Constructive Visualization

Samuel Huron, Sheelagh Carpendale, Alice Thudt, Anthony Tang, Michael Mauerer

Constructive Visualization

Samuel Huron[1,2], **Sheelagh Carpendale**[2], **Alice Thudt**[2], **Anthony Tang**[2], and **Michael Mauerer**[2]

[1]IRI, 4 rue aubry le boucher, Paris, France
[2]University of Calgary, Department of Computer Science, Calgary, Canada
samuel.huron@cybunk.com, {sheelagh, tonyt}@ucalgary.ca, {alice.thudt, damischael}@googlemail.com

ABSTRACT

If visualization is to be democratized, we need to provide means for non-experts to create visualizations that allow them to engage directly with datasets. We present *constructive visualization* a new paradigm for the simple creation of flexible, dynamic visualizations. Constructive visualization is *simple*—in that the skills required to build and manipulate the visualizations are akin to kindergarten play; it is *expressive*—in that one can build within the constraints of the chosen environment, and it also supports *dynamics* — in that these constructed visualizations can be rebuilt and adjusted. We describe the conceptual components and processes underlying constructive visualization, and present real-world examples to illustrate the utility of this approach. The constructive visualization approach builds on our inherent understanding and experience with physical building blocks, offering a model that enables non-experts to create entirely novel visualizations, and to engage with datasets in a manner that would not have otherwise been possible.

Author Keywords

Design; Visualization; Construction; Assembling; Constructivism; Constructionism; Education; Visual literacy.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI)

INTRODUCTION

In this paper, we focus on supporting the democratization of visualization [72], which is in contrast to conventional visualization work where the focus has been to support data intensive science, industry and government. We are now seeing visualizations in such places as: personal blogs [36], as part of art works [42, 56, 73], in the news media [35, 66, 77], and as a growing part of the quantified self movement [4]. These visualizations range from the intensely personal such as the visualization of the contents of one's freezer [71], to community-based visualizations such as crime in one's neighborhood [71]. This movement opens several research questions. If democratization means that people (not just experts) will design and construct their own visualizations, what kinds of tools will these people need? What questions are they asking, and how will they explore the data that interests them? We investigate these questions by exploring situations where simple tools are being used to creatively and effectively construct complicated structures, in particular examining how these ideas arose in kindergarten education. Our goal is to explore educational theory to learn how to design tools that support non-professionals in the creation of visualizations.

The democratization of visualization is readily motivated by our current data deluge. Similarly to the era following the invention of the printing press, where information became available in unprecedented amounts, response is both positive and negative. While in the media the phrase 'big data' is discussed as an unparalleled opportunity, people individually often think of this as overwhelming. Our focus here is to look at how to make this data deluge tractable to us as individuals. For example, Grammel et al. [29] show us that non-computer scientists have considerable trouble when designing visualizations particularly when selecting good data attributes, formulating visual mappings, and interpreting the visualizations produced. We examine how to address this problem by changing the design paradigm through which people manipulate the data in conjunction with visual variables to construct a visual mapping. To this end, we take a fresh look at theories that speak to how people understand concepts that are new to them. The current way to support people in the creation of visual representations is to develop code libraries, toolkits, or create visualization templates and provide an infrastructure with which the created visualizations can be shared. In this paper we discuss an alternate approach: we consider how people can craft visualizations by using familiar elements. We build our perspective on the observation of practices in non-academic situations where people are actively engaging in the construction of their own visualizations in spite of the fact that there is a lack of tools supporting these kinds of activities. Our contribution is to propose a new visual mapping paradigm, to relate it to existing theories, and to discuss how it opens new directions for visualization design.

In this paper we start by exploring the idea of democratization of visualization design: we identify and compare three existing paradigms in which one can create a visual mapping. Then we explain how educational approaches from Froebel, Piaget, and Papert point the way to a new design paradigm, which we term "constructive visualization". Then we define the components, processes and benefits of constructive visualization. Finally we illustrate it through real world non-academic examples and discuss the implications for developing new designs and research.

http://dx.doi.org/10.1145/2598510.2598566

TOWARDS DEMOCRATIZATION OF VISUALIZATION

The InfoVis research community has focused on making visual representations of data and on supporting data tasks via interaction. A considerable emphasis has been placed on leveraging perceptual skills to improve readability of visual representations [76], studying the effectiveness of visualization in regards to the intended task [9], and making progress towards assessing insight triggered through use of visualizations [63]. A major underlying goal of much of this work has been to empower information workers and data experts.

Viegas et al.'s discussion about the democratization of visualization [72] provides a fresh perspective on how InfoVis could impact society more broadly if the tools to build and use visualizations are accessible to those other than information workers. To this end, they identify three major problems that should be addressed in order to make InfoVis accessible, understandable and beneficial for the general population:

- The ability to *create* one's own visualizations.
- The ability to *publish* or make these visualizations generally available.
- The possibility for *discussion* of these visualizations.

The ManyEyes [72] project was an initial exploration in this direction. It provides facilities to upload one's data, choose from a variety of InfoVis templates to *create* a visualization of one's own data through a common website. The created visualization is then automatically *published* online, and an associated *discussion forum* is automatically generated for each of the visualizations on the site. This site has been successful and well used, but it limits possible visualization variations to the given set of templates.

Victor's [70] discussion on the creation of visualizations sheds new light on challenges to be overcome in the pursuit of the democratization of visualization, focusing on the issue of *creation*. He identifies three paradigms that people use to create visualizations:

- *Using* a pre-coded visualization (as a template),
- *Drawing* a visualization freehand, and
- *Coding* a visualization through computer programming.

