



HAL
open science

Constructive Visualization

Samuel Huron

► **To cite this version:**

Samuel Huron. Constructive Visualization: A token-based paradigm allowing to assemble dynamic visual representation for non-experts. Human-Computer Interaction [cs.HC]. Univeristé Paris-Saclay, 2015. English. NNT : 2014PA112253 . tel-02924469v1

HAL Id: tel-02924469

<https://inria.hal.science/tel-02924469v1>

Submitted on 13 Jan 2016 (v1), last revised 6 Mar 2015 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE PARIS-SUD

ÉCOLE DOCTORALE D'INFORMATIQUE
Laboratoire INRIA, Saclay, Ile de France

Discipline: Informatique

THÈSE DE DOCTORAT

Soutenue le 29/09/2014

par

Samuel HURON

Constructive Visualization:

A token-based paradigm allowing to assemble dynamic visual
representation for non-experts

Directeur de thèse :

Jean-Daniel FEKETE

Directeur de Recherche (INRIA Saclay, France)

Co-directeur de thèse :

Vincent PUIG

Directeur exécutif (IRI Centre Pompidou, France)

Composition du jury :

Rapporteurs :

Yannick PRIÉ
Andrew VANDE MOERE

Professeur, Université de Nante, France
Associate professor (Faculty of Engineering Science,
head of "Architecture and Design", Belgium)

Examineurs :

Wendy MACKAY
Jason DYKES

Directeur de Recherche (INRIA Saclay, France)
Professor School of Informatics,
City University, London United Kindom

Christian JACQUEMIN
Brygg ULLMER

Professeur (Université Paris-Sud, France)
Associate Professor (Lousiana State University, USA)

Membre invité :

Sheelagh CARPENDALE

Professor (University of Calgary, Canada)

SAMUEL HURON

CONSTRUCTIVE VISUALIZATION

A TOKEN-BASED PARADIGM ALLOWING TO ASSEMBLE
DYNAMIC VISUAL REPRESENTATION FOR NON-EXPERTS.

Copyright © 2015 Samuel Huron

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

First printing, 2014 July 29

Second printing, 2014 September 15

Abstract

During the past two decades, information visualisation (InfoVis) research has created new techniques and methods to support data-intensive analyses in science, industry and government. These have enabled a wide range of analyses tasks to be executed, with tasks varying in terms of the type and volume of data involved. However, the majority of this research has focused on static datasets, and the analysis and visualisation tasks tend to be carried out by trained expert users. In more recent years, social changes and technological advances have meant that data have become more and more dynamic, and are consumed by a wider audience. Examples of such dynamic data streams include e-mails, status updates, RSS ¹ feeds, versioning systems, social networks and others. These new types of data are used by populations that are not specifically trained in information visualization. Some of these people might consist of casual users, while others might consist of people deeply involved with the data, but in both cases, they would not have received formal training in information visualization. For simplicity, throughout this dissertation, I refer to the people (casual users, novices, data experts) who have not been trained in information visualisation as non-experts.

These social and technological changes have given rise to multiple challenges because most existing visualisation models and techniques are intended for experts, and assume static datasets. Few studies have been conducted that explore these challenges. In this dissertation, with my collaborators, I address the question: *Can we empower non-experts in their use of visualisation by enabling them to contribute to data stream analysis as well as to create their own visualizations?*

The first step to answering this question is to determine whether people who are not trained in information visualisation and the data sciences can conduct useful dynamic analysis tasks using a visualisation system that is adapted to support their tasks. In the first part of this dissertation I focus on several scenarios and systems where different sized crowds of non-InfoVis experts users (20 to 300 and 2 000 to 700 000 people) use dynamic information visualisation to analyse dynamic data.

Another important issue is the lack of generic design principles for the visual encoding of dynamic visualization. In this dissertation I design, define and explore a design space to represent dynamic data for non-experts. This design space is structured by visual tokens representing data items that provide the constructive material for the

¹ RSS is a Web content syndication format, the acronym stands for Really Simple Syndication, <http://validator.w3.org/feed/docs/rss2.html>

assembly over time of different visualizations, from classic representations to new ones. To date, research on visual encoding has been focused on static datasets for specific tasks, leaving generic dynamic approaches unexplored and unexploited.

In this thesis, I propose construction as a design paradigm for non-experts to author simple and dynamic visualizations. This paradigm is inspired by well-established developmental psychological theory as well as past and existing practices of visualisation authoring with tangible elements. I describe the simple conceptual components and processes underlying this paradigm, making it easier for the human computer interaction community to study and support this process for a wide range of visualizations. Finally, I use this paradigm and tangible tokens to study *if* and *how* non-experts are able to create, discuss and update their own visualizations. This study allows us to refine our previous model and provide a first exploration into how non-experts perform a visual mapping without software. In summary, this thesis contributes to the understanding of dynamic visualisation for non-expert users.

Acknowledgments

0.0.1 Work setup

This work was supported by a ANRT CIFRE grant (N°2010/1534). This type of grant consists of a 3 year employment contract between a Ph.D. Student in a research laboratory (INRIA) and a company or non-governmental organization (IRI POMPIDOU CENTER). During the three years, time is split equally between the two environments IRI (50%) and INRIA (50%) according to the distribution present in Figure 1.

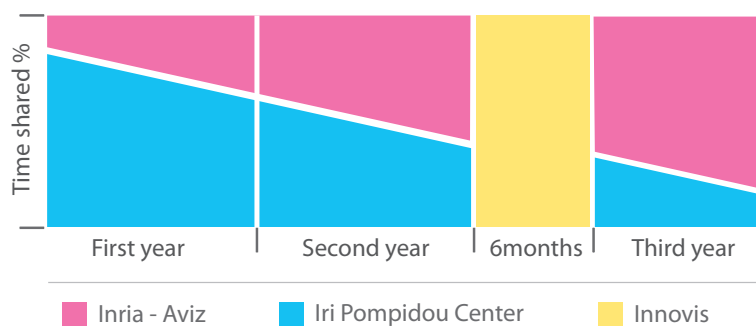


Figure 1: Time shared between different work environments.

Over these three years I had the good fortune to work in three different exceptional environments, IRI, INRIA and the INNOVIS Lab (where I did an internship). These three organisations provided me with a creative set of constraints, methods and supervisions styles.

The first, IRI² is a non-governmental research association that began in the Pompidou Center. Its purpose is to anticipate, accompany and analyse the transformation of cultural practices enabled by digital technologies, and sometimes even to contribute to their emergence. My role at IRI was to transpose the philosophical concepts discussed during seminars into interactive design proposals and prototypes.

² <http://www.iri.centrepompidou.fr/>

The second, INRIA³ is a public research laboratory dedicated to digital science and technology. At INRIA, I was involved as a Ph.D. Student in the AVIZ team⁴. AVIZ is a multidisciplinary team focusing on the improvement and analysis of visualizing large and complex datasets. My role in this team was to study, evaluate, and report the produced systems designed at IRI. My supervisor was Jean Daniel Fekete who provided me with a great deal of help in structuring my research process and learning the constraints of scientific rigor. I also want to acknowledge the incredible pedagogical skills of Petra Isen-

³ <http://www.inria.fr/>

⁴ <http://www.aviz.fr/>

berg who was always present when I needed further explanations.

These two contexts were fruitful for creating collaborations with a variety of people from different domains of expertise and with different orientations. Moreover, the theoretical part of this thesis was assembled while I was interning at the Innovis laboratory, under the supervision of Sheelagh Carpendale. This internship helped me to reflect more broadly on the research produced in the two previous environments and discover what was unique in the embedded designs. During and after this internship Dr Sheelagh Carpendale supervised and guided part of this work. She gave me much support so that I could attain confidence in the creative seeds of my ideas.

In summary these three contexts provided me with different elements: IRI provided me with a grounded experimental field for application, technical support, and philosophical ideas about “amateurs contribution”. INRIA - AVIZ provided me with expertise in the information visualisation field, a structure for thinking, and scientific methodologies. INNOVIS ⁵ helped me to attain an overview of my research and to learn design methodologies. For this reason my research sits on the boundaries of these three domains, “amateurs contribution”, InfoVis, and Design, as illustrated in Figure 1.4 in Section 1.1.

⁵ <http://innovis.cpsc.ucalgary.ca/>

0.0.2 *Thanks to*

I would like to thank Tiphaine who supported me during this work and was there through the best and the worst. I also want to thank my dad and sister, and some great friends Nicolas, Antoine and Jose. And all the friends I have neglected due to this work, Manon, Steve, Touffik, David, Céline, Cécile, Kim ...

A huge thanks to Nicolas and Antoine Mazieres, with who I spent too much time doing/discussing almost everything. A huge thanks to Francois Taddei for all the discussions which challenged my thinking, but moreover for his effort to provide a place where students could feel at home while at University. Thanks to Jean Baptiste Larbune and his inextinguishable flow of references during discussion. A great thanks to all the Fabelier community who really deserve it and teach me a lot.

People in the place I have been working (INRIA, IRI and INNOVIS), were outstanding! In IRI, I want to acknowledge the people with who I did a lot of the work present in the first part of this thesis: Nicolas Sauret, Raphael Velt, Florence Duc, Thibaut Cavalie, Alexandre Monnin, Yves-Marie Haussonne, Karim Hamidou, Alexandre Bastien... In AVIZ all the great folks I shared good times with: Romain Vuillemot, Yvonne Jansen, Pierre Dragicevic, Jeremy Boy that help me on my english, Mathieu Le Goc thanks for supporting me on beer drinking, Benjamin Bach, Charles Perin, Wesley Willett and others... I also really enjoy my period at the **Innovis** Lab, all the people there were “super cool” and inspired me a lot: Jagoda Walny, thanks for being so patient with me and being my lab buddy, Lindsay MacDonald and Kyle Hall I enjoyed our discussions a lot, Lawrence Fyfe and Alice Thudt, soon

it will be time for a beer, and also David Ledo, Bon Adriel Aseniero, Michael Maurer and Gerry Straathof...

Thanks to all the participants in my experiments and studies.

Also I wanted to thank all the incredible teachers who inoculated me with this “desire of knowing” virus. And the teachers who help me to focus on how to catalyse it. I particularly want to thank all the people of the experimental school “Le lycée du temps choisi”⁶ that provide me a perfect environment to come back to school.

⁶ https://www.ac-paris.fr/portail/jcms/p1_72258/le-lycee-du-temps-choisi-ldtc

I also want to thank a part of the colleague and friend I meet during the industrial part of my career. First the Business Lab (Valerie Legat, JC B, JC G, Laurent, Jhonatan...) team who teach me a lot and provide me confidence in my working skills. My partners during my entrepreneurship adventures, as well as some of my clients during my freelance period.

Moreover, I want to acknowledge all the people who helped to improve the quality of this document, it was useful and valuable: Chih-Chun Chen, Kyle, Petra, Wes, Lindsay, Lawrence, David and Jennifer.

Et avant tout, à ma mère, Odile.

List of Figures

1	Time shared between different work environments.	v
2	Online index of additional ressources: http://samuel-huron.gforge.inria.fr/	xxv
1.1	First timeline charts by Joseph Priestley (1765).	1
1.2	Example of a sand clock. Some Rights Reserved ("running out of time" by Rui Malheiro) http://goo.gl/1rLU30	2
1.3	Research Scope.	3
1.4	Domain Scope.	5
1.5	Overview of my research path through a modified version of MacKay and Fayard Triangulation framework. Each box is a phenomenon and each arrow is the move from one phenomenon to others.	9
2.1	Plot of the number of papers returned by IEEE digital library for the venue TVCG with the query "dynamic data", "dynamic visualization" and "data stream" research in full text by year. The percentage is computed according to the number of paper containing "ieee". The rectangle on top corresponds to the papers in which the authors use one of these keywords according to the website Keyvis.	21
2.2	Dynamic Visualization pipeline freely adapted, with object of research in blue (Diagram adapted from Cottam et al. [Cottam et al., 2012].	22
2.3	Screenshot of Information Flocking Boids, Vande Moere [2004].	25
2.4	"Dust and magnets", Yi et al. [2005], http://youtu.be/wLXwL38xek0	25
2.5	"Whisper", screenshot on details, Cao et al. [2012].	25
2.6	Krstajić et al. [2010] visualisation details screenshot.	25
2.7	"Cloudlines", screenshot details, Krstajić et al. [2011].	26
2.8	"Artifacts of the Presence Era", screenshot, Viegas [Viégas et al., 2004].	26
2.9	Screenshot of the application vox citivas [Diakopoulos and Shamma, 2010].	27
2.10	Screenshot of conference monitor. Sopan [2012].	28

2.11	Screenshot. Dork et al. [2010].	28
2.12	TwitInfo screenshot Marcus et al. [2011].	28
2.13	Visual spread sheets extract from Chi and Card [1999]	30
2.14	Screenshot of Microsoft Excel interface.	30
2.15	Screenshot of Processing interface [Reas and Fry, 2007].	30
2.16	Lark interfaces, Tobiasz et al. [2009].	30
2.17	Visual Builder example from Myers et al. [1994]	30
3.1	Information visualisation and other domains of monosemic [Bertin, 1977] visual representation systems. I derived inspiration from some of these domains (for example, instruments of measures, instruments of calculation) by studying their visual representation mappings. This is a containment diagram.	33
3.2	A clock, a thermometer, and an abacus. The first two devices are instruments for measuring, the former for measuring time, and the second for measuring temperature. The third instrument is used for calculation and is also manipulable. All three of them provide a visual representation of information.	36
3.3	A type of tally mark used in Europe, Africa, Australia, and North America.	37
3.4	Ishango bone (20,000 B.C.)	37
3.5	Chinese tally mark system.	37
3.6	Dot and line tally marks encode the accumulation of ten units.	37
3.7	Clay tokens, extract from [Schmandt-Besserat, 1996]. .	39
3.8	A. Tablet with engraved tokens. B. Tablet with engraved tokens and additional information such as strokes to delimit layout and pictographs. C. Tablet with engraved table layout sorted by value order, similar to several bar charts. Extract from Robson [Robson, 2003]	40
3.9	Clay tokens and in the background an envelope with impressed token marks.	40
3.10	Example of token counting board counting process: black dots represent tokens, all the tokens on the right of the vertical line are positive values, those on the left of the line are negative values; the horizontal line corresponds to 1,5,10 from bottom to top. In grey the notation in Roman numerals and in black in Arabic numbers.	41
3.11	Photos of a the Salamis Tablet. cf: http://www.computerhistory.org/revolution/calculators/1/1/128	41
3.12	Counting frames in different civilizations, (A) Roman abacus, likely to be derived from the counting board (B)Yupana, an Incas counting instrument based on a frame to count with tokens (C) Quipu, an Incas personal cord device to count and keep track of information.	42

3.13	How to write 6492 in Maya number system, form top to bottom a round and a line stand for $1+5=6$, below 4 round stand for 4, below 4 round and a line stand for $4+5=5$, below two round stand for 2, meaning 6492. . .	42
3.14	Maya number system: Symbol are based on additive principle. Two basic symbol one representing number one and the other number five.	42
3.15	Different board game through ages, (A) Senet 3100 BC., (B) Roman tabula 480 C.E., (C) Mancala.	43
3.16	(A) Greek coin "Drachma" of Aegina 700BC (B) Token from "La monnaie de Paris" 1756 AC (C) Automatic Cashier, 19th century (D) Coin sorter 20th century. . .	44
3.17	Froebel' "Gift" N°3 contains 8 wooden cubes of the same size.	45
3.18	Froebel's illustration of building blocks manipulation. In this case (A) a volume of 8 building blocks organized in a square and corresponding to Gift 3, (B) divide the volume by two, by pulling apart two blocks of 4 (C) look at the result over a different angle by rotating the parts.	45
3.19	A) Timeline of philosophe, Joseph Priestley. 1769. B) Barchart by William Playfair, 1786.	47
3.20	Playfair stack chart and piechart	48
3.21	(A) Gerd Arntz designing one of the isotype pictograms, (B) A mold to print the isotype pictograms, (C) a assistant cutting pictograms in order to assemble them, (D) The result of assembling a set of pictograms.	49
3.22	"Home and Factory Weaving in England", Each pictogram stands for a multiple. Each blue symbol represents 50 million pounds total production. Each black man symbol represents 10,000 home weavers, each red man symbol represents 10,000 factory weavers.	49
3.23	Explanation of the Isotype principle of countability (initially shown in London 1933 by the Isotype Institute.	49
3.24	A) a matrice in folded paper, B) a woman constructing and sorting a tangible matrix, C) the tangible matrix on top of a photocopier to capture an image of it. Extract from "la graphique et le traitement graphique de l'information"	50
3.25	Jacque Bertin, Visual variables.	51
3.26	Jacque Bertin' "domino" used to construct matrices, each domino represent a data case and could be considere as a token. In background Jean Daniel Fekete, photo by Jean Baptiste Labrune.	51
3.27	Xerox Star icons.	52
3.28	Sketchpad program. Online video demo: https://www.youtube.com/watch?v=USyoT_Ha_bA	52
3.29	Xerox Star advertising.	52
3.30	Data Montain	53

3.31	Screen shot of bumpstop.	53
3.32	Scratch interfaces.	54
3.33	Stiny Kindergarten grammar.	54
3.34	Stiny example of incremental design.	54
3.35	Structure mapping example of Gentner [1983] for the analogy “The atom is like the solar system”. Top: mapping of planet and sun to electron and nucleus. Bottom: properties common to the electron and nucleus. Not all properties of one system could apply to the second, but most of them.	55
3.36	Illustration of the three different token+constraint approach.	56
3.37	“Marble Answer Machine”, 1992, Durrell Bishop	56
3.38	Different tangible interfaces based on building blocks and constructive assembly logic. From left to right: Glume project, Smart block, AR blocks, Topobo, Roblocks, Curly bot.	57
3.39	Hans Rosling construct and update a dynamic visual representation of wealth distribution over the human population with pebbles. During the talk he explains that black pebbles represent the richest people in the world and white the poorest. Extract from : https://www.youtube.com/watch?v=UNs-ziziPyo	58
3.40	“Dust and magnets”, Yi et al. [2005] , http://youtu.be/wLxwL38xek0	59
3.41	“Sand Dance”, Steven Drucker [2013] , http://research.microsoft.com/en-us/projects/sanddance/	59
3.42	“kinetica”, Rzeszotarski and Kittur [2014] , https://www.youtube.com/watch?v=70YcGiKrmEg	59
3.43	Pindices (2005) Lucy kimbell and Andrew Barry, http://www.lucykimbell.com/LucyKimbell/Pindices.html	60
3.44	The dumpster (2006) Golan Levin. http://www.flong.com/projects/dumpster/	60
3.45	We feel fine (2006) bar chart based on visual tokens assembly.	60
3.46	Handguns, 2008, 60x92 inches, “Depicts 29,569 handguns, equal to the number of gun-related deaths in the US in 2004.” Chris Jordan.	61
3.47	Handguns artwork details.	61
3.48	US debt visual representation. http://usdebt.kleptocracy.us/	61
3.49	Vitamin Lego Calendar prototypes, screenshot from the video: http://vimeo.com/75500464	62
4.1	Additional resources: Online System: http://polemictweet.com/ Metadata player code source: https://github.com/IRI-Research/metadataplayer	67

4.2	The three stages of PolemicTweet: 1) printed information on connection & tags, 2) the live Twitter client & structured annotations, 3) video player showing tweet annotations, accessible on this link: http://polemictweet.com/rsln/polemicaltimeline.php	68
4.3	The Microsoft conference room in Paris where PolemicTweet was deployed; the use of laptops or other wifi-capable devices was prevalent and ensured active participation.	69
4.4	Screen shot of the “ligne de temps” interface.	70
4.5	Screen shot of the “DIVA” main display.	71
4.6	The final PolemicTweet backchannel interface.	74
4.7	First complete mock-up I did of the MetadataPlayer presenting each of the component used in PolemicTweet.	76
4.8	Search for “ushahidi” in the “polemic line” . The tweets referring to the section of the talk related to this query are highlighted in pink. On top of that we could see the player synchronized showing the slide of the speaker about “ushahidi”.	77
4.9	Mock up of a classic stacked bar chart.	77
4.10	Design variations.	78
4.11	Incremental token property.	78
4.12	Fisheye effect on an area chart to get an individual element.	79
4.13	Mock up of the roll over feature on a token.	79
4.14	PolemicTweet public website: A. Home page, B. Archives page, C. Past event page with the MetadataPlayer.	80
4.15	Polemic lines of A) “Les Strategies Virtuelles des Musee: l’Heure de Verite”, B) “Mashup Remix”, C) “Du vinyl au numerique”, a 2h event with different density of annotation, number of peaks, and tagging activity.	82
4.16	Flyer of a Microsoft’ event with the PolemicTweet instruction distribute at the registration desk.	82
4.17	Flyer of “Museologie seminar” organize in the Pompidou Center.	83
4.18	Flyer of “Mashup film festival”organize in “Forum des Images”. Video: http://polemictweet.com/mashup/polemicaltimeline.php	83
4.19	Tweet with and without PolemicTweet tags split by backchannel vs. other clients.	84
4.20	Tweet with and without syntax by the most popular Twitter clients.	85
4.21	A participant computer, “What to look at the stage or the screen?”. Video: http://polemictweet.com/rsln/polemicaltimeline.php	87
4.22	PolemicTweet used during a web tv application for the news paper Mediapart.	88

5.1	From Left to right: PolemicTweet metadata player (Chapter 4), Bubble-T webpage, Bubble-TV photography of the Tv show stage (Chapter 5).	91
5.2	Additional ressources: http://www.aviz.fr/Research/Bubble-tv http://www.aviz.fr/Research/Bubble-t	91
5.3	Bubble-T screenshot (21/12/2011), a video of the top of the interface is accessible here : https://www.youtube.com/watch?v=TEfLV7fX07E	95
5.4	Bubble-T visual components: A. A header, B. A control Bar, C. A real-time visualization, D. Day and week time frame visualizations, E. Tweets read area incomming token appear here, F. An example of the trajectory of a tweet, G. A column containing a set of tweets. (Screenshot from the original website 21/12/2011). . .	97
5.5	Bubble-T: Rolling over on a token shows a tooltip containing the information about the corresponding tweet.	97
5.6	Bubble-T embed into the official Youtube channel for the French presidential elections 2012.	98
5.7	Tokens would overflow due to a bug in the token disappearance script.	99
5.8	The stage with Bubble-TV as a background.	100
5.9	The building and preparation of the stage before the show.	101
5.10	The directors from the studio, authoring multiples videos sources during the show.	101
5.11	Speaker present and explain how the visualisation will work.	102
5.12	The community manager screen control.	103
5.13	The Tweet counter embedded in the stage furniture. . .	103
5.14	The community manager reporting Twitter activity. . .	103
5.15	The state of the visualisation few minute before the end of the emission.	104
6.1	The Visual Sedimentation metaphor applied to a bar chart (left), a pie chart (center), and a bubble chart (right).	107
6.2	Additional resources: A dedicated website was made for this technique. This website has source code, documentation, videos and demos for Visual Sedimentation: http://www.visualsedimentation.org	107
6.3	Illustrations of real life sedimentations process. Athabasca glacier (left), Mississippi riverbed on a map [Fisk, 2005], water riverbed on a Glacier, water surface ocillation, Hoodoo Bryce canyon, Types of sediments [global-spec.com, 2013] (right).	109
6.4	Summary of the different states of tokens in the Visual Sedimentation metaphor.	110

6.5	Summary of the design space. Time goes from right to left: 1) Data streams chunked and stored in a buffer for further reading, 2) Tokens appear from an entrance point, 3) Physical forces transport and shrink tokens until they flocculate, 4) Strata are tokens aggregated into layers.	112
6.6	Template schemas: Blue lines represent the <i>walls</i> ; red lines the <i>ground</i> ; green arrows the <i>entrance point</i> ; black arrow the <i>gravity</i> direction; light grey shapes the <i>deposit area</i> ; and dark grey shapes the <i>aggregated area</i> . (A) A common bar chart and (B) its template; (C) the pie chart template; (D) the bubble chart template. . . .	114
6.7	Summary of the case studies: (A) Bubble-T seminal applications.; (B) Bubble-TV.; (C) SediClock: a hybrid between an hourglass and a water clock.; (D) Most popular questions asked on StackOverflow, the categories being the 10 most popular tags.; (E) Birth, death and population screen capture at 5h15 pm.; (F) WikiFlow.; (G) Edits of text and code before the deadline of VisWeek 2012 using SediVN.; (H) SediMMS.; (I) Label size reflecting the recent activity (bottom left) and lines between tokens showing changes on common documents in SediVN.	115
6.8	Bar chart with Visual Sedimentation.	116
6.9	Piechart with Visual Sedimentation.	117
6.10	Bubble chart with Visual Sedimentation.	118
6.11	SediClock, Time Visualization screen shot 2014-06-09 at 04h22m20s. A snippet video is accessible at the following url: http://youtu.be/jVjN4LM7qKc?t=1m34s	119
6.12	Birth, death and population screen capture at 5h15 pm.	119
6.13	StackOverflow, Questions Visualization screen shot 2013-03-11 at 5h03 pm. A snippet video is accessible at the following url: http://youtu.be/jVjN4LM7qKc?t=1m55s	120
6.14	StackOverflow Logo.	120
6.15	Birth, death and population screen capture at 5h15 pm.	121
6.16	SVN Commits visualization	122
6.17	Bubble chart with Visual Sedimentation.	123

6.18	Summary of the case studies: (A) “Bar Chart” without walls.; (B) “Bar Chart” without ground and gravity; (C) “Drop Chart”, constraining tokens to flow inside identical complex shapes.; (D) A bar chart <i>overflow</i> due to the limited container capacity. (E) “Pie Chart without walls”; (F) The previous chart with a pixel aggregated area at 2 minute mark; (G) The previous chart with circular strata; (H) “Hoodoo Chart”.; (I) “Heap Chart” where tokens arrive from the center.; (J) “Heap Chart” where the deposit and aggregated areas are overlaid.; (K) “Trail Chart” where the tokens leave trails; (L) “Typo Chart” constraining tokens to the shape of the letters “VIZ”.; Red letters are pre flocculation variations and blue letters are post flocculation variations.	124
6.19	Evolution of a “barchart without boundaries”.	125
6.20	Different types of aggregated area, from right to left: classic aggregated area generated with a geometric primitive element, aggregated area made of flocculated tokens, aggregated area generate from the trail of tokens.	126
6.21	Token stroke and fill variation overtime.	128
6.22	Dissociation of the token collision area and the token visual mark.	128
6.23	Summary results of the Visual Sedimentation toolkit. The horizontal axis represents the number of tokens per second, from 3 to 105.	130
6.24	Bar Chart overflow.	130
7.1	Child playing and learning math.	139
7.2	Many Eyes screenshot.	141
7.3	Use, Draw, Code, Slide extracted from Brett Victor presentation, Feb.2013 at Stanford. Video: https://vimeo.com/66085662	141
7.4	Favorites sport teams of a 5th Grade girls.	142
7.5	Flow map of the number of men in Napoleon’s 1812 Russian campaign during 1812-1813. Lithograph, 62 x 30 cm, 1869, Charles Joseph Minard	142
7.6	The genealogy of pop rock music covering the time period from 1955 to 1978. 50 x 91 cm, 1978, Rebee Garofalo	142
7.7	The extended visualisation reference model [Jansen and Dragicevic, 2013], adapted to our terminology. . .	143
7.8	Paper distribution between the different paradigm. Blue for using (36), Red for Drawing(13), Orange for coding (17), Green for constructing (0).	144
7.9	A complete set of toys “Gift” designed by Froebel to learn mathematics.	145
7.10	Details of on of the toys. Here the “Gift” N°3 the box, and the eight building blocks.	145

7.11	Screenshot of a video of Piaget experiment [Piaget, 1977]: A girl of six year old stacking cube to rebuild a volume.	145
7.12	Scratch commmand block.	145
7.13	Children playing with a turtle, a pedagogical learning material to program logo by trial and errors.	146
7.14	Instruction for teacher to explain mathematical operation with Froebel Gifts.	146
7.15	Example of instruction for teacher Illustration for the Froebel Gift N°3 from the manual for teachers [Bultman, 2000]: A. Set of blocks in Gift 4, B. Operation showing division by two, C. Operation to teach multiplication. D. Operation to teach subtraction.	146
7.16	Example <i>constructive visualization</i> components.	147
7.17	Example <i>constructive visualization</i> process.	149
7.18	Plastic Cups, 2008, 60x90, Depicts one million plastic cups, the number used on airline flights in the US every six hours. Chris Jordan.	151
7.19	Different level of zoom in the piece "Plastic Cups".	151
7.20	Otto & Marie Neurath and his assistant are assembling previously clip pictograms (Section 3.3.2: page 49) into a specific spatial configuration for a chart.	152
7.21	Neurath visual representation of the car, telephon and bus production by country per 50 peoples.	152
7.22	Michael Hunger original personal time management tool.	152
7.23	Photography of the simulation of visual representation updates over time.	153
7.24	Kevin quinn's problem resolution tracking visual representation.	153
7.25	Details of Kevin Quinn tracking system zoom on.	154
8.1	Constructing a visualisation with tokens: right hand positions tokens, left hand points to the corresponding data.	161
8.2	Additional ressources: A dedicated website was made for this experiment, it contain videos, photos, figures, and others: http://constructive.gforge.inria.fr/	161
8.3	Experimental setup (all measurements are in centimetres).	167
8.4	Photograph of the experiment room.	168
8.5	Photograph of the table for the experiment.	168
8.6	Constructed visualizations of participant 1 to 4. Columns represent participants, rows represent tasks: (A) create a visualization, (B) update, (C) annotate.	170
8.7	Constructed visualizations of participant 5 to 12. Columns represent participants, rows represent tasks: (A) create a visualization, (B) update, (C) annotate.	171

8.8	A flow diagram showing the most common paths. In purple: the mental tasks, in blue: the physical tasks. The gray background rectangles illustrate the logical tasks. The gray oblongs linking two spheres represent possibly concurrent actions.	173
8.9	A participant for the pilot study achieving task C. . . .	173
8.10	View from the front camera on three participants (1) reading, (2) reading and pointing, (3) reading and grasping tokens	174
8.11	A participant GRASPS tokens (1,2,3) and BUILDS a construct (4,5,6). She is: (1) grasping some tokens from the box, (2) manipulating and transporting the tokens from the box to the canvas area (3) positioning the tokens, .	174
8.12	(4,5) starting to create a new green construct by placing tokens one by one on the canvas, (5) augmenting the construct in an organized way following her previous assembly (6) done. The pink annotation describes: #2 point to the token box. #4 point to the canvas.	174
8.13	Examples of constructs: (1) a line composed of 5 tokens, (2) square and rectangles (3) a construct resembling a horizon graph [Reijner and Software, 2008]: the tokens are layered to optimize canvas real estate. (4) a 3D bar chart column with two nested constructs in green and orange.	175
8.14	Participant 2 is testing four different combinations with the same visual constructs. During each rearrangement she adapted the organization of the constructs as necessary.	175
8.15	Participant 4, (1) after LOADING all the tokens, (2) ORGANIZING the red construct, (3) EXTENDING to other constructs.	176
8.16	Participant 12, (1) defines a assembly of construct for the first month (2) then replicates it for the next month (3), shows the result (4).	176
8.17	Participant 2, Mosaic of pictures shot at a regular interval.	177
8.18	Computations: C1: aggregate by spatial merge, categorize by proximity; C2: same categories as in C1 but different arrangement of constructs.	178
8.19	Participant 8, task C, Categorize: Colour semantic. . .	178
8.20	Computing: information preserving during aggregation;	179
8.21	Computations: A1/2: categorize by colour assignment; A2/3: aggregation by merging constructs;.	179
8.22	computation of delta with another month in blue (highlighted);	180
8.23	Computations: E: unitized construct ~ 100 ; F: unitized and merged construct to illustrate a month's budget. .	181

8.24	Storytelling: A1 : <i>unitized</i> construct; A2 : temporary SPLIT to explain the categories; B1 : <i>marking</i> of categories exceeding budget; B2/B3 <i>marking</i> and annotation how to optimize budget.	182
8.25	Sequences of actions during task A for all but participant 5.	183
8.26	Tokens of different shapes, colors, sizes.	185
9.1	From Left to right: PolemicTweet meta-data player (Chapter 4), Bubble-T web-page, Bubble-TV photography of the TV show stage (Chapter 5).	194
9.2	The Visual Sedimentation metaphor applied to a bar chart (left), a pie chart (center), and a bubble chart (right).195	
9.3	Constructing a visualisation with tokens: right hand positions tokens, left hand points to the corresponding data.	196
A.1	Consent form page 1, french version.	208
A.2	Consent form page 2, french version.	209
A.3	Pre experiment questionnaire, french version.	210
A.4	First interview guideline, french version.	211
A.5	Second interview guideline, french version.	212
A.6	Post experiment questionnaire, french version.	213

List of Tables

- 3.1 Different systems composed of discrete elements used to process abstract information before the computer era 46
- 3.2 Different computer era system composed with discreet element to process abstract information. 58
- 4.1 Syntaxe used in PolemicTweet: agreement with the speaker (++), disagreement with the speaker (-), questions raised by or proposed to the speaker (??), and references (==), e.g. quotes or URLs to related content 73
- 4.2 Percent of remote attendee versus local using the backchannel interface, ranged by average of time by domains in minutes. Based on Google analytics report for 15 events. 85
- 6.1 Parameters of the Visual Sedimentation Toolkit 113
- 7.1 Table of existing information visualization design paradigms in comparison to *constructive visualization* paradigm. NofP: Number of Papers referenced by Grammel et al. [Gammell et al., 2013] distributed across Victor's [Victor, 2013] paradigms and including the option of constructive visualizations. 144
- 7.2 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 [Neurath, 1939], Picture 3: ©Michael Hunger, Picture 4: ©General Motors Cf. <http://goo.gl/zMFK6E>. 150
- 8.1 Demographic information and questionnaire answers visualized with Bertifier technique from [Perin et al., 2014]. 166
- 8.2 Summary of identified goals, tasks, and actions. 172

List of Publications

0.0.3 Journal Articles

S. Huron, Y. Jansen, and S. Carpendale, "Constructing Visual Representations: Investigating the Use of Tangible Tokens", *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, Dec. 2014. (To appear.)

S. Huron, R. Vuillemot, and J.-D. Fekete, "Visual Sedimentation", *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, pp. 2446-2455, Dec. 2013.

S. Huron, R. Vuillemot, and J.-D. Fekete, "La Sedimentation visuelle - Outil et technique pour visualiser les flux de donnees a destination du grand public", *Ingenierie des Systemes d'Information*, HERMES, May 2014. (In press)

S. Huron, N. Sauret, and R. Velt, "Design MetaData - Retour d'experience sur un atelier de design interactif interdisciplinaire dans une demarche d'innovation ouverte", *Interfaces numeriques*, LAVOISIER, May 2014. (In press)

0.0.4 Articles in Peer-reviewed Conference Proceedings

S. Huron, S. Carpendale, A. Thudt, A. Tang, and M. Mauerer, "Constructive Visualization", in *Proceedings of the 2014 ACM conference on Designing Interactive Systems*, (Vancouver, Canada), pp. 434-442, ACM, June 2014.

S. Huron, P. Isenberg, and J.-D. Fekete, "PolemicTweet: Video Annotation and Analysis through Tagged Tweets", in *Proceedings of the IFIP TC13 Conference on Human-Computer Interaction (INTERACT)*, vol. 8118, (Heidelberg, Allemagne), pp. 135-152, IFIP, Springer, Sept. 2013.

A. Mazieres and S. Huron, "Toward Google Borders", in *Proceedings of the 2013 ACM Web Science*, (Paris, France), pp. 244-247, ACM, Apr. 2013.

0.0.5 *Articles in Workshop*

S. Huron, R. Vuillemot, and J.-D. Fekete, "Bubble-TV: Live Visual Feed- back for Social TV Broadcast", in *CHI 2013 Workshop : Exploring and enhancing the user experience for television*, (Paris, France), Apr. 2013.

S. Huron and J.-D. Fekete, "User-Supplied Sentiments in Tweets", in *IEEE Visweek : 2nd Workshop on Interactive Visual Text Analysis*, (Seattle, Washington, USA), IEEE, Oct. 2012.

0.0.6 *Posters*

S. Huron, R. Vuillemot, and J.-D. Fekete, "Towards Visual Sedimentation", In *VisWeek 2012 Electronic Conference Proceedings*, (Seattle, Washington, USA), IEEE, Dec. 2012.

B. Bach, P. Dragicevic, S. Huron, P. Isenberg, Y. Jansen, C. Perin, A. Spritzer, R. Vuillemot, W. Willett, and T. Isenberg, "Illustrative Data Graphics in 18th-19th Century Style: A Case Study", in *IEEE Conference on Visualization - IEEE VIS 2013*, (Atlanta, USA), 2013.

P. Dragicevic, B. Bach, N. Dufournaud, S. Huron, P. Isenberg, Y. Jansen, C. Perin, A. Spritzer, R. Vuillemot, W. Willett, and T. Isenberg, "Visual Showcase: An Illustrative Data Graphic in an 18th-19th Century Style", in *Visual Showcase at the Joint ACM/EG Symposium on Computational Aesthetics, Sketch-Based Interfaces and Modeling, and Non-Photorealistic Animation and Rendering*, (Anaheim, USA), Jul. 2013.

0.0.7 *Exhibits*

Here I list exhibits during the Ph.D. periode:

S. Huron and J.-D. Fekete, "Noise of ..." in *IEEE Visweek Art Show*, (Providence, Rhode Island, USA), IEEE, Oct. 2011.

List of acronyms

ANRT : Association Nationale Recherche Technologie

CIFRE : Conventions Industrielles de Formation par la REcherche

InfoVis or InfoVis : Information Visualization

INRIA : Institut national de recherche en informatique et en automatique

IRI : Institut de Recherche et d'Innovation du Centre Pompidou

Isotype : "International System Of Typographic Picture Education"

NSF : National Science Fondation (USA)

NIH : National Institutes of Health (USA)

RSS : Really Simple Syndication

URL : Uniform ressource locator

Most of the work described in this thesis has online resources such as videos, open source codes, demos, or datasets. In order to augment your reading experience and keep a link between this dissertation and all online additional materials, I use a pictogram in front of every chapter to signify the presence of online accessible resources. This pictogram will be placed in the margin as Figure 2 and all the resources URLs will be in its caption.



Figure 2: Online index of additional resources: <http://samuel-huron.gforge.inria.fr/>

Contents

Abstract	iii
Acknowledgments	v
0.0.1 Work setup	v
0.0.2 Thanks to	vi
List of Figures	ix
List of Tables	xxi
List of Publications	xxiii
0.0.3 Journal Articles	xxiii
0.0.4 Articles in Peer-reviewed Conference Proceedings	xxiii
0.0.5 Articles in Workshop	xxiv
0.0.6 Posters	xxiv
0.0.7 Exhibits	xxiv
1 Introduction	1
1.1 Research scope	3
1.2 Problem statement	5
1.3 Research approach	6
1.4 Research path and collaboration	9
1.4.1 On collaboration	9
1.4.2 Research path	9
1.5 Contributions	11
1.6 Outline	13
I Background	17
2 Related Work in Information Visualization	21
2.1 Introduction	22
2.2 Dynamic visualisation of data streams	23
2.2.1 Data streams	24
2.2.2 Summary	26
2.3 Dynamic data analysis	26
2.3.1 Crowdsourcing media annotation	27
2.3.2 Backchannel for live events	27
2.3.3 Summary	28
2.4 Authoring Dynamic Visualization	29

2.4.1	Summary of authoring tools	31
2.5	Chapter Summary	31
3	A Short History of Token as used in Visual Representation	33
3.1	Introduction	34
3.2	Pre computer era: A short history on the use of discrete elements for dynamic representation	36
3.2.1	The evolution of tally marks (20,000 B.C.)	37
3.2.2	Evolution of Tokens Systems (8000 B.C.)	38
3.2.3	Counting board (300 B.C. or before)	41
3.2.4	Counting frame: Abacus, Soroban, Yupana	42
3.2.5	Board games: Senet, Roman Tabula, Mancala (3000 BC)	43
3.2.6	Coin and money system (600 BC till the current day)	44
3.2.7	Building block: Froebel educative materials to teach numeracy (19th century A.D.)	45
3.3	The Printing Press era: from statistical graphics to the roots of information visualization	47
3.3.1	From line charts to area charts (1700-1800)	47
3.3.2	Otto Neurath Isotype. (1882-1945)	49
3.3.3	“La graphique” and the semiology of graphics (1918-2010)	50
3.4	Human computer interaction era: from icons to tangibles	52
3.4.1	Graphical user interfaces and metaphors	52
3.4.2	Assembly of visual building blocks for learning, modelling, designing	54
3.4.3	Summary on token like metaphors in GUI	55
3.4.4	Tangible user interactions	55
3.4.5	Summary	58
3.4.6	Tokens in Information Visualization	59
3.4.7	Summary	59
3.5	Information art and design based on element assembling	60
3.6	Summary	62
II	Dynamic Data Analysis for Non-Experts	63
4	PolemicTweet: Engaging Audience in a Dynamic Data Analysis Process	67
4.1	Introduction	68
4.2	Motivation	69
4.3	Video and sentiment analysis	71
4.3.1	Video content analysis	71
4.3.2	Tweet content analysis	72
4.4	Getting the right metadata to visualize	73
4.5	System Design	73
4.5.1	Backchannel to crowd source tag annotation	73
4.5.2	Engaging the crowd, social protocol for synchronous annotation	74

4.5.3	Video analysis web video player	75
4.5.4	Polemic line design rationale	77
4.5.5	Website	80
4.6	Assessing the Impact of PolemicTweet	80
4.6.1	Data Collection	81
4.6.2	Analysis	82
4.6.3	Is the system sufficiently engaging?	82
4.6.4	Does the visual backchannel incite people to add tags?	84
4.6.5	What is the impact of presenting tweets and tags on a video web player?	85
4.7	Discussion	86
4.7.1	Is it possible to process dynamic data analysis with non-expert users?	88
4.7.2	Follow up applications	88
4.8	Chapter Summary	88
5	Bubble T&Tv - Design Exploration: Combining Token With Physical Simulation & Area Chart	91
5.1	Introduction	92
5.2	Problems	92
5.2.1	Emphasis and Smooth Visual Update of Incom- ing tokens	93
5.2.2	Scalability of token-based system	93
5.3	Bubble-T: Monitoring Twitter during elections	94
5.3.1	Introduction	94
5.3.2	Context	94
5.3.3	Interesting observations	98
5.3.4	Limitations	99
5.4	Bubble-T Summary	99
5.5	Bubble-TV: Adapting Bubble-T for TV show	100
5.5.1	Introduction	100
5.5.2	Context	100
5.5.3	Design	102
5.5.4	Interesting observations from the TV show . . .	103
5.6	Chapter Summary	105
6	Visual Sedimentation - a Generic Token-based Visual En- coding for Dynamic Visualization	107
6.1	Introduction	108
6.2	Sedimentation	109
6.3	Visual Sedimentation	110
6.3.1	Tokens	110
6.3.2	Layout	111
6.3.3	Forces	111
6.3.4	Aggregated Areas	112
6.4	Visual Sedimentation Toolkit	112
6.5	Case studies	115
6.5.1	Common Area Charts with Visual Sedimentation	115

6.6	Deconstructing the Metaphor	124
6.6.1	Methodology	124
6.6.2	Variations on Force-Based Behavior	125
6.6.3	Variations on Aggregated-Area Layouts	126
6.6.4	Variations on Token Encoding	128
6.7	Technical and Design Implications	129
6.7.1	Performance	129
6.7.2	Bursty Data Streams	130
6.7.3	Tokens Deposit and Strata Visual Comparison	131
6.7.4	Token Provenance and Destination	131
6.8	Chapter Summary	132
6.9	Part Two Summary	133
III	Democratization of Dynamic Visualization Authoring	135
7	Defining Constructive Visualization Paradigm	139
7.1	Introduction	139
7.2	Towards Democratization of Visualization	141
7.3	Design Challenges	142
7.4	Inspiration from Froebel, Piaget and Papert	145
7.4.1	Froebel – Discovering the world through simple units	145
7.4.2	Piaget – Learning by construction	145
7.4.3	Papert – Constructionism applied to programming	145
7.4.4	Applying these Lessons to InfoVis	146
7.5	Elements of Constructive Visualization	147
7.5.1	Components:	147
7.5.2	Process:	149
7.6	Examples	150
7.6.1	Chris Jordan: Statistics as Engaging Art Pieces	151
7.6.2	Otto Neurath: Communicating Statistics to People	152
7.6.3	Michael Hunger: Personal Time Management	152
7.6.4	Kevin Quinn: Problem Resolution Tracking	153
7.6.5	Learning from these Examples	154
7.7	Discussion	155
7.7.1	Limitations	155
7.7.2	Applications	157
7.8	Chapter Summary	159
8	Studying Constructive Visualization & Visual Mapping	161
8.1	Introduction	162
8.2	Motivation & Background	163
8.2.1	InfoVis Democratization	163
8.2.2	The Challenge of Visual Mapping	163
8.2.3	Choosing a Methodology	164
8.3	Study Design	164
8.3.1	Participants	165
8.3.2	Setup	167

8.3.3	Procedure	168
8.3.4	Tasks	168
8.3.5	Data Collection	169
8.3.6	Data analysis method	169
8.4	Results	172
8.4.1	Construction	173
8.4.2	Computation	177
8.4.3	Storytelling	181
8.5	Discussion	182
8.5.1	Unpacking the Visual Mapping Process	182
8.5.2	Bottom-up and Top-down Procedures	182
8.5.3	The Ease of Constructing Visual Mapping	183
8.5.4	The Richness of Visual Mappings	184
8.5.5	Internalization of Data to Token Mappings	184
8.6	Limitations	185
8.6.1	Tangible tokens.	185
8.6.2	Study protocol.	185
8.6.3	Method.	186
8.6.4	Generalizability.	186
8.7	Implications for Research and Design	187
8.7.1	Supporting different construction procedures:	187
8.7.2	Exploiting processing fluency:	187
8.7.3	Tangible design:	188
IV	Conclusion and Perspectives	191
9	Conclusion and Perspectives	193
9.1	Progress on research problems	193
9.1.1	PROBLEM 1: Is it possible to engage a public into a useful dynamic data analysis process?	194
9.1.2	PROBLEM1.2: How to visually represent dynamic data for non-experts?	195
9.1.3	PROBLEM2 How can we democratize dynamic visualisation authoring tools?	196
9.2	Contributions	197
9.2.1	Major Contributions	197
9.2.2	Minor Contributions	198
9.3	Reflection	199
9.4	Perspectives	200
9.4.1	Unpacking Visual mapping	200
9.4.2	Constructive visualisation explorations, and evaluation	200
9.4.3	Contributive visualisation	201
9.5	Conclusion	201
A	Additional Materials	203
A.1	Lexicon	203
A.2	Studying Constructive Visualization	205
A.2.1	Dataset	205

A.2.2	Potocol	205
A.2.3	Consent form, Questionnaires, and Interview Guidelines	208
B	Bibliography	215

1

Introduction

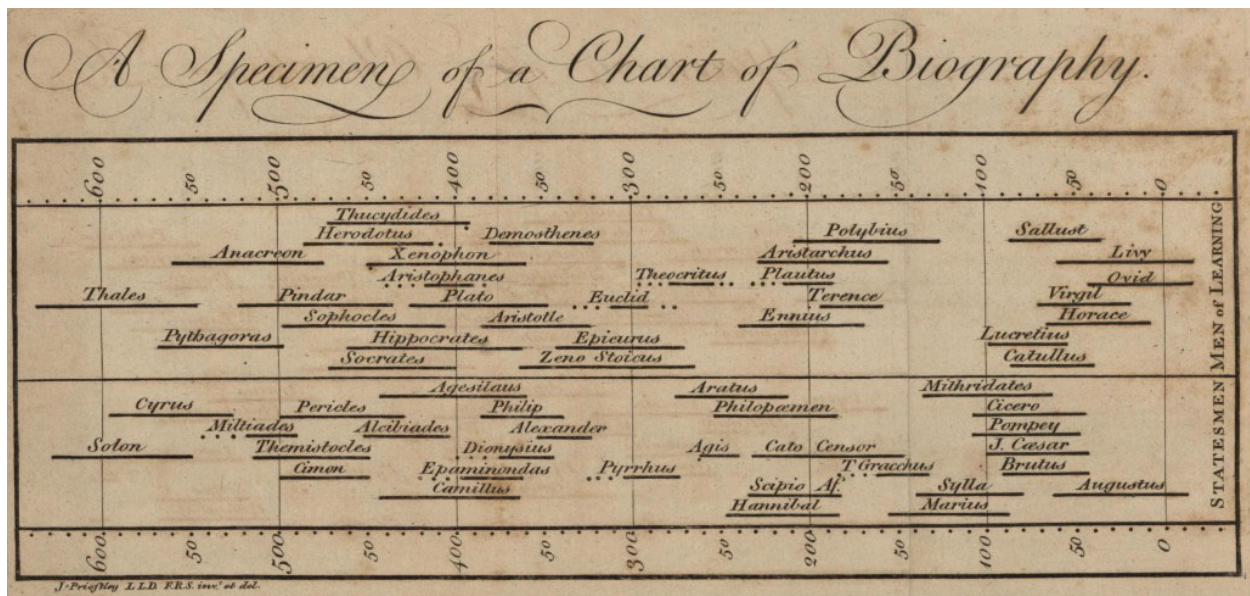


Figure 1.1: First timeline charts by Joseph Priestley (1765).

