Software Engineering, Vol. 35, Nr. 6, Nov/Dec 2009.

[MJ09] Myatt, G.; Johnson, W.: *Making Sense of Data II: A Practical Guide to Data Visualization, Advanced Data Mining Methods, and Applications*. Wiley, 2009.

[My11] Mykhashchuk, M. et al.: *Charting the landscape of enterprise architecture management - An extensive literature analysis*. In: 10.Internationale Tagung Wirtschaftsinformatik 2011, Zürich, Switzerland, 2011.

[OMG06] Object management group (OMG): *Meta object facility (mof) core specification, version 2.0*. `http://www.omg.org/spec/MOF/2.0/PDF` (cited 2011-08-23). 2006.

[Po98] Porter, M. E.: *Competitive Advantage: Creating and Sustaining Superior Performance*. Free Press, 1998.

[Se05] Sekatzek, P.: *Visualisierung von IT-Bebauungsplänen in Form von Softwarekarten - Konzeption und prototypische Umsetzung*. Diplomarbeit, Technische Universität München, Fakultät für Informatik, 2005.

[Se11] Sebis: *Building Blocks for Enterprise Architecture Management Solutions (BEAMS)*. Technische Universität München, Lehrstuhl für Informatik 19. `http://wwwmatthes.in.tum.de/wikis/beams/home` (cited 2011-08-15). 2011.

[Tu01] Tufte, E.: *Visual Display of Quantitative Information*. Graphics Press, 2001.

[Tu90] Tufte, E.: *Envisioning Information*. Graphics Press, 1990.

[WGK10] Ward, M.; Grinstein, G.; Keim, D.: *Interactive Data Visualizations: Foundations, Techniques, and Application*. A K Peters Ltd, 2010.

[Wi07] Wittenburg, A.: *Softwarekartographie: Modelle und Methoden zur systematischen Visualisierung von Anwendungslandschaften*. Dissertation, Technische Universität München, Lehrstuhl für Informatik 19, 2007.

[WP02] Ward, J.; Peppard, J.: *Strategic Planning for Information Systems*. 3rd edition, John Wiley & Sohns, 2002.