He argues that each of these approaches has pitfalls. The first one, using a *pre-coded visualization*, describes the ManyEyes [72] solution where one can choose from existing InfoVis templates (e.g. bar chart, scatter plot, etc.). The advantages here are that these templates can be well known, well understood and even well researched in terms of readability, etc. The disadvantages are that data is usually unique and often has distinct needs to achieve best results in terms of possible understanding and insight. Also using established templates can limit natural human creativity. This type of approach is exemplified by software such as Excel, Tableau [6], Spotfire [5], and also web tools like ManyEyes or Chart Editor in Google Docs [2]. In these cases variations in results are limited by the set of predefined mapping functions and the possible options to tune their parameters. These are great in that they are easy to use but they can limit the power of expression. The second approach is the use of free hand *drawing*. Victor includes within drawing that done by hand with a pen and a paper, on a whiteboard, as well as that done through using drawing software such as Photoshop or Illustrator. It is well accepted and studied that people often use freehand drawing on napkins [19], sketchbooks [30], and whiteboards [75, 74] to help them think visually [45, 11] about ideas and to create ad hoc data representations for thought and discussion purposes. With freehand drawing people have great creative and expressive freedom, which is somewhat limited by software drawing packages. However, the real limitation is the same for both: the results are static. There is no support for data dynamism, temporal changes or trends. Drawing examples include everyday ad hoc sketches and relatively famous carefully articulated ones: Napoleon's March [46], and rock music histories [1]. The third approach Victor describes is the use of *code*. Coding is probably the most common way to create new information visualizations in the research community. Arguably, coding offers considerable creative freedom in the ability to tailor visualizations directly for specific data and even for specific data tasks. Also, it can support data dynamics. In terms of democratization of visualization, the limitation is in the accessibility of the process. Coding is a skill that must be learned and is not the kind of skill that everyone possesses or that everyone would be able to take the time to develop. This is the challenge we address: Can we offer the power, expressive creative freedom and the ability to support data dynamics without requiring people to learn to code? With this basic challenge in mind we develop our design goals and then explore the ideas from theory of mind that can help us unlock this challenge.

DESIGN CHALLENGES

As noted above, we focus on the creation of the visualization from the data, or, in other words, how the mapping from the data to the visual structure is done. In this section, we first specify how this creation aspect of the visualization process relates to the InfoVis reference model [17] and Bertin's Semiology of Graphics [12]. Then, combining this InfoVis perspective with the ideas from the discussion on democratization, we define three design challenges that need to be addressed to democratize the InfoVis design.

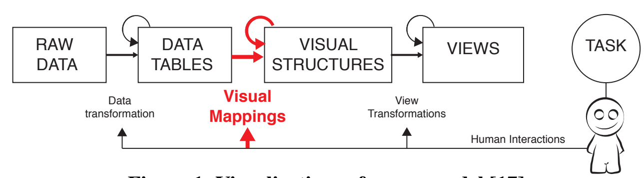


Figure 1. Visualization reference model [17].

The information visualization process has been previously modeled as a sequence of data and visual transformations in several steps. This process, known as the "infovis reference model" (figure 1), has been described by Card et al. [17], Chi and Riedl [21], refined by Carpendale [18] and extended by Jansen and Dragicevic [40]. All these models share a visual mapping process. The *visual mapping* is the process which transforms the data tables into visual structures. Card et al. [16] define this visual structure as set of marks (Point, Line, Area, Surface, Volume), their retinal encoding (Color, Size, Shape, Gray-level, Orientation, Texture, Connection, Enclosure), and their positions (X,Y,Z,T). The paradigms to

Visual mapping paradigm	Keep it Simple	Enabling expressivity	Incorporating data dynamics	Manipulation of Visual	Skills learning	CS System examples	NofP
Using	✓	✗	✓	Indirect	Medium	Excel, Spotfire [5], Tableau [6], Google Chart [2] ...	36
Drawing	✓	✓	✗	Direct	Easy/Med	Pen and paper / Illustrator, Photoshop, ...	13
Coding	✗	✓	✓	Indirect	Hard	Processing, Infovis toolkit [24], Prefuse [31], D3.js [14]...	17
Constructing	✓	✓	✓	Direct	Easy	Unknown	0

Table 1. Table of existing information visualization design paradigms in comparison to constructive visualization paradigm. NofP: Number of Papers referenced by Grammel et al. [28] distributed across Victor's [70] paradigms and including the option of constructive visualizations.

create visualizations discussed by Victor specifically examine this visual mapping process and the type of visual structure that is produced. It is this mapping that transforms the data tables to visual structures that we explore. Bertin's [12] definition of graphics as a monosemic system of signs, is particularly illuminating for explaining the visual mapping process. We identify four major actions in his definition: (1) the attribution of a signification (data properties) to a visual sign, (2) an agreement between people on the signification of this visual sign, (3) the assembly of signs, and (4) the possibility of discussing and analyzing the assembly of these signs according to this agreement. According to the context and the need, this visual mapping process could be done by a designer, or by the viewer, or defined by a programmer and done computationally. Bertin's definition is particularly useful because it opens up this data to a visual mapping process specifying details that can be explored with the ideas in Victor's visualization creation paradigms. Based on these definitions and Victor's paradigms of visualization creation we define three designs challenges, *DC*, for democratizing the visual mapping in infovis. If a new paradigm meets each of these challenges, then we may have an effective paradigm for non-computer scientists.

DC1: Keeping it simple. Here we see a strong link between simplicity and accessibility. If the actions one needs to take are similar to actions one has been comfortable with since Kindergarten, they are both simple and accessible. A good example of this is sketching, for which one of the best advantages is that we all can do it. We may not be artists or designers but making use of a few quick lines on any available scrap of paper to help us work out or explain an idea is accessible to all of us. The challenge then is to find creation activities that can be linked to data that are routed in deeply familiar activities.

DC2: Enabling expressivity. We are looking for a creation process that provides sufficient freedom to support the ability to express one's ideas. Expressivity can be defined by three dimensions according Bertin's vocabulary, (DC2.1) the degree of freedom in defining the sign, (DC2.2) the degree of freedom in attributing a signification (data properties) to a sign, and (DC2.3) finally the degree of freedom by which these signs can be assembled. Ideally this would include flexibility, plasticity and freedom to manipulate: the ideal creation process would include the possibility of incorporating changes without damage, the ability for the representation to be easily molded and remolded, providing the freedom to readily make changes. Our ideal is to support the expressivity of sketching and the flexibility of digital tools by incorporating the concept of plasticity, or the ability to re-model during the creation process.

DC3: Incorporating dynamics. One of the biggest challenges of making visualization creation more generally accessible is that thus far only through code can a visualization support adaptability to data dynamics—that the visualization can change in response to a change in the data steam. Coding remains, and is likely to remain, a skill of comparatively few. We explore the possibility of approaching this from a constructive perspective. Data dynamics incorporation can be defined with Bertin's vocabulary applied to updating the three dimensions of expressivity: (DC3.1) defining a new sign (DC3.2) updating the attribution of signification to a sign (DC3.3) updating the assembly of signs. This is a major challenge because while Victor may provide the motivation and Bertin may offer a useful vocabulary, our challenge is to develop an accessible process that incorporates dynamics.