In order to extend our human ability to memorize, process, manipulate and understand abstract information, we often encode it visually in space and also time. Historically we have used different media to hold the display of this information such as wood, clay or paper (Figure 1.1); now we also use digital media. Some of the representations we are producing are static, while others are dynamic. The term static, implies that the representation does not change over time, for example in Figure 1.1 we can see a plot realised by Joseph Priestley of the philosopher's lifetimes during the period between 600 years BC to the year 0. This visual representation is inked once and will not be updated, representing one state of this data. Other visual representations are dynamic, for instance, sand clock (Figure 1.2). A sand clock is an instrument of time measurement as well as a visual representation of this measure. When the sand clock is reversed, the sand falls down due to physical forces such as gravity, and then is constrained by the physical resistance of the glass to flow from one container to the other. During this time, one can see the full sand container and the sand flow into an empty container. A visual



Figure 1.2: Example of a sand clock. Some Rights Reserved ("running out of time" by Rui Malheiro) <http://goo.gl/1rLU30>

comparison between the two resulting sand piles allows one to assess the elapsed time between reversing the sand clock and the end of sand flow. In this case, the visual representation is updated with sand flow, and for this reason it is a dynamic representation. However nowadays it's seems we are more using digital artefacts than sand clock to process visual representation of abstract information.

Digital artefacts have played an important role in our ability to produce, gather and store data. This phenomenon is increasing rapidly. But also the amount of published data is not only rising in volume but also in update frequency, diversity in the variety of audiences. A few decades ago, data publication was asynchronous, disconnected, and stored in unconnected media. Due to the increased availability of computers, the Internet and the web, data publication activity is now increasingly synchronous, connected and stored in digital formats which can be connected. Consequently, the frequency of data publication is rapidly growing, so much than in certain domains it is in the form of a data stream. Data streams are often updated in real time, as observed for systems such as Twitter, RSS feeds, and chat streams. Unfortunately, due to limitations in our individual cognitive abilities, our ability to make sense of this data remains constant and limited and does not grow with the volume of data. Moreover, processing this data for analysis is becoming more and more challenging and expensive, often also requiring a high degree of expertise, computation, time and energy. One possible solution to these limitations could be to create piece of technologies that allow everyone to augment their own capacity to process data.

In the past few years some visual systems such as Many Eyes [Viégas et al., 2007], and later Tableau Public [Tableau, 2014b] have began to explore how web based information visualisation might be used to democratize [Viégas et al., 2007] visualisation and widen the accessibility of data to those with limited expertise in information visualisation so that they can become involved in the analysis of this data. Such platforms allow any person interested in the data to create, publish and discuss visualisation about their own data. More recent systems such as Comment Space [Willett et al., 2011] allow people to annotate, tag and comment on visualizations in support of data analysis. These systems point towards a future in which visualisation would be a new language of data analysis, and where everyone should be able to read and write on the fly in terms of data visualization, just as we do with natural language.

However these tools to create, publish and discuss information visualisation are mostly static and do not address the requirements specific to processing data from constantly updated data streams. Moreover these tools are limited to predefined established visualizations and do not provide the possibility for people to specify their own visual representation. This major challenge of enabling a wide audience to create their own visualisation has been identified by several researchers as one of the major issues for information visualization. The NIH/NSF visualisation research challenges report states:

"When visualisation researchers talk about scaling, we usually mean a large dataset. [...] We consider an alternate perspective: Instead of scaling the size of the data, what happens when we scale the size of the audience?" Fernanda Viégas et al. [2007]

“A big challenge is to create [...] a system that, while clearly not comprehensive and all-powerful, does help to enable non-experts to perform tasks [...] in any reasonable time frame. [...] The goal is to make visualisation a ubiquitous tool that enables ordinary folks to think visually in everyday activities” [Johnson et al., 2006]. Similarly, Heer and Shneiderman specifically point to the need to create new interfaces for visualisation specifications: “Novel interfaces for visualisation specification are still needed. [...] New tools requiring little to no programming might place custom visualisation design in the hands of a broader audience” [Heer and Shneiderman, 2012]. The question I address in this thesis is: **Can we take a step towards empowering non-experts in their use of visualisation by enabling this wider audience to contribute to data stream analysis as well as to create their own visualizations?** This question is challenging at several levels, ranging from the social level to the human computer interface level. At the **social level**, one of the major challenges is how to engage non-experts in a useful dynamic analysis process. At the **human computer interface level**, we need to create fluent methods and tools that allow non-experts to read and manipulate simple, dynamic, and expressive data representations.

This introductory chapter first positions the thesis domain and research scope within **democratization of dynamic visualizations** for dynamic data, in Section 1.1. Then the problem statement is defined in Section 1.2. This leads to the description of the thesis research approach and methods in Section 1.3. The contributions of this work are summarised in Section 1.5 and Sections 1.4 and 1.6 contains a brief outline of the structure of the dissertation.

1.1 Research scope

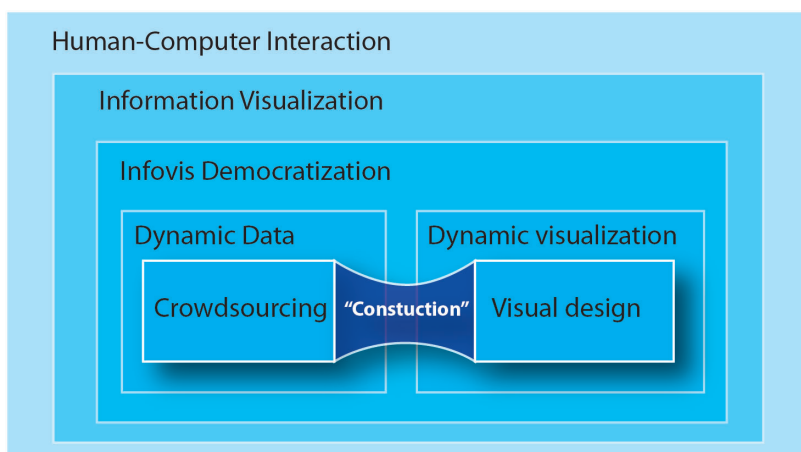


Figure 1.3: Research Scope.

The research topic of this thesis lies in the domain of information visualisation (InfoVis). InfoVis is a domain that is related to the study of human computer interaction (Figure 1.3) that endeavours to understand and provide technological solutions for the visual analysis and processing of data. The information visualisation re-

search field is grounded in the observation that human cognition is particularly efficient with respect to processing visual information. InfoVis is now a wide interdisciplinary field of research, encompassing cognitive psychology [Treisman, 1985, Healey, 2007, Ware, 2004], statistics [Beniger and Robyn, 1978, Wilkinson et al., 2006], computer science [Card et al., 1999, Spence and Press, 2000, Shneiderman, 1996], design [Vande Moere and Purchase, 2011, Neurath, 1939, Bertin, 1973, Tufte and Graves-Morris, 1983] and computer graphics.

Information visualisation is defined by Stuart Card [Sears and Jacko, 2007] as “a set of technologies that use visual computing to amplify human cognition with abstract information”. Information visualisation has been used by experts to represent, process and explore data for a number of decades in several contexts, including data intensive science, industry, and government. As these technologies have been shown to be efficient, they have started to become more widely adopted by practitioners from different backgrounds. We are now seeing visualizations in personal blogs, as art work, in web services, in the news and also as a growing part of the quantified-self movement. This increasing adoption of visualisation by different groups has led to new population -and application- specific requirements. Design and research must now create new technologies to address this new population.

Although there is no clearly defined InfoVis sub domain for addressing this problem, some pioneers such as Fernanda Viégas et al. [2007] use the term “democratization” to define the use of visual analytics tools by InfoVis non-expert users. The goal of these tools is to empower people to make sense of data by using visualisation techniques. The challenge of creating new information visualisation tools, which are easily accessible to everyone, has been emphasized by several other researchers [Heer and Shneiderman, 2012, Moorhead et al., 2006].

Coincidentally, at the same time as the increasing democratization of visualization, dynamic data stream services are becoming increasingly popular. Dynamic data, which can be defined as sequences of typed objects updating over time, are very common with social network updates (e.g. tweets, emails, RSS feeds, video streams) and in a lower level characterizations of these updates such as network logs and commits in distributed version control systems. This data presents many challenges for visualisation since it *appears* at unpredictable times, *accumulates* over time, and can be highly *heterogeneous*, containing different media such as texts, images and videos.

In this dissertation I focus on the **design of dynamic visualizations for InfoVis non-experts** (Figure 1.4). My study will address three different aspects of this approach:

- (1) If and how an InfoVis non-expert can analyse dynamic data streams. The motivation for this first aspect is to demonstrate that using simple dynamic visualizations in datastream analysis tasks can enable the inexpensive collection on rich data that can be useful for future analysis.

(2) How can better dynamic visualisation techniques be designed to empower InfoVis non-experts to manage dynamic data. This second aspect is motivated by the lack of guidelines, techniques and paradigms for guiding InfoVis non-experts in reading and authoring simple expressive and dynamic information visualizations.

(3) How can we provide the means for InfoVis non-experts to author their own dynamic visual representations.

1.2 Problem statement

Many of us encounter dynamic data streams in our everyday lives, both at an individual and at a collective scale. Examples include our social network updates (status, news articles, mails); and more personal data streams such as our bank accounts, which are composed of lines of debits and credits.

All these dynamic data streams are updated at different rates, ranging from several times per second to just once a month. Some of them are more or less public and can be used collectively (as is the case in social networks), and some are more private and personal. Processing tasks on these streams can also differ in their complexity. For example, tagging, liking or annotating a social network post is much simpler than operating the update of a visual representation.

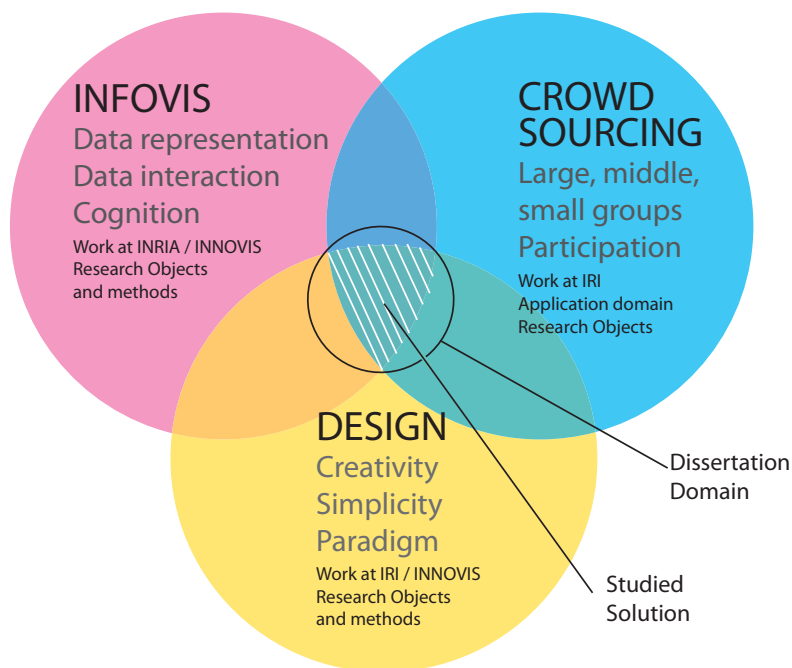


Figure 1.4: Domain Scope.

The question I address here is: How can various audiences manipulate dynamic data? To address this question we will focus on two problems:

PROBLEM 1 *Is it possible to engage the general public in a useful dynamic data analysis process?* This research begins by studying how to crowdsource dynamic data analysis during an event with the support of information visualization. This problem corresponds to an IRI application need: to gather and exploit specific metadata during the event. This implies several sub problem challenges to be taken into account: (P1.1) First, is it *possible* to leverage a crowd of InfoVis non-experts to *analyse* dynamic data for free? and (P1.2) second, *how do we visually represent* dynamic data for this specific audience?

PROBLEM 2 *How can we democratize dynamic visualisation authoring?* Solutions discovered that to solve (PROBLEM 1) may provide a means for a designer, engineer, or computer scientist to create dynamic visualisation that can engage the general public. However, we wanted to refine our research to find a solution that allows various audiences to create their own dynamic visualizations. This implies several sub problems: (PROBLEM 2.1) What are the *challenges* to create a visualisation authoring tool for non-expert users? (PROBLEM 2.2) What *means* can support a process of visual mapping activity with this population? (PROBLEM 2.3) And most of all, can we better understand how *visual mapping* is generated?

1.3 Research approach

Information visualisation as computer science is a research domain involved in the study of phenomena partially designed by the scientist. This particularity makes this science part of a design science [Simon, 1969]. Design methods [Howard et al., 2008] in the domain of information visualisation are starting to be discussed [Munzner, 2009, Vande Moere and Purchase, 2011, Sedlmair et al., 2012, Goodwin et al., 2013] but creative design [Howard et al., 2008] and ideation methods are not specifically defined in the information visualisation domain. To meet my needs of designing creative artefacts I used methods from other domains. These methods are the use of metaphors [Gentner, 1983, Blackwell, 2006], design space exploration [Pugh, 1991, Laseau, 2001, Greenberg et al., 2011], combination and mutations of systems [Cross, 1997] and inspiration [Howard et al., 2008, Osborn, 1953].

In addition to the creative design methods applied to my research,

I also applied a different set of behavioural research methods. Runkel and MacGrath [1972], Mcgrath [1995] characterized different research strategies for studying human behaviour using a framework with several dimensions: (1) from obtrusive to unobtrusive research operations, (2) from generalizable behaviour systems to particular behaviour systems. This framework also describes different points of maximum concerns: generality over actors, precision of measurement of behavior, system character, and context. Each of these strategies provides different benefits and limitations. Below are listed the different methods I used for designing (*in italics*) and for observing phenomena:

Collection. The process of collecting examples to get inspiration is a common creative design method [Herring et al., 2009, Howard et al., 2008]. This process allows one to identify features or forms of interest as well as desired features that do not exist yet and corresponds to some needs. I used this method in Chapter 3 to inspire our design statement to use “individual discreet element”, I also used this method in Chapter 7 to define the theoretical basis of this thesis.

Field Experiment. Field experiment according to McGrath [Mcgrath, 1995] is defined as “working within an on-going natural system as unobtrusively as possible, except for intruding on that system by manipulating one major feature of that system.”. This approach provides a strong external validation but is limited to a narrow context and system. In this research I was concerned with studying if the public attending an event could evolve with dynamic data analysis tasks. In order to study these phenomena with respect to the targeted population I chose to use field experiments as a less obtrusive research strategy. My rationale for this choice was the complexity of simulating engagement of a crowd in a lab experiment and the advantages of being able to observe a real-world situation. I used this method in Chapter 4 and Chapter 5. In the first case I used a comparatively formal approach, in the second I used a more relaxed one.

Combination. According to Cross [1997]:“Creative design can occur by combining features from existing designs into a new combination or configuration.”. Combining allows one to use an existing system and enhance it by adding part of the features of another system resulting in a third new system that provides features from the two parent

systems. This method supports the extension of the features of a system. I used it in different parts of this dissertation such as the application of physical rules to token systems in Chapter 5 or the combination of token systems and area charts in Chapter 6.

Metaphor. Gentner [1983] describe this process as a binding of properties between two structures (Figure 3.35). Metaphors are frequently referred to in human interaction and Blackwell [2006] has studied in detail the history of metaphors in this domain. A metaphor can be both a generative technique for creative design [Cross, 1997], or a support to explain certain properties of a system [Carroll et al., 1987]. I used metaphors in Chapter 6.

Observational Study. This type of study is a qualitative data analysis approach as described by Creswell [2012]. Our observational studies was conducted in a laboratory setting and was carefully constrained, thus it falls into the laboratory experiment category of Runkel and McGrath. This type of study does not provide precision of measurement or generalizable results but offers support for the identification of a phenomenon, instructions for future research and designs, and provides rich detailed quantitative observations. We used this method in order to get a better understanding of the visual mapping procedure within the *constructive visualization* paradigm, had defined. The details of this research are given in (Chapter 8).

1.4 Research path and collaboration

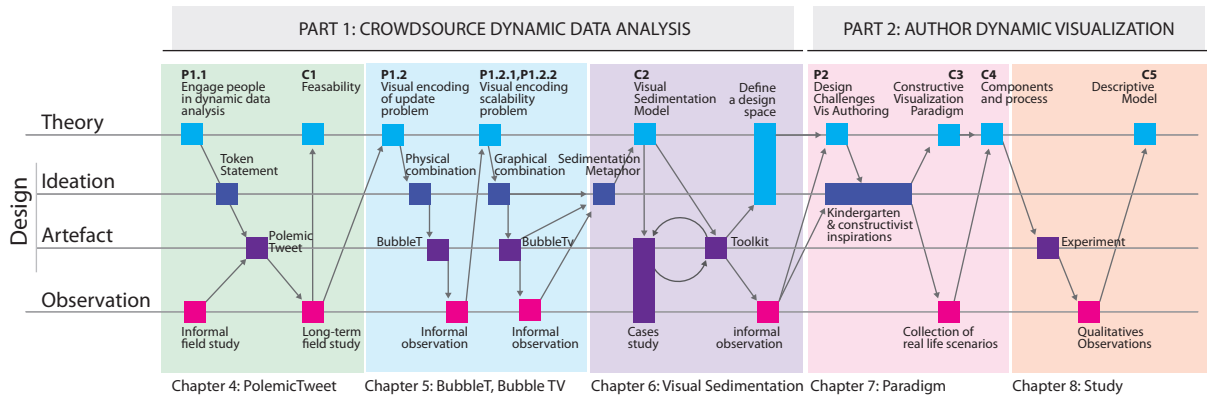


Figure 1.5: Overview of my research path through a modified version of MacKay and Fayard Triangulation framework. Each box is a phenomenon and each arrow is the move from one phenomenon to others.

In this section I present the collaboration I had the pleasure to have during this research, as well as the path that join each project of this research and form my inquiry.

1.4.1 On collaboration

Throughout these past three years, I have had the good fortune to work with people with different backgrounds, education and sensitivities. Our exchanges provided invaluable insights relating to my research and inspired the projects presented in this thesis. In the margin note in front of each chapter, I acknowledge the work of my collaborators for each project so as to clarify who was involved in each project and for which publications. At the same time however, a Ph.D. dissertation reflects a single individual's thread of thinking, and this dissertation is written from my perspective as an individual researcher. Every project was initiated by me, and in every case, I held the key role in conceiving, designing and leading the project. In the section below, I will describe the role of each collaborator as well as my own for each project. What I wish the reader to bear in mind is that these collaborations were extremely fruitful intellectual exchanges where the whole was always greater than the sum of the parts.

1.4.2 Research path

Research in human computer interaction involves different disciplines, including design, humanities, computers sciences, and experimental sciences. During my research, I iterated through the following cycle: identifying a problem, generating ideas, designing an artefact, observing this artefact impacting the situation, and finally theorizing on the effect of this impact. To describe the path I took through these different activities I extended Mackay and Fayard [1997] triangulation framework. This allowed me to represent these activities of observing, identifying, generating, designing and theorizing within a common

framework. To extend the framework to fit my process, I forked the “design” dimension of this framework into two branches, “design of artefact” and “ideation” referring to the process of creative design, generating [Cross, 1997, Howard et al., 2008] and refining [Pugh, 1991, Laseau, 2001, Greenberg et al., 2011] new design. In this section, I will use this extended framework to describe the relations between the activities in my research process.

In Figure 1.5 we can observe that the overall process is divided into two overall columns dedicated to the two problems presented in the previous section. The first part reports on the design and deployment of different crowd-sourcing applications that allow us to evaluate the possibility of engaging users in dynamic analyses processes. The second part is focused on defining and studying the *constructive visualization* paradigm to empower non-expert users in the authoring of dynamic visualizations. These two parts are divided into research phases that each correspond to a chapter of this dissertation and are described further.

I started this work from an initial research question (Figure 1.5,P1): “Can we engage non-experts people in useful dynamic data analyses processes?” To study this general problem I downgraded the question to a specific application case: the annotation and tagging activity of video conferences. Based on different informal field observations I designed a first token-based visualisation system “PolemicTweet”. Then I ¹ deployed, studied and evaluated this system during a long term live deployment. From these observations I identified limitations of the visual encoding of data updates (Figure 1.5,P1.2). To explore a solution to this problem I combined a token-based visual encoding system with physical simulation the result of this combination is the design of a new system called “Bubble-T” (Chapter 5). This system is a public application providing a visual representation of incoming tweets during a short period of time. After deploying this application on the web I ² received positive feedback which encouraged us to go further with this design approach. The system also allowed us to identify some drawbacks in my visual encoding design. The major problems (Figure 1.5,P1.2.1,P1.2.2) was the scalability of the visual representation as token visual representation system have a one-to-one mapping. The property of this visual system limited the volume of data it could represent in a limited space. On the other hand visual representations such as area charts allow different scale mappings. To address these issues I ³ decided to combine this two visual system: token and area chart. The result of this combination is a second system called “Bubble-TV”. The design and deployment of these two systems informed the creation of a more general token-based visualisation metaphor: “Visual Sedimentation” (Figure 1.5,C2). I ⁴ refined this metaphor while developing several visualisation case studies and creating a toolkit. This toolkit enabled visualisation experts to be able to create dynamic visualizations of streaming data that would engaging to the “crowd” but left open the issue of enabling the general public in the creation of their own visualizations.

¹ with the collaboration, for the implementation: of Raphael Velt, Karim Hamidou, Yves Marie Haussonne, for the deployment: Florence Duc, Nicolas Sauret, Vincent Puig, and for the evaluation and analyses: Petra Isenberg, Jean Daniel Fekete.

² with the collaboration for the design and implementation: Raphael Velt, Romain Vuillemot, Yves Marie Haussonne

³ with the collaboration for the implementation and analyze of: Nicolas Sauret, Raphael Velt and Romain Vuillemot

⁴ with the collaboration for part of the design and analysis, Romain Vuillemot and Jean-Daniel Fekete.

The first part my work focused on two major problems: **PROBLEM 1.1** how to engage people in a useful dynamic data analysis process, and **PROBLEM 1.2** how to visually represent dynamic data updates. In the first part of this research I explored different token-based visualizations in order to create dynamic data analysis systems for non-expert users. In the second part of this research I generalized the previously explored tokens based approach to dynamic visualisation authoring (**PROBLEM 2**). First, I⁵ abstracted the design challenges relative to authoring tools for non-expert users (Figure 1.5,P2). I sought inspiration from early education theory and devices such as Kindergarten pedagogical material [MacCormac, 1974] and Constructivism [Piaget, 1989]. From this inspiration I both defined the *constructive visualization* paradigm (Figure 1.5,C3) and collected observations on real life historical and contemporary examples. From these examples I extracted the components and processes of *constructive visualization* (Figure 1.5,C4).

⁵ with Sheelagh Carpendale, and in collaboration with Alice Thudt, Anthony tang and Michael Maurer.

In order to study this paradigm I⁶ then designed an experimental protocol to observe the behaviours this paradigm would engage. Observations gathered during this experiment allowed us to abstract a descriptive model of how non-experts process visual mapping by assembling tokens (Figure 1.5,C5).

⁶ with the collaboration of Yvonne Jansen and Sheelagh Carpendale

1.5 Contributions

This research makes two main contributions: a **first research part** (Part II) is focused on a social data analysis scenario: designing and deploying applications during events with the intention of engaging the public in dynamic data analysis processes, and the **second research part** (Part III) is focused on the generalization of interface design principles building on findings from the first part.

- **Part II:** Application for social dynamic data analysis
 - I gathered experimental evidence to support the following: (A) that it is possible to engage people during an event in a manner that supports them conducting analysis tasks on dynamic data, (B) that these types of analyses on dynamic data can be effective, (C) that the data produced can be useful afterwards. These contributions take a significant step towards solving the **PROBLEM 1.1** as describe in Chapter 4.
 - I identified the need for and designed several different systems for real time crowd-sourcing via public participation through information visualization. These systems were deployed in different situations ranging from medium-sized groups (20 people), to large-sized groups (300 people) and very large groups (700 000 people). As part of this contribution, I take a step towards scalability of the visualizations used **PROBLEM 1.1** and the **PROBLEM 1.2**. This is discussed in Chapter 4 and Chapter 5.

- I used a metaphor to create a model for a new family of visualisation techniques that provides a visual encoding of the update of dynamic data at different time scales. This metaphor is generative and allows me to provide a class of new visualizations. This contribution addresses [PROBLEM 1.1](#) and is mostly discussed in Chapter 6 and Chapter 5.
- **Part III: Democratization of dynamic visualisation design**
 - I identified a paradigm of constructive visualization. This new design paradigm provides a means for non InfoVis experts to author their own dynamic visualizations. The InfoVis community has previously recognized that creating a simple tool for non-expert users is important, but the application of the constructive principle to dynamic InfoVis design and usage has never been examined before. This contribution addresses [PROBLEM2.1](#) and [PROBLEM2.2](#) is discussed in Chapter 7.
 - I created an operational description of the *constructive visualization* authoring paradigm that can be used in designing and building new information visualisation systems. I built a conceptual framework based on the study of how people construct visualizations out of tokens. This framework defines the major components and processes needed to author a *constructive visualization* environment. I also described the use of these components and processes in four real life scenarios. This contribution is in response to the [PROBLEM2.2](#), and is discussed in Chapter 7.
 - I gathered empirical evidence that people can create, update, annotate and discuss visualizations that they have constructed by assembling tokens that have been mapped to data. I explored in the details of the logical tasks and actions related to the visual mapping. Finally I extracted from these observations a model that describes the actions, their purposes, and their relationships. I also studied the sequence of process actions and observe a high diversity. This contribution is in response to the [PROBLEM2.3](#) and is discussed in Chapter 8.
 - I conducted an initial exploration of details of the logical tasks and actions previously simply considered as a “black box” that was called the **visual mapping** process. From my observation I report the diversity of the sequence of actions, where though people basically used the same actions they did not adhere to the same ordering of actions. From this actions I define a model of the visual mapping process in a tangible constructive environment that describes the actions, their purposes and the interplay between them that together result in the building of visual representations. This contribution is in response to the [PROBLEM2.3](#) and is discussed in Chapter 8.

1.6 Outline

This dissertation is divided into three parts. Part I holds two chapters, one on the related work from computer science literature and the other contains a short history of the use of tokens in visual representations. Part II, starts with an exploration into non-expert participation into dynamic data analysis and ends with the development of a token-based visualisation paradigm, Visual Sedimentation, which has been designed to fit these needs. Part III explores the possibility that a token-based approach can also be used to enable non-experts in the creation of their own visual representations.

Part I : *Background*

The first part of this dissertation sets the stage for this research by providing the background. The background is divided in two chapters. One covers the relevant computer science literature providing the recent related InfoVis work. The other chapter provides an historical point of view on the dynamic visual representation systems that have inspired our design approach.

Chapter 2 : *Related Work in Information Visualization.*

This chapter covers the computer science literature that is the most closely related work and that is relevant to our areas of focus: crowdsourcing visualizations, visualisation of dynamic data, and information visualisation authoring tools.

Chapter 3 : *A Short History of Tokens as Used in Visual Representations.*

This chapter takes a more historical point of view, exploring the use of tokens in visual representations throughout the ages. This chapter presents a non-exhaustive collection of existing artefacts that have been designed to manipulate information during different periods of history. All these systems were selected because of a common factor: they are composed of individual discreet visual elements, or tokens.

Part II : *Dynamic Data Analysis for Non-Experts.*

The second part of this dissertation is related to the design and evaluation of different applications that were designed, implemented and deployed for use collectively by non-experts for analysing and visualizing dynamic data.

Chapter 4 : *PolemicTweet: Engaging the Audience in a Dynamic Data Analysis Process.*

This chapter contains the initial research, reporting on the design and analysis of, PolemicTweet, a system designed to crowd source conference video tagging with structured sentiment metadata. A long-term evaluation of this system under live deployment allowed me to study (1) whether it is possible to engage people during an event to conduct data analysis tasks, (2) if this type of analysis task can be effective, and (3) if the data produced are useful afterward.

Our findings confirm that a well-designed social protocol dedicated to the context and coupled with a customised interface does permit the gathering of useful information.

Chapter 5 : *Bubble-T & TV, a Design Exploration: Combining Tokens with Physical Simulation & Area Charts*

This chapter contains an extension of the research in Chapter 4. Based on considerations and insights into crowdsourcing and non-expert tagging of dynamic data as described in Chapter 4, two new dynamic information visualisation systems are designed to support similar social protocols and based on token manipulation. In Bubble-T, the purpose was to create an interface that could show the updates over time for an event with a longer time scale. In Bubble TV, the focus was on providing a similar tagging system through tweeting during a live event but with a different group scale (the audience of a TV show).

Chapter 6 : *Visual Sedimentation: a Generic Token-based Visual Encoding for Dynamic Visualization*

This chapter is a pivot point between the research on exploring dynamic data in social scenarios and the visual representation design considerations. From the previous scenario based research (Chapters 4 and 5) a barrier between token-based visualizations and area chart visualizations was identified. In this chapter, this problem was addressed by introducing a novel design metaphor for visualizing dynamic data streams. The metaphor is drawn from the real-world sedimentation processes and this metaphor has helped aid comprehension. In this chapter I discuss how this sedimentation metaphor addresses the specific challenges of smoothing the transition between incoming and aging data, and between a token-based visualisation and an area-based visualization. Here the metaphor's design space is explained and a toolkit that facilitates its implementation is described. The generative capabilities of the design space with this toolkit are described.

Part III : *Democratization of Dynamic Visualization Authoring.*

The first part of this research was dedicated to the creation of functional applications for different dynamic data analysis tasks and the development of interfaces dedicated to non-expert populations of different sizes. This drew attention to how the assembly of tokens could be useful for authoring dynamic visualizations. In this part of the dissertation, these observations are extended to inform the design of information visualisation authoring tools for non-experts and an observational study focusing on what this approach can teach us about visual representation process is described.

Chapter 7 : *Defining Constructive Visualization Paradigm.*

This research phase starts with the extraction of the most significant research challenges for democratizing dynamic information visualisation authoring tools from the research in previous chapters. Examining how to address this problem has led to defining a

new design paradigm through which people can manipulate the data dynamically in conjunction with visual variables to construct a visual mapping. This involved: (1) taking a fresh look at theories of how people understand concepts that are new to them, (2) observing practices in non-academic situations where people are actively engaging in the construction of their own visual representations, and (3) identifying from these practices the components and processes of this new paradigm.

Chapter 8 : *Studying Constructive Visualization and Visual Mapping.*

To better uncover the visual mapping process as well as the implications of the *constructive visualization* paradigm presented in the previous chapter, an exploratory study was designed and conducted. In this study people were asked to create, update and explain their own information visualizations using simple materials such as tangible building blocks. I learned that all participants, most of whom had no experience in visualization, were readily able to create, dynamically update and talk about their own visualizations. On the basis of these observations, I discuss the actions of our participants in the context of the development of their visual representations. From this I highlight some implications for tool design that can enable broader support for InfoVis authoring.

Chapter 9 : *Conclusion and perspectives.*

This chapter holds the conclusion and discusses possibilities for future work. This chapter summarizes the processes during this research, declares the contributions, discusses the limitations of this work, and makes suggestions about possible future work.

Part I

Background

Representation: “A representation is a formal system for making explicit certain entities or types of information.” Marr [1982].

This part of the dissertation places the research in context. The background is divided into two chapters. The first, Chapter 2, provides background with respect to computer science, in particular, the recent related InfoVis research. The second, Chapter 3, provides a historical point of view on the dynamic visual representation systems that have inspired my design approach.

Although there has been widespread recognition of the importance of the democratization of InfoVis [Viégas et al., 2007, Heer and Shneiderman, 2012, Johnson et al., 2006] in the InfoVis community, common definitions of InfoVis include the use of computer technology. Perhaps the best known is from Card et al. [Card et al., 1999] “The use of computer-supported, interactive, visual representations of abstract data to amplify cognition.” However, the purpose of the research in this dissertation is to discover and investigate a new paradigm for the way we design, update and manipulate dynamic information visualizations for all the people who are interested in the data, whether they are experts or non-experts with respect to data science and information visualization. Thus to provide an appropriate background it seemed important to consider both the related found in computer science literature and related work from more distant domains that has been used to represent information throughout the ages.

In Chapter 2, I survey the related computer science research, in particular looking at three research areas: (i) the dynamic data analysis of data streams by non-experts, (ii) the design of dynamic visualizations that represent streaming data, and (iii) the authoring tools for visualizations. Existing visualisation tools tend to be designed for use by information visualisation and data experts and for static datasets. But technological advances such as the generalization of data streams, and social change such as the democratization of data in our everyday lives, create a need for dynamic InfoVis tools that anyone can use.

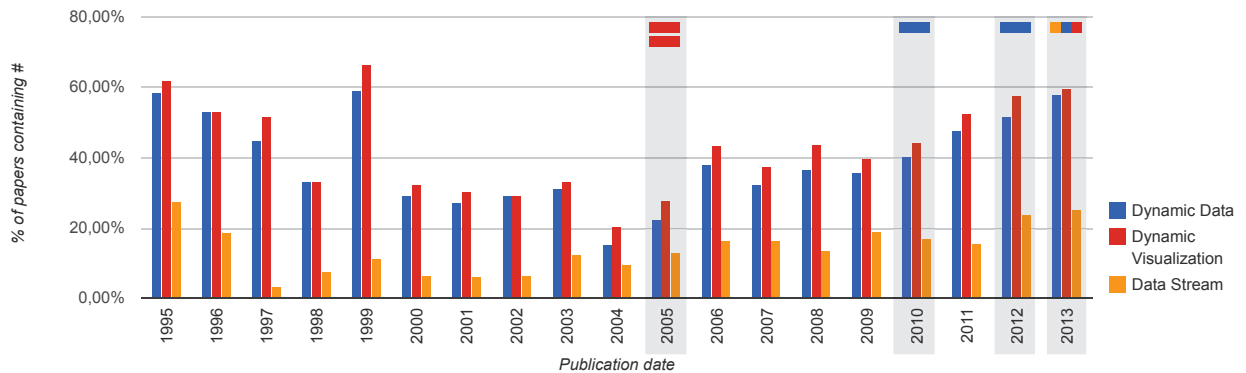
In Chapter 3, I investigate the history of the use of discrete elements in dynamic visual representations in others domains. This use of discrete elements relates to my work on authoring dynamic visualizations. Through reviewing the history of the use of discrete elements, I have been able to identify the following characteristics of current tools: (i) they are complex to use and understand, (ii) they are limited in their degree of expression and personalization, or (iii) they have difficulty supporting dynamic functions such as custom remapping and updating. From the tools currently available it is apparent that the most powerful authoring tools require coding, which in turn requires a high level of expertise. Based on these observations, the third part of this dissertation (Part III) discusses my exploration

of the challenges of developing dynamic InfoVis authoring tools for non-experts.

From these surveys (Chapters 2 and 3) I have identified that: *(i)* backchannel and crowdsourcing media annotation are extremely well-suited application domains for exploring whether the general public can be effectively involved in the process of dynamic data analysis, *(ii)* dynamic visualisation is a young area with as yet no generic vocabulary and mature techniques, although some design approaches based on discrete elements do exist, and *(iii)* dynamic information visualisation authoring tools are limited by different constraints and do not provide a simple solution that allows non-experts to author their own dynamic visualizations.

2

Related Work in Information Visualization



This chapter contains a brief survey of three different but interrelated aspects of research that are important for *dynamic information visualization* for non-experts. These are: (i) engaging non-experts in dynamic data analysis, (ii) designing dynamic information visualizations for non-experts, and (iii) authoring tools that have been designed to help non-experts create *dynamic information visualizations*. Each of these aspects is relatively new and still under investigation. However, there has been considerable research work in closely related topics, such as crowdsourcing, human computer interaction, and visual analytics. These latter topics have also been drawn upon as a basis for this research into dynamic visualisation for non-experts. In this chapter, I present literature related to: (i) the exploration, application and empirical work developed in this dissertation, (ii) the information visualisation research community and, (iii) designing *dynamic information visualizations* for non-experts.

Figure 2.1: Plot of the number of papers returned by IEEE digital library for the venue TVCG with the query “dynamic data”, “dynamic visualization” and “data stream” research in full text by year. The percentage is computed according to the number of paper containing “ieee”. The rectangle on top corresponds to the papers in which the authors use one of these keywords according to the website Keyvis.

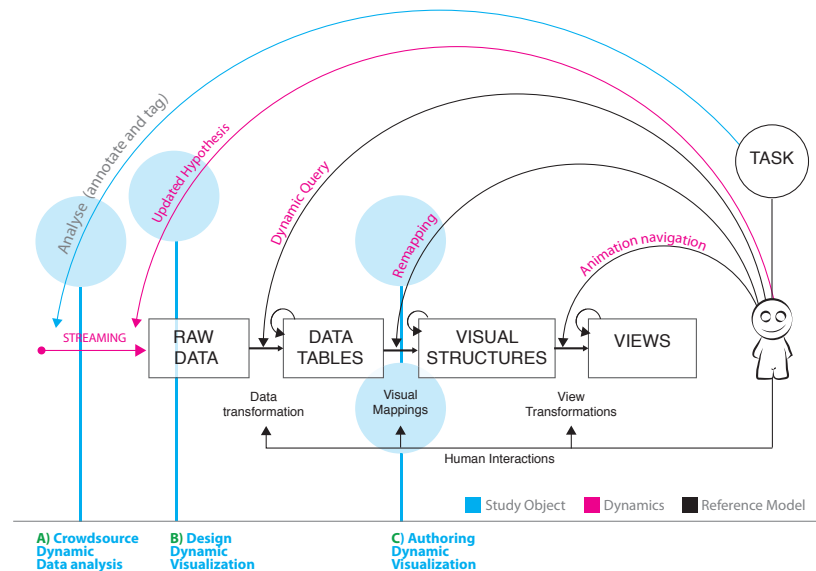
2.1 Introduction

The words *dynamic visualization* and *dynamic data* are already widely seen in information visualisation discussions, but few authors have used it to tag their own papers. According to the website Keyvis ¹, which records all the keywords from all papers from the journal IEEE TVCG ² between 1995 and 2013, very few authors used the keywords “dynamic visualization” (3), “dynamic data” (3), “data stream”(1) to classify their papers, although many authors use these terms in their papers. A study of full text over the years yields the distribution shown in Figure 2.1. Most of these papers do not use the term dynamic in the context of dynamically updating data, rather they refer to visualizations that use a fixed dataset that indicates change over time, or the inclusion of interaction techniques that can change the views. The difference between the number of authors applying the keywords and the number of papers containing these terms may suggest that the topic is not yet structured in the domain, even if it is an area of interest.

¹ <http://www.protedi.at/keyvis/>

² IEEE Transactions on Visualization and Computer Graphics is the main academic journal on information visualization.

Figure 2.2: Dynamic Visualization pipeline freely adapted, with object of research in blue (Diagram adapted from Cottam et al. [Cottam et al., 2012].



However, recently, Cottam et al. [2012] defined the term dynamic visualisation as “visualizations that change over time”, and proposed an updated version of the InfoVis pipeline reference model [Card et al., 1999] based on dynamics (Figure 2.2). According to the authors the “change” can be due either to dynamic data updates, to a user action (interaction), or to the visualisation itself (animation). In this research, I focus on three problems that appear at different levels of this pipeline. The first is to enable InfoVis non-experts to process *analysis on dynamic data*. The second is to determine how to *design a dynamic visualization* that is easy to understand and remains interpretable over time for the same population of users. The third is to study how the *visual mapping is processed* by InfoVis non-experts, and to subsequently devise a way to create some specific dynamic visualisation authoring tools that

fulfil their needs.

Few applications exist for InfoVis non-experts to process analysis on dynamic data. In order to investigate this first point I focus on digital backchannels. A digital backchannel is a thread of information that accompanies a live event. With the growing pervasiveness of real time social networks such as Twitter, digital backchannels are becoming more and more common for monitoring what is happening live during public events such as lectures, conferences [Harry et al., 2009, Hannon et al., 2011], TV shows [Doughty et al., 2011], and natural disasters. Several authors [Diakopoulos and Shamma, 2010, Dork et al., 2010] showed that this stream of information could be used as an index to analyse other synchronous datastreams such as videos, photo collections, and others. In the first section of this related work I present the research done on using backchannels to crowdsource dynamic annotation of a data stream.

While the scope of my research is confined to dynamic visualisation of textual datastreams, it relates to providing a visualisation that permits people in a public context to easily interpret visualizations of the data as it is updated. Work done in the database systems [Babcock et al., 2002] and machine learning [Gaber et al., 2005] communities tends to focus on architecture and development problems. In contrast, I focus on finding a correct visual mapping that delivers an engaging visualisation that is easy to interpret over time. In the second subsection I present previous work done in the domains of dynamic visualisation of data streams and visualisation updates.

Fry [2000] highlights the fact that in a data stream the values and structure of data can both change over time. For this reason, Fry asserts that providing consistency through an adaptive visualisation to help the user to maintain a consistent mental model. Mapping structure is a strong limitation of a dynamic visualization, even if it can handle dynamic datastreams, is mapping structure. This limitation can be a strong motivation for providing the means for people to reorganize and update their own visual mapping over time. The authoring of a visual mapping structure by InfoVis non-experts is a complex challenge [Grammel, 2012], and can be even more so when the visualisation has to be dynamically updated. The last subsection of this chapter focuses on this particular problem of authoring the visual mapping of dynamic visualisation for non-experts.

In this background section I review literature related to each of the following three points: *i)* dynamic data visualizations design, *ii)* crowdsourcing dynamic data analysis, and *iii)* authoring tools to build dynamic information visualizations. All of these are considered in respect to InfoVis non-experts.

2.2 *Dynamic visualisation of data streams*

Cottam et. al. specify a taxonomy on that complements Bertin's visual variables [Bertin, 1977] and propose a revised version of the information visualization reference model to consider dynamics at each stage

(Figure 2.2). They created this taxonomy in order understand how data changes influence the interpretability of visual representations. It classifies dynamic visualizations in a two dimensional matrix defined by spatial and retinal categories.

Cottam et al. [2012] point out that changes can arise at any stage of the InfoVis pipeline and can be provoked either by a data update from streaming data, for instance, or triggered by a user actions such as a dynamic query or remapping, or just animation. This allows us to identify different types of dynamic phenomena in a data visualisation (Figure 2.2): (i) dynamicity of the data (streaming data), (ii) dynamicity due to the interaction, (iii) dynamicity due to a remapping, (iv) dynamicity due to a view change, an animation or navigation. Next I will focus this survey on two aspects of dynamicity: how data streams are discussed in the InfoVis field and how existing dynamic visualizations are designed (Figure 2.2 B).

2.2.1 Data streams

The topic of data streams is popular in computer science disciplines such as database systems [Babcock et al., 2002], machine learning [Gaber et al., 2005], and information visualization [Norton et al., 2001, Cottam et al., 2012, Wong et al., 2003].

Abundant work exists on visualizing time-oriented data [Aigner et al., 2007]. In this section I focus on dynamic visualisation updated over time due to incoming data, and specifically data streams, which are unbounded sequences of data elements that are continuously transmitted.

Due to its novelty in information visualization, the vocabulary related to data stream visualisation is not standardized. Norton et al. [2001] used the term “streaming graphics”. Fry [2000] proposes “organic information design” for a certain type of dynamic data visualization. Finally, the term “dynamic visualization” seems to have become accepted, as it is used by several authors [Wong et al., 2003, Cottam et al., 2012] who present dynamic visualizations as nomenclature for visualisation for dynamic data, including data streams.

Norton et al. [2001] introduce “streaming graphics”, a new visualisation approach of visualisation for displaying streaming data in real time. The authors explain “the difference between streaming data sources and static data warehouses is a stream versus a pool.”. They provide several preliminary guidelines designing streaming visualizations dependent upon on whether the time-scale is continuous or discret. They also describe update strategies, and real-time versus replay scenarios. However, they discuss graphical constraints and solutions instead of metaphors or design rules.

Fry [2000] the values of the data as well as its structure can both change over time. In order to help the user maintain a consistent mental model, Fry asserts that consistency should be provided through an adaptive visualization. Fry’s major contribution is Processing [Reas

and Fry, 2007], the widely used visualisation programming environment. This authoring tool is a major step in democratizing visualisation programming, but its audience is nonetheless limited to people who can program.

Forbes et al. [2010] introduce “behaviorism”, a framework specifically for facilitating the creation of novel dynamic information visualizations. This framework is a unified system that permits the addition of behaviours to data, visual encoding, and, behaviours themselves. These behaviours can be triggered by different parameters and leads to changes in the visual encoding, the interaction, or previous behaviours. Their approach provides a coding tool for designing dynamic information visualizations.

Hetzler et al. [2005] describe user interactions and technical implementation specifically applied to dynamic data visualization. They emphasise the importance of retaining the context over time, the ability to distinguish between what is new and what is old, and the ability to identify the “now” timeframe, and understanding transitional change over time.

Dynamic information visualization

A large number of techniques for visualizing streaming data have already been published. For instance, Vande Moere [2004] uses a metaphor of a dynamic system to represent dynamic data in a dynamic visualization (Section 2.2.1). The metaphor is based on *flocking birds* to capture emerging patterns in time-varying datasets. The items representing data managed by Vande Moere are boids and curves. This technique relies on a complex behavioural algorithm to identify data clusters, different approach are more naive and uses very simple properties of the items coupled with a physics simulation like infoticles Vande Moere [2002], Whisper [Cao et al., 2012].

Vande Moere [2002] use a particle metaphor to model information representation into a tree dimensional scene rendered in a immersive environment. In this metaphor each data case is a particle, coined “infoticles”. Users could parameterise forces and surfaces to influence and filter the flow of infoticles (Figure 2.4). Infoticles allow real time data updates as well as replay of previous datastreams.

Whisper (Section 2.2.1) is a system visualizing the spatio-temporal traces of tweet diffusion according to topics in real-time. Whisper provides “Topic discs”, which are individual discreet elements that move over time. It also uses a force model to animate the visualization. Whisper is specialized to trace the diffusion of topics, which is a specific task, unlike our approach, which is generic and does not assume spatial data.

Krstajić et al. [2010] (Section 2.2.1) present a technique to visualize and monitor news streams on-line. This technique enables the representation of each single data item as a single visual mark with certain properties, subsequently to aggregates this item in another visual representation of a set of previous data items. However, the

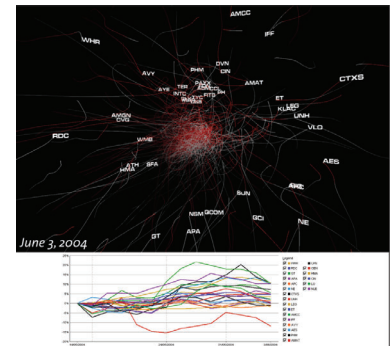


Figure 2.3: Screenshot of Information Flocking Boids, Vande Moere [2004].

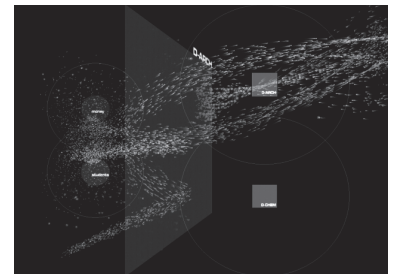


Figure 2.4: “Dust and magnets”, Yi et al. [2005], <http://youtu.be/wLXwL38xek0>

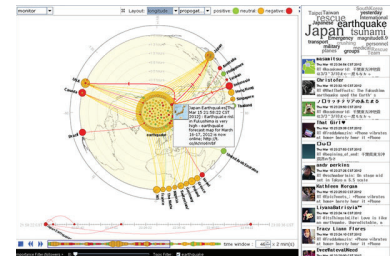


Figure 2.5: “Whisper”, screenshot on details, Cao et al. [2012].



Figure 2.6: Krstajić et al. [2010] visualization details screenshot.

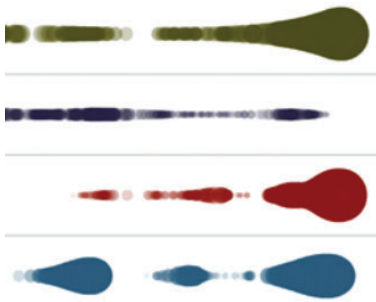


Figure 2.7: “Cloudlines”, screenshot details, Krstajić et al. [2011].

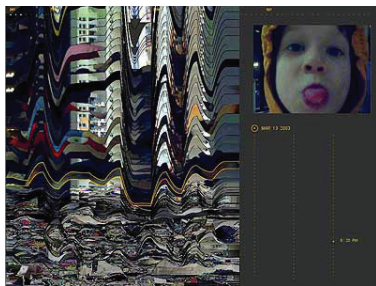


Figure 2.8: “Artifacts of the Presence Era”, screenshot, Viegas [Viégas et al., 2004].

transition between the independent visual item and the aggregated version is not smooth.

Cloudlines (Section 2.2.1) is a technique developed by Krstajić et al. [2011] incremental time-series visualizations. They introduce two major concepts that highly inspired our work: (i) the need to emphasize incoming elements and provide context based on their past, and (ii) the use of a decay function to express distortion over time.

In another context, Viégas et al. [2004] designed and deployed a visualisation (Section 2.2.1) that keeps track of peoples presence in an exhibition space. Their visual representation is inspired by a geological metaphor. Trails of the video of the instant aggregate over time in strata that represent the past. These work, strata decay and compress over time.

2.2.2 Summary

In this section I introduced several systems and concepts, which compose and define the area of dynamic visualisation and streaming data. As observed by Krstajić et al. [2010], the domains of dynamic visualisation and streaming data are still in their infancy. We saw first that the vocabulary surrounding dynamic visualisation is not yet standardized, and second that there are currently no generic techniques despite standard time-series. However, time-series does not provide a view of the new incoming element alongside an overview. Despite this, several elements of the literature reviewed inspired our research. Multiple works [Cao et al., 2012, Krstajić et al., 2011, 2010] present the idea of emphasizing incoming **independent discreet elements** and making them **decay** over time. Others present the idea of using **metaphors** [Cao et al., 2012, Viégas et al., 2004, Vande Moele, 2004] in order to provide a visual system that can cope with a dynamic system that the user has already experienced. Most of these elements structure the design of our Visual Sedimentation metaphor (Chapter 6).

2.3 Dynamic data analysis

Many scenarios, ranging from critical to casual require the analysis of dynamic data, Rohrdantz et al. Keim et al. [2013] provide different examples such as crisis management, news and stock market, server administration and log management [Girardin and Brodbeck, 1998]. All these applications concern expert users trained to monitor highly specific data in critical contexts. However, dynamic data analysis is also applied in more casual contexts with a general public audience. For example thread of information that accompanies a live event. These thread are know as backchannel. In the first part of this thesis, I narrow the scope of our investigation. My focus is crowdsourcing of media annotation on a backchannel. This type of application suits our needs to gather information analysis from non-expert InfoVis

audience during a dynamic process, in this case a live event. In this section I will first present a literature review of the domains of crowdsourcing for media annotations and backchannels for live events.

2.3.1 Crowdsourcing media annotation

According to MIT News [Hardesty, 2012] “Crowdsourcing is a technique for processing a task over the internet by splitting it into small chunks that dozens, hundreds or even thousands of people complete.”. This technique is particularly useful for analysis tasks that are trivial for humans but difficult, for computers [Quinn and Bederson, 2011]. Crowdsourcing systems have demonstrated their effectiveness with several applications such as photo selection [Bernstein et al., 2011], data analysis [Willett et al., 2012], and question answering services [Bigham et al., 2010].

Crowdsourcing activities could be processed on social networks such as Twitter with a non-expert population. It has already been shown to be effective on low latency analysis tasks. For example, Bernstein et al. [2011] developed a technique to recruit synchronous crowd and process dynamic analysis tasks in a short period of time.

Diakopoulos and Shamma [2010] showed with the application Voxcitivas (Figure 2.9) that the social structure and conversational content of tweets can provide insight into a media event’s structure and semantic content: quantitatively through activity peaks and qualitatively through keyword mining. The same authors [Diakopoulos et al., 2010], after having collected tweets during the U.S. State of the Union presidential address in 2010, they used them to annotate a video of the event. Yet, despite the benefits of crowdsourcing for annotation, there are disadvantages to consider. Marcus et al. [2011] referred to the nature of crowdsourcing as time-consuming due to the recruitment of the crowd and the time to achieve a task. To avoid this problem, the authors designed a recruitment strategy and a retainer model (paying workers to wait and respond quickly when asked). Another identified problem is the motivation of workers. Wu et al. [2011] encountered this problem in their video summarization technique via paid crowd workers and expressed the need to further investigate incentive mechanisms.

A large number of pitfalls have also been identified in crowdsourcing task quality [Snow et al., 2008, Ipeirotis et al., 2010]. Workers are often untrained and lack task context, particularly in specific domains such as science or the humanities.

In summary crowdsourcing suffers from the following pitfalls: *cost*, *speed*, (it is much slower than automated algorithms), and the fact that it is not immune from *ambiguity*.

2.3.2 Backchannel for live events

A digital backchannel can be understood as a thread of information that accompanies a live event. For more than ten years now, digital

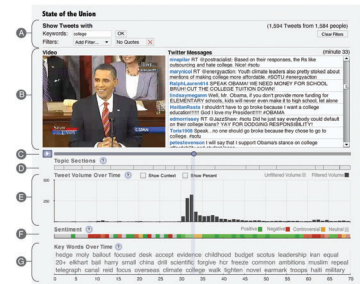


Figure 2.9: Screenshot of the application vox citivas [Diakopoulos and Shamma, 2010].

backchannels, have become a growing research area in human computer interaction [McCarthy et al., 2005, Rekimoto et al., 1998, Harry et al., 2009], computer supported collaborative work [Hannon et al., 2011, McNely, 2009, Cogdill et al., 2001] and visual analytics [Dork et al., 2010, Marcus et al., 2011].

Backchannels have been studied and used in several contexts, such as conferences [Harry et al., 2009, Hannon et al., 2011], classrooms [Bry et al., 2011], and meetings [Stefik et al., 1987]. A taxonomy of different backchannels was presented by Cogdill et al. [Cogdill et al., 2001]. The researchers present five types of backchannels: process-oriented, content-oriented, participation-enabling, tangential and independent backchannels.

A digital backchannel is useful for providing awareness [Bry et al., 2011] to local and remote participants [Rekimoto et al., 1998] of questions, comments [Hannon et al., 2011], shared work, and references, and can encourage real-world discussions [McCarthy et al., 2005]. Hannon et al. [2011] studied backchannels in academic conferences using IRC and considered different types of problems arising with different types of use. For example, they noted the cost of spreading information for a backchannel connection. McNely [2009] suggests that the problem has been partially solved by the increased availability of micro-post web services like Twitter due to their simplicity of use and the large adoption rate.

Sopan [2012] showed that a Twitter backchannel (Figure 2.10) during a conference, permits connections to be made to both local and remote participants. In another academic context, Harry et al. [2009] implemented and described the use of a backchannel to leverage participation in an auditorium to allow the audience to vote on questions and give feedback.

With the prevalence of social networks, portable devices, and wireless network connections, backchannels are no longer an emerging social phenomenon but a real trend. When this increased use of social networks and portable devices meets large scale events such as the Olympics, the Arab spring, or national elections, visual analytic techniques can help to summarize and understand what has happened during the event. Dork et al. [2010], for example, introduced the visual backchannel (Figure 2.11), a timeline-based visualisation that presents an overview of an event through social stream aggregation.

Marcus et al. [2011] presented an algorithm for event detection in a stream to produce annotations on a timeline-based visualisation of the social activity surrounding an event. Diakopoulos et al. [2009] synchronized a recorded TV show with related tweets (Figure 2.12) and provided two timelines, one for sentiment analysis over time, the other for volume of tweets.



Figure 2.10: Screenshot of conference monitor. Sopan [2012].

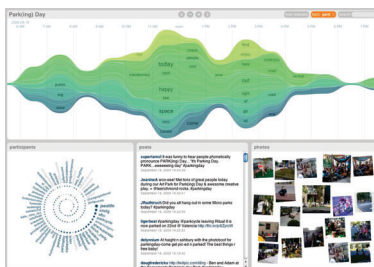


Figure 2.11: Screenshot. Dork et al. [2010].

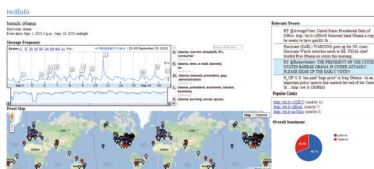


Figure 2.12: TwitInfo screenshot Marcus et al. [2011].

2.3.3 Summary

In this section I presented some related work on using backchannels for live event annotation, as well as work on crowdsourcing ap-

proaches for event analysis. These have demonstrated that backchannels are: (i) used by a varied audience of non-specialist, non-expert users, (ii) used in a wide variety of events (lectures, academic conferences, TV show and others), (iii) be a source of dynamic data for visualization, and (iv) able to support the processing of dynamic tasks such as annotating or voting. These properties enable me setting up experiments with visualisation to support dynamic data analysis for a InfoVis non-expert audience. In the first part of my work (Chapter 4 and Chapter 5), I will use similar properties but in a slightly different organization, I will create a backchannel visualization to process a specific analytical task by a general audience during different events. According to this related work, this approach is new and has not been studied before. This also suits our needs to investigate the possibility for a non-expert to analyze dynamic data.

2.4 *Authoring Dynamic Visualization*

In the pipeline provide by Cottam et al. [2012], visual remapping is identified as one of the dynamic parts of visualization. Visual mapping and remapping (Figure 2.2 C) is one of the most time-consuming process in using visualisation authoring tools. In this section, I focus our survey on this problem in order to motivate the investigation process in the second part of this dissertation.

The authoring of visualisation for InfoVis non-experts has been extensively studied by Grammel [2012]. In his Ph.D. Grammel defines InfoVis novices as: “users who create visualizations to support their primary tasks, but who are typically not trained in data analysis, information visualisation and statistics.”. I will use this definition when I refer of “non-expert users”. Together with Tory and Storey, Grammel et al. [2010] conducted a study to understand how InfoVis novices design visualizations. In this study, they found that every participant passes through different visualisation construction cycles. Each of these cycles is determined by a change of the visual mapping. These cycles may be seen as equivalent to the dynamic processes referred to as “remapping” by Cottam et al. [2012].

However during this study, Grammel et al. identified three major barriers that are related to the visual mapping process:

1. selecting which **dimensions** to map to visual variables,
2. selecting which **visual marks to use**,
3. and decoding and **interpreting** the visual result

While this study [Grammel et al., 2010] provides valuable insights, it is based on the participants choosing among pre-defined templates and thus leaves many questions unanswered about the dynamic aspect of the mapping and remapping process.

Currently existing tools to author visualisation can be classified according to their design approaches. Victor [2013] distinguishes

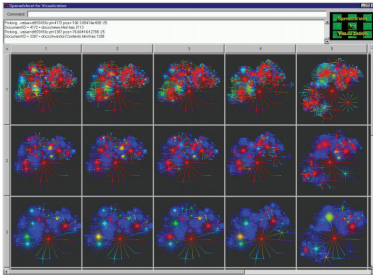


Figure 2.13: Visual spread sheets extract from Chi and Card [1999]

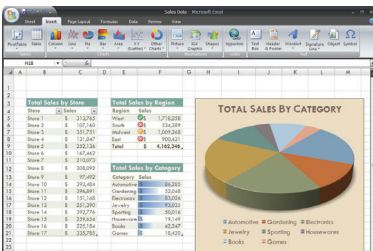


Figure 2.14: Screenshot of Microsoft Excel interface.

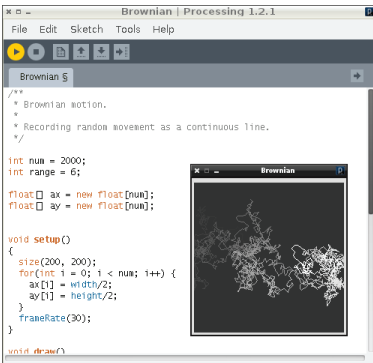


Figure 2.15: Screenshot of Processing interface [Reas and Fry, 2007].

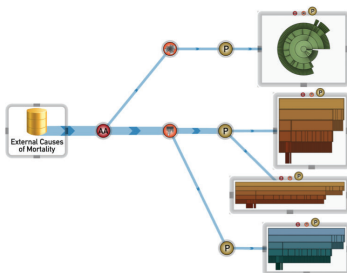
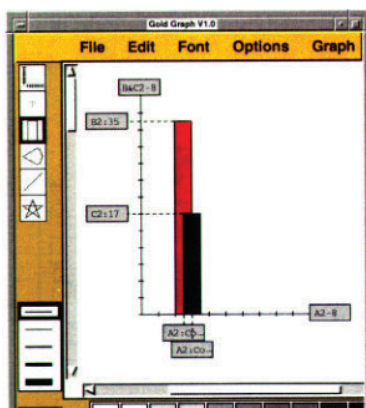


Figure 2.16: Lark interfaces, Tobiasz et al. [2009].



between three fundamentally different approaches: using, drawing and coding.

A recent, detailed survey of InfoVis authoring tool user interfaces inspects articles published in twelve major human computer interaction venues over the past twelve years [Grammel et al., 2013]. From this process the authors collected 282 research papers. They classified each of these papers into six different approaches (see the article reporting the study for more detail): visualisation spread sheets, template editors, shelf configuration, textual programming, visual dataflow, and visual builders.

Visualization spreadsheets, are small multiples interfaces showing different versions of the same visualisation and which allow the user to apply and compare different visual mappings. This interface facilitates the exploration of different visual mappings (Figure 2.13).

Template editors, are probably the most common visualisation authoring tool. These allow the user to select a standard visualisation and apply it to existing structured data. It is easy and quick to use, but does not provide flexibility with respect to the final rendering or visual mapping. Microsoft Excel fall into this category (Figure 2.14).

Shelf configuration, permits the user to map data attributes to visual properties. The visual properties are organized in a virtual shelf. This shelf allows the user to attach attributes to visual properties such as marks, colors, size, shape, etc.

Textual programming, similar to precedent coding category, textual programming gives full freedom in visual expression for an expert user. This enables them to define the visual mapping freely as well as the possible dynamic updates and interactions. However, these are expert tasks and require some time to yield a final result (Figure 2.15).

Visual dataflow user interfaces are generally node-link diagrams (Figure 2.16). These diagrams represent the pipeline of transformation from the raw data to the representation. Each node can be edited and manipulated in order to transform or filter the data, choose a visual mapping, change the presentation, etc.

Visual builders, user interface concept is similar to graphical editors such as Adobe Illustrator or Photoshop. They contain an assembly area and some toolbars with prototypes of graphical elements that one can select to use in one’s own visualisation (Figure 2.17). The way the visualisation is constructed is by selecting the graphical elements, which are visual variables, putting them into the scene, and assigning them data properties. These types of interfaces have the benefit of providing great flexibility in the visual design.

Apart the Visual dataflow, most of the tool categories are highly focused on the visual mapping specification process rather than the remapping process. Remapping can be due either to the dynamic aspects of the data or to changes in visual mapping for exploration.

2.4.1 Summary of authoring tools

Information visualisation authoring has been a major area of work for decades. A lot of different approaches have been proposed but most of them are focused on either providing a fully expressive tool for an expert user or on providing automated and less expressive tools for non-expert user. I also found that all of these tools permit updates to the dataset while keeping the visual mapping the same but are not explicitly focused on the remapping aspect of dynamic visualizations. In the second part of this dissertation I will discuss this point in more detail (Chapter 7) as well as defining and studying (Chapter 8) a new approach to dynamic visualisation authoring tools.

2.5 Chapter Summary

In this related work we present works from three different aspects of dynamic visualizations for InfoVis non-experts. Each of these aspects is positioned at different stages of the visualisation pipeline (figure section 2.2). The fields associated with these different aspects are also relatively new in the research area. The first aspect is the dynamic data analysis of data streams by non-expert users. The second is the design of dynamic visualizations to represent data streams. The third is authoring tools for visualizations.

From the related work on dynamic data analysis, we find that most existing approaches are only focused on expert users [Grammel et al., 2010]. However, crowdsourcing media annotation and backchannels both require that non-expert users engage in dynamic practices. From this previous work we note that crowdsourcing has the following properties: (i) it can be processed on social networks and used by the general public, (ii) certain dynamic analysis tasks could be done in a short period of time, (iii) existing practices on social networks already provide some useful information for visualization. Based on these observations we argue that this application domain is a rich field for experimentation in order to determine whether non-experts can conduct dynamic data analysis. In its current state, crowdwourcing suffers from pitfalls such as the cost, the speed and the possible ambiguity. In the two first chapters (Chapter 4, Chapter 5) of the second part of this dissertation I will investigate how we can reduce these problems by using dynamic visualization.

From our survey of dynamic visualizations of datastream we note that: (i) the vocabulary related to dynamic visualization is not yet standardized, (ii) no generic techniques exist to visualize dynamic data, (iii) despite some common design principles emerging between several techniques, no guidelines or general approaches exist. In this literature selection we identified common principles in several techniques was the emphasis on incoming **independent discreet elements** and their **decay** over time. Also, others, such as Viega and Vande Moere, introduce the idea of using **metaphor** to provide a visualization system to cope with dynamic systems. Based on these

observations, in the section that follows (Chapter 3), we will investigate the history of independent discrete elements used for dynamic visual representation in others domains.

Our review of related work on authoring dynamic visualizations revealed that current tools are either *(i)* complex to use and understand, *(ii)* limited in their degree of expression and personalization, or *(iii)* do not easily support dynamic functions such as custom remapping and update. We also found that the most powerful tools (coding) required a high level of expertise. This observation serves as the main motivation for the second part of this dissertation (Chapter 7) to define the design challenges for dynamic InfoVis authoring tools for non-expert users.

From these three surveys we conclude that: *(i)* backchannel and crowdsourcing media annotation is a good application domain in which to experiment whether the general public can conduct dynamic analysis, *(ii)* dynamic visualization is a young area with no generic vocabulary and no generic techniques, but some design approaches exist based on **independent discrete elements**, *(iii)* dynamic information visualization authoring tools are subject to different constraints and do not provide a simple solution for non-expert users wishing to author their own dynamic visualizations. Based on these three observations, we will study and design three aspects of dynamic visual representation based on **independent discrete manipulative elements**: *(i)* the possibility of using dynamic visualizations to engage the general public in an analysis process, *(ii)* the creation of generic techniques for dynamic visualization, *(iii)* the creation of dynamic visualisation authoring tools for the public.

3

A Short History of Token as used in Visual Representation

Token: "A thing serving as a visible or tangible representation of a fact, quality, feeling, etc." ¹.

¹ Oxford dictionary definition, online edition, s.v. "token" : <http://goo.gl/u0IadI>

Information Visualization —————

Monosemic Visual Representation —————

Some Instruments of Measure —————

Some instruments of Calculation —————

Visual Representation —————

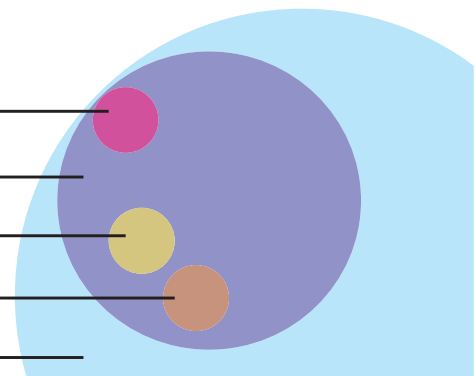


Figure 3.1: Information visualisation and other domains of monosemic [Bertin, 1977] visual representation systems. I derived inspiration from some of these domains (for example, instruments of measures, instruments of calculation) by studying their visual representation mappings. This is a containment diagram.

In the previous chapter I surveyed the dynamic information visualisation field in the domain of Computer Science. From this survey I found dynamic information visualisation to be a relatively new research area with an emerging vocabulary and no generic techniques. However, I found that different techniques share common design principles such as the use of **discrete elements** [Cao et al., 2012, Krstajić et al., 2010] that **decay** [Krstajić et al., 2011, Viégas et al., 2004], and the use of **metaphors** [Viégas et al., 2004, Vande Moere, 2004]. Motivated by this, I wanted to further investigate the use of **discrete elements** like token in history of monosemic visual representation.

From this observation I concluded that **discrete visual elements** standing for data -tokens- could provide benefit to dynamic visualizations because they allows me to identify each data items at an initial stage and keep their identity over time. This belief is grounded in the expressive power of tokens, that can be understood through their history, which is further illustrated in this chapter.

This collection uncovered a substantial amount of past related work from both near and distant history. The work was also drawn from a wide variety of domains (Figure 3.1), including pre-computer instruments, table games, computer human interaction, graphical statistics and information visualization. Most of this work shares the principle of providing a visual dynamic representation based on the assembly of *individual discrete elements*. It also included principles that are highly instructive for our research and design of dynamic information visualizations.

I have organized this history chapter as follows: (i) I will describe some of the artefacts and their use as well as the literature before the computer era, (ii) In the second section I describe the replication of discrete visual elements in different technical manifestations as well as other classic visual representations during the printing press era. (iii) Then I describe how similar approaches have been exploited in other computer domains such as human computer interaction, pedagogical coding environments, and tangible interactions. (vi) And last I will present some contemporary art and design pieces sharing the same principles.

3.1 Introduction

One of the major problems of dynamic visualizations is to process a dynamic change, which is easily predictable by the user [Cottam et al., 2012]. Designing a system that can operate a predictable change is not trivial. And it is even less trivial when the user is not specially trained to use the system. Due to the limited number of established guidelines in the domain, our research and design draws inspiration from dynamic visual representations throughout time, from antique systems to the computer era.

Visual representations of abstract information can be defined either by a visual mapping correspondence with the data [Bertin, 1977], by a binding of the data to a tangible object [Vande Moere, 2008, Jansen et al., 2013], or by providing a metaphor [Gentner, 1983, Blackwell, 2006, Lakoff and Johnson, 2008] to bind some properties of a concrete system to a representational system.

None of these ways of mapping or binding information provide a specific solution for visual dynamics representation. And even if most of our current information visualizations in the computer domain have dynamic properties due to interaction techniques [Heer and Shneiderman, 2012] such as, brushing and linking, dynamic queries, etc. these functions are limited to a fixed dataset and do not support dynamic data updates over time or dynamic remapping with the same representational system.

One of the possible reasons for these limitations, is that most of our current information visualizations are based on classical designs. For example, area charts are most likely visually inspired by 18th century graphical production in a domain called by [Beniger and Robyn, 1978] “statistical graphic”. I hypothesize that during this era, the dataset was not updated as regularly as it is today. Moreover, the technical manifestations of visual representations were mostly based on ink and paper technologies, which are not amenable to dynamic changes.

However, since the year 1000 but before printing press, humans used other technologies to produce visual representations in many different ways [Schmandt-Besserat, 1996, Crump, 1992, Hsieh, 1981, Catepillán and Szymanski, 2012, Robson, 2003]. Independently from the technology, many of them are based on the manipulation of discrete objects spatially organized to express information.

In this chapter I will present some snippets of this history as well as some details of the technologies and representation mappings. I will start the chapter by introducing the use of discrete elements in the past for counting and then discuss different token-based systems. All these systems preceded the computer era. I will then briefly present some of the classical area charts created during the printing press era, and following later developments, the use of discrete elements in human computer interfaces, tangible interfaces and contemporary art and design information representation.

3.2 Pre computer era: A short history on the use of discrete elements for dynamic representation

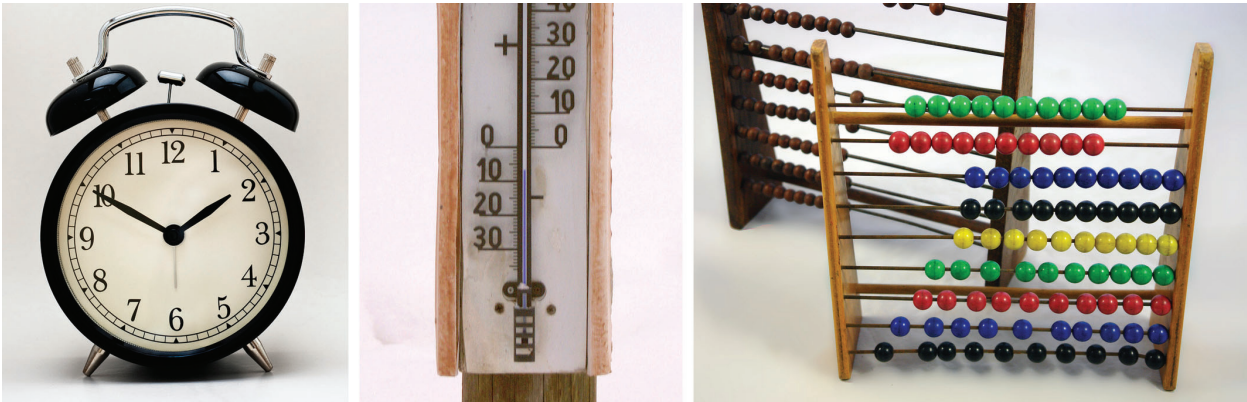


Figure 3.2: A clock, a thermometer, and an abacus. The first two devices are instruments for measuring, the former for measuring time, and the second for measuring temperature. The third instrument is used for calculation and is also manipulable. All three of them provide a visual representation of information.

Before the computing era, humans created many instruments to visually represent dynamic data and also to count, measure, and process data. Examples of these can be found in the artefacts we find in our daily life, for example clocks, thermometers, abacuses, monetary currency, among others (Figure 3.2). All of these instruments have specific types of data input functions, data storage functions, and provide a visual representation of the processed data.

In this section I will focus on a particular subset of these instruments, a subset regrouped by the common use of a *discrete - sometimes manipulable - elements which stand for an information unit*. In the majority of cases, these discrete elements are organized spatially and assembled with dynamic operations on the information units. These information units might be of any type: quantitative, qualitative or both.

In this category of instruments we can list different types of systems: tally marks, clays tokens (Figure 3.7), coins for money, beads and abacus, pawns in board games, pictograms, digital icons on virtual desktops, building blocks in educational games. These artefacts have different functions, for example simplifying counting, calculating, paying etc... Some of them are the result of a long history of design refinement and selection throughout the ages. Most of them are also dependent on specific cultural and technical rules. For instance, abacus processing is different according to culture, coins are part of the wider system of money regulated by law, and virtual icons are elements of a desktop metaphor environment. Despite these differences, these systems all provide a simple and pervasive means for manipulating abstract information dynamically. In this section I will present a short and non-exhaustive history of these instruments.

3.2.1 The evolution of tally marks (20,000 B.C.)



Figure 3.3: A type of tally mark used in Europe, Africa, Australia, and North America.

Perhaps one of the oldest and still in use forms of dynamic visual representation systems are tally mark systems. Tally marks (Figure 3.3) are discrete inscriptions of objects representing single units. In most cases, these are clustered spatially in groups of five for legibility and counting. These systems are useful for counting on-going events such as cyclic natural phenomena or scores in a game.



Figure 3.4: Ishango bone (20,000 B.C.)

The oldest traces of such a system has been identified by Marshack [1972] on Ishago bones (18,000 to 20,000 BC.) and Lebombo bones (35,000 B.C.), which are sets of bones dating back to the middle Palaeolithic to the Mesolithic era. Alexander Marshack interpreted the notches of the Ishago bones (Figure 3.4) as the first tally mark counting system hypothesizing that they could represent a lunar calendar. Even though the hypothesis that the notches are tally marks is accepted, there is no unanimously agreed upon interpretation of what these marks represent. For instance, Thompson [1996] hypothesised that these notches marked the menstruation cycles of the author.



Figure 3.5: Chinese tally mark system.

The spatial encoding of tally marks evolved in different ways over time with different local cultures. Some used symbols to represent larger units, for example, the system used in China, Japan, and Korea [Hsieh, 1981] encodes the accumulation of five units in the character meaning “stop” (Figure 3.5). More modern tally systems have a more complex spatial organisation for compactness while providing greater cardinality. For example, two types of signs could be used: dots and lines (Figure 3.6).

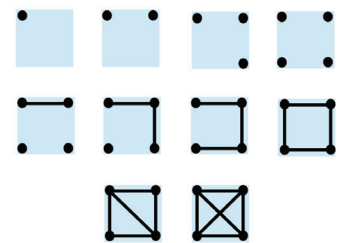


Figure 3.6: Dot and line tally marks encode the accumulation of ten units.

The initial motivation for tally mark systems came from the need to provide the following:

1. **A incremental writing system:** The intermediate inscriptions do not need to be erased or discarded to execute additive operations (e.g. adding an extra unit).
2. **A countable representation:** Even if a tally cluster indicates a fixed sum and offers a way of efficiently getting an idea of the count or quantity, it is still possible to see the smaller units to get more precision.
3. **One to one correspondence between the tally marks and the objects being counted:** for most of them the representation of the natural number N is represented by N repetitions of the chosen symbol.

Tally marks constitute a visual counting system composed of the incremental addition of discrete elements. Dynamic data are reflected

in changes in the visual representation, allowing it to serve both as both a means of recording data over time (input) and a means of representing data with visual cues over time (output). This dual input-output capability of tally marks informs the designs of our systems presented in Chapters 4 and 5; these systems also provide a way to **record** and **represent** dynamic data over time.

3.2.2 *Evolution of Tokens Systems (8000 B.C.)*

Like tally marks, token-based systems have been in use for a long time, from 8000 B.C. to the current day. The appearance of the first tokens coincided with the invention of agriculture; according to Schmandt-Besserat [1996].

Unlike tally marks, which are not semantically related to what they represent, the first tokens were semantically defined: their shapes were determined by the kinds of goods they represented. Denise Schmandt-Besserat conjectures that they were used for trading. Today the majority of our use of tokens is in local trading, taking place independently from the mainstream monetary system, for example, in casinos, subways or automatic delivery machines. The primitive system of tokens is likely to have inspired the creation of coins. Coins differ from tokens in that they are part of the monetary system and semantically free, standing simply for units.

It is likely that token-inspired systems have had a long life in human history because they are very simple to understand, allow us to easily manipulate small quantities of data, and permit flexible computations.

In this subsection I present a selection of different artefacts that used tokens for different purposes before the computer era.

Clay tokens (8000 BC)

The first trace of a token-based system dates back to between 8000 and 3000 B.C. and was identified as such by Schmandt-Besserat [1996]. In her book “How writing came about”, she studies a collection of 8,162 tokens discovered in the Middle East (Iraq, Iran, Turkey).



Figure 3.7: Clay tokens, extract from [Schmandt-Besserat, 1996].

Clay tokens were an archaic counting system based on small objects of many shapes representing units of different types of goods. A semantic value was assigned to the shape and etching of the token. For example, an ovoid shape represented a jar of oil, and a round shape represented a sheep. These tokens were used to count, process, and record the exchange of goods. As with tally marks, the tokens had a one to one correspondence with the goods they represented, one jar of oil was represented by one ovoid, two jars of oil by two ovoids, and so on. There is some evidence that these tokens had a socially defined “syntax”, Schmandt-Besserat [1996] reports that “tokens were arranged in lines of [tokens] of the same kind, with the largest units placed at the right”. It is possible that this spatial ordering already permitted visual comparison of two lines.

Tokens pre-dated the abstract numerical system; the token was a tool for concretizing the counting process. Other concrete counting routines still exist in our society today, for example, the use of our fingers. These methods have the benefit of functioning without the need to invoke our memory.

Clay tokens had many benefits compared to their previous technologies. They provided more flexibility than tally marks in information manipulation such as addition, subtraction, undo operations and corrections because they were tangible independent objects.

Storing Clay tokens: Envelops and Tablets (4000 BC)



Figure 3.8: A. Tablet with engraved tokens. B. Tablet with engraved tokens and additional information such as strokes to delimit layout and pictographs. C. Tablet with engraved table layout sorted by value order, similar to several bar charts. Extract from Robson [Robson, 2003]

The spatial flexibility of the clay token system makes it well suited to basic computations. On the other hand, this spatial flexibility also means that the positions of tokens are not fixed, thus making the system unsuitable as a means of storage.

Schmandt-Besserat [1996] points to the invention of token envelopes (Figure 3.9) as the first solution to address the increasing need for token storage in archives. A token envelope was basically a clay ball in which the tokens were placed and confined. Before placing the tokens in the envelope, the shapes of the tokens were impressed into the surface of the envelope. The number of units was reflected in the number of impressed shapes. The envelope could be unsealed to verify the count.

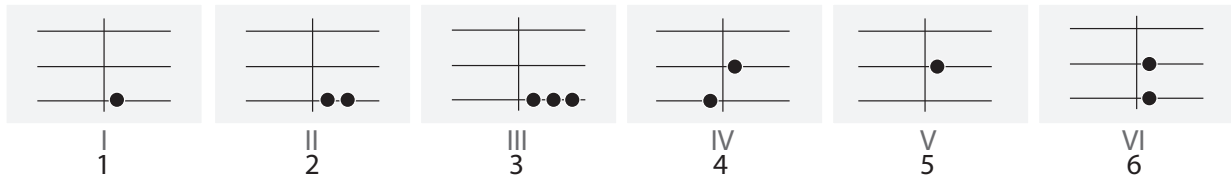
Four millennia later, a second method appeared when people realized that given the presence of the impressions, it was unnecessary to keep the tokens inside the envelope. Consequently, the clay tablet replaced the envelope, and the impressed tablet became the conventional means for recording token counts. Making markings on tablets with a pointed stylus subsequently became a system in its own right. This system had several new features, including (i) “pictographs” of the tokens, (ii) the ability to add layout elements such as strokes to create tables, and (iii) the ability to encode metadata so as to provide contextual information, for example, the names of people involved in an exchange.

As Robson [Robson, 2003] notes, some of the layouts were used to produce ordered data tables; in Section 3.3.2 I observe similar layouts in a different technical system.



Figure 3.9: Clay tokens and in the background an envelope with impressed token marks.

3.2.3 Counting board (300 B.C. or before)



After the invention of impressed tablets and abstract notation to record counting information, but before the invention of written numbers, token systems evolved to support calculation. One example of such a system is the counting board (Figure 3.10) used by Babylonian, Greek and Roman civilizations to perform calculations. The board provided information-mapping units to tokens. In this system, unlike in the Sumerian system, tokens did not need different shapes. Simple pebbles without specific semantics replaced tokens. For this reason token no longer needed to be manufactured. Their value was defined by their position on the board. One of these boards was discovered on the island of Salamis (Figure 3.11).

The Salamis board was a marble tablet containing different marks. The most visible ones in the center of the tablet were a set of parallel lines equally divided by a vertical line. An X sign is marked every three intersections between the parallel lines and the horizontal line, and a semicircle is marked at the top of the intersections. Three sets of Greek symbols representing numbers are marked on the left, right and bottom of the tablet. The horizontal lines are used to assign the units to the tokens. The vertical line is used to separate the space for negative and positive numbers. A particular value is expressed by the position of the tokens on the tablets.

By abstracting the semantics of the tokens and dynamically assigning the units to tokens according to the spatial position, this system delivered several improvements to the previous system. On the counting board a token stands for a unit and not for a particular type of good as in the previous clay token system. Also, the assignment of the value is dynamic since the unit value of a token change according to its position on the lines. The counting board therefore enabled processing of larger numbers without requiring a large number of tokens; it also enabled more complex operations such as multiplication and division.

According to [Stephen K. \[2005\]](#) the roman numeral notation system is derived from the optimisation of a notation based on the counting board as shown in Figure 3.10.

Figure 3.10: Example of token counting board counting process: black dots represent tokens, all the tokens on the right of the vertical line are positive values, those on the left of the line are negative values; the horizontal line corresponds to 1,5,10 from bottom to top. In grey the notation in Roman numerals and in black in Arabic numbers.

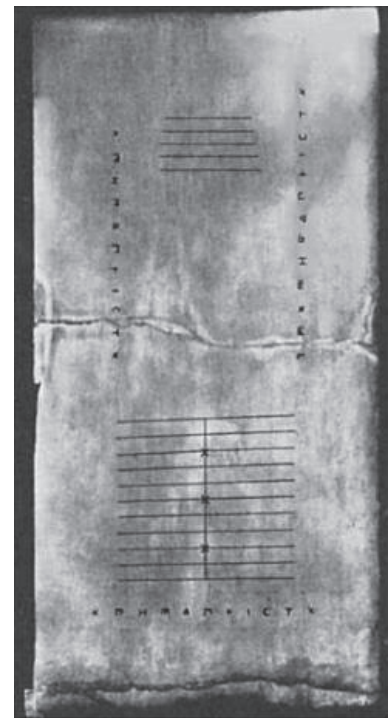


Figure 3.11: Photos of a the Salamis Tablet. cf: <http://www.computerhistory.org/revolution/calculators/1/1/128>

3.2.4 Counting frame: Abacus, Soroban, Yupana

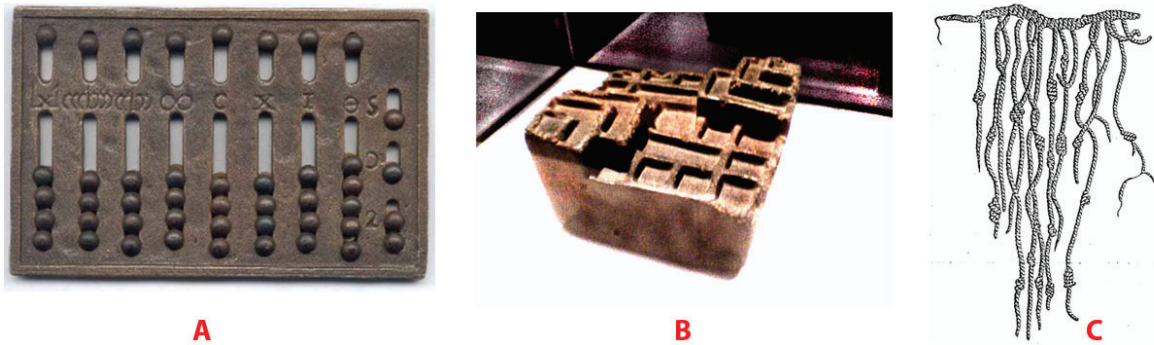


Figure 3.12: Counting frames in different civilizations, (A) Roman abacus, likely to be derived from the counting board (B)Yupana, an Incas counting instrument based on a frame to count with tokens (C) Quipu, an Incas personal cord device to count and keep track of information.

It is likely that the counting frame and abacus evolved from the counting board (Figure 3.12). A counting frame is a class of device having a frame that holds axes or a container with freely sliding tokens mounted on it. In the case of the abacus the tokens are beads, in the case of Yupana they are corn seeds, and in the case of kipu they are knots. The layout and logic of the abacus remains close to that of the counting board. The abacus mostly fixes the lines from the counting board into mechanical constraints. Abacuses were used before the written numeral system and are still in use today. Abacuses constrain token moves to a single dimension, which makes them easier to manipulate but restricts their flexibility. Similar to the counting board, the abacus supports addition, subtraction, multiplication and division. Yupana is a word meaning “count” in Quechua. This device is a two or three-dimensional frame delimiting several containers holding corn seeds, which are used as tokens. It has been conjectured that the value of a token is based on the position of corn in the containers.

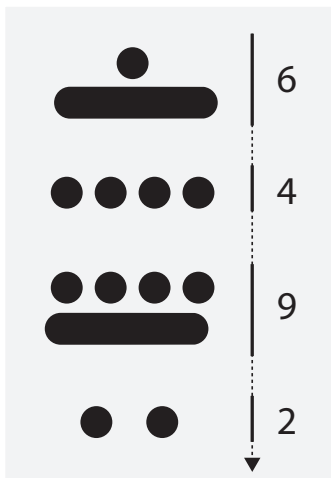


Figure 3.13: How to write 6492 in Maya number system, from top to bottom a round and a line stand for 1+5=6, below 4 round stand for 4, below 4 round and a line stand for 4+5=5, below two round stand for 2, meaning 6492.

However, the precise details of how the Incas used this device are yet not known. In a similar geographic area, the Maya used a counting system (Figure 3.14) based on two symbols which can be identified as tokens. George Ifrah [2001] describe this specific system of counting and its spatial organisation is explained inFigure 3.13.

Quipu came from the same civilisation and appears to provide a storage device. According to Garcilaso de la Vega [Joseph, 2011], the value of the knot also depends on its position on the cords. Each string has a different color, which might encode other information according to O’Connor and Robertson [2001]. It is conjectured that the just as the Yupana had subsidiary containers, the Quipu had subsidiary cords.

Figure 3.14: Maya number system: Symbol are based on additive principle. Two basic symbol one representing number one and the other number five.



3.2.5 Board games: *Senet*, *Roman Tabula*, *Mancala* (3000 BC)

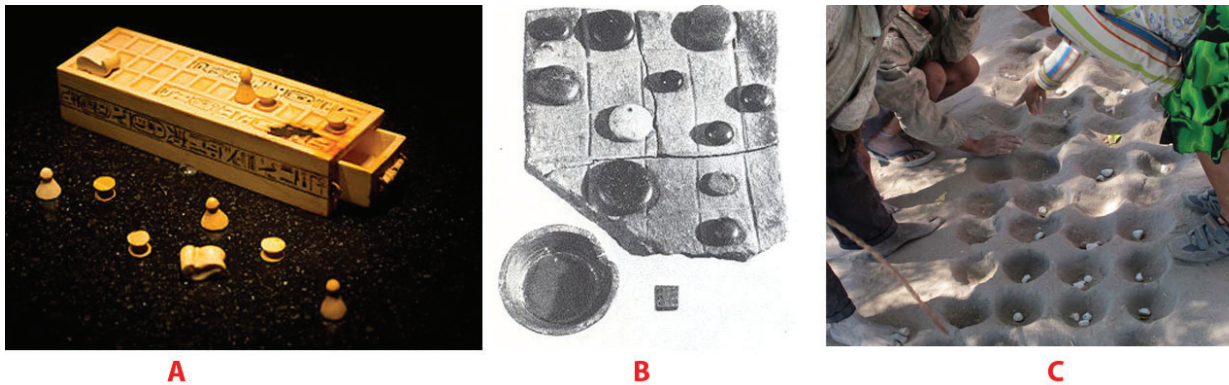


Figure 3.15: Different board game through ages, (A) *Senet* 3100 BC., (B) *Roman tabula* 480 C.E., (C) *Mancala*.

Board games originating from Ancient Egypt have been identified as close relations of the counting board. The board games consist of boards with marks or physical constraints, and a set of rules. These boards are accompanied by small pieces similar to tokens, called pawns that are used either as counters, avatars, or other types of information. In most cases, movement of the tokens express state changes over time. In these board games, play is a sort of iterative dynamic computation, with the score changing with each step due to the action of one of the players.

One of the first board games discovered was the *Senet* (Figure 3.15 A.), which was played by Egyptian nobles. Several paintings of Pharaohs playing this game, as well as the board games themselves, have been discovered in the tombs of Pharaohs. The board is a grid of 30 squares arranged in three rows of ten. The precise rules of the game are still unknown.

Roman tabula (Figure 3.15 B.) is considered to be the ancestor of backgammon; a comprehensive description of how to play is available online ². According to Kowalski [1998], *Roman tabula* was a popular game played by soldiers as well as nobles.

Mancala is a popular family of board games (Figure 3.15 C.) played in different continents, from Asia to Africa. The earliest traces of this game come from Africa 6 century AD. This family of board games consists of several lines of containers (the numbers of lines and containers can vary between different versions of the game). Two containers are larger than the other and used to hold the tokens won by the player. Tokens are pass from one container to the other, and the winner is the one who has the most tokens at the end of the game.

² <http://www.bkgm.com/variants/Tabula.html>

3.2.6 Coin and money system (600 BC till the current day)

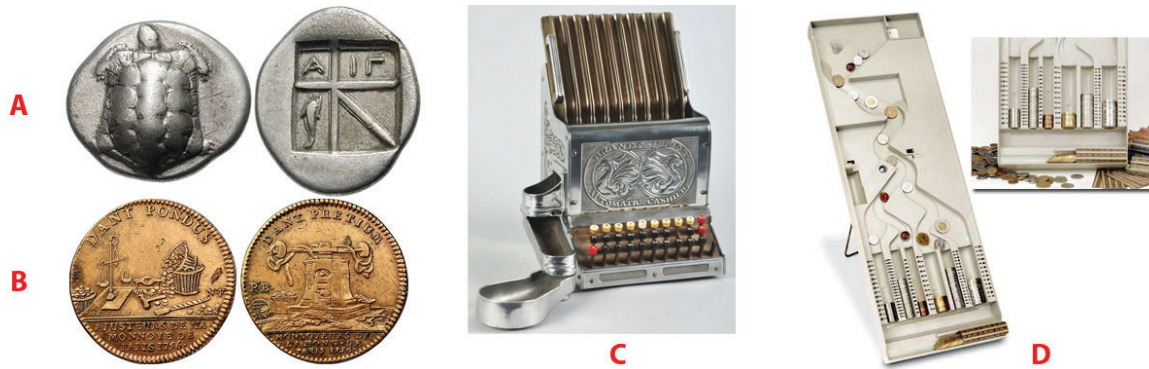


Figure 3.16: (A) Greek coin "Drachma" of Aegina 700BC (B) Token from "La monnaie de Paris" 1756 AC (C) Automatic Cashier, 19th century (D) Coin sorter 20th century.

A coin (Figure 3.16,A,B) is a round piece of hard material used to exchange a value. The earliest coins discovered are dated around the 6th century BC. Shortly after, coins from different cities of Greece appeared and spread around the Mediterranean. Since that time coins have been the most common embodiment of money. We still use coins today as a part of our monetary system.