INSPIRATION FROM FROEBEL, PIAGET AND PAPERT

Here we discuss the ideas from three leading thinkers, Froebel, Piaget, and Papert, to show how the development of constructivist approaches can shed light on our challenges.

Froebel – Discovering the world through simple units

Thanks to Froebel's ideas, for almost two centuries Kindergarten has been a place for children to learn complex and abstract concepts such as math and geometry by playing and manipulating objects. Froebel's legacy has extended beyond educators to architects [44], designers [23, 68] and computer scientists [61, 60, 59]. Froebel's Kindergarten used pedagogic activities called "Gifts". Aleeb-Lundberg [8] describes these activities as a series of geometrical primitives or building blocks presented to a child in a sequence. Each "gift" comes with associated manipulation methods that are designed to teach simple mathematical operations such as sorting, counting, adding, subtracting and fractions. It is this idea that simple manipulation of blocks can illustrate mathematical concepts (Figure 2), that we extend to constructive visualization.

Piaget – Learning by construction

Piaget used similar building blocks to the ones designed by Froebel [3, 55] in his experiments to study the cognitive processes of children. According to Piaget manipulating and experimenting with physical objects is the main way in that children learn. Piaget provides a solid framework that helps us understand the learning stages during children's cognitive

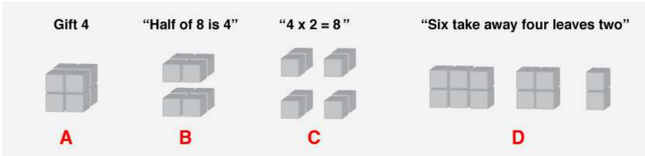


Figure 2. Illustration of Froebel Gift 4 from the manual for teachers [15]: A. Set of blocks in Gift 4, B. Operation showing division by two, C. Operation to teach multiplication. D. Operation to teach subtraction.

development. More recently, Chapman's studies [20] have shown how this ability to grasp new ideas through construction applies to people of all ages, not just children.

Papert – Constructionism applied to programming

Building on constructivist theories [53], Papert [51] extended the idea of pedagogical manipulative materials to computer programming. The first result of this fruitful approach was the programming language Logo [52]. Then, Papert founded a research group at MIT Media Lab, whose name was inspired by the Froebel system. The "Lifelong Kindergarten Group" published several major works such as Scratch [60, 43, 59] and Mindstorm [51]. Scratch follows this inspiration by transforming the building block idea into a visual representation of the "command block". These blocks represent variables, statements, expressions and control structures; the essential process is to snap it together to do the programming. This approach was so successful, that Scratch now has more than a few hundred thousand followers and its logic is integrated in the programming interface of commercial products such as Lego Mindstorm.

Applying these Lessons to InfoVis

From Froebel, Piaget and Papert we learn: (1) that understanding of abstract and mathematical concepts can be developed through the manipulation of simple elements such as wooden blocks, balls, etc. (→ DC1: keeping it simple); (2) that this approach has continued to prove accessible and effective as it has spread across the world, and is still in use today (→ DC1: accessibility); (3) that this approach is highly creative and generative (→ DC2: expressivity); (4) that this approach also allows people to modify and understand their constructions over time (→ DC3: dynamics).

To put this in the context of visualization, Froebel's gifts relate to Bertin's definition of visualization a monosemic system. The blocks are the signs, the significations are the numeric units, the agreement of the attribution between sign and signification is defined in the manual for teachers or in visualization by the mapping between the visuals and the data. Since the assembly of blocks provides a unique way to learn numeracy, the assembly of signs that signify data may let us learn about data.

ELEMENTS OF CONSTRUCTIVE VISUALIZATION

In this section we incorporate lessons from Froebel, Piaget and Papert to define a new method for visualization design: constructive visualization. As an operational definition, the constructivist approach to designing information visualization is the act of constructing a visualization by assembling blocks, that have previously been assigned a data unit through a mapping. We explain the central idea of manipulating building blocks for information visualization in two parts: first we present the various components (figure 3) that comprise this approach; second, we describe the processes through which these components can be used to assemble visualizations specifically by non-experts.

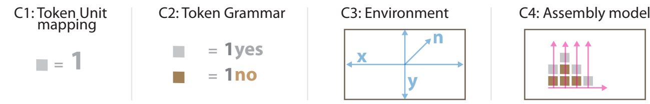


Figure 3. Example constructive visualization components.

Components:

C1: The basic unit: a token – We will call the basic unit a token, which, in visualization, is a discrete visual mark representing a data unit. This token can be physical or virtual, can have any type of shape, volume, surface, area, texture, or colour, etc.. A token can also have an assigned visually interactive direct manipulation functionality, for example, drag and drop, select and move. In the visualization, tokens represent data elements, though only after a data mapping has been assigned. For instance, a Lego block can be a token as it is a discrete unit, has a visual manifestation and supports direct manipulation. This token parallels Bertin's mark in his vocabulary of graphics, being a visual sign that can assigned varying visual properties and data mappings.

C2: Token Grammar – The token grammar defines a set of different types of tokens, and the mapping of their visual properties to aspects of the data. For example, the size or position of a token can be mapped to the quantity of the data, while the colour of tokens might be mapped to different aspects of the data. This mapping between properties of the tokens and aspects of the data is made by the individual constructing the visualization. According to Bertin's vocabulary, the token grammar could be considered as the list of semantic assignations of each visual variable.

C3: Environment – The environment is the space that provides constraints on how tokens can be assembled together using the token grammar. The properties of this space could include many different types of constraints such as gravity, 2D, 3D, space limitations, grids or others. These constraints can provide a structure to help the assembly or define limits of how tokens can be assembled. According to Bertin's vocabulary, the environment is what he terms the graphic system, though we also include use of volumes (in tangible and 3D environments) as well as 1D and 2D environments.

C4: Assembly model – The assembly model defines the rules of the construction process. This is the internal model of how the constructing and deconstructing of the visual representation is carried out. The final result contains the data articulated and represented through the assembly of the tokens.

Process:

P1: Environment initialization. The first step involves the choice and establishment of the environment with its associated constraints and its relationship to the tokens.

P2: Mapping data to "tokens", and data properties to token properties. Decisions must be made about how to assign one or several units of data values to one or more token (this comprises the token grammar). Furthermore, data properties must also be assigned to the token properties, for example, relating position to some quantity/property of the data.

P3: Assembling the tokens. Assembling consists of manipulating the tokens in a way that is valuable for those who are involved in the activity. This assembly may be for the purpose of inspection, exploration, or visualization. This assembly occurs in the environment, which may be defined in such a way to constrain the space of possible assembly methods, or allow for entirely free construction.