Contrary to clay tokens, at one point in their history, coins became flat. This property permitted them to be stacked in columns, initially manually and later mechanically. These physical properties and others such as the diversity of sizes, weights, and values, led people to optimized their size and make specific arrangements and machinery to process them. During the 19th century, coin machines were created to automatically perform tasks similar to those performed by current computer processes with data, such as dispensing, counting, sorting, packaging. These systems sometimes provided a visual representation of the quantity of tokens by simple stacking. Coins can be considered to have evolved from the token system.

3.2.7 *Building block: Froebel educative materials to teach numeracy (19th century A.D.)*

Token-based systems have also been used for teaching mathematics, geometry and modelling to children. In the 19th century, schooling was only provided for children who already knew how to read and write. To democratize mathematical and geometrical education for illiterate children, Froebel [Kraus-Boelte and Kraus, 1892] designed educational materials, called "Gifts"(Figure 3.17). Some of the "Gifts" (3 to 10) are basically wooden blocks mapped to units according to their shape, as in the Sumerian clay token system. Unlike the clay tokens in the Sumerian system however, even if each shape corresponds to a specific unit, this unit is not semantically assigned to a specific type of good.

The mappings from unit values to Gifts are defined in a manual for teachers (Figure 3.18), with three types of exercises, those for learning geometry, those for learning mathematics, and those for modelling and aesthetics. The exercises are designed to teach mathematics and geometry using concrete counting and tangible manipulation to explain abstract numbers. This wooden block and unit mapping system is similar to the archaic clay token system cited before (Section 3.2.2) in several ways, but also differs in some respects to make it more suited to their application context.

Similarities:

1. has independent discrete elements
2. has values and semantics determined by shape
3. permits manipulation for counting
4. is simple to understand

Differences:

1. is related to a defined process
2. has stackable and 3d constructions



Figure 3.17: Froebel' "Gift" N°3 contains 8 wooden cubes of the same size.

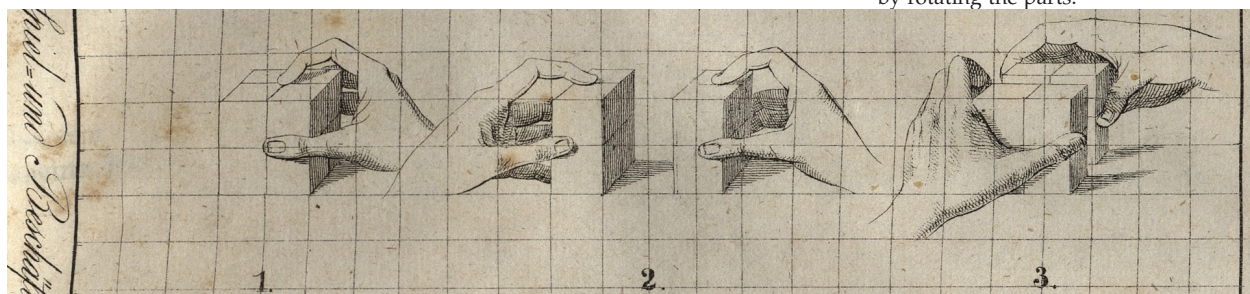


Figure 3.18: Froebel's illustration of building blocks manipulation. In this case (A) a volume of 8 building blocks organized in a square and corresponding to Gift 3, (B) divide the volume by two, by pulling apart two blocks of 4 (C) look at the result over a different angle by rotating the parts.

Summary

In this subsection I have presented a short pre-computer era history of how people manipulated discrete elements in order to process abstract information. I have presented a variety of systems (Table 3.1) that enabled humans from different cultures and civilisations to count, store data, play, educate children, and perform trade. What these different systems share is their manipulation of discrete elements to represent more abstract information.

In all of these systems, the manipulation of these elements are bound to certain abstract functions and constrained by different types of limitations. For example, a clay token cannot be stacked and a pebble does not have meaning outside of the counting board. These systems were used by a population that was highly diverse in terms of educational level and social status, from the Sumerian farmers to the Egyptian aristocracy, from the Roman soldier to the German children of the 19th century. Finally we saw that most of these systems provide a dynamic representation as well as a data input and computational support.

While this survey is limited, it provides an overview of how people manipulated representations based on discrete elements before the computer era. From this survey we get a glimpse of the diversity in systems supported by these types of objects (Table 3.1) and also the dynamic, pervasive and simple nature of these systems. Throughout this dissertation, the designs of our visualisations are inspired by the properties and uses of such discrete elements, which bind to data.

Table 3.1: Different systems composed of discrete elements used to process abstract information before the computer era

Discreet element	System	Data types
Tally	Tally marks	Numerical
Clay token	Free manipulation	Numerical + categorical
Clay token	Envelope	Numerical + categorical
Clay token	Tablete	Numerical + categorical + metadata
Pawn	Board games	Categorical
Coins	Money	Numerical
Pebble	Counting board	Numerical
Knot	Quipu	Numerical + categorical
Corn	Seeds	Numerical
Yupana	Quantitative	Numerical
Beads	Abacus	Numerical
Building block	Educational toys	Numerical + categorical + aesthetics

3.3 The Printing Press era: from statistical graphics to the roots of information visualization

As we saw in the previous section, visual representation has a more extensive history than information visualisation in the computing domain. It is very likely that humans used visual representations to count and communicate information before they even had words to describe the concept of counting.

However, during the 18th century, states and organisations began to systematically collect data consisting of demographic and economical measures, which led to the beginning of “statistics”. Beniger and Robyn [1978] report that one of the tasks was to transform the data into visual representations, a domain named “statistical graphics”. During this period many new representations were developed that still prevail in our culture. An important example is the area chart, which includes barcharts, piecharts and others.

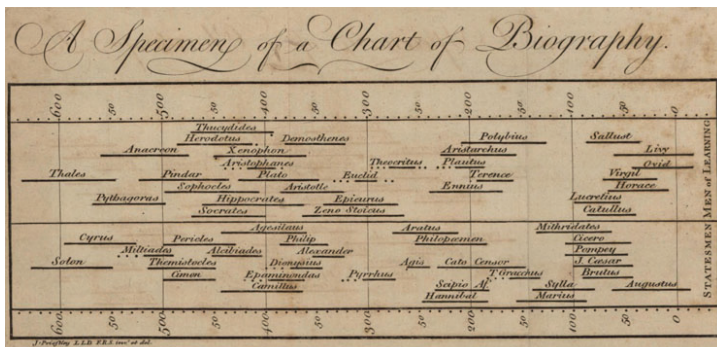
All of these charts were designed and produced on paper without manipulating discrete elements. The motivation for presenting this work here is to inspect the difference between these artefacts and previous ones. Furthermore, in Chapter 6 we will propose a technique for transforming for transforming these classic static charts into dynamic ones. And in Chapter 8 we will study how people can construct dynamic chart by manipulating discrete elements.

3.3.1 From line charts to area charts (1700-1800)

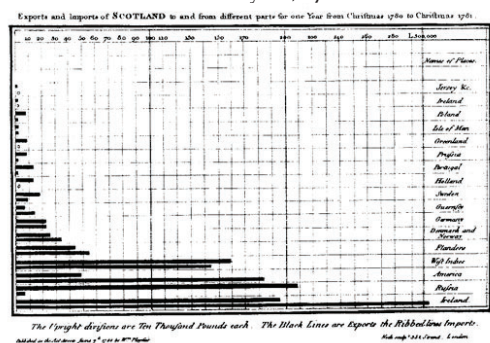
As observed by Friendly and Denis [Friendly and Denis, 2001], the 18th century was a very productive period for abstract graphs and visual representation. This period was the early beginning of the systematic collection of data and statistics, and new visual representations were required in order to understand and represent this data. These new practices and requirements were also simultaneous with several technical innovations such as colour printing and new lithographic procedures, which introduced new graphical possibilities.

From this period we can identified some of the first area-based charts. For instance Joseph Prietley (Figure 3.19) is credited with having produced full timeline charts (1765). According to Beniger

Figure 3.19: A) Timeline of philosophe, Joseph Priestley. 1769. B) Barchart by William Playfair, 1786.



A



B

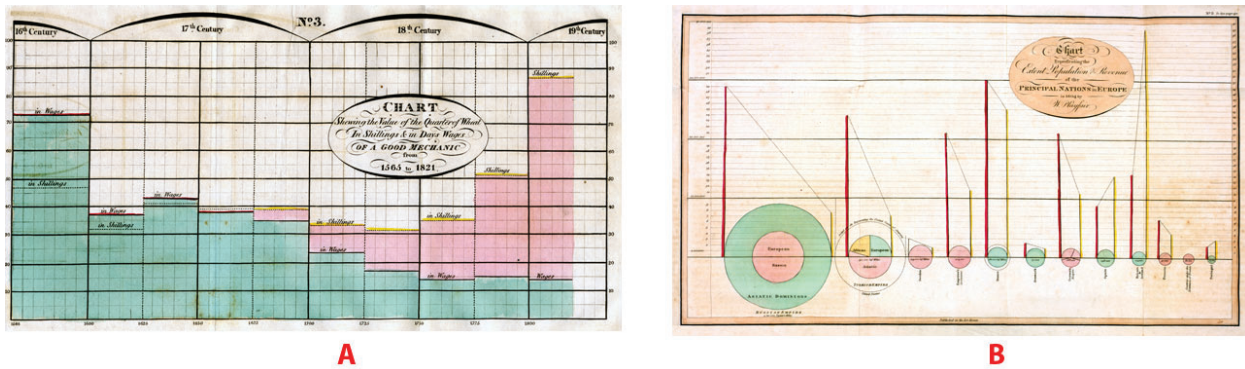


Figure 3.20: Playfair stack chart and pie chart

and Robyn [1978], a few years later in 1786, William Playfair wanted to reproduce these charts but ironically, due to lack of times data, he was constrained to producing the first paper-based barchart (Figure 3.19). Later he created the pie chart and the circle graph (Figure 3.20). Alexander von Humbolt, influenced by Playfair, combined these ideas to create the stack graph and the superimposed square for comparing areas.

These works had a great influence on information visualisation in general and are now part of common culture; we are able to easily produce them and use them in many contexts to communicate. And their widespread adoption is evidenced by the fact that most of them are implemented in generic computing tools such as Microsoft Excel.

Beniger and Robin describe this work as useful for “discrete quantitative comparisons” [Beniger and Robyn, 1978]. But apart from the timeline, all these charts are based on expressing an ensemble or sum of categorical data by mapping their values to an area filled by a graphical mark. We will later refer to this class of representation as an “area chart”.

During this period of history, collecting data, analysing it and producing a graphical representation, took long periods of time. It is perhaps for this reason, as well as the constraints posed by technology, that these charts were based on fixed elements. In Chapter 6 we will use the layouts previously defined for barcharts, piecharts and others to transform them into dynamic representations.

3.3.2 Otto Neurath Isotype. (1882-1945)

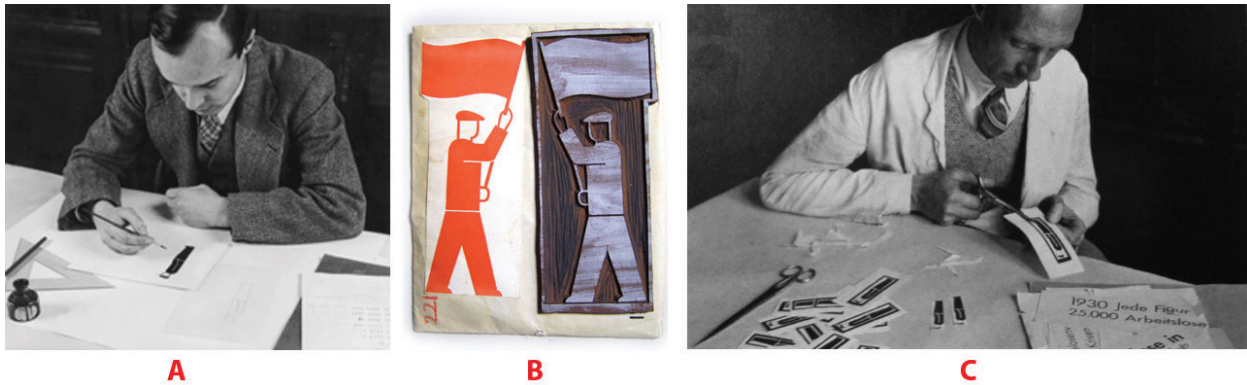


Figure 3.21: (A) Gerd Arntz designing one of the isotype pictograms, (B) A mold to print the isotype pictograms, (C) a assistant cutting pictograms in order to assemble them, (D) The result of assembling a set of pictograms.

Like Playfair, Prietley and Von Humbolt, Otto Neurath used paper for his visual representations, but instead of using area, he re-introduced discrete elements to make his chart more accessible and easy to compose. Otto Neurath [Cat, 2010] was an Austrian professor of political economy in Europe during the last century. He strongly believed that statistical data and information such as scientific results should be accessible to everyone. This led Neurath to define a graphical system for informing citizens and children about their position in the world according to statistics. He named this method Isotype for “International System Of Typographic Picture Education”) [Neurath, 2009]. The Isotype method is composed of two mains components. One is to show numeracy via countable units. The other is to employ simple pictograms to indicate the semantics of these units.

The unitization of visualizations generated a segmented layout by the repetition of elements in lines. This recalls the layout structures displayed on the antique clay tablets (Section 3.2.2). Neurath argued that his approach of making a countable chart was easier to read than the previous area charts (Section 3.3.1). In addition, he claimed it provided further benefits:

1. the pictogram provides a semantic signifier for the unit
2. the division of the area into sub units permits one to count units and makes it easier to compare quantities between two categories (Figure 3.22)
3. the division of the area into equal sub units between two charts permits comparisons which wouldn't be possible otherwise (Figure 3.23)

Neurath’s approach is based on keeping countability accessible both within a chart and between charts, and providing semantics for each countable unit. To achieve this goal, he subdivided the sum of the values into discrete sub units organized in a layout permitting comparison and countability between categories. The resulting pictures (Figure 3.22) with common graphical patterns recall the way

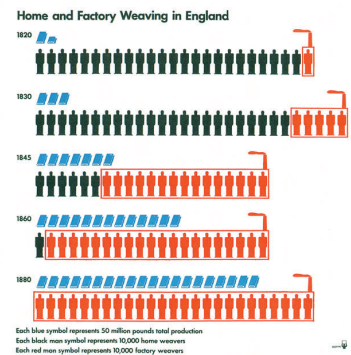


Figure 3.22: “Home and Factory Weaving in England”, Each pictogram stands for a multiple. Each blue symbol represents 50 million pounds total production. Each black man symbol represents 10,000 home weavers, each red man symbol represents 10,000 factory weavers.

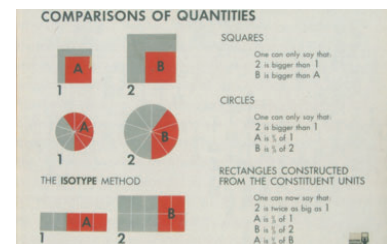


Figure 3.23: Explanation of the Isotype principle of countability (initially shown in London 1933 by the Isotype Institute.

Sumerians organized their clay tablets (Figure 3.8 C). The design of the visualisation we present in Chapter 4 was inspired by this countability principle. Later, in Chapter 6 we design a method for making the transition between a countable element representation and an area chart representation. Also, in Chapter 7 we analyse the approach of Otto Neurath to define a new visualisation design paradigm.

3.3.3 “La graphique” and the semiology of graphics (1918-2010)



Figure 3.24: A) a matrice in folded paper, B) a woman constructing and sorting a tangible matrix, C) the tangible matrix on top of a photocopier to capture an image of it. Extract from “la graphique et le traitement graphique de l’information”

Jacques Bertin (1918-2010) is a famous cartographer, well known for investigating the theoretical foundations of the modern information visualisation domain [Card et al., 1999]. He is also recognized as having identified visual variables (Figure 3.25) which are the basic components of visualization. He classified these into two groups, the spatial variables (position of the mark) and the retinal variables (properties of the mark). The spatial variables are the positions on the plan, the retinal are the visual aspects. In his book “The semiology of graphics” [Bertin, 1973] he defined “graphics” as a monosemic system of signs based on those variables:

A system is monosemic when **the meaning of each sign is known prior** to observation of the collection of signs. An equation can be comprehended only when the unique meaning of each term has been specified. A graphic can be comprehended only when the unique meaning of each sign has been specified (by the legend).

(...)

On the other hand, in graphics, with a diagram or map, for example, each element is defined before hand. The perceptual process is very different and translates into the question: Given that such a sign signifies such a thing, what are the relationships among all the signs, among all the things represented?

(...)

This agreement is then used **to discuss the assembly of signs** and chain the proposals in a sequence of evidence. Visualization can become indisputable, that is to say related to the three dimensions of spatial perception. On this point, mathematics and graphics are similar and allows rational processing.

Bertin’s 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 meaning (data properties) to a visual sign (visual mark)
2. an agreement between people on the meaning of this visual sign
3. the assembly of signs
4. the possibility of discussing and analysing the assembly of these signs according to this agreement.

Bertin's "Semiology of graphics" was published just before the introduction of the first computer with a graphical user interface. Even if he saw potential in computers for providing the technical means to analyse and visualize data, he used mainly tangible media, such as printed papers and photography, to construct his visualizations (Figure 3.24). In the book "La graphique et le traitement graphique de l'information" [Bertin, 1977], he described how he dynamically manipulated and assembled marks printed on papers (Figure 3.24,A) and how he used sets of dominos (Figure 3.26) to construct and re-order 2d matrices (Figure 3.24,B). After manipulating these tangible media, he captured the version he wanted to communicate using a camera or photocopier. Like Neurath, in these two cases Bertin dynamically assembled a representation through operations such as sorting, ordering and others, and then fixed the visual representation, which also captured the different operations of sorting, ordering etc. Another point that is important to note is that Bertin used both discrete elements that could be manipulated (dominos in matrices) and grouped data elements (printed barchart) to construct the composition of a visualization. However we do not know whether he used a similar approach for dynamically updating datasets.

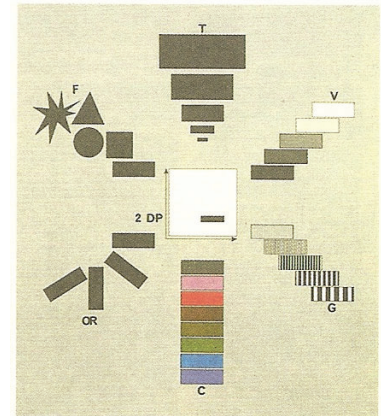


Figure 3.25: Jacque Bertin, Visual variables.



Figure 3.26: Jacque Bertin' "domino" used to construct matrices, each domino represent a data case and could be considere as a token. In background Jean Daniel Fekete, photo by Jean Baptiste Labrune.

3.4 Human computer interaction era: from icons to tangibles

Figure 3.27: Xerox Star icons.

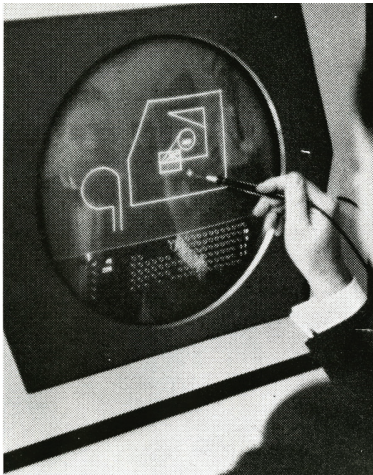
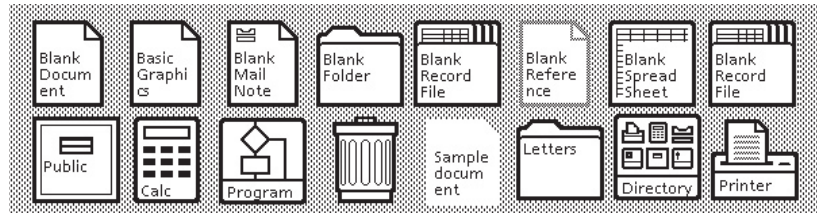


Figure 3.28: Sketchpad program. On-line video demo: https://www.youtube.com/watch?v=USyoT_Ha_bA

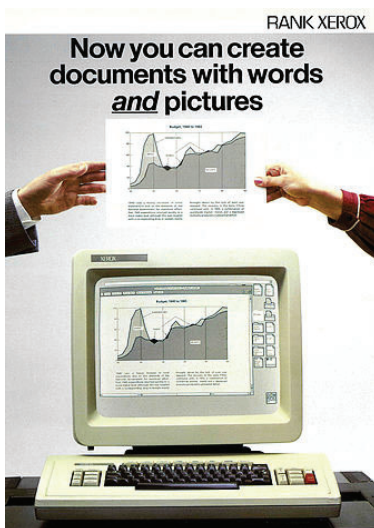


Figure 3.29: Xerox Star advertising.

In the previous subsection (Section 3.2.1) we saw how discrete elements were used before the computer era, then we saw how these techniques were applied using paper as the medium (Section 3.3). In this section I will survey the design of a similar approach for different purposes in human computer interaction. Human computer interaction is a field that emerged around 1970 [Myers, 1998, Myers et al., 2000] from the need to understand and predict how humans control a computer, with the objective of minimising the occurrence of errors and optimising actions.

Before 1970, people interacting with computers were trained by professional experts. The introduction of devices such as personal computers with a screen, a mouse and a keyboard, as well as the introduction of interaction metaphors such as the desktop, windows, icons and menus, stimulated a radical change and democratized computer technology, making it available to less expert audiences. In this subsection I will review some of the concepts and techniques related to *discrete individual elements*, which stand for information items.

3.4.1 Graphical user interfaces and metaphors

The currently popular interaction paradigm of having an interface that can be directly manipulated, i.e. where the elements of interest are visible on the screen and can be manipulated using a pointing device, was first introduced by Ivan Sutherland with his Sketchpad [Sutherland, 1963] (Figure 3.28) prototype in 1963. With the aid of a light pen device, the user can create shapes, select them, move them, and change their size.

Later, in 1973, Kay [1989] introduced the desktop metaphor at Xerox PARC to design the Xerox Alto computer. However, the interface was based on only a menu and windows system; the icon system had not yet been introduced. Two years later, Kay's Phd student David Canfield Smith, invented icons: "*(a) visual entities called icons, subsuming the notions of variable, reference, data structure, function and picture.*" [Smith, 1975]. Pygmalion, the system he was working on at that time, was the first custom visual programming interface. Later, Smith became the chief designer of the Xerox Star. The Xerox Star was the first commercial personal computer with a Graphical User Interface [Smith et al., 1987](Figure 3.29), and when working on

this device, Smith added icons to the interface. In this interface, the icons (Figure 3.27) refer to a paper metaphor: files are represented by rectangles with text inside like a paper sheet, the folders follow the same metaphor structurally – they contain files – and graphically – their function is signified by an icon that graphically represents a folder. Icons are symbolic representations that are particularly interesting. They are discrete marks on a desktop that graphically signify independent elements and can be manipulated to deliver particular functions. Concretely, a user of the Xerox Star can select an icon simply by pointing to it with a mouse and re-organise their icons on the desktop, although the original spatial organisation of icons was constrained by a grid. Later the drag-and-drop function was invented and the grid was relaxed.

Icons have some similarities to the Sumerian tokens such as the fact that they: (i) have independent discrete elements (ii) have fixed semantics defined by symbols (iii) can be manipulated into different spatial organisations, (vi) are simple to understand. On the other hand, they also differ from Sumerian tokens in that they: (i) are visual and not tangible, (ii) consist of textual as well as pictorial information, (iii) do not provide the means to count, even though they provide the means to organise sets (folders). (vi) and can be generated or deleted as needed.

The icons and desktop metaphor have been very successful. To cope with the flow of information on desktops, some authors such as Mander et al. [1992], created a “pile” metaphor for supporting casual organization of information flow. This technique is based on the idea of demarcating areas to represent sets of individual elements.

To address the accumulation of icons on the desktop, designers and computer scientists began to create other metaphors that could provide more space than a single desktop. For example, Data Mountain [Robertson et al., 1998] is based on a metaphor that models a 3d desktop environment (Figure 3.30), allowing document icons to be positioned anywhere on a virtual inclined plane. The metaphor provides more space for icons and creates a setup where they don't overlap.

Agarawala and Balakrishnan [2006] introduced the BumpTop system (Figure 3.31) to explore the application of physical properties to the virtual desktop metaphor. In BumpTop, icons have physical properties so that gravity and collision management are incorporated. BumpTop also offers different layout interaction techniques to organize icons into comparable assemblies, such as piles or mosaics.



Figure 3.30: Data Mountain



Figure 3.31: Screen shot of bumptop.

3.4.2 Assembly of visual building blocks for learning, modelling, designing

Figure 3.32: Scratch interfaces.

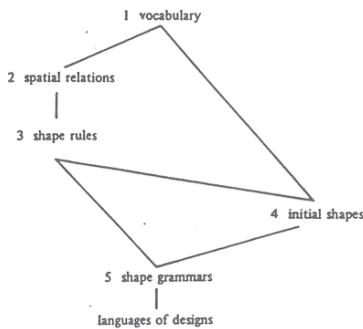
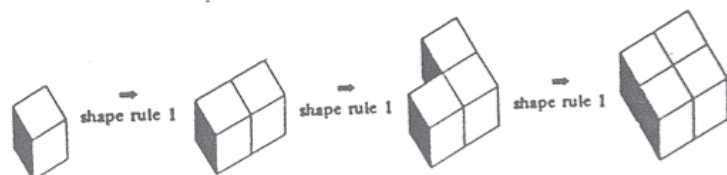


Figure 3.33: Stiny Kindergarten grammar.

Influenced by the work of Froebel (Section 3.2.7) some researchers adapted this idea to programming languages to create “Scratch” [Resnick, 2007]. They used the building block idea as a visual metaphor for programming, created the “command block” [Papert, 1999, Resnick et al., 2009].

These blocks represent variables, statements, expressions and control structures; the process is to snap them together to do the programming. This approach was so successful that “Scratch” now has a few hundred thousand users [Monroy-Hernández, 2012] and its logic is integrated into the programming interfaces of commercial products such as Lego Mindstorms. In their exploration of the design space of “Scratch”(Figure 3.32). Maloney et al. [Maloney et al., 2010] report the following dimensions as a possible factors of success: 1) a single window interface, 2) liveness and tinkability, 3) Making execution visible, 4) No error messages, and 5) Making data concrete.

Figure 3.34: Stiny example of incremental design.



In computer graphics, the adaptation of Froebel visual building blocks was also beneficial. The best example is the work of Stiny [1980]. Stiny defined a computable shape grammar (Figure 3.33) based

on the Froebel Gifts to assist computer-aided design (Figure 3.34). The generative properties of this work inspired several works on procedural modelling [Müller et al., 2006], computer aided design [Renner and Ekárt, 2003] and mechanical engineering design [Hsu and Woon, 1998]. Stiny’s contribution is to define a constructive computable language of shape designs. Our work is inspired by the same idea but the goal of our approach is specific to information visualisation and not to computer graphics in general.

3.4.3 Summary on token like metaphors in GUI

In this subsection I have presented a some elements of history of graphical user interface development for providing simpler ways for non-experts to manipulate information using computers. This evolution took place around a set of technologies, including the mouse, icons, desktop metaphors and the idea of direct manipulation. I argue that successful democratization of computer usage was in large part due to the use of *metaphors* and *discrete elements* that can be manipulated with simple procedures. We also presented different projects that attempted to provide solutions for managing constant flow of information and masses of documents, or simply supported new styles of desktop organization based on physical simulations.

Most of these extensions to the desktop metaphor are based on mimicry of physical forces and/or phenomena to provide ways of organizing, arranging and assembling the icons in ways that could be more easy to process. Finally, I also presented a few approaches using the Froebel Building block idea either for learning programming or for creating a computable grammar shape. Common to all these GUI examples (desktop metaphor, programming environment, and shape grammar) is the use of discrete elements. These elements stand for abstract data and constitute part of a metaphor for manipulating abstract systems. I summarise this view in Table 3.2. We also see how using analogy and metaphors is a common and effective process for defining computer user interfaces. As Carroll et al. [1987] notes, metaphor provides different benefits with respect to simplifying the interactions.

3.4.4 Tangible user interactions

In response to graphical user interfaces, in 1997, Ishii and Ullmer [1997] proposed a new view on interfaces. From the observation that we are living in a physical world and we have developed various work practices for manipulating information in this environment, they proposed the creation of tangible interfaces going beyond the traditional graphical user interface paradigm and embodied in tangible objects. A few years later, Ullmer et al. [2005] discuss, in the context of tangible interfaces, the concept of tokens to manipulate digital information. His model is based on several examples I already presented, such as board games (Section 3.2.5), abacuses (Section 3.2.4) and others. Ullmer proposed a tangible interaction model in which digital in-

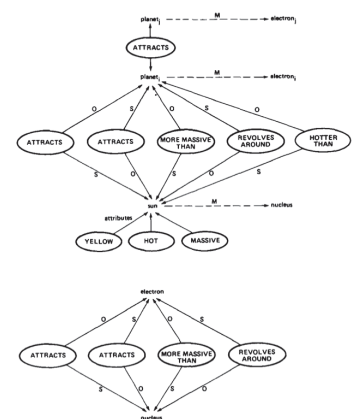
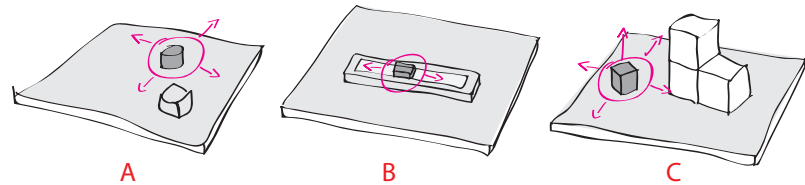


Figure 3.35: Structure mapping example of Gentner [1983] for the analogy “The atom is like the solar system”. Top: mapping of planet and sun to electron and nucleus. Bottom: properties common to the electron and nucleus. Not all properties of one system could apply to the second, but most of them.

formation is represented by physical tokens. In this system, logical constraints on the information are mapped to physical constraints. Ullmer explained that the relation between these two elements could be interpreted as an interaction event and as a trigger for digital actions. This approach is very powerful, since it can also be applied to analyses of previously introduced dynamic visual representations such as slider bars.

Figure 3.36: Illustration of the three different token+constraint approach.



In this model Ullmer et al. distinguish between three types of token-based approach¹ (Figure 3.36):

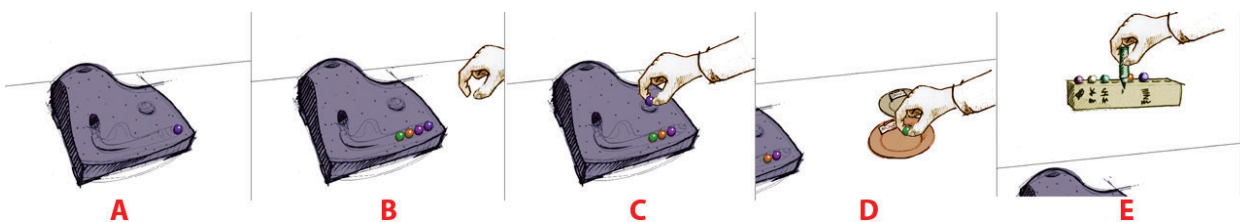
A Interactive surfaces are the devices or the techniques for manipulating physical objects on an augmented planar surface [Schäfer et al., 1997, Jordà et al., 2007]. Objects are used as physical controllers of digital objects.

B Token + constraint systems are similar to abacuses. They are systems that are most often used for manipulating abstract data that doesn't have a physical counterpart. These systems are based on two types of elements: tokens, which can be manipulated, and constraints, which provide a frame limiting the possible interactions between the tokens. Constraints are also providing clues for interpreting these interactions. In chapter Chapter 6, our visualisation design is inspired by a token + constraint approach in a visual environment.

C Constructive assembly is based on the idea that tokens can be used as modular interconnecting building blocks to model more complex systems such as the Froebel system (Section 3.2.7). In the second part of this thesis (Part III) I explored the potential for using constructive assembly to define and study a token-based paradigm for information visualisation design.

A famous example of a tangible interface, token + constraints design project is Bishop's "Marble answering machine" (Figure 3.37). In

Figure 3.37: "Marble Answer Machine", 1992, Durrell Bishop



this mock-up of a telephone answering machine. Each incoming voice message is represented by a marble of a different colour. The Marbles are released from a hole in the machine and physical constraints guide their flow to an area where they can stack. Then by grasping a marble and putting it into a container one can hear the message. Marbles can also be annotated or stored in groups inside other types of containers. The marble answering machine can be analysed as a token + constraints system, marbles standing for tokens and physical constraints creating a stacking assembly representing the number of remaining messages.

Building block approach in tangible interfaces

In the domain of tangible interfaces a lot of work has been done on manipulating discrete elements mapping to data to form a constructive assembly (Figure 3.38). Some of these interfaces are based on the idea of building blocks [Parkes et al., 2006, Girouard et al., 2007, Roberto et al., 2011, Schweikardt and Gross, 2006, Raffle et al., 2004, Wyeth and Wyeth, 2001] and some directly refer to the original idea of Froebel gifts [Anderson et al., 2000, Frei et al., 2000, Girouard et al., 2007, McCormack et al., 2004, Schweikardt and Gross, 2006, Raffle et al., 2004]. What they share is the basic idea that independent discrete elements can be assembled by construction to provide a more complex element. Most of these devices are used for pedagogical purposes. For instance, Smart block [Girouard et al., 2007] is a material to learn the mathematical principles behind volume. Ar block [] is a tangible programming environment. Topobo [Raffle et al., 2004] provides a means to model kinetic relations between blocks by example so that children can efficiently program the movement of robots. Roblock [Schweikardt and Gross, 2006] and Curly bot [Frei et al., 2000] both serve as pedagogical material. In these examples, the uses of discrete elements simplify complex abstract concepts for children.



Figure 3.38: Different tangible interfaces based on building blocks and constructive assembly logic. From left to right: Glume project, Smart block, AR blocks, Topobo, Roblocks, Curly bot.



Figure 3.39: Hans Rosling construct and update a dynamic visual representation of wealth distribution over the human population with pebbles. During the talk he explains that black pebbles represent the richest people in the world and white the poorest. Extract from : <https://www.youtube.com/watch?v=UNs-ziziPyo>

Table 3.2: Different computer era system composed with discreet element to process abstract information.

Use of discrete elements also began to be applied by different people to information visualization. I identify several examples of people discussing the use of tangible building blocks such as Lego [Hunger, 2008, Wilson, 2012] to build their own visual representations. In these cases I can identify different user profiles in the population, such as InfoVis non-experts [Hunger, 2008, Wilson, 2012] and data scientists (Figure 3.39) who exploit the simplicity and pervasiveness of these tools. In the second part (Part III) of this dissertation I analyse some of these practices.

3.4.5 Summary

Discreet element	System, Metaphor	Data types
Icons	Operating system, Desktop	Files, Set of files, program
Command blocks	Programming environment, Building blocks	Variable, function, statement
Building blocks	Computable shape grammar, Building blocks	Null
Glume project	Modelling and visualizing, null	Three dimensional data sets
Smart block,	Mathematical Manipulative, null	Unit
Topobo,	Educational activities, null	Kinetic information

In this section I considered two technical media for manipulating discrete elements: interaction on screen, and interaction with tangible elements. Different approaches to manipulating discrete elements are listed by Ullmer: *i)* Interactive surfaces, *ii)* tokens + constraints *iii)* and constructive assembly. Constructive assembly was exemplified in the survey by building blocks, sometimes as a metaphor, sometimes as a device. All of these elements provide a means of manipulating and controlling different types of abstract data, such as files, sets of files, variables, functions, statements, units, or kinetic information. Most of these systems are used to reduce the degree of complexity of a previous system. They are all used for interacting in a dynamic fashion. This collection of systems has inspired me to use similar types of approach in designing my dynamic visualisation systems for non-experts.

3.4.6 Tokens in Information Visualization

Recently, token-based information visualizations have emerged as a trend in the InfoVis community [Jetter et al., 2011, Steven Drucker, 2013, Rzeszotarski and Kittur, 2014, Yi et al., 2005, Vande Moere, 2002]. However, few studies exist on how people manipulate tokens as part of a *visual mapping process*. Here we summarize existing token-based visualisation techniques.

Yi et al. [2005] use a magnet metaphor to map multivariate data. Data dimensions are mapped to “magnets” and by moving these, data points are attracted depending on their value for the respective data dimension (Figure 3.40). Each data point can be seen as a token, though tokens are not interactive objects here and can only be indirectly manipulated through the “magnets”.

During the time of this research, several new token-based visualisation techniques have been presented in conferences; some based on screen such as SandDance [Steven Drucker, 2013], and Kinetica [Rzeszotarski and Kittur, 2014] and other with tangibles tokens [Jetter et al., 2011].

Jetter et al. [2011] use physical tokens to represent queries in a faceted search on a tabletop. Each token is a physical object representing a search facet. By manipulating the tokens, one can create a query through a hybrid interface.

SandDance [Steven Drucker, 2013] by Steven Drucker et al. is a information visualisation system (Figure 3.41). Focused on a “natural user interface”, Sanddance allows users to see individual records (data points map to grain of sand) and their overall structure. From a menu, users can map data attributes into different visual token (“sand”) to reveal patterns within the dataset. The system is design to manipulate hundreds of thousands of tokens. The manipulation is operated by touch interactions and is never directly on the tokens. The transitions between the view are animated. Each grains of “sand” is moving to the new mapping position to constitute the new visual representation.

Rzeszotarski and Kittur [2014] visualize multivariate data as tokens and combine these with physical affordances (Figure 3.42). In their multitouch system, data are mapped to points, which are equivalent to tokens. The system provides a set of virtual tools that allow users to build a representation by changing points’ positions, appearances, and interactivity.

3.4.7 Summary

In this section I considered several dynamic tangible and on screen visualisation based on discreet manipulative element. Most of the on-screen visualizations are using either a metaphor [Steven Drucker, 2013, Yi et al., 2005] to aid the user comprehension, or physically simulated constraints [Steven Drucker, 2013, Yi et al., 2005] to manipulate the visual tokens. None of these systems provide a way to visualize a datastream or to author the visualization.

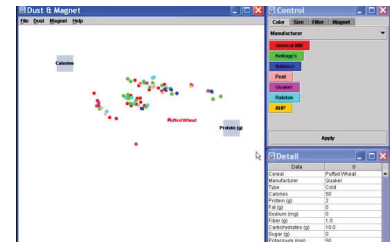


Figure 3.40: “Dust and magnets”, Yi et al. [2005], <http://youtu.be/wLXwL38xek0>



Figure 3.41: “Sand Dance”, Steven Drucker [2013], <http://research.microsoft.com/en-us/projects/sanddance/>

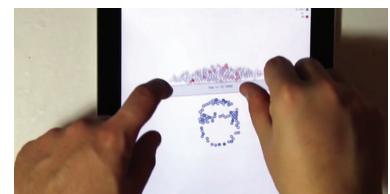


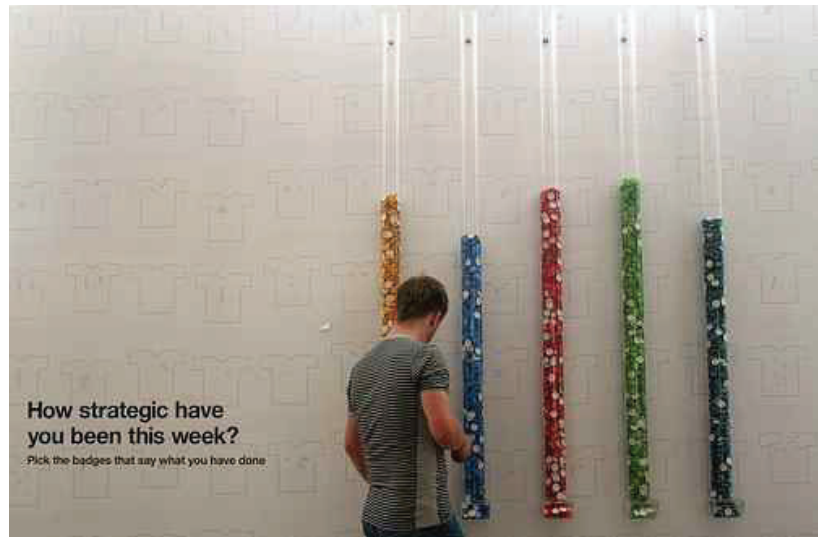
Figure 3.42: “kinetica”, Rzeszotarski and Kittur [2014], <https://www.youtube.com/watch?v=70YcGiKrmEg>

3.5 Information art and design based on element assembling

Lau and Vande Moere [2007] describe an information aesthetic continuum that spans from information visualisation to visualisation art. They observed that visualisation art often focuses on novel techniques for mapping data. In this subsection I focus on visualizations created by artists and designers.

In this domain, there is a broad range of representations involving the mapping of discrete elements to quantities. In the following section I described examples that explicitly use this approach. Some of these examples are used to represent dynamic data.

Figure 3.43: Pindices (2005) Lucy Kimbell and Andrew Barry, <http://www.lucykimbell.com/LucyKimbell/Pindices.html>



Pindices [Barry and Kimbell, 2005] (Figure 3.43) is a work of art created by Lucy Kimbell in collaboration with the sociologist Andrew Barry. The objective of the project was to “make political activity visible”. To do so the artist installed several 2m-high plastic tubes in a public space during an event. They filled the tubes with badges, each tubes containing badges of a different colour. They asked the public to reply to a question by picking a badge.

The badges in each tube stand for a particular response. In this way, the quantity of badges in the tubes would decrement over time and reveal the distribution of responses to the question.

“The Dumpster” [Levin, 2006] (Figure 3.44) is a good example of this aesthetic trend. The work visualizes a collection of 20,000 blog posts, extracting posts related to breakups of romantic relationships over the course of a year. “The Dumpster’s” visual encoding uses “bubbles” (similar to our previously presented tangible tokens) for presenting each romantic behavior collected in blog posts. These bubbles are organized according to physical forces. The “bubble visualization” is linked to a pixel based visualisation which provides an aggregated view over the course of the year but does not provide a transition between the aggregated area and the tokens.

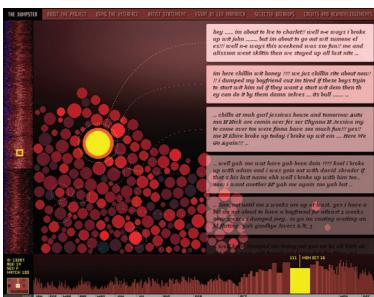


Figure 3.44: The dumpster (2006) Golan Levin. <http://www.flong.com/projects/dumpster/>

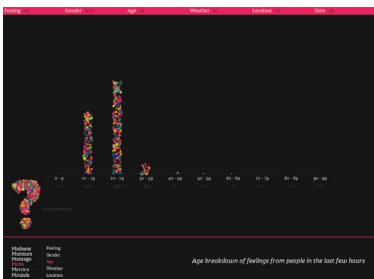


Figure 3.45: We feel fine (2006) bar chart based on visual tokens assembly.

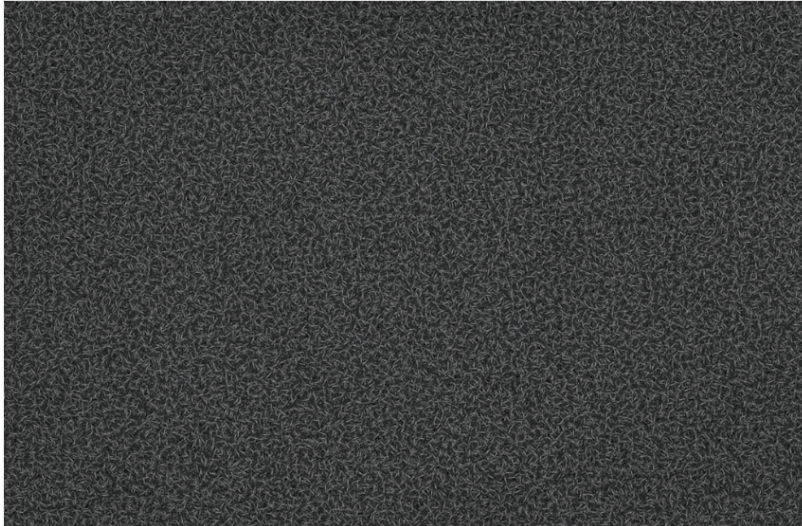


Figure 3.46: Handguns, 2008, 60x92 inches, “Depicts 29,569 handguns, equal to the number of gun-related deaths in the US in 2004.” Chris Jordan.

“We feel fine” [Harris, 2012] (Figure 3.45) is a well-known website which defines itself as “An exploration of human emotion, in six movements”.

Like “The Dumpster”, this website represents extracted blog-posts containing emotional words collected over the world. To show these elements in a visualization, the author, Jonathan Harris, uses rounded shapes that are spatially organized around templates such as bar charts. These templates permit quantitative analyses and serendipitous findings. Clicking on each token allows the user to view details.

Jordan [2008], in his artworks series “Running the Numbers” (Figure 3.46), produces monochromes based on accumulation of elements to express statistics about these elements. For instance, to represent the number of gun-related deaths in the US in 2004 (29,569), he painted 29,569 handguns on a canvas (Figures 3.46 and 3.47). Some of his canvas have a layout, some do not. This work shows how accumulation of individual discreet element (Figure 3.47) can be highly adapted to communicate emotions.

“US debt” [demonocracy.info, 2012] (Figure 3.48) is a famous infographic of the US debt, based on the accumulation of \$100 bills. This infographic video uses a storytelling process to provide a sense of scale between a unit token (100 dollars), which is commonly known, and the amount of US Debt at that time, which is not commonly known, \$122.1 trillion. To bridge the gap between the unit and the total sum, no aggregation process is used; instead, a viewport zooms out and large objects are displayed so as to allow visual comparison. However, the aggregation is only achieved by scaling down since the purpose of the infographic is to convey the scale of the debt by relating it to the size of \$100 bills.

Recently the design studio Vitamin [Vitamin, 2013] presented a device called the “Lego Calendar”(Figure 3.49). The Legocalendar is a time and resource planner for teamwork based on Legoblocks. It is positioned vertically on a wall. Each row represent a month,

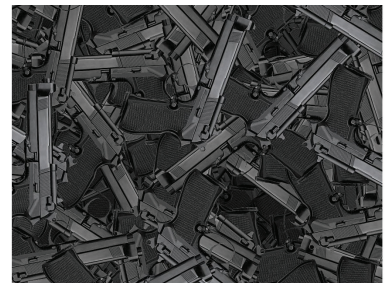


Figure 3.47: Handguns artwork details.



Figure 3.48: US debt visual representation. <http://usdebt.kleptocracy.us/>

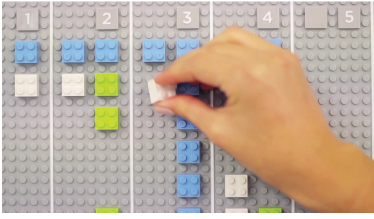


Figure 3.49: Vitamin Lego Calendar prototypes, screenshot from the video: <http://vimeo.com/75500464>.

each rectangle represents a day of the week. One Legobrick represent a human resource. Software automatically synchronizes a digital version online based on the tangible Legoverion.

There also exist other initiatives in which calenders are built from Lego bricks, such as Knud Molenbach's Lego wall prototype [George Fitzmaurice, 2009], or Pinata design agency's similar system in [Pinata, 2009]. We present two further instances of this type in Chapter 7.