P4: Evolution over time. The initial assembly constitutes a single state of the visual representation. This state can be updated as needed by a person, group of people or computer algorithm, depending on how the environment is defined.

EXAMPLES

While our notion of constructive visualization is not yet in use digitally (table1), there are examples in the physical world. To provide a better understanding of how this visual mapping paradigm can work in practice we describe four real world examples. This illustrates the wide space of applicability, expressive and generative capabilities of constructive visualizations. We explain how each example implements the process of constructing the visualization and how each fulfills our three design challenges.

Chris Jordan: Statistics as Engaging Art Pieces

Jordan is an artist who provokes thoughtfulness about everyday environmentalism. Jordan explains [65]: "what I'm trying to do with my work, is to take these numbers, these statistics from the raw language of data, and to translate them into a more universal visual language, that can be felt". He starts by choosing an object to illustrate a unit of the statistics (→token), then he assembles it (→assembly model), and photographs his results. For instance, the image in Table Table 2 line 1, shows a part of an image of the number of plastic cups used every six hours on airline flights in the US (1 million plastic cups). Jordan does not create a mapping from data to the constructive element; instead he uses the data item as the token or basic construction unit. His pieces are made with the appropriate number of data items for the specific statistic he has chosen to reveal. Since this practice makes for enormous constructions, Jordan's art works are ultimately photographs of these constructions. He uses constructions of basic data tokens to communicate statistics in a manner that is understandable to everyone and has emotional impact.

Otto Neurath: Communicating Statistics to People

Neurath [49, 50] was a professor of political economics in Europe during the last century. His firm belief that information such as scientific results and statistical data should be accessible to everyone led him to create infographics to inform all citizens and school children about their position in the world according to statistics. There are three basic principles in his assembly model: one, show numeracy via countable units (→token); two, employ simple pictograms as the countable units using each pictogram to encode different information (→token grammar); and three, position these pictograms in comparable layouts (→assembly model). Neurath's work is internationally successful and still used in infographics to represent numerical values understandably. This example shows usage of the constructionist principle applied to printed graphics, which suggests that the approach is transferable to a 2D system.

Michael Hunger: Personal Time Management

Hunger created a tool to help himself manage (record, report, and plan) his time effectively when working on many different projects. To find a solution to his time management problem, he did an extensive review of software and artifacts that have been designed for this purpose including spreadsheets, browser based time tracking, Outlook, popup applications, as well as tangible solutions such as diaries, sticky notes, paper, tally sheets, and notebooks. However, since none of these solutions worked well for him, he decided to design his own personal time management system which he could tailor to his needs. He created his time management tool out of Lego bricks (Table 2,S3) as follows: on top of a Lego baseboard, he placed five Lego row-bricks (8x1 pins)(→token). In his *token grammar* each row-brick represents a day of the week differentiated by colour (red for Monday, then orange, etc.). Then he partitioned the time into quarters of an hour, using a 4 pin brick. The colour of these hourly 4 pin bricks encodes the project on which the time was spent. These hourly rows are stacked on top of each other to represent the amount of work done during a day (→assembly model). For Hunger the benefits of his technique are obvious. It helped him where none of the previous solutions he had tried had and after four months he is still using it. In addition he says he finds it playful, pleasant, and fast with little to no overhead and that it allows him to report his time used and to plan his work time. To summarize, Hunger invented his own constructed tangible visualization. It provides him with an easy assembly process where Lego bricks are his tokens (simplicity). These Lego tokens offer considerable building freedom (→expressivity) and allow him to update his visualization at will (→dynamics).

Kevin Quinn: Problem Resolution Tracking

Quinn [7] is an engineer in the automobile industry. His job is to manage the vehicle engineering operations crossover team. Having an overview of the full car production process is a necessity for him. He needs to understand if there is a problem in the production process, and to know which resource to allocate to each part of the process. While he was provided with a visualization to work with, he and his colleagues found the visualization frustrating for several reasons: the visualization did not show what they "really needed to see", and that they could not "grasp it or reshape it"(→DC1,DC3). To address their need for reshape-able information overview, they designed a visualization based on a Lego board (2). The Lego board is used as an array (*the assembly model*) according to the following: the horizontal axis corresponds to the time of production - each column is a week, and the last two columns are dedicated to canceled and closed issues. Each row corresponds to a special area of the vehicle (e.g. Body, Chassis, etc.), and the last row to cost savings. *The token grammar* is defined as follows: each Lego brick represents an issue in the vehicle; the size of the block represents the severity of the problem; and the colour the area of the vehicle. The tokens are labeled with the ID of the issues and a progress bar

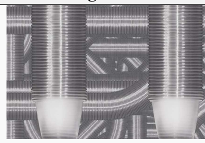
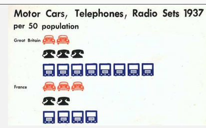
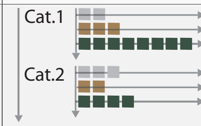
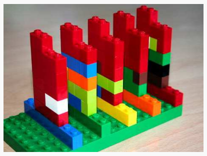
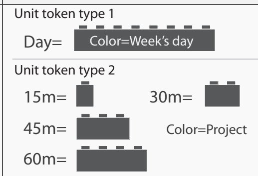
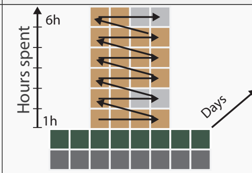
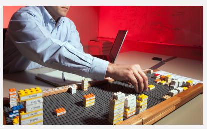
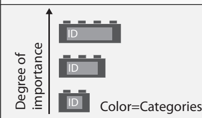
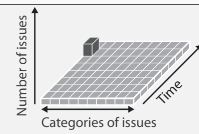
Examples	Picture fragment	Type	C1: Token	C2: Token Grammar	C4: Assembly model	C3: Environment
1. Chris Jordan		Artistic	Object / picture of object	1 picture: = 1 plastic cup, used on airline flights in the US during last six hours	Artistic, the assembly model in this case does not follow the definition of a monosemic system. The assembly is not described as processing the data, but as providing a feeling about the data.	2D Paper canvas
2. Otto Neurath		Analytic	Pictogram	3 Pictograms: = 1 car per 50 people, = 1 bus per 50 people, = 1 phone per 50 people		2D Paper canvas
3. Michael Hunger		Analytic	Lego bricks			3D Physical tangible Lego board
4. Kevin Quinn		Analytic	Lego bricks			3d Physical tangible Lego board

Table 2. Four real world analytic and artistic examples of constructive visualization made from assembling unit tokens, and their respective components. Picture 1: ©Chris Jordan, Picture 2: extracted from [50], Picture 3: ©Michael Hunger, Picture 4: ©General Motors Cf. `http://goo.gl/zMFK6E`.

is printed on the side. The company has adopted this system, and Quinn and colleagues continue to work with the Lego visualization board. For instance, if a special part of the vehicle such as the brake does not function properly during the durability testing, the person in charge will make a paper report as usual and also provide hand-operated data dynamics by adding a Lego brick to the board, with the ID of the report on the side. One colleague said: "The teams either want to see their Legos moving in a positive direction or have a solid action plan for addressing one that is red".

Learning from these Examples

Each of these examples is simple to construct and to read. Each also exhibits the principles of constructive visualization, as is summarized in Table 2, though each does so in a different way, providing instances within the broader design space. In this subsection we evaluate the constructive visualization paradigm using these examples by discussing how they address the design challenges outlined earlier.

DC1: Simplicity - All four examples address this design challenge in the following way: (1) they have been made by people who are not visualization experts, they are simple to make, understand and reproduce. They have simple and rich token mappings (→C1), and grammar (→C2), and in each case, the correspondence between the token and the data unit is easy to understand and to assign. Moreover, the assembly model is understandable (→C3).

DC2: Expressivity - All examples address the expressivity design challenge in the following way: first, in each case the author was freely allowed define the signs; second, the data attribute mapping to these signs was also something the author could decide, and finally the author could freely assemble these signs. Jordan and Neuraths examples illustrate the first point, where they design their own signs/tokens (→C1=pictogram, object or picture) to express some dimension of the data. Hunger's token grammar (→C2) shows the second point: how with the same type of token (Lego brick), one can define a grammar with several types of semantics: a day token (ordered data) with a color for each day, and the quarter of hour token (quantitative data) with a color for each project. Even though Jordan takes artistic freedoms in his assembly model (→C3), the constructions can be made and understood by everybody, and this exemplifies our third point. Jordan's assembly model expresses only one data dimension, the magnitude of his data. Neurath shows at least two data dimensions: amount and categories. Hunger expresses three dimensions in his model: day of the week, time of the day, and identity of the project and Quinn shows four dimensions: time spent, category of the issue, the ID of issues, and issues' degree of completion. Finally, this diversity of token grammar shows the ease of assigning a data attribute (→DC1).

DC3: Dynamics - Hunger's and Quinn's examples address this design challenge in the way that they create their assembly model (→C3) to support collective or individual updates. Hunger and Quinn both use their visualization as an input method of keeping track of their data over time and as a source of information both to make reports and to inform decisions. They both update their visualizations several times in a day by moving, adding, and removing tokens (→C1). In addition, Quinn's also updated by his group. They apparently do not update their token grammar (→C2). However, the possibility of dynamics (→DC3) is also dependent on the environment. While Lego is updatable and adjustable, photographs and printed graphics are less so. Reproducing Neuraths approach computationally could enable dynamics.

RELATED WORK IN VISUALIZATION

As mentioned earlier, the ideas of Froebel and Piaget have already inspired considerable research in computer science: learning programming environments [43, 52, 60, 59], computer graphics [68, 44], tangible interfaces [10, 54, 25, 58, 64, 65], and graphical user interfaces [41].

Research exploring tangible interfaces is also heavily influenced by this approach. Some refer to this idea in general [10, 25, 27, 44, 65, 58], while others specifically employ the building block idea [54, 27, 62, 65, 58, 78]. The latter generally use the blocks to simplify complex ideas for education, or focus on environments emphasizing construction: for example, to enable concrete manipulations on abstract concepts in math and programming.

Computer graphics research has also made use of ideas in constructive thinking. For example, Stiny [67] defined a computable shape grammar based on Froebel‘s Gifts [68] to assist computer-aided design. The generative properties have also appeared in procedural modeling [47], computer aided design [44] and mechanical engineering design [34]. Stiny‘s contribution was to define a constructive, computable language of shape designs, which aligns with our goals, though we apply this approach specifically to InfoVis.

The idea of making use of building blocks has been applied both as a metaphor, and as a tangible device. As described by Gentner [26], using a metaphor is a frequent and powerful human process. The history of use of metaphors in human computer interaction have been extensively studied by Blackwell [13], and his application to the domain of information visualization has been studied by Ziemkiewicz et al. [80].

InfoVis design focuses on the mapping between data and visual encoding (e.g. [12, 16, 22]). Designing information visualizations is a complex problem: the community has worked to create and study information visualizations [6, 5, 2] and built toolkits to ease their implementation [24, 14, 31]. However, the usual target audience for InfoVis design support are experts such as programmers, researchers, and scientists. A growing voice in the community is arguing that we need to challenge this approach and some are arguing the need to create new tools that allow a broader audience to create custom InfoVis design [33]. Others are discussing democratization [72], novice users [29, 32], and some use the phrase “casual viz” [56] to describe visualization as used in non-work environments. In spite of these calls for the articulation of new approaches for the design of these tools, InfoVis still primarily focuses on experts. Our work is also related to Grammel et al. [28]’s articulation of the “Visual Builder”. Visual primitives can be manipulated to customize the visual outcome of the data representation without writing code. Few existing systems allow for direct manipulation of visual entities directly bound to data. Existing visual builder provide only a rough approximation to this idea. For example, manipulating organizational elements such as data axes or data categories [79], or using heuristics to create visualizations out of a manually defined outline of data points [48]. Pretorius et al.’s [57] tool allows direct manipulation of visual primitives that can be assigned to data variables. While this approach meets our goal of direct manipulation of visual data representatives, it still requires the manual mapping of variables to visual primitives, which contradicts our goal of simplicity.

One of our goals is to extend Ullmer et al.’s ideas [69] to provide a deeper understanding of how constructive assembly can be applied to infoVis. For example, Huron et al. [37, 38], explore visualization of streaming data, where data is automatically chunked into tokens. Those tokens are essentially equivalent to our tokens, and their description of templates/-constraints maps to our concept of an assembling model. However, this work does not allow non-expert populations to actually design or construct their own visualizations.