3.6 Summary

In this section, I sketched from various snippets through time, a brief history of visual representation systems based on manipulative discreet individual elements. This history was given through four different time frames. First, I considered artifacts from the pre-computing era and the printing press era; then I surveyed the early days of human-computer interaction to the current day; finally I reviewed some pieces from contemporary art and design. From these four time periods I identified artifacts from different domains: counting devices, memorizing devices, table games, pedagogical material, exchange devices, computers interfaces. This collection of artifacts has spread through time, space, cultures and populations, and strongly supports the claim that discrete elements have been one of the primary instrument for manipulating dynamic visual representations of information over time. This overview provides the motivation for the research and designs presented in this dissertation.

We also noted that discrete elements were previously treated as tokens by Ullmer et al. [2005] in the domain of tangible interfaces. The same author also recognised the possibility of using such elements to manipulate computer systems. Ullmer et al. define three styles of mapping for token-based tangible interface architectures: *Spatial*, *Relational* and *Constructive*. In this research, I will apply this approach to the visual encoding of information on screen. In the first part of this thesis, I will explore the *Spatial* and *Relational* token-based approaches to inspire my design of dynamic information visualizations. In the second part of this dissertation, I will investigate the token-based *Constructive* assembly approaches to define a new information visualisation design paradigm inspired by the examples presented in this history.

This token-based paradigm will instruct all my design and research. My motivation for that is: (i) this approach seems to withstand the test of time and is widespread, (ii) this approach gives us a typical pattern for simplifying complex elements in computer science, and (iii) this approach has yielded successful outcomes in many dynamic visual representation systems.

Part II

Dynamic Data Analysis for Non-Experts

In the previous related work chapter (Chapter 2) we identified that backchannel use and crowdsourcing could be a great application domains to experiment with dynamic visualizations for the general public. To the best of our knowledge previously there was no generic visualisation technique to visualize dynamic data for the general public. This part of the dissertation is focused on our first research **PROBLEM1** posed in the question: could dynamic information visualisation be useful for the general public? To study this problem we first decomposed it into two sub questions: **PROBLEM1.1** Is it feasible for the general public to process and analyse dynamic data? and **PROBLEM1.2** is how can dynamic data be visualized for this audience?

In order to gain knowledge about the feasibility of using dynamic data visualisation for the general public (**PROBLEM1.1**) we present the design, deployment and evaluation of three different systems. Each of these systems have been deployed “in the wild”. We use the phrase “in the wild” as commonly used in human computer interaction domain to denote real world situations, where the researchers do their best to not interfere and to impose no controls. Two of the three - PolemicTweet and Bubble-TV - were specifically designed to engage audiences into a dynamic annotation and tagging process during the event. The first system, called “PolemicTweet” (Chapter 4) was design to engage the audience of a conference to produce structured tag through microposting in order to index the video tape of the event. The second system “Bubble-T” (Chapter 5), was designed to visualize in real time the opinion of Twitter users about the French presidential candidate during the 2012 national presidential campaign. From this second system we derived a third one “Bubble-TV” (Chapter 5). Bubble TV was designed to process a dynamic poll along with a TV show and gather metadata during the TV show. All of these systems used a social network as a communication channel to process the set of pre-defined tasks.

The visual encoding design of all of these systems were inspired by a token-based approach describe in previous chapter (Chapter 3). They all apply this idea in a different manner, and in progress through these three systems this token-based approach was refined incrementally. The first system, “PolemicTweet” (Chapter 4), visually encoded each tweet into a little square that were stacked one after the other into a barchart setup. Updating of this visual encoding was abrupt and lacked continuity. This motivated us to introduce smooth transitions in the next system by using a physical simulation. We implemented this in “Bubble T” (Chapter 5), where each tweets was encoded into as round shape and a physical simulation was used to assemble them in a barchart. Both of these visual encodings limited the amount of data that could be displayed over time, to solve this problem we merge these techniques with classical area chart in “Bubble-TV” (Chapter 5). From these different iteration we structured and defined a novel design metaphor: “Visual Sedimentation”.

Visual Sedimentation (Chapter 6) is a design metaphor that can be used to visualize dynamic data. This metaphor unifies our three previous design iterations and addressed our second sub problem: (PROBLEM1.2) how to visually encode dynamic data for the general public. This metaphor provides a way to illustrate the various stages of dynamic data: appearing of new data, accumulating with previous data, and the aggregation of data to provide a context over time. In the Chapter 6 we defined the sedimentation process as it appears in the physical environment and describe how we selected some physical sedimentation properties to applies as a metaphor for dynamic visualization. To facilitate the design of visualizations using this metaphor we implemented a toolkit. We then explored the design space provided by this toolkit and present a group of variant visualizations.

4

PolemicTweet: Engaging Audience in a Dynamic Data Analysis Process

The literature review discussed in Chapter 2, allow to identified few systems that already used the general public social activity surrounding an event to annotate and index videotape of the event. However in the context of these systems, the data analysis and visualization process were mostly static. In those cases, when the event happened, people's social activities were recorded and later the data gathered was processed and tagged either by an algorithm or by crowdsourcing. Finally the visualized data were used to index a videotape of the event. We also identified in Chapter 2 that this process either generated some *cost*, or was *slow*, and it was not immune from *ambiguity*.

This situation seems to be a great application domain to evaluate our first problem **PROBLEM1** Is it feasible for the general public to perform dynamic data analysis? In this chapter we ¹ Extend the previously describe process by introducing a different dynamic data visualisation component. For this purpose we designed a system and a protocol that engage the audience of the event in structured tagging activity of their own social activity.

Comparing this dynamic data analysis with the existing static data analysis approach suggests that dynamic data analysis by non-experts *is possible*, and also provides *useful*, *quick* and *economic* information. It also suggests that dynamic data analysis by non-experts may benefit both the public and the organisation of the event by providing instruments that involve the audience into the interpretation process of the event.

¹ Portions of this chapter were previously published in [Huron et al., 2013a]. Thus, any use of "we" in this chapter refers to Samuel Huron, Petra Isenberg and Jean Daniel Fekete, and also Raphael Velt, Karim Hamidou, Yves Marie Haussonne from IRI development team.



Figure 4.1: Additional resources:
Online System:
<http://polemictweet.com/>
Metadata player code source:
<https://github.com/IRI-Research/metadataplayer>

4.1 Introduction

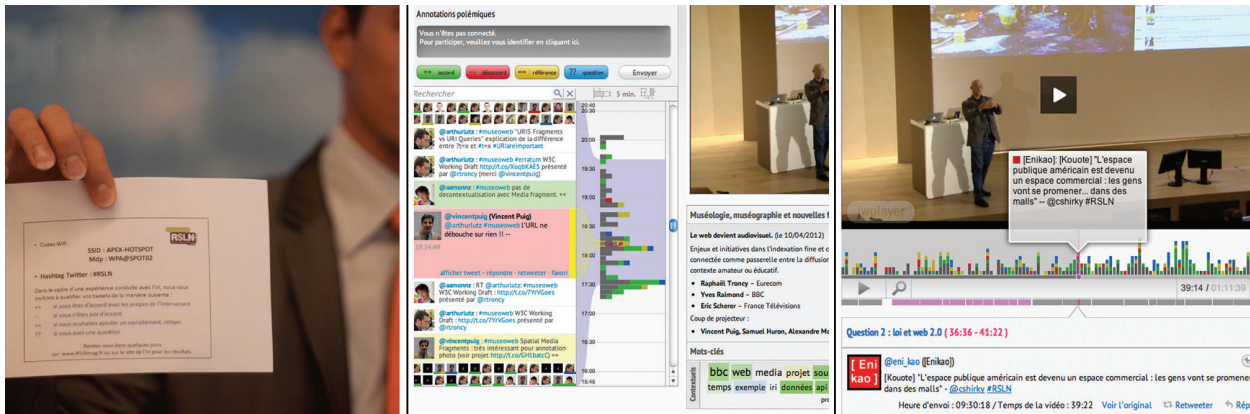


Figure 4.2: The three stages of PolemicTweet: 1) printed information on connection & tags, 2) the live Twitter client & structured annotations, 3) video player showing tweet annotations, accessible on this link: <http://polemictweet.com/rsln/polemicaltimeline.php>.

In this first case study, I designed a new system with the goal of crowdsourcing dynamic data analysis. This was achieved by the design and deployment of PolemicTweet. PolemicTweet is a web-based structured video stream tagging, browsing, and analysis system. We present in this chapter the design the results of a long-term evaluation of the system under live deployment. The system is unique in that it supports a wide variety of related activities ranging from video tagging to analysis and provides its different components offering a variety of benefits for different kinds of audiences. PolemicTweet was developed in response to the need for effortless but rich video tagging for event recordings and their subsequent browsing and analysis. A number of organizations in the world, including the Pompidou Center, regularly organize mid-sized conference-style events, with audiences of 25-300 attendees. Many of these events are recorded using videos for archival purposes or for sharing on the web. For example, a rapidly growing global audience regularly follows the TED events series [TED, 2013]. However, with current online video players it is difficult to get additional information on a recorded video. Video metadata does not generally give good information about the events unfolding during the course of a video, such as the topic(s) covered, controversies that may have arisen, or reactions of the audience to a speaker shown in the video. This is because videos are complex to summarize, analyze, search and browse. Yet, we argue that providing additional information is highly valuable to external viewers for both casual browsing and deeper video analysis.

We thus, designed PolemicTweet to capture structured annotations in a cost-effective manner and provide this information for perusal. The captured metadata is meant to help both casual browsing and analysis of video content. PolemicTweet also provides visual summarization of annotations captured during live events and includes additional features for search and browsing. We specifically contribute a web-based solution to video annotation and analysis that is



Figure 4.3: The Microsoft conference room in Paris where PolemicTweet was deployed; the use of laptops or other wifi-capable devices was prevalent and ensured active participation.

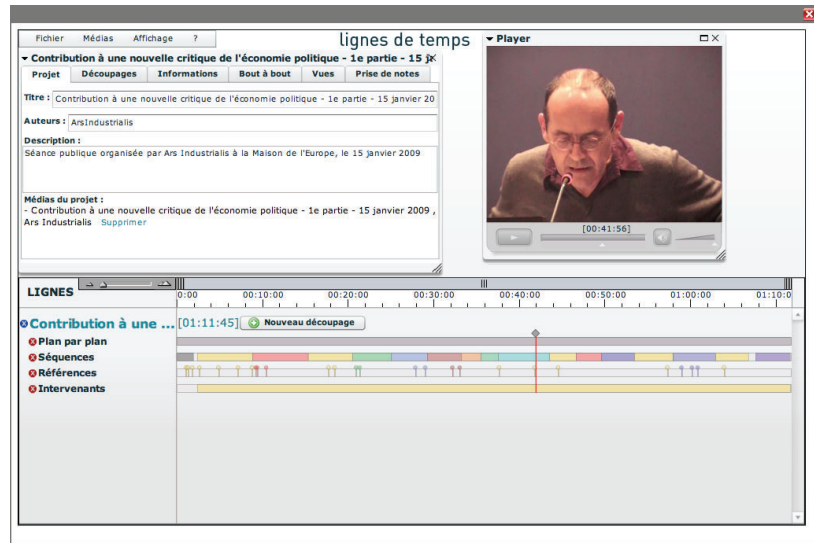
easy to deploy and use. It is engaging in that the annotation phase offers benefits to participants beyond their involvement in a tagging activity. PolemicTweet is a well-rounded system of integrated components and has been evaluated in a long-term study involving the capture and annotation of video from 27 different events over one year.

4.2 Motivation

Our design of PolemicTweet was motivated by the proliferation of microblogging activities at live events associated with our organization: the “Institut de recherche et d’innovation du Centre Pompidou” (IRI). IRI is a research association focusing on the study of cultural practices enabled by digital technologies. A part of IRI’s activities is to organize seminars (Figure 4.3) on subjects such as philosophy, design, and digital cultures. A typical seminar is video recorded. A formal procedure of human annotation and tagging on the video footage was used to follow the event in order to make the videos more searchable and comprehensible. This video annotation task was generally assigned to a interns (often archivist students) using a custom software called “Ligne de temps” [Puig et al., 2007]. This process had several major disadvantages: without having attended the event, the job was tedious and time-consuming for the student, and costly for the organization.

After this internal annotation process, the video recordings were published online on a web platform displaying the same annotation interface as “Ligne de temps” (Figure 4.4). In addition to the first set of annotations, the attendees of the event were allowed to add further annotations. However, the participation remained low: the interface for annotation was designed for an expert user and did not entice the public participation we had wished for. Also at the end the annotation was based on just one person’s interpretation of the video which is not representative of how the public appreciated the video content.

Figure 4.4: Screen shot of the “ligne de temps” interface.



The previous process was a typical a static analysis process: data (the video) is produced at one time and then later it is analysed and annotated by a someone. This process takes a long time. To optimize this problem IRI had already started to transform the static process into a dynamic one by developing a note-taking mode in “Ligne de temps”. The intern or somebody else at IRI then had to take some notes during the conference, the software timestamped these notes, and later these notes were re-synchronized with the video. This dynamic system was useful but its best use was limited to the organizers.

From these past experiences, we wanted to expand the dynamic analysis process by including the conference in the process. To do this we decided to take advantage of the micropost data produced by the audiences attending along our events. The resulting system, PolemicTweet, was, thus, motivated by three goals:

1. Engage the audience to tweet and to crowdsource video annotation and tagging,
2. Provide a visual backchannel to incite people to tag their tweets with predefined tags and thus provide structured annotations,
3. Provide a easily accessible tools in an encompassing system that spans the whole process from annotation to video analysis.

In order to reduce the complexity of the annotation process we decided to make it more engaging and to open the process to a wider audience. As indicated in Goal 1 above, we opted for a crowd-sourced solution. Crowdsourcing was a promising solution since the behaviour of participants in live events has evolved due to two digital enablers: microblogging and portable devices. The speaker nowadays is no longer the only information channel during an event, a second information channel (backchannel) is now regularly used:

the audience’s tweets. This discussion is the data source we wanted to leverage for video annotation and analysis in our tool PolemicTweet.

4.3 Video and sentiment analysis

To analyze or study an event, video recordings and/or audio tapes are frequently used [Allen, 1989]. For this reason the study and design of video analysis software is a popular topic in information retrieval [Hauptmann, 2005], computer human interaction [Diakopoulos et al., 2009, Abowd et al., 2003, Mackay and Beaudouin-Lafon, 1998, Roschelle and Goldman, 1991], computer supported collaborative work [Cockburn and Dale, 1997, Schroeter et al., 2011, Engström et al., 2012], and visual analytics [Daniel and Chen, 2003, Ruppert and Kohlhammer, 2010, Tonomura et al., 1993, Parry et al., 2011] as well as hypertext [Aubert and Prié, 2005].

In this section, we discuss efforts most closely related to our goals and solutions in regards to our video analysis interface and tweet sentiment analysis.

4.3.1 Video content analysis

Mackay and Beaudouin-Lafon [1998] designed DIVA, DIVA was one of the earliest systems on video annotation, tagging, and analysis (Figure 4.5). Mackay synchronized records of metadata with her video records, such as the movements of a mouse on a screen, with her video records.

Nowadays recorded metadata is often taken from real-time web social services like chat or microblogging. PolemicTweet relates to this trend but asks Twitter users to tag their own tweets for structured annotation. Another solution that aids in the process of video analysis has been to support multiple analysts in parallel. Cockburn and Dale [1997], for example, designed and developed CEVA a synchronous collaborative video annotation system that focuses on supporting parallel analysis. Parallel analysis is based on the idea that multiple video analysts (five in their prototype) share their analyses in real-time for distributing the workload. The authors argue that this synchronicity property offers two potential benefits: 1) synergy of group participation and 2) distributing the analysis workload. We were particularly interested in supporting the synchronicity property for the PolemicTweet tagging phases to similarly take advantage of the synergy of the audience during event recording. The Videolyzer tool [Diakopoulos et al., 2009] also includes an asynchronous formal semantic tagging tool to increase information quality by allowing users to collaboratively organize their comments on the video. Their tool differs from ours in that we did not want to rely on a complex tagging interface in order to more easily spur user tagging in real-time during the event. Nevertheless, Videolyzer inspired the social use case scenario of PolemicTweet.

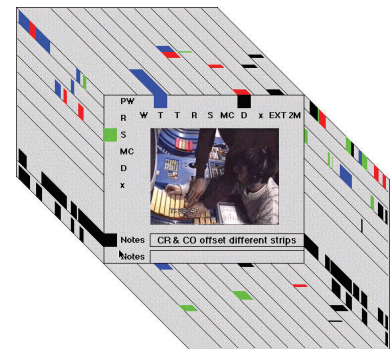


Figure 4.5: Screen shot of the “DIVA” main display.

4.3.2 *Tweet content analysis*

Transcripts, tweets, and other temporal data streams are often used to annotate video. In the case of temporal data composed of text, it is common to use named-entity recognition, sentiment analysis, and natural language processing to recognize and tag events of interest. When we started developing PolemicTweet, sentiment classification methods used on tweets were not as effective as on classical natural language corpora due to the limited length of tweets and the common use of shortened and non-standard English words [Pang and Lee, 2008]. Now, the combination of different techniques from machine learning [McCarthy et al., 2005], semantic rule based approaches [Jiang et al., 2011], and graph-based optimization [Speriosu et al., 2011], have improved sentiment classification of tweets significantly, and can achieve 85.6% accuracy [Jiang et al., 2011]. In this last work, the authors considered three types of sentiments: negative, positive, and neutral. The reported accuracy rate was achieved on an English tweet corpus of the most popular queries on Twitter (Obama, Google, iPad, Lakers, Lady Gaga), and on a clearly targeted subject. We did not take advantage of this approach as it only works for English tweets, it is complex to set up, and has only been tested on a topic-limited tweet corpus. Instead, for PolemicTweet, we needed to find a solution that would work in multiple and mixed languages, was simple to deploy, and would work on a variety of specialized corpora like philosophy, aesthetics, or design. Diakopoulos and Shamma [2010] used another approach for sentiment classification: they used Amazon Mechanical Turk to perform hand-annotated sentiment classification on tweets. Turkers were compensated \$0.05 per ten tweets analysed. Turkers were asked to tag four types of sentiments: negative, positive, mixed, and “other”. The corpus of tweets was in English and about politics (2008 US presidential debate). In a second article [Diakopoulos et al., 2010], the authors described the use of machine-learning algorithms to perform the same analysis with lower accuracy and cost but higher speed. Crowdsourcing sentiment classification on tweets with its higher success rate and relatively low overall cost is becoming more and more common. For instance, it is one of the products of CrowdFlower [Crowdfower, 2012] a well-known crowdsourcing platform. However, this approach cannot be used to tag tweets in real-time. Whether or not to use crowdsourcing as a tool depends on one’s annotation goal [Quinn and Bederson, 2011] since there is a speed vs. quality trade-off to consider. For PolemicTweet, we wanted to have the best of both worlds: classify tweets with a low cost (both in computer processing and in money) and have a high precision rate, even if the recall rate depended on the adoption of our tool. We also wanted to have this classification in real time.

4.4 *Getting the right metadata to visualize*

As we have presented in previous section, most of the actual systems are based either on asynchronous formal semantic tagging or use of social network conversations. In the case of social network conversation, tags are extracted by crowdsourcing or by natural language processing. We proposed a new approach based on user engagement that allows us to gather different types of information according to our need. This approach consists of engaging users in an active position. We ask them to tag and annotate the video stream dynamically by using their own social activity. The metadata produced is then each user's own interpretation.

4.5 *System Design*

PolemicTweet is made of four interconnected components: 1) the definition of four tags to annotate tweets and the backchannel interface to read and write tagged tweets during the conference event, 2) a social protocol to set up and run an event live, and 3) a web video player synchronized to a tweet visualisation to navigate and replay conference video recordings, 4) a website to provide a fluid browsing between all PolemicTweet components. We describe the components in this order.

4.5.1 *Backchannel to crowd source tag annotation*

To provide an enriched visualisation of video annotations we wanted to collect tweets structured by specific annotation tags. We chose to provide a limited number of tags with a simple syntax and a specific colour code (Table 4.1). Our choice of tags was informed by observation of the types of tweets typically used in the conference-style events that we were targeting with PolemicTweet. For choosing our set of tags, we had to balance expressive power and simplicity. To provide a memorisable set of tags, we decided to use the simplest possible tagging format. At the same time, our interface needed to provide enough incentive for using the tags.

Semantic	Syntax	Colour
Agreement	++	■
Disagreement	-	■
Question	??	■
References	==	■

Table 4.1: Syntaxe used in PolemicTweet: agreement with the speaker (++), disagreement with the speaker (-), questions raised by or proposed to the speaker (??), and references (==), e.g. quotes or URLs to related content

The screenshot shows the PolemicTweet backchannel interface. It features a top navigation bar with a login prompt (A), a search bar, and a list of tweets. A timeline on the right side shows the sequence of tweets, with a red circle (B) highlighting a specific tweet. On the right side, there is a video player showing a presentation slide (C) titled "Muséologie, muséographie et nouvelles formes d'adresse au public". Below the video, there is a text box (D) with the title "Muséologie, muséographie et nouvelles formes d'adresse au public" and the subtitle "Le web devient audiovisuel". At the bottom right, there is a "Mots-clés" section (E) listing various keywords related to the content.

Figure 4.6: The final PolemicTweet backchannel interface.

The design rationales for choosing the tag syntax were:

- **Simplicity** of use and memorability, to facilitate its use and adoption;
- **Brevity**: to cope with the 140 character limitation of Twitter and for fast typing;
- Client and language **neutrality**: to be usable from any Twitter client with any kind of text input method, usable in any language;
- **Ease of parsing**: for automatic processing tools;
- **Expressiveness and univocality**: to allow clear statement of intent and for machines to be able to interpret it unambiguously.

4.5.2 *Engaging the crowd, social protocol for synchronous annotation*

One very unique feature of PolemicTweet is a social protocol we developed in order to help event organizers make best use of PolemicTweet. The main goal of our protocol is to inform the audience about the capture and future use of their tweets and that to introduce the four

specific tags. The protocol suggested that event organizers to follow three phases: Before the talk: Instructing the crowd. We designed a “connection package” to inform the audience. It consists of a flyer (Fig. 4.2) given to attendees during the registration process, containing information about the network access policy, the PolemicTweet tags, the website URL, and the Twitter hashtag for the event. Additionally, we asked organizers to make an announcement prior to the conferences. During the talk: Crowd source sentiment analysis and video annotation. Attendees send tweets with the PolemicTweet tags, a program records all the tweets relative to the Twitter event’s hashtag. The visual backchannel website is set up to provide real-time visual feedback of tweets and to give an incentive for participation and easy access to people who do not use Twitter (Fig. 4.6). Organizers can also make use of this interface to get informed about questions to ask at the end of a talks. After the talk: Publishing. Organizers synchronize the recorded data with the timestamps of the video streaming server and the tweets timestamps. Then they publish the video on the web (copying and modifying a configuration file) with the MetadataPlayer, which we discuss in the next section. Synchronization of the video and tweets could be complex due to the lack of reliable video timestamps. In the case of Diakopoulos et. al. [Diakopoulos et al., 2009], synchronization was simple because the video was provided by an official TV and had a well-known timestamps. In the academic context, the event could be recorded by different actors and not broadcasted. And most of the time if the video is not streamed the video timestamps is not reliable enough. We, thus, deploy several strategies for synchronizing video recording and tweet activity: we video record a clock before the event, we have a special tweet at the beginning of the event, and we re-synchronize the video with a custom script. Of course for resynchronization of tweets to a videotape it is important to use the originally captured video footage and not an already cut and edited version.

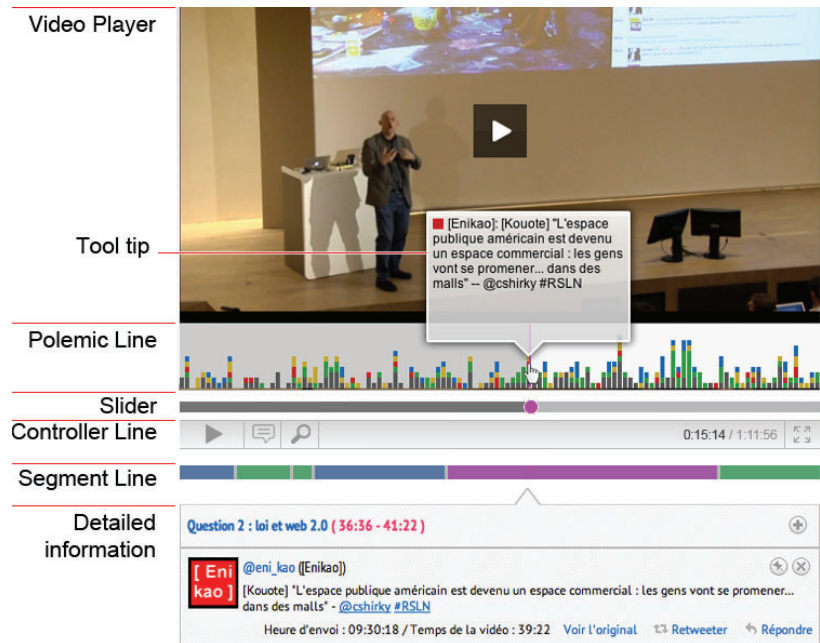
4.5.3 *Video analysis web video player*

The PolemicTweet video player (Fig. 4.7) is part of a bigger project at IRI and his named MetadataPlayer². The MetadataPlayer is designed to play a video while showing different types of information such as the tagged and untagged tweets sent with the conference hashtag during the time of video recording. Our specific design goal for PolemicTweet was:

1. to provide a visualization that gives an overview of the tags and activity spikes,
2. to design a compact player that can be easily embedded, just like the YouTube player,
3. to augment the player’s video navigation techniques to get more benefit from the annotations.

² The MetadataPlayer is a free open-source project available on GitHub: <http://goo.gl/eKr0QU>

Figure 4.7: First complete mock-up I did of the MetadataPlayer presenting each of the component used in PolemicTweet.



We designed all the time-related components as graphical horizontal projections on the time axis. All components have the same width and horizontal scale to allow for a vertical alignment between them. Navigation of the video is performed using the standard video player components and controls: a controller line with the buttons to play, pause, change the audio level, and a seek bar. Two novel components show context and allow for contextualized interactions: a segment line for quick access to chapters of the video such as topics of a presentation or name of presenters for a panel. The main novel component is the polemic line. It visualizes the tweet activity during the event, positioned on the time axis; it provides contextual feedback on the tweet activity, a direct access to video segments highlighted by tweets, and direct access to the tweet contents through hovering. The polemic line has the following characteristics:

- **Polemic line visualization:** The visualisation is a bar chart composed of colored graphical tokens. Each token is a square that represents a tweet colored according to the tweet's tag or grey if the tweet does not contain any tag. The token square has a fixed size of 5x5 pixels to facilitate its pointing and selection without taking too much screen real estate. Depending on the length of the video, each 5px slot represents a certain time range. At each timeslot each recorded tweet is represented by a token square sorted top-to-bottom by type (green, red, blue, yellow, and grey) and by time, so that the most expressive tweets are at the top of the bars.
- **Interaction:** Moving the mouse over a token square shows a tooltip with the tweet's text, author name, and tag colour. Clicking on it seeks the video corresponding to the time of the tweet, and makes meta information appear in the details information component



Figure 4.8: Search for “ushahidi” in the “polemic line”. The tweets referring to the section of the talk related to this query are highlighted in pink. On top of that we could see the player synchronized showing the slide of the speaker about “ushahidi”.

(Fig. 4.7). The position of each token provides information about when the tweet was emitted but also about the contextual activity level at the moment.

- **Search:** A search text-field (Fig. 4.7) in the controller bar allows finding text in the tweets. When a search string is entered, the visualisation is updated to highlight each token representing a tweet containing the string. As shown in Figure 4.8 this feedback shows the distribution of tweets containing the specified string (in pink).

4.5.4 Polemic line design rationale

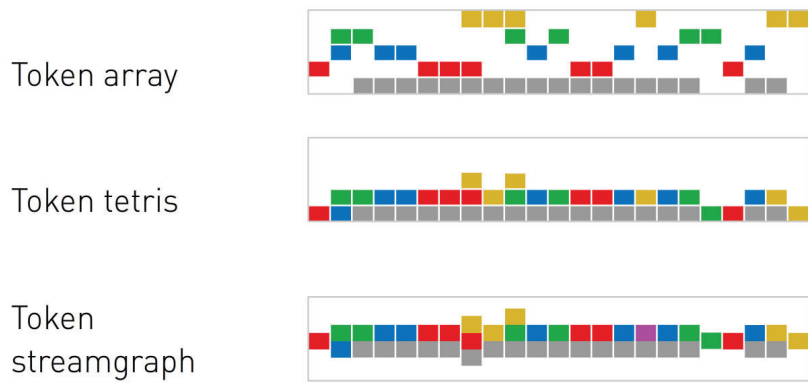
Classical
stack bar chart



Figure 4.9: Mock up of a classic stacked bar chart.

To define the design of the polemic line, we explored different design alternatives. In this section we relate this exploration to the rationale. The PolemicLine requirement was to visualize categorical time stamped data. The distribution of data over time was defined by IRI as an important factor from which some points of interest could be extracted. The first solution for this requirement was to use a classic stacked barchart as presented in Figure 4.9. Keeping a

Figure 4.10: Design variations.



homothetic mapping between the video player control bar and the chart of the polemic line (Figure 4.7) helped to avoid problems with scale comparisons. For this reason the x-axis was used to represent time, the y-axis to express the quantity of tweets by timeslot, and finally, stacked areas were used for the category comparison. This barchart design benefited from being culturally shared, well known, and providing a simple tried and tested solution that had already been implemented many times before.

However there were two desirable features that this chart could not provide: **1)** the ability to point to individual annotations inside the chart and get information directly, **2)** the ability to update the chart iteratively (Figure 4.11) during the event so the tokens could operate as tally marks (Chapter 3.2.7).

To address the first problem, we considered using a fisheye [Furnas, 1986] view to show individual elements inside the surface, as illustrated in the mock up in Figure 4.12. However, this solution is complicated to implement and does not address the second problem. Based on these two goals, the idea of being able to point to individual elements, and the inspiration of previous token-based visualizations (Chapter 3), I generated several different designs variations as shown in Figure 4.10. The first iteration of this series was “Token array”. Influenced by the “ligne de temps” interface I decided to use horizontal lines by category, and inside these add tokens to represent the position of the annotation. this design was not selected because it did not effectively show the frequency of annotation in the same category within the same time frame. I derived other designs from

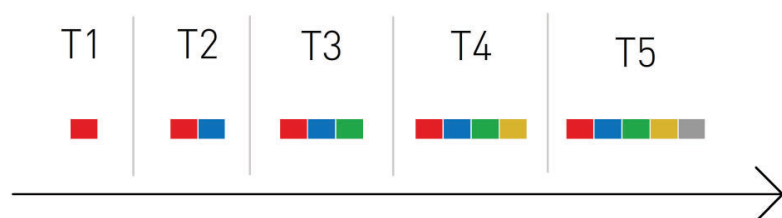


Figure 4.11: Incremental token property.

Roll over
- fish eye view

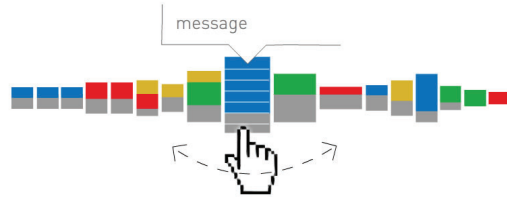


Figure 4.12: Fisheye effect on an area chart to get an individual element.

other inspirations such as “Token Tetris” and “Token Streamgraph”. “Token Tetris” is close to the “Token array” design, but the tokens are stacks on a vertical axis and can show the frequency of annotation by time slot.

Our specific design goal for the Polemic line visualisation was to provide:

1. a visualisation that would be simple to understand for a non-expert user and that could be explained in a single short sentence such as “Each square represents a tweet, and the horizontal axis is the time.”
2. a visualisation where one could see the overview first and get the details [Shneiderman, 1996] of the tweets just by mousing over (Figure 4.13)
3. a visualisation that could be used as an interface widget to control the video player,
4. a visualisation that could be updated iteratively

After presenting the different chart variations and specific design goals to the IRI team, we selected the “Token tetris” variation. The “Token Tetris” variation fulfilled all of the above criteria and is also close to the “barchart” reading convention.

Roll over

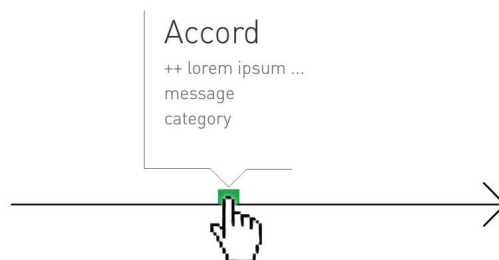


Figure 4.13: Mock up of the roll over feature on a token.

4.5.5 Website

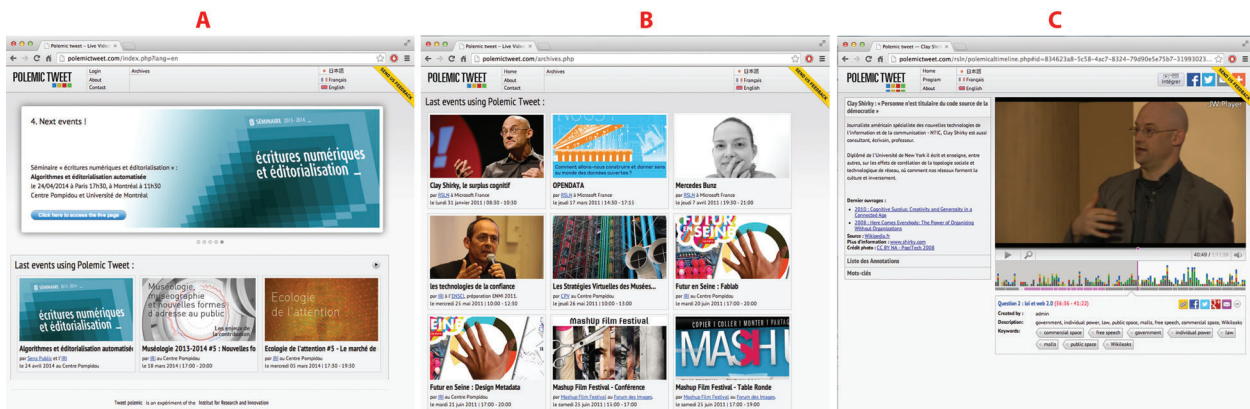


Figure 4.14: PolemicTweet public website: A. Home page, B. Archives page, C. Past event page with the Metadata-Player.

The next step was to provide a fluid browsing experience between all of the PolemicTweet components and to build a public website composed of three main pages which embed the previous components as follows:

1. a home page presenting the past, present and future events,
2. an archive page to find older events,
3. a backchannel webpage which is only available during a live event, and
4. a page for past event that show the annotated videos.

We could have a lot more pages to enrich the user experience. For instance, a summary of annotations by Twitter users, or a user page, but we were limited in resources and choose to stay focus on the dynamic part.

4.6 Assessing the Impact of PolemicTweet

In order to better understand how PolemicTweet would support our goals of inexpensive, useful, and engaging video annotation, we studied the tool using a mixed methods approach, incorporating both qualitative and quantitative approaches under a long-term deployment. We successfully deployed PolemicTweet in 27 real conferences, seminars, and events at different locations, on different topics, and with varying sizes of audience during 2012. During these 27 events we recorded 46 tracks (video and tweets), each track being one speaker or one session. 20 events took place at the Pompidou Center in our conference room, 7 events took place at various places in the city of Paris. Nine events had between 25-50 attendees, 13 events 51-100 and five events 101-290. The topics of the events varied greatly, including academic topics such as science, technology, philosophy, and design.

The duration of 24 events was limited to one day while three events took place over several days. The recorded tracks from each event ranged from 30 to 395 minutes. Overall, we received and collected 9,088 tweets from 1,012 unique accounts. Over the year 2012, the web site (player, backchannel, and also other additional pages like home and archives) received about 157,000 page views from over 15,000 unique visitors with an average visit duration of 3 minutes which is comparable to National video websites with substantially more content and public coverage like Francetvod.fr [Mediametrie, 2012] which had a 3.2min average visit duration during February 2012. Except for the video recording, sound capture, and the communication support for the tags (flyers and poster boards), the components of PolemicTweet were deployed on the web, which greatly simplified the process for the organizers. The recorded videos are all published on the open site ³.

³ <http://www.polemictweet.com>

4.6.1 Data Collection

To study the use of PolemicTweet we gathered three types of data: all tweets using the various conference's hashtags, logs from the usage of the PolemicTweet MetadaPlayer and the backchannel interface, and two questionnaires sent to various users of PolemicTweet, either as designers, organizers, speakers, or audience participants. We used Google Analytics to log the website pages and a logging system ⁴ to collect low-level traces of user interactions with the video player that Google Analytics could not provide. On our backchannel interface, we recorded usage logs only via Google Analytics to give us information about where users came from (direct access, Twitter, devices, others) and where the users were physically located (inside or outside the conference room). This last information does not take into account attendees not using the provided free Wifi connection (e.g. using their mobile's own 3G network) and possible errors of domain detection. Though there were a few tablets in use. Device most frequently used for connection were mostly computers. We also recorded the web client's signature for each tweet to understand where tweets were sent from. We sent two web surveys, the first consisted mainly of close ended questions sent by Twitter to 140 randomly chosen attendees of one of the recorded events; the second questionnaire was sent to understand how PolemicTweet fared from the organizer's point of view. This questionnaire consisted of open-ended questions sent by email to five members of the event organization team in our institute. For the first questionnaire we received 47 responses, 27 of which completed all parts of the survey. 70% (19) of the respondents attended at least one of the conferences. 96% (26) had already used Twitter at conferences, 3% had only used it for reading. This shows a clear positive feedback towards the goals of PolemicTweet. For the second questionnaire we received four completed responses.

⁴ We used Aubert's ktbs4js. Source code: <https://github.com/oaubert/ktbs4js>

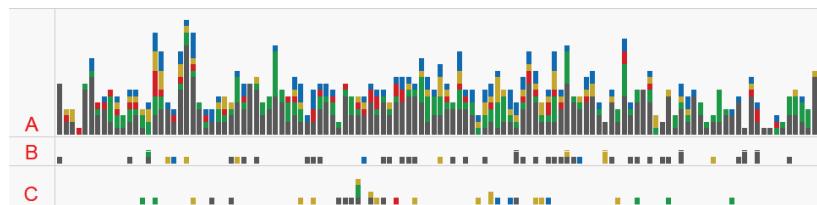
4.6.2 Analysis

In this section we report on the analysis of our collected data according to three main questions:

1. Is the system sufficiently engaging to collect data for crowd sourced video tagging?
2. Does the visual backchannel incite people to tag their tweets and thus provide annotation structure?
3. Is the video web player with annotations useful?

4.6.3 Is the system sufficiently engaging?

Figure 4.15: Polemic lines of A) “Les Strategies Virtuelles des Musee: l’Heure de Verite”, B) “Mashup Remix”, C) “Du vinyl au numerique”, a 2h event with different density of annotation, number of peaks, and tagging activity.



Through our live deployment of PolemicTweet, we found that engagement in the crowdsourced video tagging was highly diverse. We recorded from 0.20 to 7.94 tweets per minute (mean at 1.47, median at 0.89), with 0.12 to 2.13 tags per minutes (mean 0.47, median 0.32). Averaged across all events tweets were tagged using our syntax 40% of the time. Considering tagging a tweet as an additional cost, we conclude that the system provided enough tagging incentive on average. Otherwise, we found that engagement in the annotation process was highly varied and depended on several factors including the type of audience, the content of the event, the number of Twitter users, and whether the PolemicTweet protocol had been respected or not. We discuss these in more detail now: Type of Audience. Our audiences were varied in terms of size, culture, equipment, incentive to achieve a task, and distribution of local and distant attendees. All of these factors played a role on the participation and engagement of attendees in the tweeting and tagging task. In our collected data, we had a group of events in which the number of recorded Twitter accounts in comparison to the number of attendees was very high (between 64% to 187% on 7 events with 825 tweets on average, (e.g. Fig. 4.15. A).

Groups with values higher than 100% represent those in which more Twitter accounts were recorded than participants in the events. Here remote attendees also used the conference hashtag to participate in the backchannel. The recorded events can be categorized according to two participation types: medium participation (51% to 20% on twelve events with 209 tweets on average) and low participation (18% to 1%, on seven events with 61 tweets on average, Fig. 4.15. C). Type of Speaker and Twitter Reaction. Some attendees reported through

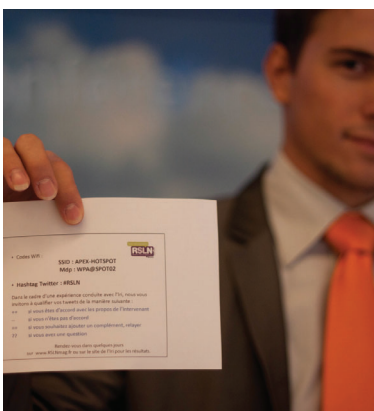


Figure 4.16: Flyer of a Microsoft’ event with the PolemicTweet instruction distribute at the registration desk.

our informal interview that writing a tweet and tagging it took more attention in-situ than when following the event online. We had originally hypothesized that a captivating speaker would probably lead to less Twitter activity, but from our observations organizing and attending our events we could not confirm this hypothesis. On some events we observed a high level of tweet activity during a captivating talk and others with a captivating speaker had hardly any tweet activity.

On the other hand we also noticed that in cases of less captivating talks, a peak of tweet activity arose because the audience was no longer focused on the speaker. Overall, regardless of the quality of the speaker we found that the audiences of our events were polite and only used negative tags 14% of the time. Type of Content. We found that the type of content of an event and its audience could be highly related factors when it comes to annotation activity. For instance, one of the events' topic was the websites of the Pompidou Center⁵. In accordance to the topic, the audience consisted of people active on the web and social networks. During this event we saw a very high participation (Figure 4.15. A) in terms of tweets and tags: 6.06 tweets per minute and 2.13 tags per minute. On the other hand, we were also surprised to see low participation for certain events where the content would have suggested high Twitter usage. For example the "Muséologie 2.0"⁶ had the topic: museum digitization, and preservation policies (Figure 4.17). Here, the level of participation was extremely low: 0.20 tweets per minutes and 0.12 tags per minute. Observance of the Polemictweet Protocol. We found that the protocol was generally well applied by the event organizers (Figure 4.16). Yet, in some cases we observed that deviance from the protocol could impact user activity. In particular we found problems related to hashtag selection, instruction diffusion, and connectivity.

For the event "L'Open Data, et nous, et nous, et nous ?", organizers chose to use an existing hashtag to boost interaction between audience and remote Twitter users. They used #opendata, a common hashtag, which had been in use daily before the event by this community and others. The effect was that the tweet number per minute exploded to 7.94 but the number of tags per minute stayed really low at 0.88 as the tags were not known outside of the event. The backchannel was crowded by other tweets relative to the subject but not to the conference. This was a source of frustration for some attendees and made our system less useful in real-time and as a video annotation tool.

Choosing a specific hashtag is important if video annotation is a goal, even if the event should stay connected to existing Twitter communities. Another issue regarding the observance of the Polemictweet protocol was that the instruction distribution to the audience was always different. Sometimes flyers were put on a seat, sometimes handed out with oral instructions, and sometimes placed somewhere on a table at the entrance. Despite these differences, most of the time attendees found the information and tweeted. Yet on some events



Figure 4.17: Flyer of "Museologie seminar" organize in the Pompidou Center.

⁵ <http://polemictweet.com/CPV/polemicaltimeline.php> and the Tate gallery

⁶ <http://polemictweet.com/2011-2012-museo-structured-data/polemicaltimeline.php>

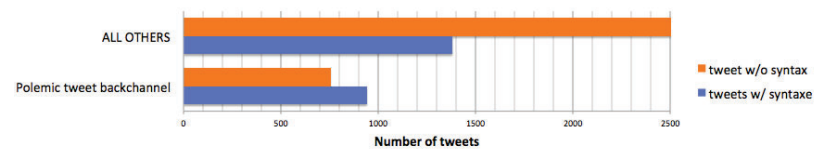


Figure 4.18: Flyer of "Mashup film festival" organize in "Forum des Images". Video: <http://polemictweet.com/mashup/polemicaltimeline.php>

we observed that flyers were not provided and the announcement of instructions was made during the event and not at the beginning. This dramatically impacted the audience participation. For example this resulted in only 0.27 tweets per minute for the event “Du vinyl au numérique” (Figure 4.15.C). The last and most obvious factor that highly impacted the audience participation was network connectivity. For instance, in the event “Mashup and Remix” (Figure 4.18) mobile networks (3G and others) were hidden because the conference room was usually a cinema. A wifi network was provided but connectivity information was just given orally and not on the flyer. Thus, despite an audience of 70 people only five sent any tweets (0.85 per minute).

4.6.4 Does the visual backchannel incite people to add tags?

Figure 4.19: Tweet with and without PolemicTweet tags split by backchannel vs. other clients.



Of the 27 responses to our first questionnaire, 89 % (24) reported that they remembered PolemicTweet tags from attending the event. While our general goal in making the tags known was reached, we were interested in learning more about people’s tagging practices and in particular in regards to our backchannel interface. We observed that tweets sent by the PolemicTweet backchannel interface had a higher rate of tag adopt (55% of tweets were tagged) than for the top ten other clients used (20% of tweet were tagged) (Figure 4.19.). A Welch’s t-test was conducted to compare percentage of tagged tweet emitted through the backchannel and by all other clients. There was a significant difference in the score for the backchannel ($M=70\%$, $SD=26\%$) compared to all other clients ($M=24\%$, $SD=11$): $t(23)=6.74$, $p<.001$. Attendees sent significantly more tagged tweets through the backchannel than through all other clients, suggesting that the backchannel interface did engage people to participate in tagged tweeting. Remote attendees spent more time on the backchannel interface than local attendees. Of those attendees who spent more than 5 minutes on the interface, 62% (Table 4.2) were remotely connected through outside the conference wifi. From informal feedback from attendees we hypothesize that local attendees did not need some of backchannel interface features like video streaming and preferred their own Twitter client but that these features were useful for remote attendees.

Based on a Welch’s t-test, we observed that, on average, attendees who used our tags on more than one tweet per event tweeted significantly more than those who did not tweet using our tags (average of 5.3 tweets per person without tags, 15.5 with tags, $p < .01$). This shows a strong correlation between tweeting activity and their use

of the tags. Although we cannot conclude on causality at this point, the two alternative explanations are either that 1) attendees using the syntax were tweeting more meaning they were more engaged, or that 2) attendees who tweeted frequently had no difficulty adopting our tags. Informal feedback leads us to think that both situations happened.

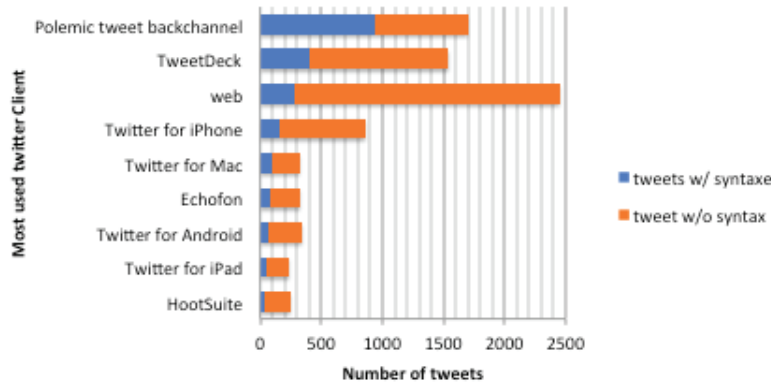


Figure 4.20: Tweet with and without syntax by the most popular Twitter clients.

	> 1 minutes	> 3 minutes	> 5 minutes
Min	72%	30%	22%
Median	86%	80%	62%
Max	93%	89%	81%

Table 4.2: Percent of remote attendee versus local using the backchannel interface, ranged by average of time by domains in minutes. Based on Google analytics report for 15 events.

4.6.5 What is the impact of presenting tweets and tags on a video web player?

To measure if the tweet visualisation component was useful, we logged user interaction on the video player as described in Section 4. To identify impact on the user activity on six events, we conducted a Welch’s t-test to compare if there is a correlation between where the video was played and the density of tweets present at these locations. According to these measures, viewers seemed to specifically seek out spikes of Twitter activity on five of the six videos: three trails followed the Twitter activity significantly ($p < .01$), two showed trends ($.01 < p < .05$) and one did not follow ($p > .1$). All trails that were correlated to the tweets had on average more than one tweet by bar in the chart. According to the attendees survey sent after the end of the event 40% (11) reported to have used the Polemic line Visualization. 90% (10) of these respondents found that the interface provided the following useful information:

- Item summary of Twitter activity: 90 % (10).
- To view a part of video with:
 - Many tweets: 81 % (9)
 - Positive opinions (++): 36 % (4)

- Negative opinions (-): 54 % (6)
- Questions (??): 45 % (5)
- References (==): 36 % (4)

81 % (9) respondents agreed or strongly agreed that the visualization helped them to browse the video and 100 % (11) would recommend PolemicTweet to a friend. We received additional feedback from open questions in our surveys and report the most interesting here. One particularly prominent comment was the notion of hotspots and points of interest, which also resonates in the answers above. Several comments were similar to this one: “PolemicTweet is useful to browse videos faster and find a hotspot (polemic, debate, synthesis, and minutes)”. Another frequent comment related to video overview. Five users wrote that PolemicTweet was useful for “viewing the mood (of the audience)”, and that this overview helped them to “construct an opinion”, and to put the talk “in perspective to the audience’s opinions and references”. Others commented on the usefulness of the player after an event to “re-find some information and share it with people who could not attend.” We also received a comment from one of the organizers that the player was useful to “prepare the chaptering, for taking notes of a speaker change, and references (to others resources).” On the same point, probably one of the most enthusiastic organizers reported that during the event he used the system for note taking: “it replaced my pencil and paper for note taking.”

4.7 Discussion

Evaluating PolemicTweet as a deployed system over a long period of time was challenging because we had little control over our audience and settings. Yet, our analysis and feedback of PolemicTweet has been predominantly positive on both usability and usefulness. Was PolemicTweet successful? According to our analysis, the three main goals of PolemicTweet were achieved:

1. The system engaged the audience, encouraging them to provide data for crowdsourced video tagging,
2. The visual backchannel incited people to tag their tweets and thus provided an annotation structure, and
3. The media player augmented with the annotated tweet visualization, was considered a significant improvement over traditional video players

We found that PolemicTweet was successful beyond its originally intended purpose of video tagging. In particular remote participants were able to get real-time visual feedback about audience sentiment and could be involved in the event as a commentator and annotator and not just a passive listener. The PolemicTweet environment and

backchannel interface provide a friendly tool for augmenting the public debate around conferences; this is something we consider valuable for improving the communication around important societal issues such as citizens and science or politics in the digital era. Beyond that, the PolemicTweet outcomes indicate that we can crowdsource an annotation task using social networks (and not only with dedicated crowdsourcing platforms) depending on the content and community. Last, the use of the backchannel interface can aid by providing an incentive to achieve a task, like tagging during a collaborative synchronous activity. In summary we found a considerable amount of excitement among participants and event organizers. PolemicTweet can be considered as a first step into exploring real-time information visualisation of crowdsourced tasks. Our design is applicable to a variety of different scenarios and events such as classroom presentations, MOOC ⁷ [Aubert et al., 2014], synchronous web seminars, public debates, and social TV. We believe our experience could be applied on popular video websites to leverage the crowd for annotating the videos to provide a richer user experience. Despite this, some of our choices should be re-assessed:

- **Possible optimization.** We collected a large number of annotations through tweets but not all were tagged. With our pre-defined tags we could achieve a better precision than sentiment analysis algorithms, but for some events had only low recall. More work is needed to research incentives for the audience to tag tweets.
- **Shared attention.** Asking the audience to tweet during an event and moreover to tag their tweets comes with challenges. First, it requires additional work to remember the tags and to add them. The danger is that attendees may be losing the focus of their main activity (listening to the conference) and even of their secondary activity such as writing a tweet (Figure 4.21). Can we further simplify the interface so that it would require less attention?
- **Tags Property.** We received a lot of feedback from Twitter users on our tags. Some of them criticized their restrictiveness; others felt that tags should be based on existing usage for expressing sentiment. We have several reasons not to rely on existing tags or practices: tags like #fail, #happy, as well as Smileys, are not universal, are ambiguous and are longer than our syntax.
- **Scalability,** User interface design and system. PolemicTweet comes with some scalability problems. Due to the homothetic representation of tweets, the interface is limited by temporal density. Our design works well for events in which Twitter activity is between 0.5-5.0 tweets per minute However we met a graphical scalability problem with two of our recorded events: the standard representation took too much screen real estate. To solve this problem, we used a classical aggregated stacked area chart.
- **Update Mechanism,** Because at this time there are no design guideline we designed the visualisation update to be as simple as pos-

⁷ MOOC stand for Massive Online Open Courseware.

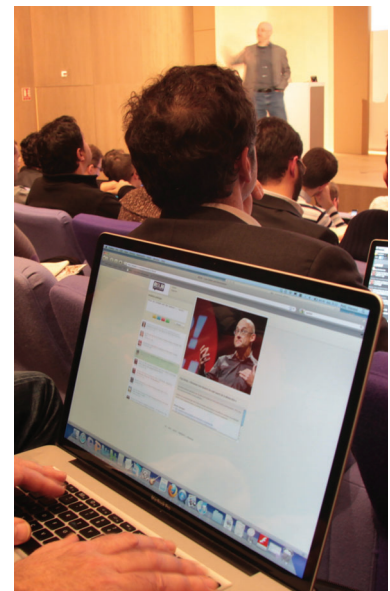


Figure 4.21: A participant computer, “What to look at the stage or the screen?”. Video: <http://polemictweet.com/rsln/polemicaltimeline.php>

sible. The update is incremental without animation. When the visualisation is updating a new token square simply appears, but the way it appears does not provide any emphasis that is status is new. We think that improvement have to be done on that topic.

4.7.1 *Is it possible to process dynamic data analysis with non-expert users?*

We deployed PolemicTweet during 27 events with between 25 to 300 attendees per events. During these events 1012 people used Twitter to send a tweets with the hashtag of our event and in total 9088 tweets was recorded. Over these 9088 tweets 40% used our syntax to tag their tweets and annotate the video. During the event part of the public was engaged in the activity, allowing us to collect structured annotation of the video data stream. During these events the visual backchannel seems to provide a great support to the dynamic data annotation activity. After the event, the public changed their video watching comporment, by giving more importance to the tagging part of the video. This system allows people to gather, structuring and represent useful information for the public of this event.

This system demonstrates the feasibility of achieving dynamic data analysis with non-experts during live event. However, our study suggests that the system is highly dependant on the context, the public engagement, the visual backchannel design, and the respect of the protocol.

4.7.2 *Follow up applications*

Due to the success of this application several organizations contacted us to deploy the system in different scenarios such as the speech of a minister, a Web TV of a newspaper (Figure 4.22), and a Cinema projection, etc.⁸. This variation in deployment allowed us to see that changing the context of deployment could have an important impact on different components. For instance sometimes wifi is not accessible to the public or not efficient, or the host communication around the system is not enough clear or visible, sometimes the event has its own backchannel and publicize it in concurrence with ours. Sometimes the speaker have has a position of authority with the public, which could produce a social reserve. All of these factors, internal to the event organization but external of the system, could drastically impact the annotating and tagging activities by the public. We thinks that adaptation could be done to solve some of these factors as we specifically do in chapter 5 for a real television show.



Figure 4.22: PolemicTweet used during a web tv application for the news paper Mediapart.

⁸ Most of these events can be found on the PolemicTweet archive page: <http://goo.gl/Mprnn9>

4.8 Chapter Summary

In this chapter we presented PolemicTweet a system and a social protocol that allows to engage conference public into a dynamic data analysis process. PolemicTweet is a deployed system, used regularly

now at least once a month. This system was deployed for more than one year and co evolved during several cycles with its users (audience and organizers). This is the reason for its success as well as for the difficulty of evaluating it according to its design goals: they have evolved. We assessed the success of PolemicTweet and the factors influencing adoption, participation, and use in a long-term deployment. We recorded a wide range of different events and despite the differences in audience, topic content, and event location. The system showed its robustness and effectiveness: 1) to produce *useful structured annotation and tagging* in most cases, 2) to provide a method and tool to *engage the audience* in a live tagging activity through a visualization, 3) to support web-based video browsing activity while providing useful landmarks with a simple yet powerful navigation tool.

The main contribution of this chapter is the design, deployment and analysis of the system. PolemicTweet is reducing video annotation complexity and cost for events of approximately 25-300 attendees, such as academic conferences. This system provide a *proof of concept* that general public could perform dynamic data analysis task. In this really specific situation we solve the problem [PROBLEM1.1](#) “Is it possible to leverage a crowd of data stack holders to analyse dynamic data for free?”.

However despite the success of PolemicTweet, this system allow us to identify some problems relative to [PROBLEM1.2](#) the way we designed the visual encoding such as the update mechanism and the scalability. In order to explore possible solution to this problems we will explore other designs in the next chapter with two new systems “Bubble T” and “Bubble Tv” (Chapter 5). In this design we visualize similar data stream, and support similar activity, but with a larger population scale and bigger datastream.

5

Bubble T&Tv - Design Exploration: Combining Token With Physical Sim- ulation & Area Chart

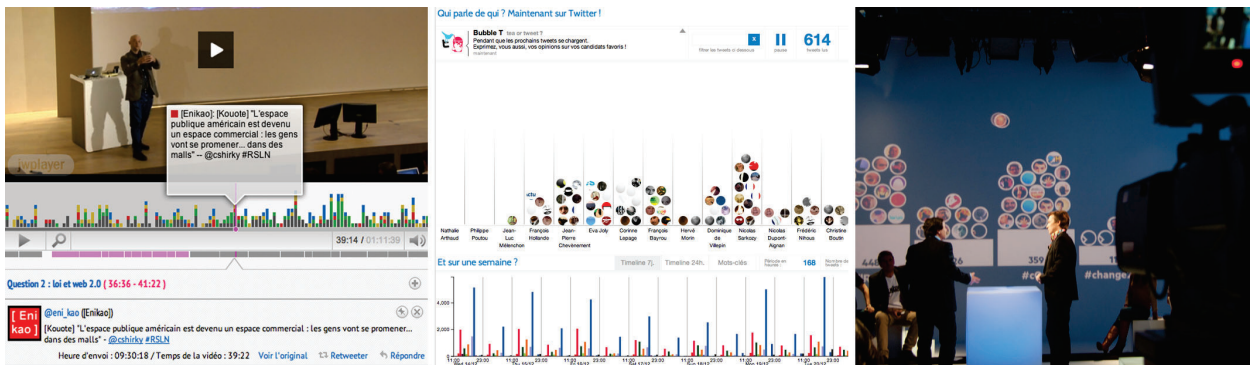


Figure 5.1: From Left to right: PolemicTweet metadata player (Chapter 4), Bubble-T webpage, Bubble-TV photography of the Tv show stage (Chapter 5).

In Chapter 3, I presented a set of historical examples of token-based dynamic visual representation. In Chapter 4 an initial token-based visualisation system, PolemicTweet, was presented. This system allowed the audience of an event to analyse tag and annotate the video stream from the event dynamically. As such, this system addressed **PROBLEM1.1**. However PolemicTweet partially addressed **PROBLEM1.2**: “How do we visually represent also dynamic data for this specific audience?”.

By analysing PolemicTweet, we ¹ were able to identify several limitations and sub-problems such as: (i) the abrupt update mechanism, (ii) the scalability of the visual representation (iii) the applicability to other contexts and (iv) the scalability to more than 300 people. The focus of this chapter concerns the two sub-problems that we characterized (Section 5.2), and the visual representation design solutions that we explored (Section 5.3, Section 5.5).

To explore the possible solutions for the two different sub-problems, we iteratively designed, deployed and studied two visualizations based on dynamic data streams similar to those used by PolemicTweet. The first design, Bubbel-T, addresses the problem of visually representing the update with a token-based approach. The second, Bubble-TV,



Figure 5.2: Additional resources:
<http://www.aviz.fr/Research/Bubble-tv>
<http://www.aviz.fr/Research/Bubble-t>

¹ Portions of this chapter were previously published in [Huron et al., 2013b]. Thus, any use of “we” in this chapter refers to Samuel Huron, Romain Vuillemot, Raphael Velt, Nicolas Sauret, and Jean Daniel Fekete.

addresses the problem of the scalability limit of token-based visual representation.

In this chapter I first describe the problems. I then describe our new designs, their contexts of deployment, and some observations. Finally I discuss the possible solutions provide by this design and there limitations

5.1 Introduction

Here we focus on two problems related to using dynamic visualizations based on tokens: (i) the mechanism by which the visualization is updated, and (ii) the scalability of the visual representation. To find visual mapping solutions to these problems, we conducted two short design studies, which we present in this chapter. Each of these design studies was meant to address one of the aforementioned visual representation problems.

The first design study focused on a visualization system called “Bubble-T”. Bubble T was designed to provide a live visualisation of what French people were saying on Twitter about presidential candidates during the 2012 French presidential campaign. With Bubble-T, we sought to create a token-based visualization system that addressed the visual update problem. We released Bubble-T for several months during the presidential campaign and it gathered significant positive feedback during this time. Bubble-T allowed one to visualize incoming data, but did not provide a solution to our second problem: how to represent past data in a dynamic token visualization system with a limited screen real estate, i.e. the scalability of the visual representation.

The reception of Bubble-T was so positive that the innovation service of the National TV broadcast contacted us to apply it to a Social Tv application. This provided an additional application case that we used to extend our design and work on the second problem. To this end, we designed “Bubble-TV” as a live visualisation for the tweets sent by TV viewers. The visual encoding used in Bubble-TV is similar to that of Bubble-T, but it provides a solution to the second problem. The visual encoding of Bubble-TV does so by combining classic visual marks for representing aggregated amounts of data and token-based visual representation.

In this chapter, we first describe the two problems, (i) and (ii), in greater details. We then provide details about the visualization systems that we designed to address these two problems, Bubble-T and Bubble-TV, and our results from our subsequent deployment of the two systems.

5.2 Problems

Visualizing streaming data, such as tweets, is challenging. In particular, the way to update the visual representation is nontrivial. The constant and unpredictable data updates make it difficult to allocate

an optimal portion of the screen to the visualization. This is compounded by the requirement that the visualisation should be able to reflect changes at different temporal scales while keeping the visual mapping continuous for the viewer, in order to support as tasks such monitoring.

5.2.1 *Emphasis and Smooth Visual Update of Incoming tokens*

In PolemicTweet description (Chapter 4), when a new tweet was available, the visual representation displayed a new square next to or on top of the previous one. This visualization updating process occurred without transition from the first visual state to the second visual state. Consequently, if the user was not paying enough attention during this updating process, he could have been unable to detect the new token and unaware of the corresponding change to the visualization. Therefore, this early design was a simple, but naïve approach to updating a visualisation of streaming data.

In their taxonomy for dynamic visualization, Cottam et al. [Cottam et al., 2012] explained that the ability to recognize an element as representing the same entity at multiple time points is important for comparing the representation across time. They use the term “identity preserving” to describe a visualization that exhibits this property. In a token-based system, each token represents a single element, so it is easier to keep track of a token’s identity over time. However, the appearance of new tokens into the display area should be identifiable so as to avoid incorrect interpretations in terms of the current state of the visualization, and to avoid confusion between incoming tokens and old tokens.

PROBLEM 1.2.1: This visual update problem can be summarized as trying to find an answer to the question: “How can incoming data in token-based dynamic visual representation be highlighted?”. To address this problem, we created and studied “Bubble-T” (Figure 5.3), which is described in Section 5.3.

5.2.2 *Scalability of token-based system*

Most visual representations based on area charts are scalable to different dataset sizes. This is because most of these visual representations can be adapted to different mapping scales. Here the term mapping scale describes the ratio between a given unit of area in an area chart and the amount of data that the unit of area represents.

By changing the mapping scale, a designer can increase the amount of data that is presented in a visualisation while simultaneously ensuring that the visualisation occupies a fixed amount of screen space. This approach to increasing the amount of data provided in a visual representation has already been utilized in *static* token-based visual representations by simply changing the token-to-data ratio. For instance, Otto Neurath used different types of scale in all his isotype charts (Section 3.3.2).

In a *dynamic* visual representation, the scale of a token while it is

being displayed on the screen could affect the perceived mapping of the token. This could in turn affect the user's ability to recognize the token's value at different points in time. Consequently, changing the mapping scale of a token in a dynamic visual representation, as a means to deal with scalability issues, would go against the recommendation of creating "identity preserving" visualizations. However, not changing the scale of the token would quickly lead to a screen real estate problem as shown in Figure 5.7.

PROBLEM 1.2.2: The scalability problem can be summarized as trying to find an answer to the question: "What is the appropriate visual representation for dynamic token-based visualizations such that these visualizations are scalable?". To address this problem, we created and studied "Bubble-TV", which is described in Section 5.5.

5.3 *Bubble-T: Monitoring Twitter during elections*

5.3.1 *Introduction*

During political election campaigns, different channels of information, e.g. newspapers and the political parties, provide information to citizens. Most of the time newspapers discuss the positions of the political candidates, provide the results of polls showing people's support for the different candidates, or cover related events. Political parties generally publicize their political agenda and candidates. In contrast, the social networks allow people to express their **own** perspectives on candidates in addition to spreading information from the two previous sources. In contrast to the abundant amount of information provided by these different sources, there are relatively few places where one can build an idea of what people are expressing about a candidate right now. In this section, we describe our experiences while designing and deploying Bubble-T.

5.3.2 *Context*

Tweets can provide an informative pulse of what is being said on various public subjects. Their quantity, rhythms, and conversational structures are ways to understand a general opinion despite the fact they are short messages. As discussed in Section 5.2.1, one challenging aspect of monitoring near real-time updates is that one needs to integrate the incoming data into the visual representation of data that has already arrived in some visual manner. This challenge could arise in different scenarios such as during a meeting or a TV show, but also during a major event like presidential elections.

What is Bubble-T ?

It addresses the challenges of updating a visualisation in real-time by using a particle simulation to fill a bar chart. Each particle is a new piece of information that we refer to as a "token" (Figure 5.5). In the current context, each Tweet is mapped to a token, and each column corresponds to a presidential candidate. Therefore, once a tweet is

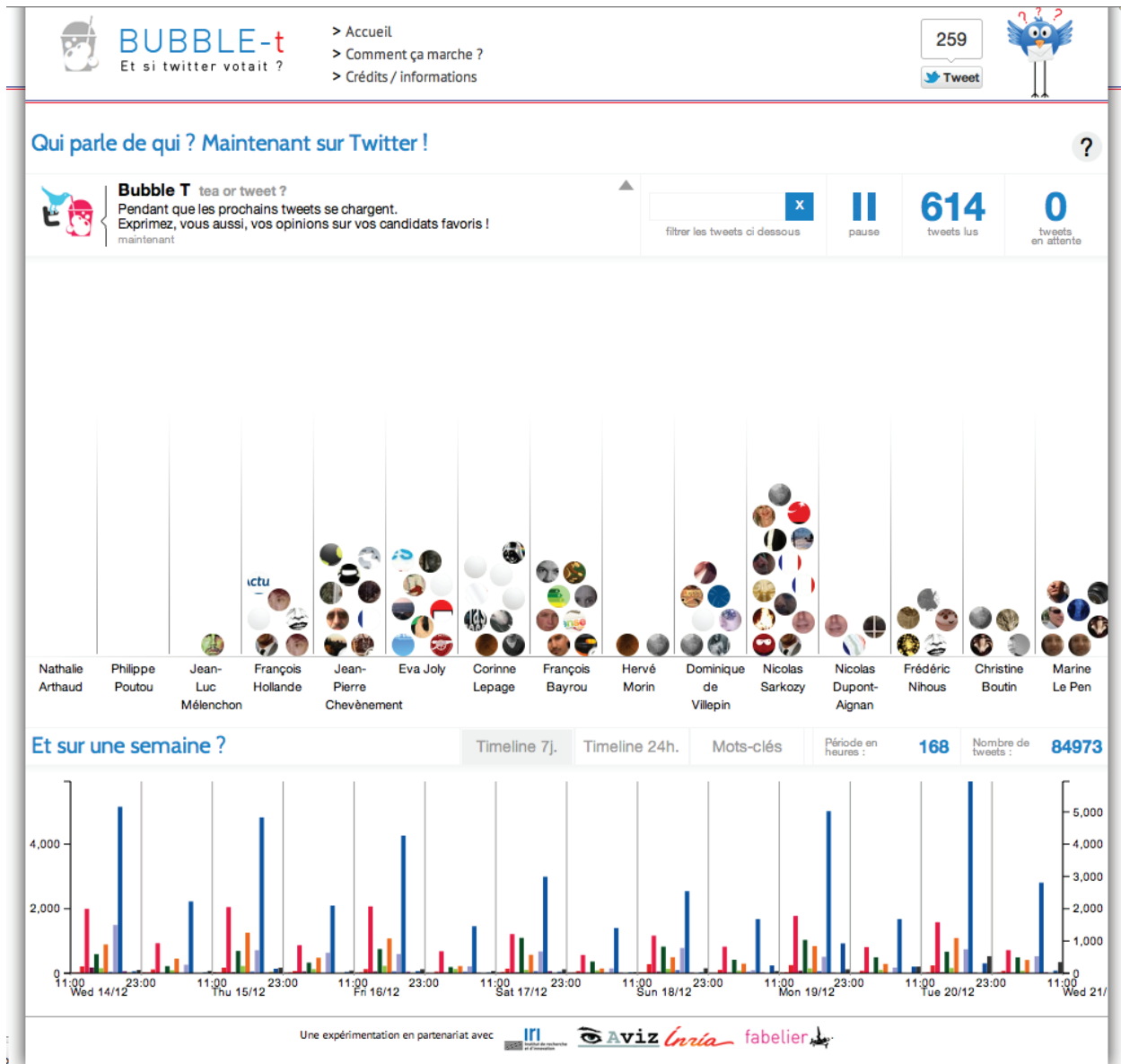


Figure 5.3: Bubble-T screenshot (21/12/2011), a video of the top of the interface is accessible here : <https://www.youtube.com/watch?v=TEfLV7fX07E>.

sent about a candidate, it is transformed into a token (Figure 5.4 E) that is thrown into the candidate's column (Figure 5.4 F,G). A physical simulation engine processes the throwing animation. The columns are progressively filled as they accumulate tokens. When a column is full, then tweets that first arrived are flushed. We mapped the Twitter user's avatar to the token for additional information. A static bar chart at the bottom of the visualisation captures all tweets during two different time periods, seven days and 24 hours (Figure 5.4 D).

Visual components

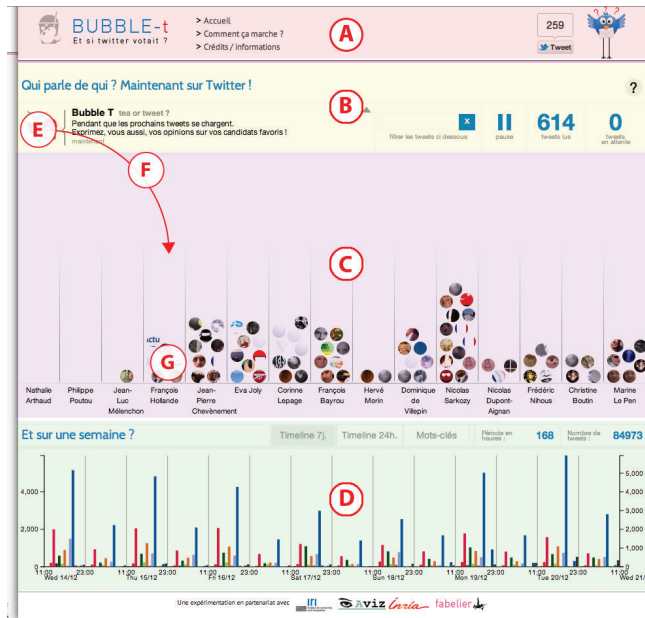


Figure 5.4: Bubble-T visual components: A. A header, B. A control Bar, C. A real-time visualization, D. Day and week time frame visualizations, E. Tweets read area incoming token appear here, F. An example of the trajectory of a tweet, G. A column containing a set of tweets. (Screenshot from the original website 21/12/2011).

The Bubble-T interface, is composed of 6 different components:

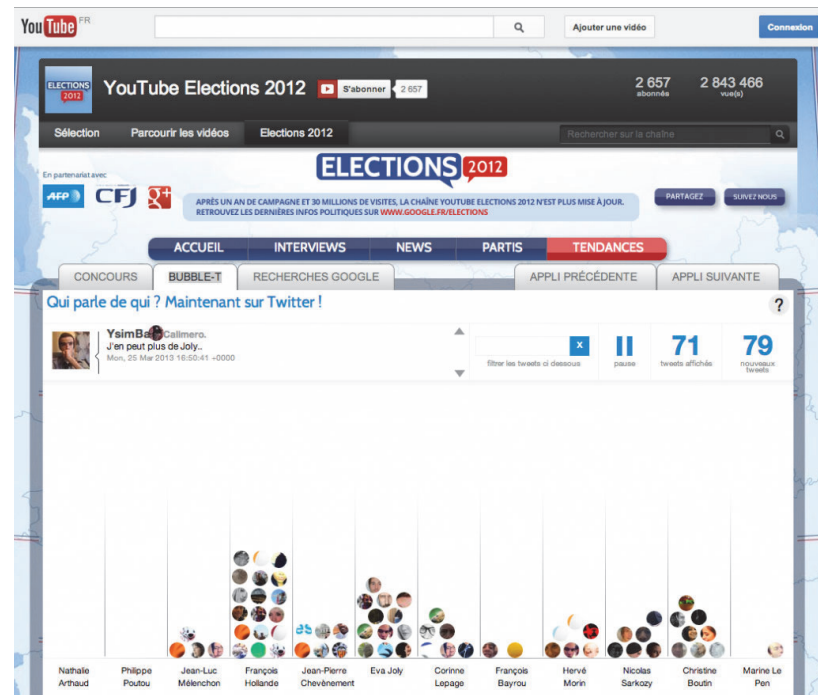
- A. **Header** The header provides a visual identity and a short menu that allows people to access to the explanation page, to get the credits, and to expose the tweeter sharing button.
- B. **Control bar** The control bar is composed of three areas: the tweet reader, which presents the incoming tweet and avatar before they are transformed into tokens, the search filter, a play pause button to pause the data stream if needed; a counter for tweets that have been read; and a counter for waiting tweets.
- C. **Real-time visualization** This space is divided into columns. Each column represents a candidate to the elections. Incoming tokens appear from the tweet reader area (Figure 5.4,E). Each token falls into the column corresponding to the candidate cited in the tweet (Figure 5.4,G). Each tweet could be a point and a tooltip will show the content of that tweet (Figure 5.5). Each column contains additional information through a tooltip to present the candidates.
- D. **Day and week time frame visualizations** This area allows one to switch between three different views: a 24-hour bar chart, a seven-day bar chart and a tag cloud representation.



Figure 5.5: Bubble-T: Rolling over on a token shows a tooltip containing the information about the corresponding tweet.

5.3.3 Interesting observations

Figure 5.6: Bubble-T embed into the official Youtube channel for the French presidential elections 2012.



Bubble-T received very positive feedback. It was selected among the finalists for the DataViz Google Elections 2012 competitions and hosted on the YouTube page dedicated to the French presidential elections (Figure 5.6). Also 1800 tweets referenced the URL when it was published (not the YouTube one), and the average visit duration on our server is 4:21 minutes, which is far longer than similar pages [Mediametrie, 2012]. According to Google analytics, up to 81 273 unique visitors browsed our website in the six months during the presidential elections, and this does not even take into account the YouTube version of the application. Moreover, several national media wrote articles about Bubble-T on their web version like LeMonde.fr [LeMonde, 2012], Radio France [Martin, 2012], 01.net [le Bourlout, 2012]. Also we collected comments on Twitter from various people, all were positive and people used words such as “Nice”, “excellent”, “hypnotic” to describe the visualisation. We also get some explicit direct feedback on Twitter from domain professionals. For example, this digital project manager from the French Senate’s broadcast team tweeted, “Bravo for the Bubble-T, the interface is innovative and relevant.”.

Finally, one interesting and unexpected behaviour that we observed was that when the visualisation was publicly presented, the audience in the room sent tweets, not only to test the system but also to show or see their own avatar inside the chart. We interpreted this behaviour as a potential engaging property of this system. Despite the public success of this visualisation, we observed that the visual updating process for Bubble-T, based on a physical simulation of tokens, seemed to be easy to read and sufficiently emphasized the incoming elements.

Moreover, as a solution to [PROBLEM1.2.1](#), Bubble-T seemed to be appealing and engaging.

5.3.4 Limitations

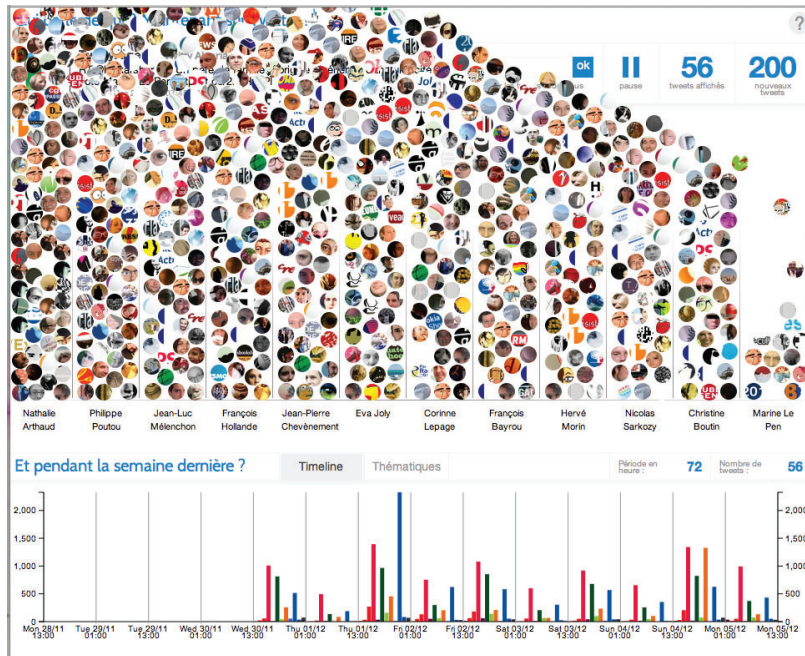


Figure 5.7: Tokens would overflow due to a bug in the token disappearance script.

Despite our success with applying animations and physical simulations to a token-based dynamic visualization, we did not solve the scalability problem [PROBLEM1.2.2](#) in Bubble T. In particular, the time period represented in the token-based bar chart is limited to a rather short window of time due to the limited amount of screen real estate that we chose to dedicate to this visualisation . Figure 5.7 shows what could happen if the columns were not flushed frequently.

5.4 Bubble-T Summary

Bubble-T was designed in order to response problem [PROBLEM1.2.1](#) “How can incoming data in token-based dynamic visual representation be emphasized?”. To arrive to a solution, we combined particle physical simulation with a visual representation of data based on tokens. Each token is bound to a particle with physical properties and then the physical simulations pilot its spatial position. This simulation allows one to create predictable animations for new tokens appear into the scene. This visual representation design received positive feedback from people on Twitter as well from French national media. However, Bubble-T, our solution to problem P1.2.1, is not a scalable visual representation, i.e. it does not solve [PROBLEM1.2.2](#). In the next section, we will present “Bubble-TV”, which responds specifically to [PROBLEM1.2.2](#).

5.5 *Bubble-TV: Adapting Bubble-T for TV show*

5.5.1 *Introduction*



Figure 5.8: The stage with Bubble-TV as a background.

Live visualizations connected to Twitter are an opportunity to engage television (TV) viewers by providing them with visual feedback, social awareness and the possibility to interact with others. In this section, we describe our experiences while deploying Bubble-TV, an extension of Bubble-T. Bubble-TV was intended to be a live visualization of TV viewers' Tweets. It was used as a background for the French TV show: "Le grand Webzé". Specifically, Bubble-TV allowed the hosts of "Le grand Webzé" to explore and comment on the social activity that took place around the show. In addition, we found that Bubble-TV impacted the viewers' experiences, since half of all the viewers who Tweeted actively used the system guidelines that were given to them.

5.5.2 *Context*

What is Bubble-TV?

Bubble-TV is a dynamic visualisation that encodes Tweets as small tokens called *bubbles*. Similarly to Bubble-T, the motion of the tweets is determined using a physical simulation. Tokens continuously fall from the top of the screen and fill up the bins of a bar chart. After a small amount of time, each bin shows the total number of

Tweets (Figure 5.8), divided into two areas: 1) the tokens representing tweets that have been sent in the last 30 seconds, and 2) an aggregate representing older tweets, which is updated every time a bubble disappears for being more than 30 seconds old.

This offers a double reading: latest tweets are expressed by the density of tokens, and the total count by the bar chart beneath. Thus one can easily visualize social networking activities. This visualisation technique has already been used to monitor real time evolution of 2012 French presidential candidates' popularity on Twitter.

For which live TV show?

"Le Grand Webz  " was the live TV show for which Bubble-TV was deployed. While in production, "Le Grand Webz  " aired on "France 5", a national public TV channel,. Moreover, each month there was one instalment of the show that would air at 11 pm on a Friday and would last 1h30min "Le Grand Webz  " was presented by two hosts and a community manager, and had an average of 115 060 viewers with its peak number of viewers being 708 000. The main goal of "Le Grand Webz  " was to promote emerging talents on the Web (YouTubers, bloggers, ...), which is why Bubble-T [Huron et al., 2012] was picked up by the show. The episode in which Bubble-TV appeared aired on February 24th, 2012.

Design and deployment constraint for the show

TV shows have a well-established complex production chain. Many people are involved: scenarists, directors, producers, TV hosts, guests, and in our particular case, a community manager. The first preparation meeting we had with the TV show team was held 45 days before the show in which Bubble-TV was to appear, so the design and the deployment of the project were quick. We designed and deployed Bubble-TV in the following three steps:

1. **Preparation of the TV Show (Weeks before).** We discussed with the scenarists and the directors when Bubble-TV would appear on screen, when the TV hosts and community manager would have to introduce it, what role the visualisation would have, and what guidelines would be given to viewers. From this discussion, we designed and developed the visual mapping and the user experience of Bubble-TV.
2. **Before the TV Show (Hours before).** We set the system up on stage and in the video control room (Figure 5.9). We explained to the community manager how to interact with the visualisation feature by showing her how to point at and display details of an individual Tweet.
3. **During the TV Show.** We gave the community manager, TV hosts, and directors autonomy. They followed the initial script we agreed upon, but had room to improvise (Figure 5.10).

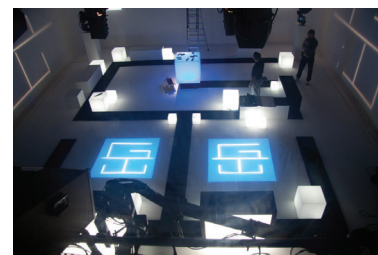


Figure 5.9: The building and preparation of the stage before the show.



Figure 5.10: The directors from the studio, authoring multiples videos sources during the show.

5.5.3 Design

The scenarists and the directors of Bubble-TV episode of “Le Grand Webzé” were interested in actively engaging viewers using Twitter and including the viewers in the TV show itself. Traditionally, viewers would be actively engaged in a TV show by inviting them to be physically present during filming, by having them call into the TV show, through the use of SMS or via emails.

Scenario and User Experience

Scenarized and explaining Bubble-TV to a very diverse audience was not straightforward. The timing had to be precise. The description of Bubble-TV had to be short, easily understood, and repeated throughout the show both regularly as well as consistently.

Scenario: Instead of monitoring the whole Twitter activity, a specific scenario was chosen, which corresponded to a real time poll system would be used during the TV show. This is quite common, but is usually done with phone calls or SMS, which are hard to visualize collectively and expensive. At the beginning of the show, the speaker introduced Bubble-TV (Figure 5.11, duration 2min15). Next, the speaker asked the polling question to the public, “Does the second host have to be fired?” and provided them with four possible answers (#yes, #no, #hehastochange, and #whoisthisguy). The speaker then quickly explained how to send tweets with hashtags, and how to read Bubble-TV. During the show, the community manager made quantitative and qualitative reports (Figure 5.14, between 30 and 50 seconds longs). At the end of the show, the hosts and the community manager discussed the final results of the poll; they ended up keeping the second host.



Figure 5.11: Speaker present and explain how the visualisation will work.

Audience Instruction: During the broadcast, the hosts commented on Bubble-TV as if it were a weather forecast with the difference that they had to introduce what a Tweet was and its meaning in the context of the question being asked. They had to repeat both questions and instructions to engage in Bubble-TV frequently as the show was quite long and people could join in at any time. The community manager reported several interesting Tweets and interacted them live.

Visual Mapping Design: For the TV show, tweets had to be mostly presented on TV in aggregated forms, i.e. total numbers of Tweets for a given answer to the question, or one Tweet at a time. In contrast, the original Bubble-T visual representation showed a group of tweets as an assembly of individual token without providing an aggregated view. The TV show was a perfect application case to investigate our scalability problem, P1.2.2. According to the poll scenarios, we thought that it would be preferable to have the both views, token and aggregated, represented in a dynamic way. The TV show’s team

wanted the token view to engage Twitter users because it indicated the source of every Tweet based on individual avatar face of each token. They also wanted an aggregated view that could provide an idea of the polling trends overtime. Our approach to solve this problem was to combine an area chart with a token-based chart. After several iterations of possible combinations, we chose to use the visual constrains of the area chart to simulate physical constrains on token trajectories. The rationale to our design was to keep the area chart visual identity as well as the token identity safer as possible by avoiding: *i*) occlusion: not have a visual mark on top of a other, *ii*) layout identity: not have a visual mark entering a space that is already structured for reading without following it's constraints, *iii*) predictability: having always the same type of token movements.

Different interfaces: Bubble-TV was created iteratively to make it compatible with the TV format and the different roles of each of actor on the TV show, e.g. the hosts and scenarists. Stability was a prime issue since no crashes, or reboots were allowed, and we had to ensure that the system would not freeze. This design process resulted in the development of three different interfaces dedicated to the viewers and hosts, the community manager and the producers. These were:

- **The stage visualization** (Figure 5.8), which used the Bubble-TV visualisation introduced earlier. It presented Twitter activity in real time.
- **The community manager interface** (Figure 5.12), which was similar to the stage visualisation but fit for a computer screen, and augmented with interactive features used to monitor, read, select, and display Tweets. This interface also controlled the visualisation on stage.
- **The tweets counter** (Figure 5.13), which allowed the viewers, hosts, and producers to know the exact number of tweets throughout the show.

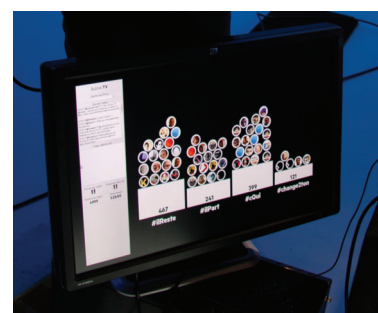


Figure 5.12: The community manager screen control.



Figure 5.13: The Tweet counter embedded in the stage furniture.

5.5.4 Interesting observations from the TV show

During the approximate 20 minutes (out of 1h30) Bubble-TV² was visible and in focus Figure 5.8, a series of unexpected events occurred. For instance, one of the two hosts voted on Twitter to fire the other (Figure 5.14). At one point, the visualisation suddenly refreshed while a host was pointing a specific bubble making bubble disappear. At another, one of the hosts made a connection between an avatar and a person in the audience.

After the TV broadcast

During the show in which Bubble-TV was deployed, twice as many Tweets were sent compared to the previous one (4658 versus approximately 2000). 23% (1094) of the tweets were sent using the

² A montage of interesting sequences is available here (in French): http://www.youtube.com/watch?v=jn9ief50_lgQ

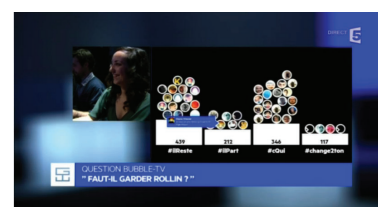


Figure 5.14: The community manager reporting Twitter activity.

answer hashtags, and the viewers sending these tweets accounted for 518 out of the 1268 active Twitter users. We believe that this high level of adoption, which was 40% of the Twitter accounts sending tweets related to the TV show, is linked not only to the polling scenario that occurred during the TV show, but also to viewers wanting to see their avatar on TV. Overall, we received positive feedback; we were congratulated by the producers, viewers and broadcaster of the show. Unfortunately, *Le Grand Webzé* has since been cancelled for casting issues. However, Bubble-TV has since appeared in the France Television 2012 innovation teaser for programs, and is still under consideration for further use.

Bubble-TV summary



Figure 5.15: The state of the visualisation few minute before the end of the emission.

Our experiences with Bubble-TV (Figure 5.15) indicate that social networking activity could be highly impacted by the experience provided by TV broadcasting. We believe that live social network interactions driven by a broadcaster will enable tasks such as live tagging. Indeed, people seem motivated to tag their own tweets if their opinion is asked for [Huron and Fekete, 2012]. This demonstrates that similar scenarios to that of *PolemicTweet* (Chapter 4) could be applied to even larger public groups such as 700 000 viewers in the case of Bubble-TV. The benefits of this include providing real-time feedback and accuracy that other systems may not deliver. It would also provide quantitative and qualitative data from which broadcasters could benefit. For example, such data could be used for sentiment analysis, video annotation, or specific metadata collection for future applications. We believe visualizing tweets in real time will have an important role to play. However, such visualization will have to be carefully designed and used in well thought scenarios.

It seems that our visual mapping was successful for the goals

of Bubble-TV. This visual mapping has the advantage of providing detailed and aggregated views of the data while emphasizing the dynamic arrival of each new datum, i.e. tweet in the case of Bubble-TV. The Bubble-TV visualisation was a major improvement on Bubble-T as the design of Bubble-TV allowed us to encode the total number of tweets in different categories while also providing the latest tweets to the viewers. This visual representation enabled us to solve the scalability problem, [PROBLEM1.2.2](#), define in previous Section [5.2.2](#). The combination of two visual systems, token charts and area charts, seems working with this constrain. The Bubble-TV visualisation provided a first step towards creating visualizations that incorporate both a token-based system and area chart for dynamic visualization. However, the major limitation of Bubble-TV's visualisation was its generalizability. Bubble-TV's visualisation was just one instance of an infinite number of combinations of token-based systems and area charts that one could use to make visualization. A framework to guide the generation of this new style of visualization, which combines token-based charts and area charts, would be useful.

5.6 Chapter Summary

One of the challenges considered in this dissertation is *How can dynamic data be visually represented for non-expert* [PROBLEM1.2](#). In this chapter, we presented two visual encoding sub-problems related to [PROBLEM1.2](#) and our token-based approach to visualizations. These problems were: [PROBLEM1.2.1](#) "How can incoming data in a token-based dynamic visual representation be emphasized?", and [PROBLEM1.2.2](#) "What is an appropriate visual representation for dynamic token-based visualizations such that these visualizations are scalable?".

During this chapter, we presented two visualisation solutions designed specifically to address these visual encoding problem. In our solution to [PROBLEM1.2.1](#), we introduced animation and the use of *aphysical simulation* to represent data updating process visually in a token-based system. In our solution to [PROBLEM1.2.2](#), we combined our previous design that solved [P1.2.1](#) with the use of *visual area to represent aggregate data*. This combination allowed us to represent incoming data as well as past data. Our visualisation solutions to [PROBLEM1.2.1](#) and [PROBLEM1.2.2](#) allowed non-expert audiences to monitor Twitter feeds for different tasks and in different contexts. Both of them met with great interest from the general public. Both Bubble-T and Bubble-TV received positive feedback and the attention of a large number of people (10 000 to 700 000). Finally, the two improvements to our token-based approach of dynamic visualisation seem to have been adopted by a large group of people. These improvements were (i) the use of physics-based animation for updating the visualization, and (ii) and the transformation of tokens to visual area of aggregated data. However, despite the success the visualizations presented in this chapter, Bubble-T and Bubble-TV are not

generic, but based on specific application case. Therefore, we have only partially solved our problem [PROBLEM 1.2](#). In the next chapter, we will present a metaphor and a toolkit to unify our approach to using tokens for dynamic visualization.

Visual Sedimentation - a Generic Token-based Visual Encoding for Dynamic Visualization

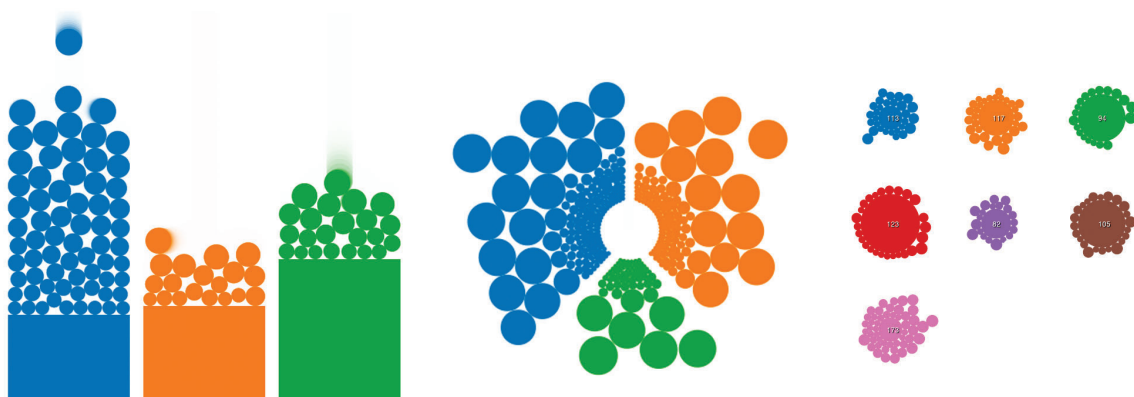


Figure 6.1: The Visual Sedimentation metaphor applied to a bar chart (left), a pie chart (center), and a bubble chart (right).

Chapter 4 focused on how one can engage people in dynamic data analysis through the use of a token-based visualisation system. In Chapter 5, two case studies were presented, which were dedicated to determining an appropriate visual encoding for dynamic data. The first case study was designed to explore how one can use tokens and physical simulations to represent the data updates that occur in a data stream in a visual and smooth manner. The second case study focused on how to increase the amount of data that could be represented using token-based visualizations; this was achieved by combining a token-based visualisation with a classic bar chart.

The visualisation techniques developed during the two aforementioned case studies received enthusiasms, but the techniques were created for specific applications and were consequently not generalizable.

In this chapter, we ¹ study and group the specific design features into a more generalizable visualisation technique for dynamic data. In order to achieve this, we present “Visual Sedimentation” as a metaphor that allows us to define and describe a design space that



Figure 6.2: Additional resources: A dedicated website was made for this technique. This website has source code, documentation, videos and demos for Visual Sedimentation:
<http://www.visualseedimentation.org>

¹ Portions of this chapter were previously published in [Huron et al., 2013c]. Thus, any use of “we” in this chapter refers to Samuel Huron, Romain Vuillemot and Jean Daniel Fekete.

encompasses both “Bubble-T” and “Bubble-TV”. This metaphor and design space allows us to achieve visual updating in classical area chart visualizations such as bar charts, bubble charts and pie charts. Visual Sedimentation also provides a framework within which we can extend token-based visualizations further by generating new visualizations based on token assemblies aggregating area charts. In this chapter, we discuss our exploration of the design space described by Visual Sedimentation, as well as the generation of new visualizations using Visual Sedimentation.