DISCUSSION

To briefly recap, the initial call for democratization of visualizations suggested that the functionality to make this possible was the ability to create one‘s own visualizations, the option of publishing these visualizations and the possibility of sharing and discussing them with others [72]. Victor [70] raised questions about what it meant to be able to create ones own visualizations. He described the exiting tools as supporting three visualization creation paradigms: using existing templates, drawing freehand, and coding. However, if one thinks the visualization creation process should be simple, expressive and support data and interaction dynamics, the three existing creation paradigms fall short. On the other hand, our suggestion, *constructive visualization*, can offer all three: it is *simple* in that the basic skills are akin to kindergarten play, it is *expressive* in that one can build freely within the constraints of the chosen environment (e.g. if it is tangible one will have to contend with gravity), and since visualizations created in this manner can be rebuilt and adjusted, it also supports *dynamics*. However, while it does suggest a new visualization construction paradigm that is simple, expressive and supports dynamics, it does not solve any other long standing visualization problems.

Limitations

One of the strengths of constructive visualization is its reliance on direct manipulation of tokens as primitives, however, this does cause some of the typical visualization challenges to have a new twist. We will discuss the challenges of defining a data unit, handling data amplitude changes, data dynamics, and legibility by others.

Defining a data unit. When developing a visualization, one of the first steps is to define how data units map to visual tokens. If we provide constructive visualization tools, or even use Lego blocks, the problem of mapping data units to token definition must be handled and the choice will affect the usefulness of the resulting visualization. For example, consider a visualization of a home’s energy usage: should the unit be a joule, a kilowatt hour, “hours a light is left on”, or perhaps “carbon footprint”? The interpretability of this choice has an impact on how the visualization can be perceived. Layered atop this, should a unit token be mapped to a single unit of the measure, one hundred units, or a million units (e.g. watt hour vs. kilowatt hour)? There will be a new challenge in constructive visualization tools of how or whether to provide defaults or a sub-set of choices or some other solution. The way to optimally choose and define this unit remains an open research question.

Data amplitude changes. Visualizing data with extreme or rapid changes of data range or data amplitude can pose challenges to mapping data unit with token. Changing token unit mapping can radically change the meaning of the assembly model and create a lack of consistency. Alternatively, not making such changes could render data variations sub-visible. These types of visualization problems are usually approached through various non-linear presentations, however, such solutions are still under debate and are often discussed as not desirable for optimal readability. In addition, how such solutions could be incorporated into a constructive visualization environment remains an open question.

Data dynamics. In our physical examples, the data dynamics have to be executed by hand, however, in a software implementation there are different ways to automatically integrate data dynamics, for example, direct manipulation [39], or algorithmically placing or removing new blocks like in visual sedimentation [38].

Legibility by others. The way people assemble a visualization (i.e. C3: assembly model) is based on a set of choices made by those building the visualization. Some assembly models like Neurath's [49] make it easier to read and compare the quantities while for others this can be less apparent like Hunger's, Jordan's. Furthermore, some may adhere to conventional visualization approaches such as a bar chart, making themselves more easy to read by others.While it may be that environment constraints could assist the readability the specifics have yet to be studied, it is also possible that keeping the flexibility of the assembly model to enable freedom of expression, will remain an important tenet in this constructive visualization approach.

Applications

In our examples we have provided concrete illustrations of some applications of this approach, showing that it is general enough to be applicable in a number of application domains. For example, based on the background theory, an interesting application would be using these ideas to learn and teach visual literacy. The method can be applied to allow children, as well as adults, to learn about the relationships between visual variables and the data that underlies them.

The clear options for personalization can make this kind of approach appropriate for visualization of day-to-day social or personal data, where individuals can explore and manipulate data from their everyday lives. For instance, the Quantified Self community has focused on data collection (e.g. personal health), but this form of data manipulation and data visualization construction may help support this community to develop richer visualizations of personal data, and more importantly, perhaps provide a deeper understanding of their data.

We see this approach being amenable to integration with touch interfaces, tangibles, and multi-modal interaction. As these kinds of interfaces come with an inherent "environment" with constraints that we are familiar with (e.g. gravity, friction, etc.), this should help provide a frame for manipulation freedom, which would benefit the assembly task.

While we see a proliferation of this basic idea both among the research community, and among people in general, in that visualizations are becoming a more regular part of daily media, there are still stumbling blocks in the matter of widespread freedom to create visualizations. We can see through the ideas of Froebel, Piaget and Papert that creation through mapping data to unit tokens may be capable of offering this inclusion. We do think that we, the human computer interaction community, can realize this idea for a wide range of analytical and artistic visualizations.

Most visualization creation paradigms focus first on creating a data representation and then developing interaction to suit data needs and tasks. The basic approach for constructive visualization is different. The focus is on creating an interactive environment where people can assemble, from modular data-linked units, visualizations that directly fit their needs.

This paradigm reveals new perspectives on the visualization design process: (1) A new set of studies will need to be done to investigate this paradigm: How do people choose their unit blocks and why? How complex and generative are their visual grammars? How do people construct and code their assembly model? Is there a recurrent pattern in these assembly models? What are their limitations? (2) A new set of guidelines will be needed to drive innovative design of constructive visualization tools. (3) Based on these guidelines, new tools that support this approach could be produced to create, assist and support visualization construction.

CONCLUSION

In this paper we have presented constructive visualization as a paradigm which can help realize the democratization of visualization. Constructive visualization offers the possibility of providing people with means to construct visualizations of their own in a simple, expressive and flexible ways. To construct a visualization the necessary components are:

- A set of basic units or tokens, which can be mapped to data,
- A token grammar, which declares how the attributes of the tokens can signify data,
- An environment in which the tokens can be placed,
- An assembly model, which describes the constraints and freedom with which the tokens can be assembled.

The process of developing the constructed visualization starts from initializing the environment in which the construction will take place. Then the data units are mapped to the tokens and the tokens visual attributes are assigned meaning according to the data. These tokens are then assembled in the environment. Changes in data can subsequently be expressed by manipulating the data token.

Our new design approach to visualization was inspired directly by ideas from three other domains that all share a core tenet of simplicity. From the invention of the Kindergarten Gifts by Froebel, we imported the logic of using tokens to manipulate units. From the constructivism theory of Piaget, we learned that manipulation can help us better understand the world. From Papert's research, we saw how these lessons can be applied to computer science in general and visualization in particular. In applying the concepts of constructionism to information visualization design, we have defined its components, token, grammar, environment and assembly model, and outlined the processes necessary to this approach: initialization, mapping, assembling, and evolution.