6.1 Introduction

This chapter introduces *Visual Sedimentation*, a novel design metaphor inspired by the physical process of sedimentation. This process is the result of objects falling due to gravitational forces and then aggregating over time into compact layers. We show how the physical properties of the sedimentation process can be applied to the design of effective visualizations of data streams through metaphor: new items are the equivalent to falling objects, animated by virtual forces, and aggregating over time in area charts.

Data streams are sequences of typed objects, and are very common with social networks updates, tweets, emails, network logs, RSS feeds, or updates in distributed version control systems. The challenges of visualizing data streams match many characteristics of the physical sedimentation process: data *appear* at unpredictable times, *accumulate* until they are processed, and need to be kept in *aggregated* form to provide historical and contextual information over time. Designing visualizations to convey those various stages is not straightforward as many visual representations may have to be displayed simultaneously.

We are specifically interested in smoothing the transition between the data stream’s focus—recent data—and the context—older data.

In this chapter, we define the sedimentation process as it appears in physical environment, followed by an explanation of how it applies as a metaphor for data streams visualizations. To facilitate the design of visualizations using this metaphor, we implemented a supporting toolkit; we applied this toolkit to design and implement case studies of both classical and novel visualizations based on real-life datasets.

We then explore the Visual Sedimentation design space by modifying the toolkit parameters and creating a group of variant visualizations. This process of deconstructing the design space has provided us with creative results that we present. We finally address technical issues related to the implementation of the Visual Sedimentation metaphor before discussing future works.

6.2 Sedimentation



Figure 6.3: Illustrations of real life sedimentations process. Athabasca glacier (left), Mississippi riverbed on a map [Fisk, 2005], water riverbed on a Glacier, water surface oscillation, Hoodoo Bryce canyon, Types of sediments [globalspec.com, 2013] (right).

Sedimentation is commonly understood as the deposition of a solid material from air or water. Biologists define it as “the tendency for particles in suspension to settle out of the fluid in which they are entrained, and come to rest against barrier” [Wikipedia, 2013]. Geologists extend this definition to “deposits from glacial ice and those materials collected under the impetus of gravity alone, as in talus deposits, or accumulations of rock debris at the base of cliffs” [of Encyclopædia Britannica, 2013].

Many factors contribute to the sedimentation process; we focus here on several elements that are applicable to our metaphor. The central element of the sedimentation process is the **sediment** itself: rock, dust or particles that vary in composition, size and weathering stage (Figure 6.3, center). External **physical forces** are applied to sediments transporting them according to the forces’ directions and magnitudes. Furthermore, depending on weathering and collisions, sediments may split or become compressed. The process that compresses sediments over time is called **decay**. Sediments end their journey as they accumulate on each other and on **barriers**, settling and aggregating into sedimentary rocks; this process is called **flocculation**. Sedimentary rocks may separate into distinctive layers or **strata**, with the oldest strata on the bottom and the youngest on the top. These strata reflect the different nature of sediments, time periods, or compression over time with **deformations** in organic or liquid-like shapes.

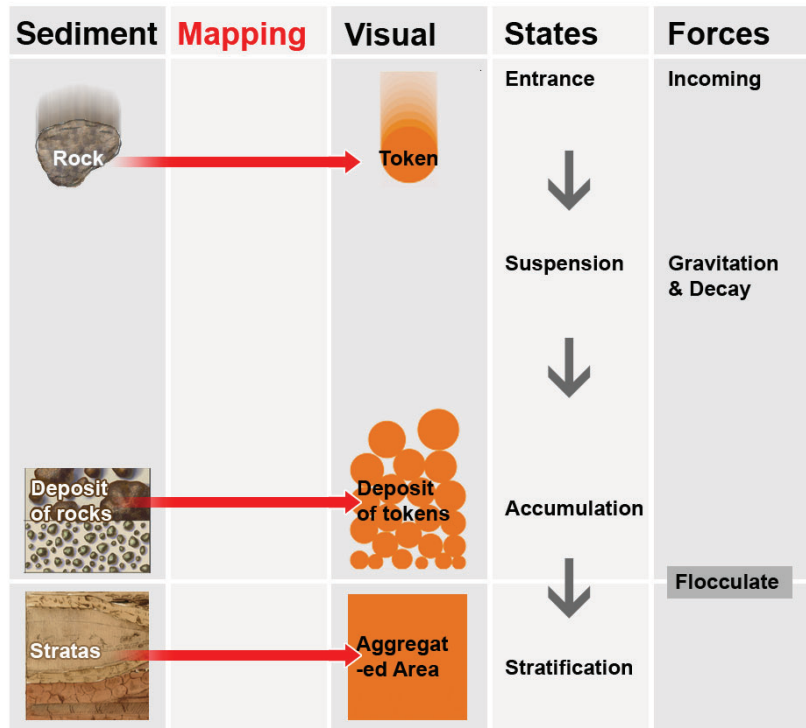
As sedimentation is part of a global physical environment, physical forces also play a role in this process and the deposit of visible traces. For example, glaciers leave **trails** of heterogeneous sediments in their wake. Similarly, riverbed deposits show historical meanders in a river’s floodplain (Figure 6.3, right); finer grain sediments deposits in suspension visually mark the maximum water level, an example of naturally-occurring **contours**. Fluids may also leave traces of past interactions between its flow and the sediment deposits with **ripple** marks. Sedimentation not only leaves visual cues or traces of the past, but also captures snapshots of living ecosystems through fossils.

Humans are familiar with the visible parts of the sedimentation process, such as mountain formations, dunes or riverbeds occurring at the surface of the Earth. The underground portion of sedimentation

(i.e. strata) previously had only been revealed to us as exposed cliff faces. During the industrial revolution, digging and coring techniques were developed that brought sedimentary layers to the surface. It was not until the late eighteenth century that James Hutton revealed a complete understanding of the laws that govern the formation of sediment on the Earth [Hutton, 1788]. Strata became a familiar concept when cut representations in cartography became incorporated into our standard educational material. Our work builds upon the general popular’s widespread understanding of sedimentation.

6.3 Visual Sedimentation

Figure 6.4: Summary of the different states of tokens in the Visual Sedimentation metaphor.



Visual Sedimentation applies the metaphor of the process of sedimentation to the visual representation of data streams. The following section explains how sedimentation concepts can be mapped to their visual counterparts, particularly: tokens (corresponding to sediments), layout, physical forces, and aggregated areas.

6.3.1 Tokens

Similarly than in our previous design studies (Chapter 4: page 77 ; Chapter 5: pages 94 and 100) a *token* is a visual mark [Bertin, 1977] representing a data item arriving from the stream. For the purposes of this chapter, each data item appears at a particular time t , and carries attributes (e.g. size, weight, texture); one of these attributes is categorical and is considered as the token’s category. The token may be as graphically rich as needed in order to visually express the

data it represents. Each token undergoes four *states* during its lifetime (Figure 6.4):

- *entrance* into the scene from an *entrance point*;
- *suspension* while falling to the ground;
- *accumulation* on the ground or on top of previous tokens;
- *decay* and finally merging or *flocculating* into an aggregated areas.

The token's visual appearance can change during its lifetime to reflect its current *state*, but it should remain recognizable and identifiable through some consistent visual encoding to clearly keep its identity or *category* until it flocculates.

6.3.2 Layout

The *layout* roughly corresponds to a two-dimensional geological cross-section of strata layers in which the tokens are shown in their multiple forms. The layout is comprised of four components that satisfy these constraints:

- *Walls* are separators used to control tokens' trajectories and to separate tokens according to possible categories while maintaining their arrival order visually.
- The *Ground* is the baseline or barrier on which tokens accumulate and generate deposits over time.
- *Aggregated areas* lie under the ground and reflect the history of tokens previously accumulated underground.
- *Containers* are compound physical elements made of walls and ground. They are semi-closed shapes that contain categorized tokens.

6.3.3 Forces

Physical forces specify the movement and behavior of tokens, for example their trajectories and reactions to collisions with other tokens or obstacles. They indirectly impact the tokens' states as their position changes and form decays. Properties of physical forces are specified as follows:

- *Gravity* moves tokens from their entrance point to the ground; it is probably the most intuitive and predictive type of force. Furthermore, token dynamics under gravity forces can be influenced by friction, such as fluid viscosity or dry friction.
- *Decay* simulates the weathering of a token. It is effectively an aging function that begins to work once tokens start to accumulate. The application of this force can produce different effects on tokens, including shrinking, flattening, or darkening.
- *Flocculation* is the transformation of individual tokens into a continuous aggregated area. This transformation can be rendered with various visual effects.

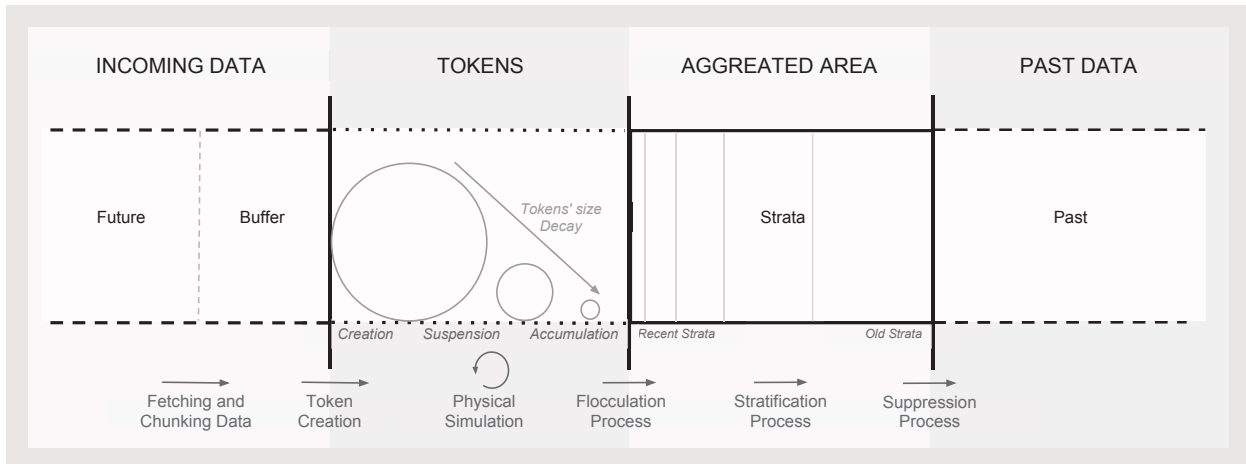


Figure 6.5: Summary of the design space. Time goes from right to left: 1) Data streams chunked and stored in a buffer for further reading, 2) Tokens appear from an entrance point, 3) Physical forces transport and shrink tokens until they flocculate, 4) Strata are tokens aggregated into layers.

6.3.4 Aggregated Areas

The final state undergone by tokens is *aggregation* with previous tokens into *strata* (Figure 6.4, bottom). *Stacked layers* represent a period of time, with a height proportional to the number of flocculated tokens. Layers are stacked but visually separated by a line or a different visual encoding. *Deformation* is caused by physical forces that change the shape and thickness of the layer, and its visual representation (e.g. colour or texture).

6.4 Visual Sedimentation Toolkit

To facilitate the exploration of the Visual Sedimentation design space, we implemented an open-source toolkit, written in JavaScript to run in modern web browsers. It provides templates for using the common charts described in the next section. For example, the bar chart on the left in Figure 6.1 can be programmed with the parameters details shown in Listing A.2. The toolkit reads parameters following the design space described in Table 6.1 to configure the Visual Sedimentation engine. The capabilities of the toolkit can be extended through a programming API to control the data sources, visual encodings, dynamic behavior, layout, and interactions.

The Visual Sedimentation toolkit relies on two existing libraries: the physics engine Box2DWeb [Catto, 2010] to manage force-based animation and collision, and the visualisation toolkit D3 [Bostock et al., 2011] for implementing the visual encodings of the aggregated areas and their deformations. Token management is handled by the physics engine in a dynamic fashion, while the aggregated areas are handled using D3 according to the strata model and created once for all when the visualisation starts.

The Visual Sedimentation toolkit relies on tokens being discrete,

however datastreams can carry continuous data or discrete high volume-rate data produced at high-speed, both of which can be unsuitable for most types of visualization. To use Visual Sedimentation with such data streams, the programmer will need to first split the continuous or fast stream into discrete chunks that the user will be able to see and understand. Furthermore, when the data flow rate is irregular or bursty, it may need some regulation; the toolkit provides some pre-processing facilities to smooth the stream throughput.

The toolkit's parameters are summarized in Table 6.1. They can be set individually to explore numerous design variations. To help designers get started, we predefined a series of templates that can be loaded by setting the chart type to type: 'StackedAreaChart' for bar charts. Other charts (pie and bubble charts) can be loaded in the same way and then be customized to meet the designers' needs.

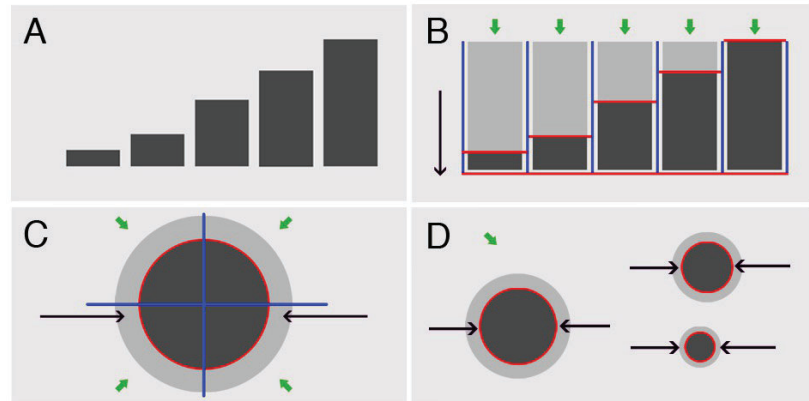
However, not all parameter values will produce visualizations compliant with the sedimentation metaphor. Parameter setting should be done carefully since parameters may depend on or influence each other. For example, changing the incoming point position will impact the accumulation location, forcing tokens to land on the wrong part of a chart. To ensure that the resulting visualisation will comply with the metaphor, designers should observe the following high level guidelines that are not enforced by the toolkit:

1. The accumulation area and the aggregated area should look continuous at their interface;
2. The trajectory of tokens in the accumulation area should be consistent with the orientation of the aggregated areas (the perceived "time direction"), at least locally around their interface;
3. The encoding of strata sizes in the aggregated area should decrease monotonically starting at the interface following the "time direction";
4. To support the metaphor, the flocculated area and the aggregated area should be visually consistent around the interface.

(t) TOKEN	(a) AGGREGATED AREA	(f) FORCE	(l) LAYOUT
1 - Visual encoding	1 - Geometric primitive	1 - Gravity	1 - Wall
2 - Incoming point	2 - Stacked geometric primitive	2 - Decay	2 - Ground
3 - Trajectory		3 - Flocculation	3 - Container
4 - Target	3 - Trail		
5 - Impulse	4 - Pixel		
6 - Physical parameters	5 - Isocontours		
	6 - None		

Table 6.1: Parameters of the Visual Sedimentation Toolkit

Figure 6.6: Template schemas: Blue lines represent the *walls*; red lines the *ground*; green arrows the *entrance point*; black arrow the *gravity* direction; light grey shapes the *deposit area*; and dark grey shapes the *aggregated area*.
 (A) A common bar chart and (B) its template; (C) the pie chart template; (D) the bubble chart template.



Listing 6.1: Bar Chart implementation based on a template

```

1 mySettings = {
2   width: 300, height: 300,
3   chart: {
4     type: 'StackedAreaChart', ...
5   }
6   data: {
7     model: [
8       {label: "Column A"}, ...
9     ],
10    strata:[
11      [ {initValue: 100, label: "Bar A"} ], ...
12    ],
13    stream: {
14      provider: 'generator', refresh: 10000/4
15    },
16  }, // data:
17  sedimentation: {
18    token: {
19      size: {original: 6, minimum: 2}
20    },
21    aggregation: {type:"stacked"},
22    suspension: { decay: {power: 1.01} },
23    flocculation:{...}
24  }, // sedimentation:
25 };
26 var barChart = $("#myDivChartContainer")
27   .vs(mySettings);

```

6.5 Case studies

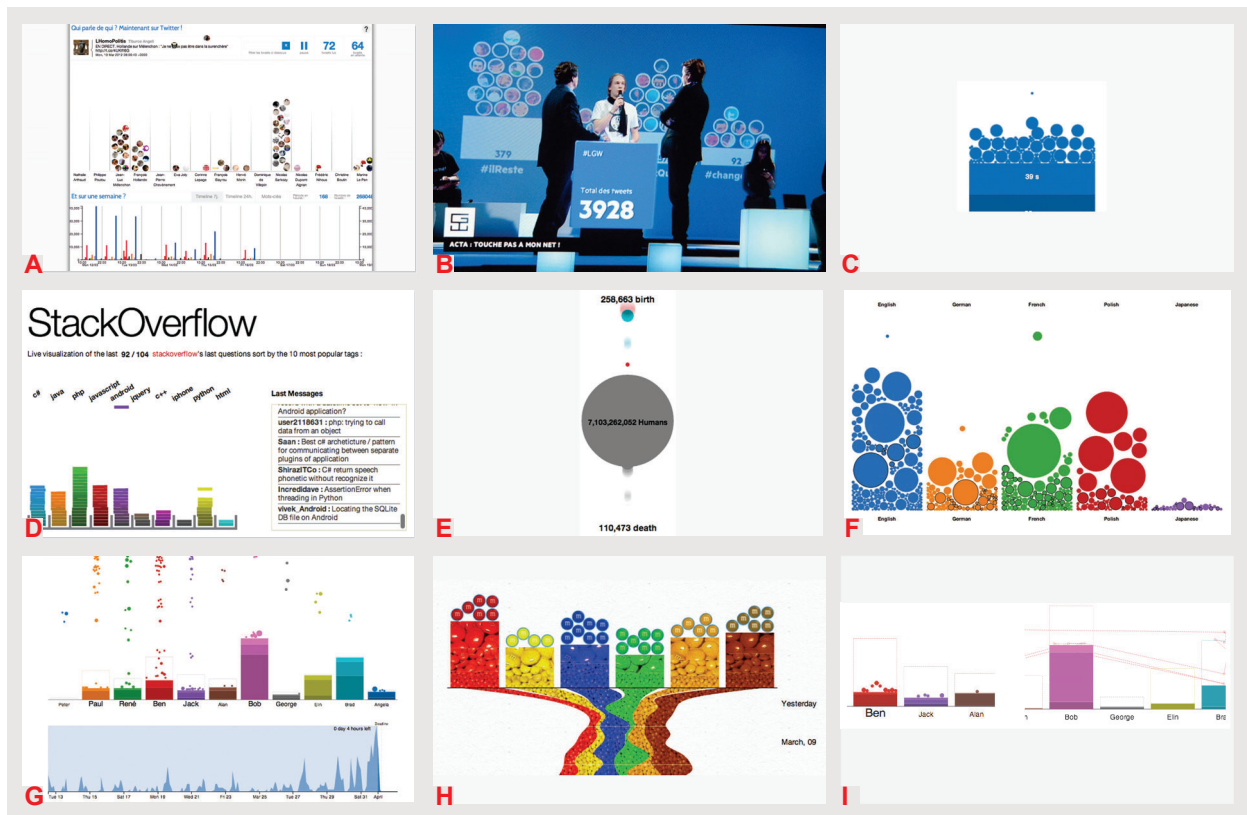


Figure 6.7: Summary of the case studies: (A) Bubble-T seminal applications.; (B) Bubble-TV.; (C) SediClock: a hybrid between an hourglass and a water clock.; (D) Most popular questions asked on StackOverflow, the categories being the 10 most popular tags.; (E) Birth, death and population screen capture at 5h15 pm.; (F) WikiFlow.; (G) Edits of text and code before the deadline of VisWeek 2012 using SediVN.; (H) SediMMS.; (I) Label size reflecting the recent activity (bottom left) and lines between tokens showing changes on common documents in SediVN.

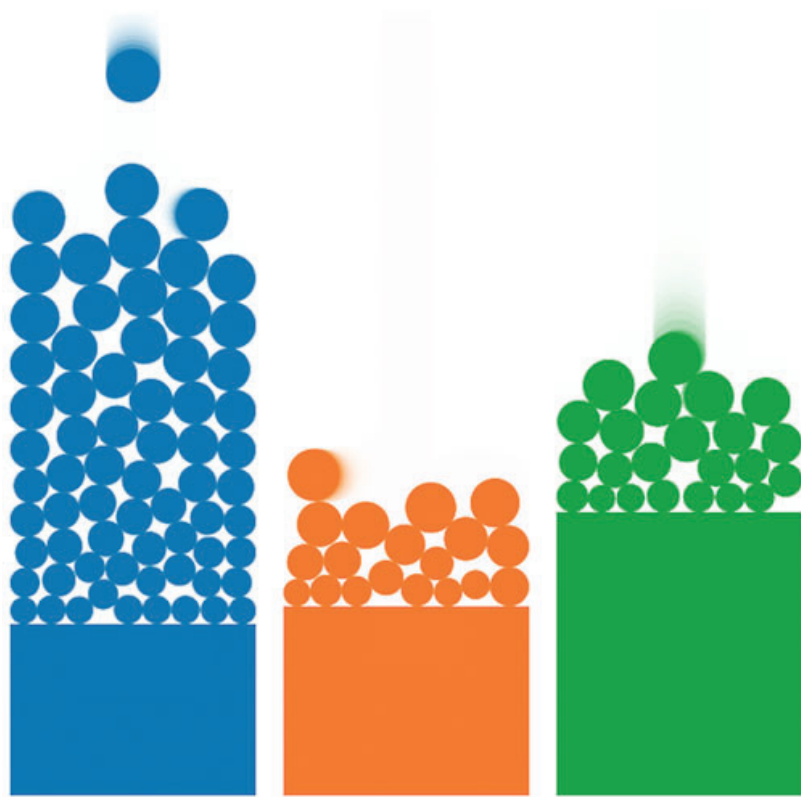
We implemented a series of examples and cases studies using the Visual Sedimentation toolkit to demonstrate which parameters combinations can be effectively used in real settings. The Visual Sedimentation toolkit not only facilitates the implementation of the sedimentation metaphor; it allows its exploration in a guided way by parametric exploration of the design space.

6.5.1 Common Area Charts with Visual Sedimentation

The toolkit supports the application of the Visual Sedimentation metaphor to common area charts such as bar charts, pie charts and bubble charts. We explain how these charts are specified and how to control their visual encoding.

Bar Chart

Figure 6.8: Bar chart with Visual Sedimentation.



A simple way to use a bar chart with Visual Sedimentation (see Figure 6.8, left), is to consider each vertical bar as an aggregated area, with a height proportional to the number of tokens stratified. Walls are added to separate the chart into bins, each containing a different category of tokens (Figure 6.6 B). The top of the bar becomes the sedimentation ground where incoming tokens will bounce and deposit before they flocculate. The upper area of the bin, above the bar, becomes a container which is to be filled up by incoming tokens. Tokens fall from an entrance point that can be right above each bin, moved by gravity. To accommodate entrance points and tokens in suspension, the maximum bar height should remain below the top of the bin containers.

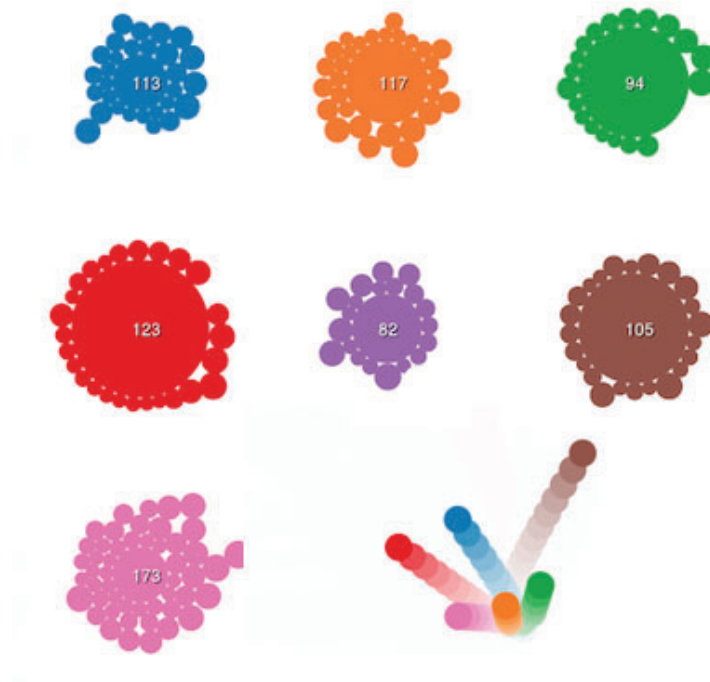
Pie Chart

Figure 6.9: Piechart with Visual Sedimentation.

Similarly, a pie chart with Visual Sedimentation (see Figure 6.9) considers each sector as an aggregated area, where the angle is proportional to the number of stratified tokens. Just as with bar charts, walls are added between sectors. The ground is represented as an arc in each sector that forms a container delimited by two walls crossing at the center. The arc radius should be small enough to leave room for entrance points. Tokens enter in the containers from an entrance point that can be in front of each sector, and move towards the center of the chart, as though tokens were gravitationally drawn to by the pie center.

Bubble chart

Figure 6.10: Bubble chart with Visual Sedimentation.



A bubble chart with Visual Sedimentation is slightly different than the other two types of area charts (Figure 6.10). There is one center per token category, organized in a grid. Aggregated areas are discs with a radius proportional to the number of tokens stratified. Bubble charts have no walls; the ground is the external circle of each disc where tokens are deposited. Tokens can enter from any point; they are gravitationally drawn to the center of the category to which they belong.

Note that for these 3 charts, the perceived size of the areas is slightly perturbed by the tokens that float outside and their non-linear aggregation over time. Value retrieval tasks cannot be performed accurately but comparisons between area sizes remain valid. If accurate value retrieval is needed, a corrected aggregated area can be drawn translucently over the deposit and aggregated areas.

SediClock

SediClock is a hybrid of an hourglass and a water clock (Figure 6.11). Like an hourglass, SediClock uses sand particles as tokens to depict seconds. Like a water clock, it simulates a liquid filling-up a translucent recipient. SediClock encodes minutes and longer time units as strata. Tokens representing seconds flocculate each minute; therefore the minute container is filled with the latest count of seconds from the starting time. In addition to second tokens, SediClock also shows time in zooms units using smaller tokens. Interestingly, these extra tokens have no consequence on the layout: they fill up the empty space left between the stacked second tokens when they fall in the container.

Real-Time World Population

Worldometers² is a website that provides real-time statistics on global issues such as demography, ecology, health, and economics, as counters updated in real time. Their real-time counters are streams of numbers using a specified unit, updated every millisecond; unfortunately the scale and context of their numerical data is difficult to understand without a proper visualization.

We used Visual Sedimentation to visualize the real-time world population, using the aggregated area to represent the living population. Unlike the previous examples, birth tokens enter and quickly flocculate to the circle representing the world population, while grey death tokens leave the disc, fall down, and vanish (Figure 6.12).

Due to the high frequency of updates, we do not use any accumulation; the speed of tokens during suspension conveys the scale of the counters clearly showing that the birth rate is higher than the death rate.

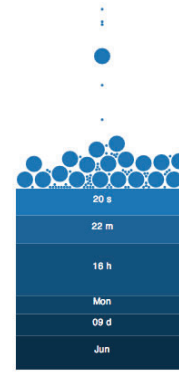


Figure 6.11: SediClock, Time Visualization screen shot 2014-06-09 at 04h22m20s. A snippet video is accessible at the following url: <http://youtu.be/jVjN4LM7qKc?t=1m34s>

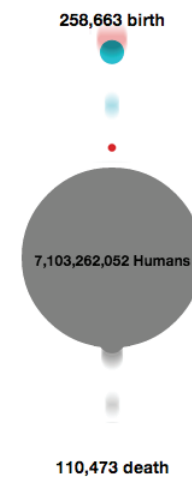


Figure 6.12: Birth, death and population screen capture at 5h15 pm.

² <http://www.worldometers.info/>

StackOverflow Questions Visualization

Figure 6.13: StackOverflow, Questions Visualization screen shot 2013-03-11 at 5h03 pm. A snippet video is accessible at the following url: <http://youtu.be/jVjN4LM7qKc?t=1m55s>

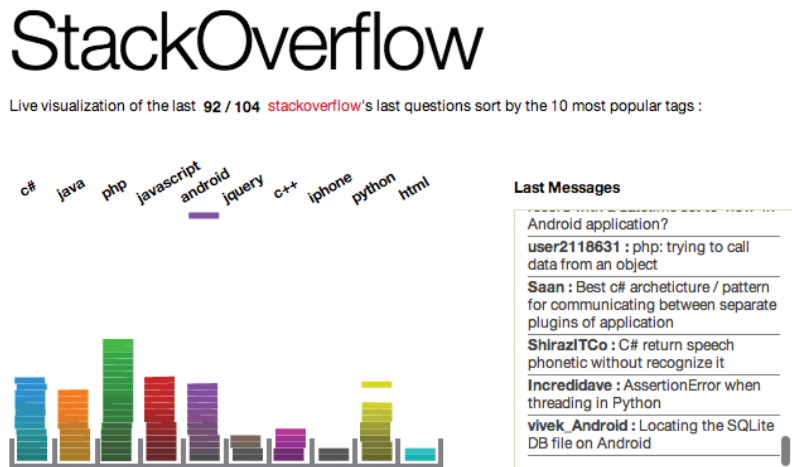


Figure 6.14: StackOverflow Logo.

³ <http://stackoverflow.com/>

StackOverflow³ is a popular website where users ask technical questions and get answers from the Web community; the questions are freely tagged by users for categorization and search [Stackoverflow]. Questions and answers flow in rapidly: as of the submission of this paper of this article, the ten most popular tags are associated with 428,345 to 160,826 questions and the site receives 20 to 500 new questions every day. Applying Visual Sedimentation to this stream, questions map to tokens and the 10 most popular tags. Our visualisation uses the classical bar chart layout (Figure 6.13). To visually match StackOverflow's logo (Figure 6.14), we use flat rectangles instead of circles as token shapes. This factor highly impacts the organization of the chart, because the stacking and grouping behaviour of squared shape are radically different than rounded shape. This force us to revise the layout of the chart. We choose to force the falling rectangular tokens to remain approximately flat when the physics engine could make them rotate, we limited the height of the deposit area above the bars and tuned the force model. To match the logo style we also set the decay function to not change the shape of tokens but instead darken their colour over time. Consequently, there is no flocculation and tokens are removed after their decay time.

To control the data flow rate, a buffer containing the last 100 questions is loaded when the application is launched. This buffer is asynchronously filled, while each second, a question is taken out of the buffer and enters the visualization.

WikiFlow

Wikipedia in live

Visualization of real time wikipedia edit on the top 5 language version.
Language are sort by importance according to wikipedia 2008 statistics.
Token size is map on the edit length. StokeColor correspond to the user.

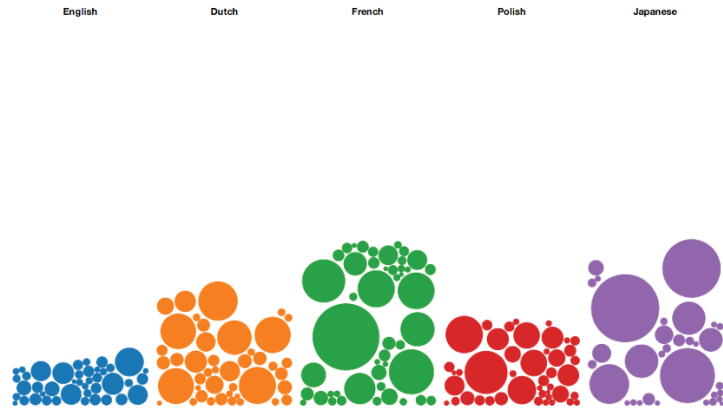


Figure 6.15: Birth, death and population screen capture at 5h15 pm.

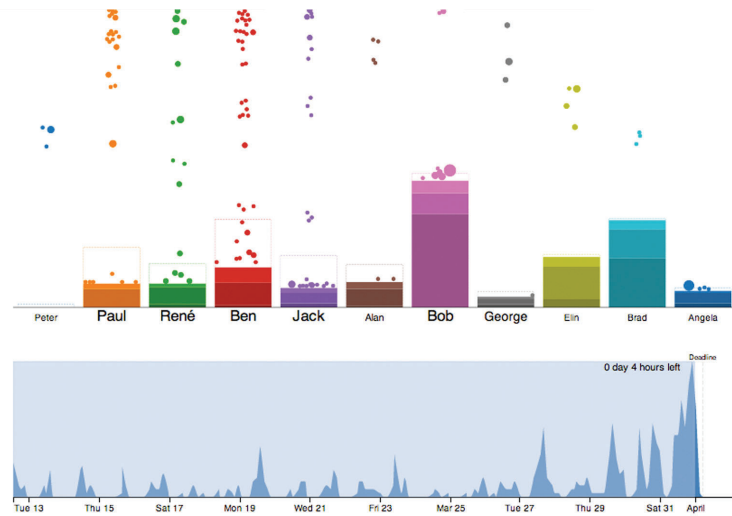
Wikipedia⁴, the free encyclopedia online, manages each of its languages as a different Web site, resulting in differences in the number of articles, coverage, style, and frequency of edits. We apply Visual Sedimentation to create the WikiFlow visualization, showing real-time edits made in five languages: English, German, French, Polish, and Japanese (Figure 6.15). Languages correspond to categories, page edits correspond to tokens with a size proportional to the edit's length.

⁴ <http://www.wikipedia.com/>

Interactions with tokens allow users to get details on demand: hovering on a token shows the page name, author, and summary of the edit. After edit details are shown, a black outline is drawn around the token. Clicking on the token opens the edited page. For managing the updates, new data is buffered and appear every 30 seconds to give users enough time to detect changes.

SVN Commits Replay and Visualization

Figure 6.16: SVN Commits visualization



We extracted edit logs from our research team SVN repository, reflecting the writing and programming of 11 researchers; relating to activity both on articles and programming during the 20 days preceding the VisWeek 2012 deadline. To visualize this data stream, we designed *SediVN*, a Visual Sedimentation bar chart where researchers correspond to categories, edits correspond to tokens with a size proportional to the edit's length, and the sequence of edits is replayed in the original order at higher speed (Figure 6.16). Tokens had a 24-hour lifetime before flocculating into three strata that encoded the token count during the past 24 hours, the previous week and the overall period of time since the beginning of the recording, respectively. Since data was known in advance, the maximum bar heights (representing the total quantity of edited text for one researcher) were shown as bar contours that fill up when tokens flocculated. To provide more control and an overview, we added a static visualization under the animated bar chart showing the distribution of edits over time during the total 20-day period; it clearly shows the increase of activity in the run-up to deadline. The progression of the *SediVN* animation is depicted as a temporal slider, with a darker translucent background from the starting time (left) to the current time. The current time position can be interactively moved to replay the animation, jump to interesting times, or control the animation speed.

As tokens may appear simultaneously, we reinforced token arrival notifications both visually and with sound. Visually, we increased the label size of the bin that will fall into. The size then decreased slowly until another token arrives. Thus, label sizes reflect the recent activity of each bin (Figure 6.7I, left). Each entering token makes a short audible beep, with a different pitch associated with each bin category. This effect is designed to evoke the vocal pitch of different researchers

working in the same room; it also expresses accurately how frequently edits can occur. All these feedback modes are reminiscent of a *ripple* effect.

We received feedback from many researchers within and outside our team; they commented that they could feel the heat of the deadline by watching SediVN. However, they sought more information relating to collaborations. We made an adjustment to SediVN by adding lines between tokens that are related to the same edited files. Since the lines also added visual clutter, they only appeared when hovering the mouse over a particular researcher's bin (Figure 6.7I).

Twitter Popularity of M&M's Colors

During two months, we recorded 19,852 tweets containing both the word "M&M's" and a colour name among red, yellow, blue, green, orange and brown. The SediMMS application visualized them in a Visual Sedimentation bar chart with colors as categories (Figure 6.17).

Each tweet is a token, represented as a real-looking M&M's. They initially appear as their tweet's text in the upper part of the chart and then morph into the well-known M&M's rounded shape. This visual effect enforces continuity between the textual and graphical representations. During initial design feedback, a user suggested that the visual transformation should give a momentum to the tokens. Therefore, we added an initial force with a random magnitude to simulate a propelling force. To show the strata, we used a background texture of M&M's at multiple scales to simulate token accumulations. Lower-level strata showed denser M&M's using a scaled-down texture to express compression over time.

One major design issue was how to display of all the recorded tweets. Standard stratification applied independently for each category made it difficult to compare the evolution of strata sizes. To preserve the strata alignment, each bar was rotated by 90 degrees at the end of each day and stacked, resulting in a streamgraph [Byron and Wattenberg, 2008]. This created an organic shape that showed the varying trends of M&M's colors over time. We advertised the visualisation through Twitter, explained the streamgraph to users as the temporal **trail** left by the bar chart over time.

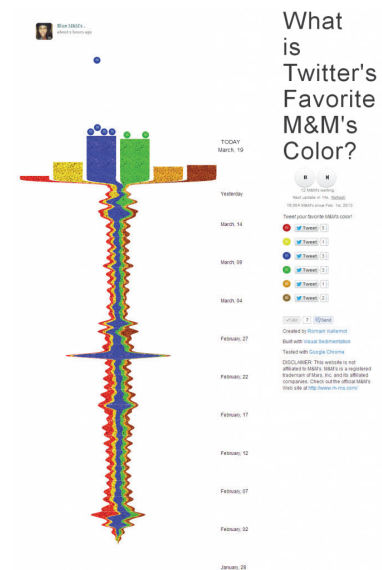


Figure 6.17: Bubble chart with Visual Sedimentation.

6.6 Deconstructing the Metaphor

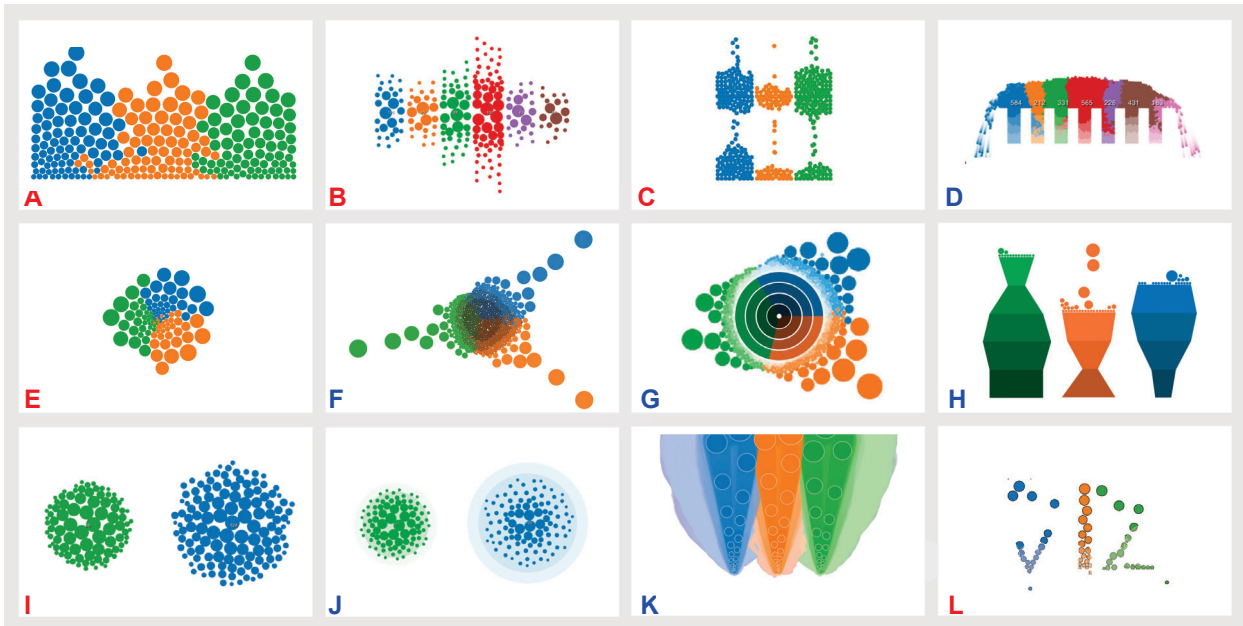


Figure 6.18: Summary of the case studies: (A) “Bar Chart” without walls.; (B) “Bar Chart” without ground and gravity; (C) “Drop Chart”, constraining tokens to flow inside identical complex shapes.; (D) A bar chart *overflow* due to the limited container capacity. (E) “Pie Chart without walls”; (F) The previous chart with a pixel aggregated area at 2 minute mark; (G) The previous chart with circular strata; (H) “Hoodoo Chart”.; (I) “Heap Chart” where tokens arrive from the center.; (J) “Heap Chart” where the deposit and aggregated areas are overlaid.; (K) “Trail Chart” where the tokens leave trails; (L) “Typo Chart” constraining tokens to the shape of the letters “VIZ”.; Red letters are pre flocculation variations and blue letters are post flocculation variations.

The aforementioned case studies are based on pre-configured range parameters from the toolkit. In these standard cases, the aggregated area always defines the visualization template and tokens fall and flocculate within these areas.

Despite the abundant literature on designing user interfaces (*e.g.* [Shneiderman and Plaisant, 2010]), there are few accepted methodologies in information visualisation to explore design spaces. In this work, we adhere to the design methodology described by Buxton [Buxton, 2010]: starting from our initial experience on Bubble-t and Bubble-TV (Chapter 5), we generalized the design space by creating the Visual Sedimentation metaphor. We then applied the design space to several examples that are instantiations of the metaphor (Section 6.5). We now further extend the metaphor by deconstructing it. We will progressively modify toolkit parameters that imitate physical-constraints and are implicit in the sedimentation metaphor; physical constraints are not necessarily required. This deconstruction is compelling for two reasons: first, from a design standpoint, it extends the possibilities of Visual Sedimentation to richer and more creative solutions; second, from a technical standpoint, it adapts Visual Sedimentation to simulate conditions that go beyond a literal physical metaphor. Moreover the design space exploration is also useful to provide some guidelines for designers and developers.

6.6.1 Methodology

To explore the design space, we relied on the parametric space provided by the toolkit (Table 6.1). One simple way to explore it would

be to vary all the parameters systematically and observe the results. However, this is not possible due to the strong constraints linking the parameters. Instead, we created variations by focusing on three parameters: force, layout, and aggregated area. We created a large number of designs with variations of these parameters, selected the “best” samples, and iteratively improved them. The criteria we used to select the best samples were:

- consistency with the metaphor,
- readability of the mapping function, and
- interesting or surprising results.

6.6.2 Variations on Force-Based Behavior

The pie chart, bar chart, and bubble chart standard templates (Figure 6.6) are provided to the designer as starting points to produce variations (Table 6.1, a[1,2], l[1,2,3]; f[1,2,3]). The three first variations (Figure 6.18A, 6E and 6I) are based on changing some parameters of these templates. The *Bar Chart without boundaries* (Figure 6.19, Figure 6.18A) uses the general container but without the walls that indicate a column structure (Table 6.1, f[2,3]). The same process was applied to (Figure 6.18E). In this case, the three categories show the same dataset. Because the categories are colour coded, the chart remains meaningful, although less readable than the original. Because the number of categories may be unknown at start-up time, we found this chart useful in allowing new category creation or removal. Despite the missing wall constraints, and even if tokens can organize freely around their landing position, the overall result remains close to a regular chart with the exception of a few tokens. In these two charts, we have also removed aggregated areas (Table 6.1, a[6]); the resulting token accumulation encodes information beyond the chronological order of the tokens, showing trends in the data stream. For example, a large number of green tokens are overflowing near the center of the pie chart, indicating that there was an abundance of green tokens early in the data stream.

Heap Charts (Figure 6.18I) are constructed like a bubble chart but with no container and a distributed entrance point. Because the tokens have full freedom, they even out in a circle; they produce a temporal fisheye effect, as tokens shrink and decay over time. We discuss the mechanisms to preserve the metaphor in Section 6.6.3.

Removing the ground and gravity in a bar chart results in a *Silo Chart* (Figure 6.18B, Table 6.1, f[1], l[2]). Having no ground removes the physical force of gravity (Table 6.1, f[1], l[3]), allowing the infinite spread of tokens. The chart preserves the separation between categories, and still provides a potentially infinite space for the tokens: only walls direct the tokens’ placement inside infinitely long strips. No gravity means that as the tokens enter through their entrance point, they push away older tokens, resulting in a growing circular shape. As the tokens decay over time, the outer ring of the chart ends-up containing smaller tokens than the center, making the more

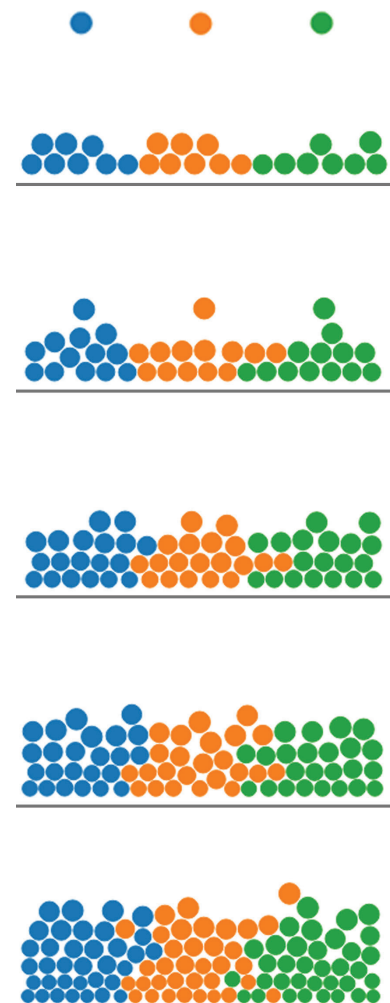


Figure 6.19: Evolution of a “barchart without boundaries”.

recent tokens more prominent. Multiple contiguous silo charts can be compared visually as the tokens fill the strips additively.

The metaphor can also be deconstructed by *perforating* layouts with holes that enable tokens to escape through the holes and prevent unexpected overflows. A *Drop Chart* (Figure 6.18C) is a bar chart with a hole on the ground that is filled with tokens from the top and emptied at the bottom. A second container underneath is progressively filled with the falling tokens. This chart allows token paths to be specified, showing the time windows before the tokens flocculate.

Varying the container shapes produces a wide variety of layouts, sometimes requiring the force models to be adapted. Another variation is to assemble various shapes as obstacles to slow down and control the path followed by the tokens (Table 6.1, l[1]). In the *Typo Chart* (Figure 6.18, L), tokens follow predefined paths that are typographic characters or signs. In this example, tokens draw the letters “V”, “I”, and “Z” as they are constrained by invisible walls. These letters are drawn as tunnels in front of the entrance points. They are filled with tokens from the top and emptied from the bottom. Despite the high level of expressivity in this example, it is difficult to compare the quantities and arrival time of tokens when the containers have different shapes.

6.6.3 Variations on Aggregated-Area Layouts

We have found three possible strategies (Figure 6.20) for creating the area chart:

1. generate an area chart from scratch with the support of the D3 toolkit,
2. generate a layout from the flocculated tokens, and
3. generate a layout from the tokens’ trails.