Constructive visualization is a novel approach to InfoVis that can provide non-experts with a method for constructing novel visualizations. As it builds on our inherent understanding and experience with physical building blocks, the simplicity in the model enables non-experts to engage with datasets in a manner that would not have otherwise been possible.

ACKNOWLEDGMENTS

We would like to thank Bon Adriel Aseniero and Lindsay MacDonald for their insightful comments. This research was supported in part by NSERC, SMART Technologies, AITF, Surfnet, GRAND NCE and ANRT CIFRE Nº 2010/1534.

REFERENCES

1. Genealogy of Pop Rock Music. Last access: 18 Jan. 2014. `http://www.historyshots.com/rockmusic/`.
2. Google Chart Editor. Last access: 10 Dec. 2013. `https://support.google.com/drive/answer/63824`.
3. Piaget on Piaget, Part 3. Last access: 17 July 2013. `http://www.youtube.com/watch?v=x9nSC_Xgabc`.
4. Quantified Self data visualization archives. Last access: 09 Dec. 2013. `http://goo.gl/uvVOv5`.
5. SpotFire. Last access: 10 Dec. 2013. `http://goo.gl/qrbN9L`.
6. Tableau. Last access: 12 July 2013. `http://goo.gl/KFFBGH`.
7. How GM Is Saving Cash Using Legos As A Data Viz Tool Last access: 17 July 2013. , 2012. `http://www.fastcodesign.com/1669468`.
8. Aleeb-Lunddberg, K. Kindergarten mathematics laboratorynineteenth-century fashion. *The Arithmetic Teacher 17*, 5 (1970), 372–386.
9. Amar, R., and Stasko, J. Knowledge precepts for design and evaluation of information visualizations. *IEEE TVCG 11*, 4 (2005), 432–442.
10. Anderson, D., Yedidia, J. S., Frankel, J. L., Marks, J., Agarwala, A., Beardsley, P., Hodgins, J., Leigh, D., Ryall, K., and Sullivan, E. Tangible interaction + graphical interpretation. In *Proc. of SIGGRAPH*, ACM (NY, USA, 2000), 393–402.
11. Arnheim, R. *Visual thinking*. UofC Press, 1969.
12. Bertin, J., and Barbut, M. *Sémiologie graphique*. Mouton; Paris: Gauthier-Villars, 1973.
13. Blackwell, A. F. The reification of metaphor as a design tool. *ACM TOCHI 13*, 4 (2006), 490–530.
14. Bostock, M., Ogievetsky, V., and Heer, J. D3 data-driven documents. *IEEE TVCG 17*, 12 (2011), 2301–2309.
15. Bultman, S. *The Froebel Gifts 2000*. Kindergarten Messenger, 2000.
16. Card, S., and Mackinlay, J. The structure of the information visualization design space. In *Proc of. Information Visualization, IEEE* (1997), 92–99.
17. Card, S. K., Mackinlay, J. D., and Schneiderman, B. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.
18. Carpendale, M. S. T. *A framework for elastic presentation space*. PhD thesis, Simon Fraser University, 1999.
19. Chao, W. O., Munzner, T., and van de Panne, M. Poster: Rapid pen-centric authoring of improvisational visualizations with napkinvis. *Posters Compendium InfoVis* (2010).
20. Chapman, M. *Constructive evolution: Origins and development of Piaget's thought*. Cambridge University Press, 1988.
21. Chi, E. H. A taxonomy of visualization techniques using the data state reference model. In *Proc. of InfoVis 2000. IEEE* (2000), 69–75.
22. Cleveland, W. S., and McGill, R. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association 79*, 387 (1984), 531–554.
23. Cross, A. The educational background to the bauhaus. *Design Studies 4*, 1 (1983), 43 – 52.
24. Fekete, J. The infovis toolkit. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on* (2004), 167–174.
25. Frei, P., Su, V., Mikhak, B., and Ishii, H. Curlybot: Designing a new class of computational toys. In *Proc. of CHI*, CHI '00, ACM (NY, USA, 2000), 129–136.
26. Gentner, D. Structure-mapping: A theoretical framework for analogy. *Cognitive Science 7*, 2 (1983), 155 – 170.
27. Girouard, A., Solovey, E. T., Hirshfield, L. M., Ecott, S., Shaer, O., and Jacob, R. J. K. Smart blocks: A tangible mathematical manipulative. In *Proc. of TEI*, ACM (NY, USA, 2007), 183–186.
28. Grammel, L., Bennett, C., Tory, M., and Storey, M.-a. A Survey of Visualization Construction User Interfaces. *Eurovis* (2013), 1–5. `http://goo.gl/Chs0vu`.
29. Grammel, L., Tory, M., and Storey, M. How information visualization novices construct visualizations. *IEEE TVCG 16*, 6 (2010), 943–952.
30. Greenberg, S., Carpendale, S., Marquardt, N., and Buxton, B. *Sketching user experiences: The workbook*. Elsevier, 2011.
31. Heer, J., Card, S. K., and Landay, J. A. Prefuse: A toolkit for interactive information visualization. In *Proc. of SIGCHI*, CHI '05, ACM (USA, 2005), 421–430.
32. Heer, J., Ham, F., Carpendale, S., Weaver, C., and Isenberg, P. Creation and collaboration: Engaging new audiences for information visualization. In *Information Visualization*, vol. 4950 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008, 92–133.
33. Heer, J., and Shneiderman, B. Interactive dynamics for visual analysis. *Queue 10*, 2 (2012), 30:30–30:55.
34. Hsu, W., and Woon, I. M. Current research in the conceptual design of mechanical products. *Computer-Aided Design 30*, 5 (1998), 377 – 389.
35. Hullman, J., and Diakopoulos, N. Visualization rhetoric: Framing effects in narrative visualization. *IEEE TVCG 17*, 12 (2011), 2231–2240.
36. Hunger, M. On LEGO Powered Time-Tracking; My Daily Column Last access: 17 July 2013. `http://goo.gl/iAj3wM`.
37. Huron, S., Vuillemot, R., and Fekete, J.-D. Towards Visual Sedimentation. In *VisWeek 2012*,
38. Huron, S., Vuillemot, R., and Fekete, J.-D. Visual sedimentation. *IEEE TVCG 19*, 12 (2013), 2446–2455.
39. Hutchins, E. L., Hollan, J. D., and Norman, D. A. Direct manipulation interfaces. *HumanComputer Interaction 1*, 4 (1985), 311–338.
40. Jansen, Y., and Dragicevic, P. An Interaction Model for Visualizations Beyond The Desktop. *IEEE TVCG 19*, 12

(2013), 2396 – 2405.
41. Kay, A. User interface: A personal view. *The art of human-computer interface design* (1990), 191–207.
42. Kosara, R. Visualization criticism - the missing link between information visualization and art. In *Information Visualization, 2007* (2007), 631–636.
43. Maloney, J., Resnick, M., Rusk, N., Silverman, B., and Eastmond, E. The scratch programming language and environment. *ACM TOCE 10*, 4 (2010), 16.
44. McCormack, J., Cagan, J., and Vogel, C. Speaking the Buick language: capturing, understanding, and exploring brand identity with shape grammars. *Design studies* (2004).
45. McKim, R. H. Experiences in visual thinking.
46. Minard, C. La carte figurative des pertes successives en hommes de l'arme franaise dans la campagne de Russie 1812-1813. Last access: 18 Jan. 2014., 1869. `http://fr.wikipedia.org/wiki/Carte_figurative`.
47. Müller, P., Wonka, P., Haegler, S., Ulmer, A., and Van Gool, L. Procedural modeling of buildings. In *ACM SIGGRAPH 2006*, ACM (NY, USA, 2006), 614–623.
48. Myers, B. A., Goldstein, J., and Goldberg, M. A. Creating charts by demonstration. In *Proc. of the SIGCHI*, ACM (NY, USA, 1994), 106–111.
49. Neurath, M. *The Transformer: Principles of Making Isotype Charts*. Princeton Architectural Press, 2009.
50. Neurath, O. *Modern Man in the Making*. Knopf, 1939.
51. Papert, S. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, Inc., NY, USA, 1980.
52. Papert, S. What is logo? who needs it. *Logo philosophy and implementation* (1999).
53. Papert, S., and Harel, I. Situating constructionism. *Constructionism* (1991), 1–11.
54. Parkes, A., LeClerc, V., and Ishii, H. Glume: Exploring materiality in a soft augmented modular modeling system. In *Proc. of CHI*, ACM (2006), 1211–1216.
55. Piaget, J. *Six études de psychologie*. Ed. Denoël, 1989.
56. Pousman, Z., Stasko, J., and Mateas, M. Casual information visualization: Depictions of data in everyday life. *IEEE TVCG 13*, 6 (2007), 1145–1152.
57. Pretorius, A., and van Wijk, J. Multiple views on system traces. In *PacificVIS IEEE* (2008), 95–102.
58. Raffle, H. S., Parkes, A. J., and Ishii, H. Topobo: A constructive assembly system with kinetic memory. In *Proc. of CHI*, ACM (NY, USA, 2004), 647–654.
59. Resnick, M. All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. In *Proc. of C&C*, ACM (USA, 2007), 1–6.
60. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y. Scratch: Programming for all. *Communication of the ACM 52*, 11 (2009), 60–67.
61. Resnick, M., Martin, F., Sargent, R., and Silverman, B. Programmable bricks: Toys to think with. *IBM Systems Journal 35*, 3.4 (1996), 443–452.
62. Roberto, R., Freitas, D., Lima, J. a. P., Teichrieb, V., and Kelner, J. ARBlocks: A Concept for a Dynamic Blocks Platform for Educational Activities. *2011 XIII Symposium on Virtual Reality* (2011), 28–37.
63. Saraiya, P., North, C., and Duca, K. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE TVCG 11*, 4 (2005), 443–456.
64. Scharf, F., Winkler, T., and Herczeg, M. Tangicons: Algorithmic reasoning in a collaborative game for children in kindergarten and first class. In *Proc. of IDC*, ACM (NY, USA, 2008), 242–249.
65. Schweikardt, E., and Gross, M. D. roblocks: A robotic construction kit for mathematics and science education. In *Proc. of ICMI*, ACM (NY, USA, 2006), 72–75.
66. Segel, E., and Heer, J. Narrative visualization: Telling stories with data. *IEEE TVCG 16*, 6 (2010), 1139–1148.
67. Stiny, G. Introduction to shape and shape grammars. *Environment and planning B 7*, 3 (1980), 343–351.
68. Stiny, G. Kindergarten grammars: designing with froebel's building gifts. *Environment and Planning B 7*, 4 (1980), 409–462.
69. Ullmer, B., Ishii, H., and Jacob, R. J. K. Token+constraint systems for tangible interaction with digital information. *ACM TOCHI 12*, 1 (2005), 81–118.
70. Victor, B. Drawing Dynamic Visualizations Talk.Last access: 12 July 2013. `http://goo.gl/NlKOg5`.
71. Viegas, F., Wattenberg, M., McKeon, M., van Ham, F., and Kriss, J. Harry potter and the meat-filled freezer: A case study of spontaneous usage of visualization tools. In *Proc. of HICSS* (2008), 159–159.
72. Viegas, F., Wattenberg, M., van Ham, F., Kriss, J., and McKeon, M. Manyeyes: a site for visualization at internet scale. *IEEE TVCG 13*, 6 (2007), 1121–1128.
73. Vigas, F., and Wattenberg, M. Artistic data visualization: Beyond visual analytics. In *Online Communities and Social Computing*, vol. 4564 of *Lecture Notes in Computer Science*. Springer Berlin, 2007, 182–191.
74. Walny, J., Carpendale, S., Riche, N., Venolia, G., and Fawcett, P. Visual thinking in action: Visualizations as used on whiteboards. *IEEE TVCG 17*, 12 (2011), 2508–2517.
75. Walny, J., Lee, B., Johns, P., Riche, N., and Carpendale, S. Understanding pen and touch interaction for data exploration on interactive whiteboards. *IEEE TVCG 18*, 12 (2012), 2779–2788.
76. Ware, C. *Information visualization: perception for design*. Elsevier, 2012.
77. Weber, W., and Rall, H. Data visualization in online journalism and its implications for the production process. In *Proc. of IV* (2012), 349–356.
78. Wyeth, P., and Wyeth, G. Electronic blocks: Tangible programming elements for preschoolers. *Proc. of the Eighth IFIP TC13* (2001).
79. Yi, J. S., Melton, R., Stasko, J., and Jacko, J. A. Dust & magnet: multivariate information visualization using a magnet metaphor. *IV 4*, 4 (2005), 239–256.
80. Ziemkiewicz, C., and Kosara, R. The shaping of information by visual metaphors. *TVCG 14*, 6 (2008), 1269–1276.