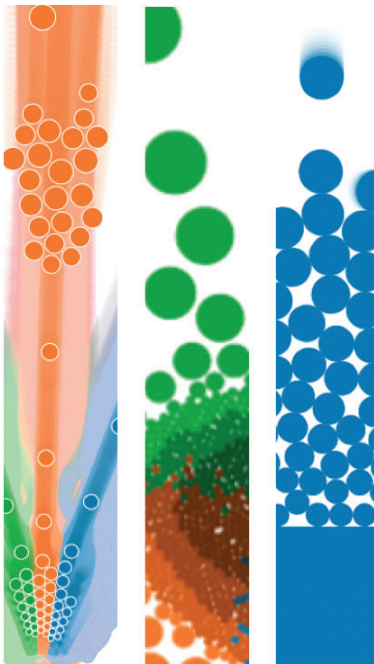


Figure 6.20: Different types of aggregated area, from right to left: classic aggregated area generated with a geometric primitive element, aggregated area made of flocculated tokens, aggregated area generate from the trail of tokens.

The first strategy is provided by the default templates for the standard aggregated areas (Table 6.1, a[1,2]). It can either use extensions of templates to produce more complex area charts (as in SediMMS), or use new variations as in Figure 6.18G and 6H. To be consistent with the metaphor, the generated area chart should follow the requirements described in Section 6.4, but the toolkit cannot enforce these requirements: automatically generating area charts consistent with the tokens’ behavior and with the requirements remains an open research question.

For the second strategy, the toolkit makes it possible to use the tokens’ dynamic behavior against physical barriers to generate shapes that can be used in the area chart. For example, the flocculation process can be further simulated to generate future generations of flocculated tokens that will constitute the aggregated area (Table 6.1, a[4]); the toolkit allows exploration of several parametric configurations for that strategy. This is shown in Figure 6.18F where a pie chart is generated by the composition of flocculated tokens which, instead of disappearing, fill pixel by pixel the aggregated area at the center

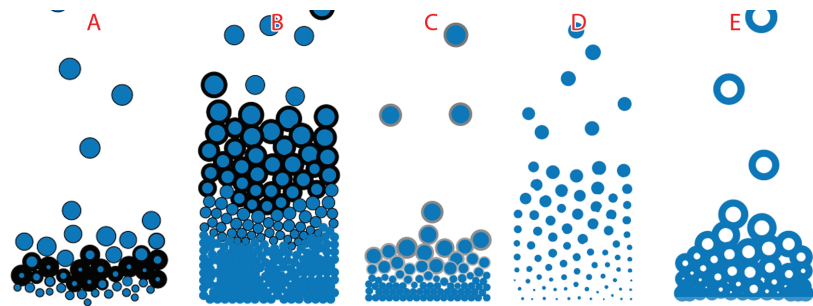
of the chart. Fading the colour of post-flocculated tokens visually creates different strata levels (Table 6.1, a[4,2]). Interestingly, this mechanism results in a rendition similar to Figure 6.18G which is based on the distribution of flocculated tokens, but requires the full execution of the simulation and may not result in a chart consistent with the flocculation stage.

The third strategy is to not only provide a visual persistence once the token is settled, but also while it is in motion starting from its incoming point. *Trail Chart* (Figure 6.18B; Table 6.1, a[3]) shows the trails of every single token trajectory, with the thickness decreasing as the token decays. This results in overlapping strips where colour encodes the token's dimension and fades out after a certain amount of time. Overlapping trails may thus indicate trends thanks to transparency. Large trails indicate that a data burst happened, which is not visible anymore once the tokens become deposited in the chart. Trails have the same properties as strata as they show previous tokens' activity. They can even systematically be used in any type of chart as a way to add motion blur and facilitate token tracking, as well as to capture screenshots of charts (Figure 6.18, I). Once they disappear, tokens may also leave a trail, which also fades away over time resulting in distinct horizontal layers similar to strata (Figure 6.18, E, I).

In addition to aggregated areas, dynamically generating strata according to token positions can be done by adding *isocontours* (Table 6.1, a[5]): area boundaries containing the tokens just before they flocculate. These boundaries provide a persistent visual footprint of maximal radii during a sliding time window (*e.g.* last minute, hour, and day). Isocontours explicitly encode the maximum value of their perimeter of deposits, they are useful for visual comparison tasks. In Figure 6.18, J, we display them low opacity so as to keep them both visible because they share the same visual space as tokens. Isocontours can only be used when the aggregated areas are generated in a way consistent and continuous with time, unlike *e.g.* Figure 6.18, F. Just like obtaining consistent isocontours, obtaining aggregated areas suitable to perform specific reading tasks is not straightforward since the aggregated area depicts the number of flocculated tokens modulated by their age. To address that issue, Figure 6.18, H visualizes the total number of tokens as bar heights and number per time window as bar widths, allowing comparisons between categories.

6.6.4 Variations on Token Encoding

Figure 6.21: Token stroke and fill variation overtime.



Token visual encoding variations over time could provide a high degree of expressivity to process different encodings or effects. One encoding could highlight different time periods in the token deposit area while keeping the visual identity of each individual elements without adding additional information layers. One way to create such an encoding, while still remaining consistent with the metaphor, would be to change the stroke thickness over time according to a function of interest. We can see sketches of this approach in Figure 6.21A,B,C.

An important benefit of token-based visual encodings is the possibility to select and manipulate directly a token that corresponds to one datum. This effect, i.e. selecting one or a set of tokens, could be achieved with various strategies. One strategy could be to vary, e.g. reduce or augment, the collision area of the token while keeping its visual mark stable. This can be seen in figure 6.22 with token C and D. If the collision areas of the tokens were reduced, then the visual marks corresponding to the tokens could visually merge to form one collective visual mark as can be seen by the partially formed group of tokens in figure 6.21,B and E inside the deposit area. Allowing reduction could render the token hard to perceive and a high reduction would radically impact its identity. If the collision areas of the tokens were increased, then it may be easier to select individual tokens. Such a situation where the tokens have an increased collision area can be seen in 6.22 with token E, and also in figure 6.21 with column D. In the latter example, the collision areas of the tokens are increasing as the tokens age and the visual marks shrink.

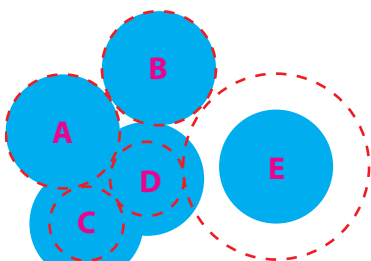


Figure 6.22: Dissociation of the token collision area and the token visual mark.

Token shape is an important factor to consider when designing a visual encoding to use in combination with the Visual Sedimentation metaphor. Token shape strongly affects the spatial organisation of the tokens in the deposit area, i.e. how tokens will group and stack. For instance, in the stackoverflow case study (Section 6.5.1), we chose to use rectangular tokens for aesthetic reasons.

As such, we were obliged to review our default layout so as to add more constraints to it. This could happen with every other potential token shape. Token shape could provide rich new visual properties. In the case of stackoverflow, rectangular tokens were easier to count than

round ones due to how the rectangular tokens organized themselves in the deposit area.

6.7 Technical and Design Implications

We now relate and comment on *technical* and *design* observations from our experience in the implementation of the previous case studies and extended designs. These issues arise because, while the physical world is mostly self-regulated (except occasional catastrophes), the digital world is not. Therefore, implementations should take measures to adjust to scale variations in an adequate way. These scale variations can relate to the tokens rate, the token sizes, the deposit time, or flocculation time.

To mitigate these issues, Visual Sedimentation uses a physics engine that behaves in understandable ways when pushed over the tokens capacity. Very much like a dam overflow, tokens overflow their containers and walls but continue to behave consistently so the metaphor is still valid to some extent. However, visualizations can become cluttered and unpredictable, sometimes in creative ways but not under the designer's control. We discuss the current performances of the toolkit and its mechanisms to improve control.

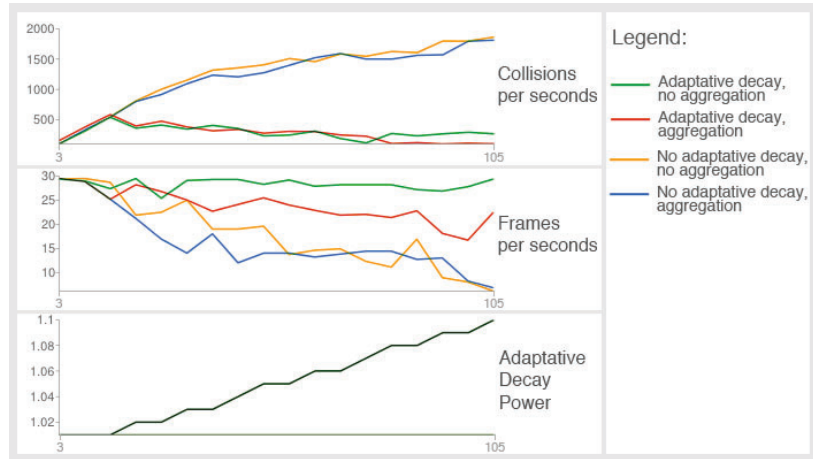
6.7.1 Performance

As the primary purpose of the toolkit is to enable the exploration of the metaphor's design space, its current implementation has not yet been optimized for performance. Still, we tested it and identified strategies to improve its performance with different parameters using the following configuration: MacBookPro, processor 2.4Ghz Corei7, 8GB Ram, Mac Os X 10.7.5, Chrome Version 27.0.1453.93. We ran the test on the default chart of the library (a 300×300 pixel canvas with a 3-bin bar chart). We recorded the number of frames per second (fps) as a measure of visual rendering quality, and also the number of collisions per second (cps) as we hypothesized that this could be a bottleneck for fps drops. We implemented various scenarios (see results Figure 6.23) with multiple strategies to keep the fps consistent. All scenarios are based on the same range of the number of incoming tokens per second (tps), from 3 to 105 (tps).

Two factors seem to account for the drop of performance: 1) increases in cps, and 2) triggers of the token aggregation function (which requires JavaScript code processing by the browser). Beyond 541 cps (measured with 15 tps), the refresh rate drops under 25 fps, which is below the standard quality for animation. To limit the number of collisions per second and maintain a decent frame rate some strategies can be adopted. We call these strategies *adaptive* as they are activated according to dynamic parameters such as tps or cps.

The first strategy is to change the decay rate according to the number of tps to limit the number of tokens in the deposit area and, indirectly, the number of collisions. The second strategy is to delay

Figure 6.23: Summary results of the Visual Sedimentation toolkit. The horizontal axis represents the number of tokens per second, from 3 to 105.



the aggregation shape update to a limited number of tokens. In this type of configuration the simulation can go beyond 25 *fps* for more than 105 *tps*. The pitfall of these two strategies is that they introduce a variation of the encoding according to the data flow rate. To avoid confusion for these adaptive solutions we recommend that designers indicate any change of encoding to the user.

6.7.2 Bursty Data Streams

The main mechanism for managing bursty data streams is buffering tokens to control their entrance rate. Buffering implies adding delay or forcing a constant interval between token, creating lag between what the data stream sends and what the visualisation displays.

Visualizing tokens immediately after a new data arrives from the stream is the simplest and most sensible strategy, when possible. Variations in token rate become visible, for example, in SediVN, an increasing edit rate as the deadline approaches. However, SediVN tunes the token sizes and gravity forces optimally for the stream because it replays recorded data and can prepare in advance.

For online cases where the data stream rate is unknown, we adopted an hourglass buffering and reading strategy: each token is displayed for a short time before it flocculates. This is the case for Bubble-t, Bubble-TV, StackOverflowChart, WikiFlow and SediMMS. This short time allows users to read the token data and it also ensures the deposit area does not overflow (Figure 6.18, I, Figure 6.24).

Other strategies include enlarging the deposit area, increasing the suspension time, and decreasing the flocculation time. These parameters affect the look and feel of the visualisation and are therefore design trade-offs. Alternatively, modifying the buffering strategy has no impact on the look and feel but on the faithfulness of the interface: if it pretends to be near real time but it is severely delayed, the interface is not faithful for the user. A simple way to fix the problem consists of showing a count of buffered items or a visual feedback representing them, such as drops ready to fall. Interactive control can

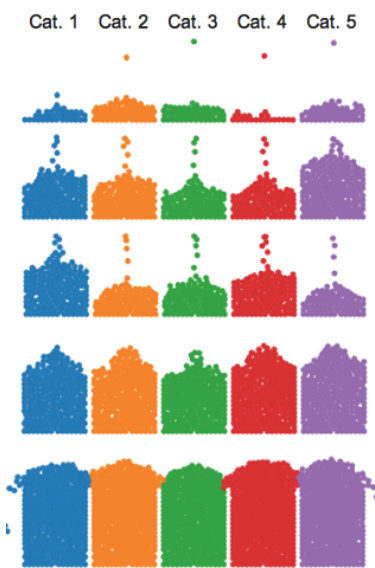


Figure 6.24: Bar Chart overflow.

be added to flush all the buffered tokens.

6.7.3 Tokens Deposit and Strata Visual Comparison

Comparing stream throughput between categories requires choosing deposit and aggregated areas that are easy and accurate to compare.

In Bubble-t, as containers share a *common ground* and no decay, the comparison is facilitated, in contrast to Bubble-TV, SediMMS and SediVN where strata elevate the ground level, making the comparison task difficult. To solve this problem, the deposit areas could share an aligned ground level from which the deposit would grow up and the aggregation would grow down. This is the design choice made in SediMMS.

In addition to positions, visual variables encoding tokens such as size and shape impact the comparison, and even the overall readability since tokens filling speed is not uniform. For example, during one of the first iterations of StackOverflowChart, tokens were horizontal rectangles (4×12 pixels) in a container twice as wide. This configuration gave too much freedom for token accumulation in containers and resulted in unexpected stacking positions. By reducing the container's width, the tokens became ordered in the next generation.

6.7.4 Token Provenance and Destination

Tokens may arrive from several entrance points, forcing users to scan many deposit areas to detect new incoming tokens. An appropriate visual encoding facilitates the *provenance* tracking of tokens. Adding labels for each entrance point is the natural solution, but is redundant with existing labels of the charts. Alternatively, using an arbitrary entrance point with no label has led to unexpected interpretations. In SediVN, a user suggested making the tokens fall from the top of the web page, as it seems more natural than the tokens arriving from nowhere. Indeed, users tend to associate visual elements surrounding the entrance point with the origin of token data. If details about tokens are provided before token creation (*e.g.* tweet text) then a visual effect expressing causality has to transform the details into tokens: in Bubble-t, tokens are propelled in the direction of their destination column, in SediMMS a smooth transition changes the tweet's text before it gets to its destination bin.

Deciding on the destination of tokens is not straightforward either. For example, in Bubble-t, the tokens are thrown similarly as a basketball shoot from a single entrance point to the final container.

The angles and forces applied to the tokens were manually tuned in order to create aesthetically pleasing token trajectories before entering the container, as well as to prevent collisions with other tokens and to reach the correct container. Having an interactive tool to test the parameters would have helped tremendously.

6.8 Chapter Summary

In this chapter, I have introduced a new design metaphor, *Visual Sedimentation*, which was directly inspired by the physical process of sedimentation. Using this metaphor, meaningful chunks of data are visualized as tokens, which enter the visualisation at an entrance point. These tokens remain in suspension while they fall to the ground based on a virtual gravitational force. The tokens then accumulate on the ground, decay, and finally aggregate into areas or strata. We have demonstrated the expressivity of this metaphor by applying it to a wide variety of dynamic data sources including: Twitter topics, social media sites such as StackOverflow or Wikipedia, and user activities in versioning system repositories.

In this chapter, I have also discussed how the metaphor reacts when adjusting design parameters that connect the visualisation to the physical constraints of physical sedimentation processes. This metaphor opens up a design space, that generalizes the possible bridge between token-based and area-chart visualisation while simultaneously providing great expressive potential.

We have created an open-source toolkit that enables designers to implement novel visualizations quickly using the Visual Sedimentation metaphor. This toolkit allows the community to explore further the design space that has been opened up by Visual Sedimentation. The exploration of this design space has already allowed us to create a broad spectrum of creative and unexpected visualizations.

The main contribution of this chapter is the integration of previous token-based visualisation systems into a generalizable design space that is supported by a metaphor that unifies token-based dynamic visualisation and classical area chart. This design space has allowed me to provide a solution to research **PROBLEM 1.2**: How can one represent dynamic data updates for non-expert audience visually?

Visual Sedimentation provides new perspectives and a new level of expressiveness when visualizing dynamic data. However, it also raises several questions, I will address in the third part of this thesis:

- (1) How can one exploit the simplicity of the token-based incremental assembly processes for visualization?
- (2) How can one provide a graphical way for non-experts to build their own dynamic visualizations by making use of tokens?
- (3) How the direct manipulative properties of a tokens could be use actively in visualization?

6.9 Part Two Summary

In the second part of this dissertation I designed a token-based visual representation system: PolemicTweet (Chapter 4: page 77). Analysis of the system allowed us to validate and discuss the possibility of using this type of visual encoding to engage people in dynamic data analysis. We then reported on the design and development of two different token-based systems (Chapter 5: page 100). Based on these different applications we derived a generic visual mapping metaphor, Visual Sedimentation (Chapter 6: page 110). This metaphor allows the combination of two different types of visual representation token and an area chart by visually encoding data updates as the assembly of tokens into an area chart, over time. It also allowed me to set up a design space that incorporates classic information visualization, such as bar charts, pie charts and others.

The study with PolemicTweet and replicated at a different scale with Bubble-TV addresses our first sub research **PROBLEM1.1** *“Is it possible to leverage a crowd of InfoVis non-experts to analyse dynamic data?”*. However, we cannot be certain that our findings are generalizable and valid in other contexts. Rather, these studies serve as proofs of concept which can be built upon to further validate the approach.

The work presented on Visual Sedimentation (Chapter 6) addresses our second sub research **PROBLEM1.2** *“How do we visually represent dynamic data updates to non-experts?”*. Even if this approach is limited to certain types of data stream, the technique provides a new approach for composing area charts with tokens assembling over time.

Despite the new perspectives on dynamic data visualisation opened up by Visual Sedimentation, it does not provide a viable means for people without coding skills to author their own dynamic visualization. In the second part of this dissertation we will explore *how* we can provide a token-based authoring paradigm for non-experts, *if* people can construct their visualisation out of tokens, and *how* people can create and update token-based visualizations.

Part III

Democratization of Dynamic Visualization Authoring

“C’est la mobilité interne de l’image qui caractérise la graphique moderne. On ne ‘dessine’ plus un graphique une fois pour toutes. On le ‘construit’ et on le ‘reconstruit (on le manipule)’ jusqu’au moment où toutes les relations qu’il recèle ont été perçues.”
 Jacque Bertin [1977]

During the previous part of this dissertation we demonstrated how token-based visual representation could be useful for dynamic visualization. All the visualizations presented in the previous part have a specific visual encoding. This encoding could be described as an assembly of discreet individual manipulative element into continuous aggregated visual area. We explored how this visual encoding could be useful to update dynamic data visualisation over time. We evaluated the feasibility to use such technique to engage a non-expert audience (Chapter 4) of different size (20,300,700 000) into a dynamic analysis process. Then we derive a generic technique to visually encode dynamic data for this population: Visual Sedimentation (Chapter 6). This technique is supported by a metaphor, a toolkit, and a design space. Despite the generalizability of this visual mapping, this technique doesn’t provide a mean for non-expert to design his or her own dynamic visualization, which is our second most important problem (P2).

To address P2, I hesitated between two different research strategies. The first one was to consolidate the Visual Sedimentation toolkit and create an on-screen interface for non-expert to author their own Visual Sedimentation mapping and remapping. The second one was: *i)* see if it is possible to formalize token assembly as a possible generic authoring approach to dynamic visualization, and *ii)* to investigate *if, what* and *how* non-expert could author their own visual representation by assembling tokens. I chose the second one.

In this part of the dissertation I will first define the challenges of providing visualisation authoring tools for non-expert. Then I explore how token assembly have been considering as a valid authoring tool for the information visualisation community. Because I found that this approach was never consider before in InfoVis community, I attempt to provide a first definition of this authoring paradigm. I build this definition by combining two sources of inspirations: previous constructivist approaches in other domains and real life examples.

The result of this research was the definition of a new information visualisation authoring paradigm: *Constructive Visualization* (Chapter 7: page 139). I will describe his different components, as well as his process. Then I will illustrate this paradigm, components and process through the description of four real life scenarios. This new authoring paradigm raises new questions. Among the most important, how non-expert people are performing visual representation authoring? But

also can I validate if *Constructive Visualization* paradigm empirically?

To study empirically if non-experts could author their own visual representation based on previously defined paradigm, I designed and conducted a observational study. In chapter Chapter 8 I present this study in which I asked people to spontaneously transform data into visual representations by using tangible tokens (Section 8.3: page 164). This study allowed us to learn more about the visual mapping process and to refine my ideas about the approach of constructive authoring of visualizations.

7

Defining Constructive Visualization Paradigm



Figure 7.1: Child playing and learning math.

In Chapter 3 I provide a set of historical and design considerations for token-based visual representations. In the first part (Part II) of this thesis, I apply some of these considerations to the design of different visual representations for dynamic data analysis systems for the non-experts.

In this chapter we ¹ analyse the different existing InfoVis paradigms to extract important design challenges. We then create a new paradigm to dynamic visualisation authoring tools. We describe the components and process, and then analyse different real life applications.

¹ Portions of this chapter were previously published in [Huron et al., 2014a]. Thus, any use of “we” in this chapter refers to Samuel Huron, Sheelagh Carpendale, Alice Thudt, Anthony Tang, and Michael Mauerer.

7.1 Introduction

In this chapter, we focus on supporting the democratization [Viégas et al., 2007] of visualization, which is in contrast to conventional visualization work where the focus has been to support data intensive science, industry and government. We are seeing visualizations in such places as: personal blogs [Hunger, 2008], as part of art works [Kosara, 2007, Lau and Vande Moere, 2007, Pousman et al., 2007,

Viégas and Wattenberg, 2007], in the news media [Hullman and Diakopoulos, 2011, Segel and Heer, 2010, Weber and Rall, 2012], and as a growing part of the quantified self movement [quantifiedself.com, 2013]. These visualizations range from the intensely personal such as the visualization of the contents of one's freezer [Viégas et al., 2008], to community-based visualizations such as crime in one's neighbourhood [Viégas et al., 2008]. This movement opens several research questions. If democratization means that people (not just experts) will author 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 (Figure 7.1). Our goal is to explore educational theory to learn how to design tools that support non-professionals in the creation of visualizations.

For example, Grammel et al. [Grammel et al., 2010] 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 chapter we discuss an alternate approach: we consider how people can author 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 authoring.

In this chapter we start by exploring the idea of democratization of visualization design: we identify and compare three existing approaches in which one can author 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.

7.2 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 [Ware, 2004], studying the effectiveness of visualization in regards to the intended task [Amar and Stasko, 2005], and making progress towards assessing insight triggered through use of visualizations [Saraiya et al., 2005]. 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 [Viégas et al., 2007] 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 project (Figure 7.2) 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 visualisation variations to the given set of templates.

Victor's [Victor, 2013] 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 (Figure 7.3) that people use to create visualizations:

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

He argues that each of these approaches has pitfalls. The first one, using a *pre-coded visualization*, describes the ManyEyes [Viégas et al., 2007] 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

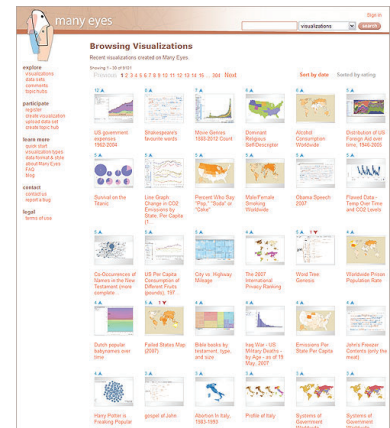


Figure 7.2: Many Eyes screenshot.

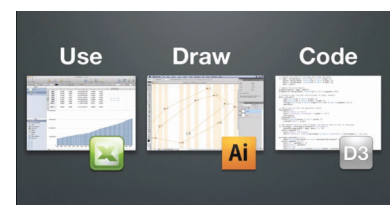


Figure 7.3: Use, Draw, Code, Slide extracted from Brett Victor presentation, Feb.2013 at Stanford. Video:<https://vimeo.com/66085662>

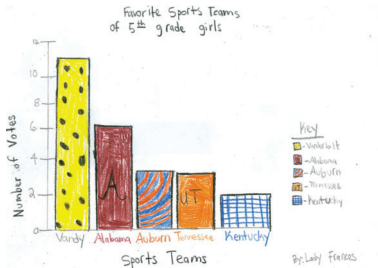


Figure 7.4: Favorites sport teams of a 5th Grade girls.

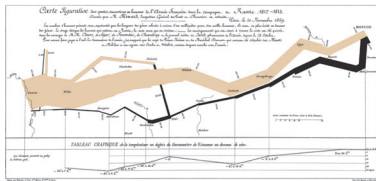


Figure 7.5: Flow map of the number of men in Napoleon's 1812 Russian campaign during 1812-1813. Lithograph, 62 x 30 cm, 1869, Charles Joseph Minard

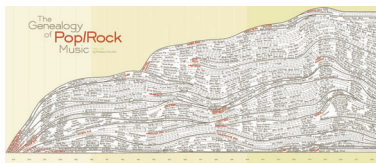


Figure 7.6: The genealogy of pop rock music covering the time period from 1955 to 1978. 50 x 91 cm, 1978, Rebee Garofalo

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 [Tableau, 2014a], Spotfire [SpotFire, 1990], and also web tools like ManyEyes or Chart Editor in Google Docs [Google, 2013]. 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 effective 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, done by hand with a pen and a paper, on a whiteboard, as well as using drawing software such as Photoshop or Illustrator. It is well accepted and studied that people often use freehand drawing (Figure 7.4) on napkins [Chao et al., 2010], sketchbooks [Greenberg et al., 2011], and whiteboards [Walny et al., 2012, ?] to help them think visually [McKim, 1972, Arnheim, 1969] 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 [Minard, 1869] (Figure 7.5), and rock music histories [Garfalo, 1979](Figure 7.6).

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.

7.3 Design Challenges

As noted above, we focus on the authoring of the visualisation from data, or, in other words, how the mapping from data to the visual structure is done. In this section, we first specify how this creation aspect of the visualisation process relates to the InfoVis reference model [Card et al., 1999] and Bertin's Semiology of Graphics [Bertin, 1973]. Then, combining this InfoVis perspective with the ideas from

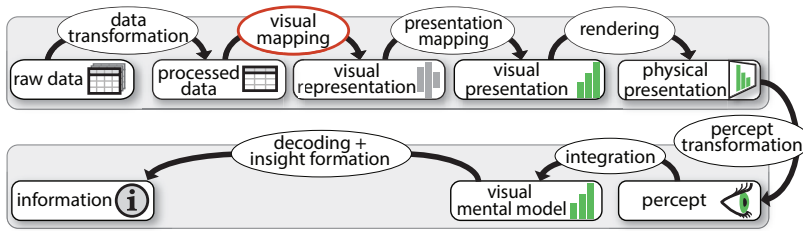


Figure 7.7: The extended visualisation reference model [Jansen and Dragicevic, 2013], adapted to our terminology.

the discussion on democratization, we define three design challenges that need to be addressed to democratize the InfoVis design.

The information visualisation 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 7.7), has been described by Card et al. [Card et al., 1999], refined by Carpendale [Carpendale, 1999], and Chi and Riedl [Chi, 2000], and extended by Jansen and Dragicevic [Jansen and Dragicevic, 2013]. 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. [Card and Mackinlay, 1997] 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 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 [Bertin, 1973] 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 visualisation creation paradigms. Based on these definitions and Victor’s paradigms of visualisation creation we define three design 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.

Paradigm	DC1 Simple	DC2 Expressive	DC3 Dynamic	Manipulation of Visual	Skills	Technical system	NofP
Using	✓	✗	✓	Indirect	Medium	Excel, Spotfire [SpotFire, 1990], Tableau [Tableau, 2014a], Google Chart [Google, 2013] ...	36
Drawing	✓	✓	✗	Direct	Easy /Medium	Pen and paper, Illustrator, Photoshop, ...	13
Coding	✗	✓	✓	Indirect	Hard	Processing, InfoVis toolkit [Fekete, 2004], Prefuse [Heer et al., 2005], D3.js [Bostock et al., 2011], and others...	17
Constructing	✓	✓	✓	Direct	Easy	Unknown	0

Table 7.1: Table of existing information visualization design paradigms in comparison to *constructive visualization* paradigm. NofP: Number of Papers referenced by Grammel et al. [Grammel et al., 2013] distributed across Victor’s [Victor, 2013] paradigms and including the option of constructive visualizations.

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 to 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 visualisation creation more generally accessible is that thus far only through code can a visualisation support adaptability to data dynamics—that the visualisation can change in response to a change in the data stream. 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

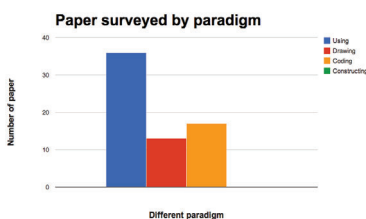


Figure 7.8: Paper distribution between the different paradigm. Blue for using (36), Red for Drawing(13), Orange for coding (17), Green for constructing (0).

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.

7.4 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.

7.4.1 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 [Manning, 2005, McCormack et al., 2004], designers [Cross, 1983, Stiny, 1980] and computer scientists [Resnick et al., 1996, 2009, Resnick, 2007]. Froebel’s Kindergarten used pedagogic activities called “Gifts” (Figure 7.9). Aleeb-Lundberg [Aleeb-Lundberg, 1970] 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 (Figure 7.14) 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 7.15), that we extend to constructive visualization.

7.4.2 Piaget – Learning by construction

Piaget used similar building blocks to the ones designed by Froebel [Piaget, 1977, 1989] in his experiments to study the cognitive processes of children. According to Piaget [Piaget and Inhelder, 1962, Piaget and et Niestlé, 1948] 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 development. More recently, Chapman’s studies [Chapman, 1988] have shown how this ability to grasp new ideas through construction applies to people of all ages, not just children.

7.4.3 Papert – Constructionism applied to programming

Building on constructivist theories [Papert and Harel, 1991], Papert [Papert, 1980] extended the idea of pedagogical manipulative materials to computer programming. The first result of this fruitful



Figure 7.9: A complete set of toys “Gift” designed by Froebel to learn mathematics.

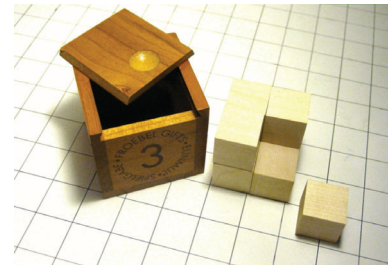


Figure 7.10: Details of one of the toys. Here the “Gift” N°3 the box, and the eight building blocks.



Figure 7.11: Screenshot of a video of Piaget experiment [Piaget, 1977]: A girl of six year old stacking cube to rebuild a volume.

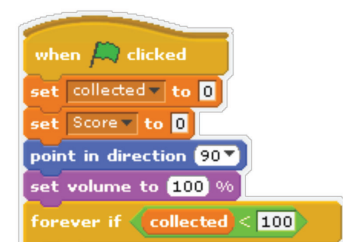


Figure 7.12: Scratch command block.

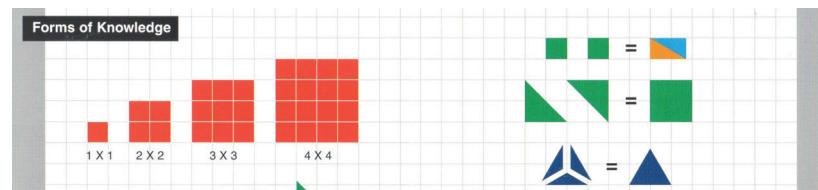


Figure 7.13: Children playing with a turtle, a pedagogical learning material to program logo by trial and errors.

approach was the programming language Logo [Papert, 1999] and turtles (Figure 7.13). 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 (Figure 7.12) and Mindstorm [Papert, 1980].

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.

Figure 7.14: Instruction for teacher to explain mathematical operation with Froebel Gifts.

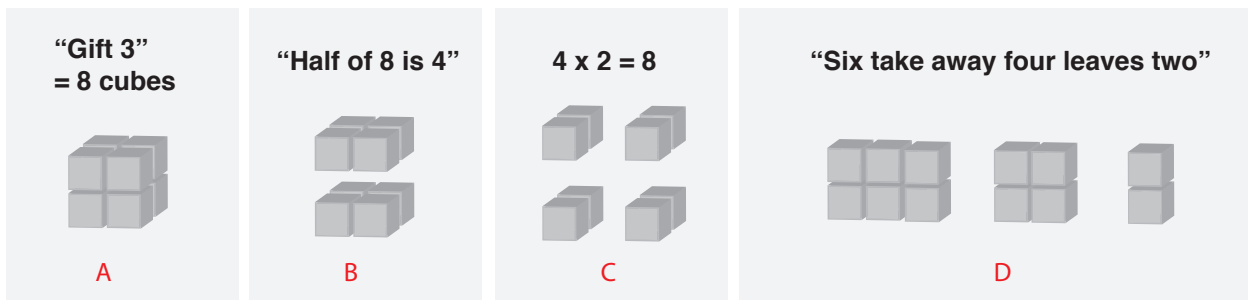


7.4.4 Applying these Lessons to InfoVis

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

Figure 7.15: Example of instruction for teacher Illustration for the Froebel Gift N°3 from the manual for teachers [Bultman, 2000]: A. Set of blocks in Gift 4, B. Operation showing division by two, C. Operation to teach multiplication. D. Operation to teach subtraction.



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.

7.5 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 visualisation 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 visualisation 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.

7.5.1 Components:

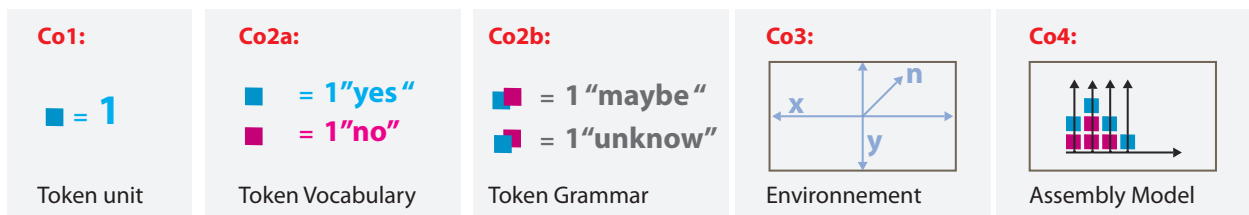


Figure 7.16: Example *constructive visualization* components.

Co1: 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.

Co2a: Token Vocabulary

The token vocabulary 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 assignments of each visual variable.

Co2b: Token Grammar

The token grammar defines a set of relations between types of tokens or tokens properties, and the mapping to the data. For example, in the case of two tokens different in sizes, one twice bigger the second, this size relation could be used to map a difference in quantity in the data. This relations between properties of the tokens and aspects of the mapping data could also be made by the individual constructing the visualization. According to Bertin's vocabulary, the token grammar also fall down into the list of semantic assignments of each visual variables.

Co3: Environment

The environment is the space that provides constraints on how tokens can be assembled together using the token grammar and vocabulary. 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.

Co4: Assembly model

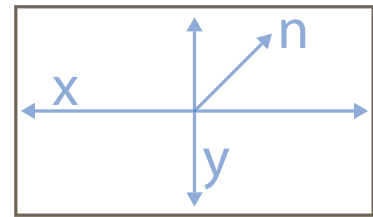
The assembly model defines the rules of the construction process. This is the internal model of how the constructing and deconstructing of the visual representation is carried out. The final result contains the data articulated and represented through the assembly of the tokens.

7.5.2 Process:

Pr1: Environment initialization.

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

Pr1:



Environment Initiatlisation

Pr2: 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 and vocabulary). Furthermore, data properties must also be assigned to the token properties, for example, relating position to some quantity/property of the data.

Pr2:



Mapping data to tokens

Pr3: 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.

Pr3:

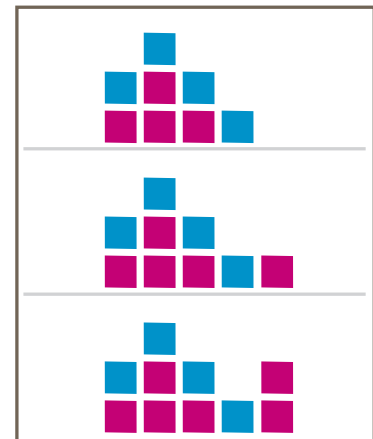


Assembling Tokens

Pr4: 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.

Pr4:



Evolution over time

Figure 7.17: Example *constructive visualization* process.

7.6 Examples







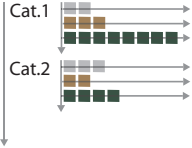
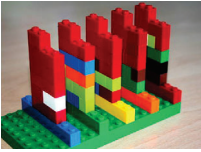
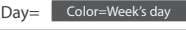




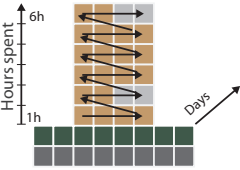

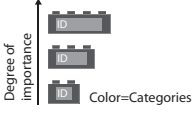
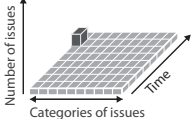
Examples	Picture fragment	Type	Co1: Token	Co2,a,b: Token Vocabulary & Grammar	Co4: Assembly model	Co3: Environment
1. Chris Jordan		Artistic	Object / picture of object	1 picture:  = 1 plastic cup, used on airline flights in the US during last six hours	Artistic, the assembly model in this case does not 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	Unit token type 1 Day=  Color=Week's day Unit token type 2 15m=  30m=  45m=  Color=Project 60m=  in this case the Unit token type 2 have a specific grammar based on the mapping between the time and the size of the tokens.		3D Physical tangible Lego board
4. Kevin Quinn		Analytic	Lego bricks	 Degree of importance ID ID ID Color=Categories		3d Physical tangible Lego board

Table 7.2: 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 [Neurath, 1939], Picture 3: ©Michael Hunger, Picture 4: ©General Motors Cf. <http://goo.gl/zMFK6E>.

While our notion of *constructive visualization* is not yet in use digitally (Table 7.1), 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 visualisation and how each fulfills our three design challenges.

7.6.1 Chris Jordan: Statistics as Engaging Art Pieces

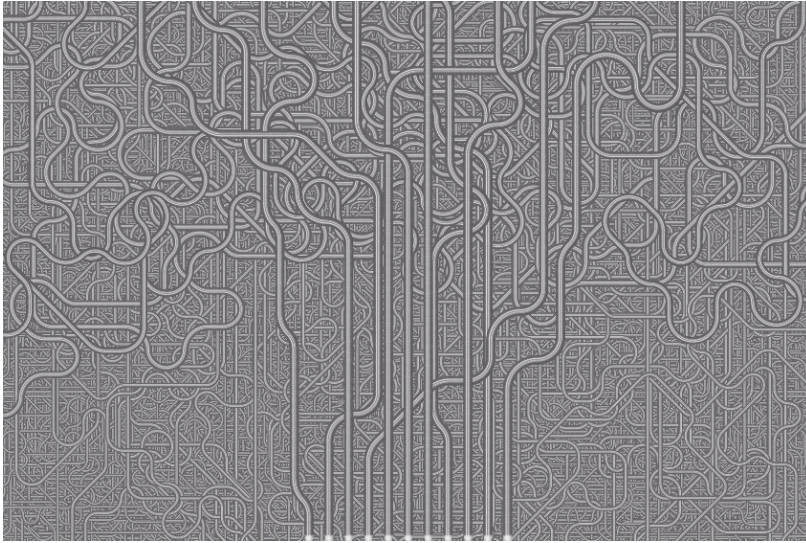


Figure 7.18: Plastic Cups, 2008, 60x90, Depicts one million plastic cups, the number used on airline flights in the US every six hours. Chris Jordan.

Jordan is an artist who provokes thoughtfulness about everyday environmentalism. Jordan explains [Jordan, 2006]: “*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 7.2 and fig. 7.18 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.

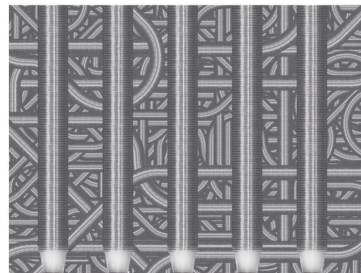
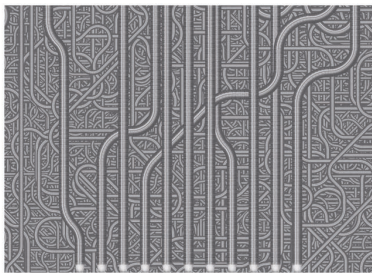


Figure 7.19: Different level of zoom in the piece “Plastic Cups”.



7.6.2 Otto Neurath: Communicating Statistics to People

Figure 7.20: Otto & Marie Neurath and his assistant are assembling previously clip pictograms (Section 3.3.2: page 49) into a specific spatial configuration for a chart.

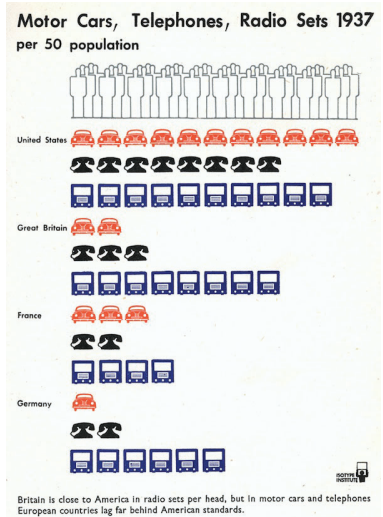


Figure 7.21: Neurath visual representation of the car, telephone and bus production by country per 50 peoples.

Neurath [Neurath, 2009, 1939] 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 vocabulary); 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.

7.6.3 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 (Figure 7.22) 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

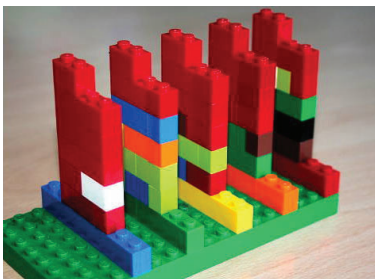


Figure 7.22: Michael Hunger original personal time management tool.

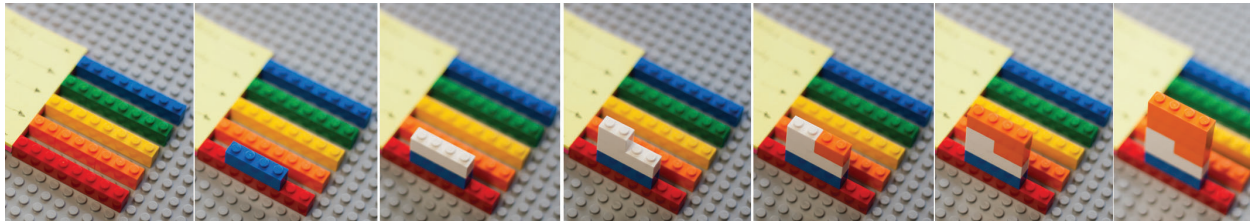


Figure 7.23: Photography of the simulation of visual representation updates over time.

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) (Figure 7.23).

7.6.4 Kevin Quinn: Problem Resolution Tracking

Quinn [Wilson, 2012] 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 visual representation based on a Lego board (7.24). The

Figure 7.24: Kevin quinn’s problem resolution tracking visual representation.

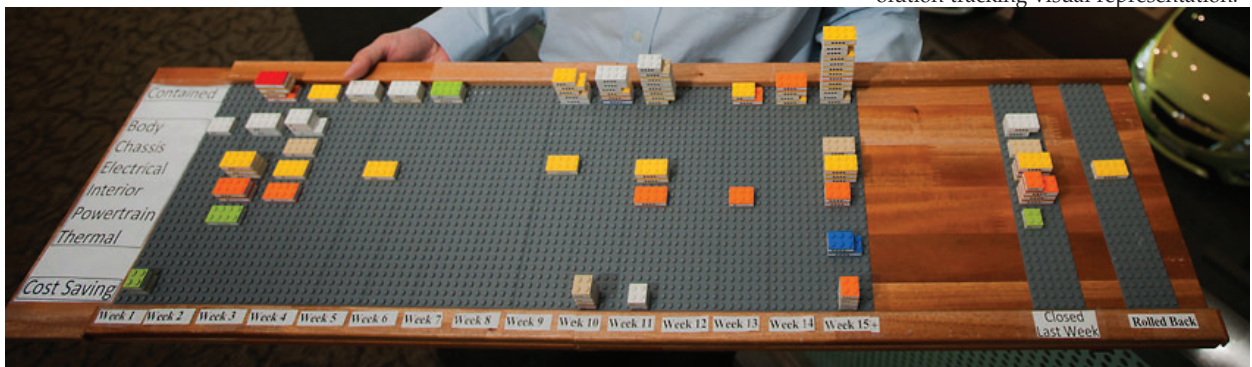




Figure 7.25: Details of Kevin Quinn tracking system zoom on.

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 vocabulary and grammar* are 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 is printed on the side.

The company has adopted this system, and Quinn and colleagues continue to work with the Lego visual representation 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 (Figure 7.25). 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”.

7.6.5 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 7.2, 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.

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 ($\rightarrow Co1$), and grammar ($\rightarrow Co2$), 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 ($\rightarrow Co3$).

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 Neurath’s examples illustrate the first point, where they design their own signs/tokens ($\rightarrow Co1$ =pictogram, object or picture) to express some dimension of the data. Hunger’s token grammar ($\rightarrow Co2$) 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 colour for each day, and the quarter of hour token (quantitative data) with a colour for each project. Even though Jordan takes artistic freedoms in his assembly model ($\rightarrow Co3$), 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 ($\rightarrow DC_1$).

Hunger's and Quinn's examples address this design challenge in the way that they create their assembly model ($\rightarrow Co_3$) 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 ($\rightarrow Co_1$). In addition, Quinn's also updated by his group. They apparently do not update their token grammar ($\rightarrow Co_2$). However, the possibility of dynamics ($\rightarrow DC_3$) is also dependent on the environment. While Lego is updatable and adjustable, photographs and printed graphics are less so. Reproducing Neurath's approach computationally could enable dynamics.

7.7 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 [Viégas et al., 2007]. Victor [Victor, 2013] raised questions about what it meant to be able to create one's own visualizations. He described the existing 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.

7.7.1 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 relative to the selecting 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 a top 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 visualisation 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 [Hutchins et al., 1985], or algorithmically placing or removing new blocks like in Visual Sedimentation chapter 6.

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 [Neurath, 2009] 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.

7.7.2 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 (Section 3.5).

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.

7.8 Chapter Summary

In this chapter I have presented constructive visualization, a token-based paradigm, which can help democratize the authoring of visualisations. Constructive visualisation offers potentially provides people with the means to construct visualisations on their own in a simple, expressive and flexible way. To construct a visualization, the necessary components are:

- A set of basic units or tokens, which can be mapped to data,
- A token vocabulary and grammar, which declares how the attributes of the tokens can signify data, and how the tokens relate to each other,
- An environment, which holds the tokens,
- An assembly model, which describes what is permitted and forbidden in assembling the tokens.

The process of developing the constructed visualisation 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 tokens.

This new design approach to visualisation was inspired by ideas from three domains that all share a core tenet of simplicity. From the invention of the Kindergarten Gifts by Froebel, I imported the logic of using tokens to manipulate units. From the constructivist 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 visualisation in particular. In applying the concepts of constructivism to information visualisation design, I have defined its components, token, vocabulary, grammar, environment and assembly model, and outlined the processes required for this approach: initialization, mapping, assembly, and evolution.

In the next chapter, I will gather empirical evidence to evaluate *if* and *how* people can create, update and discuss the visualisations they produce.

8

Studying Constructive Visualization & Visual Mapping

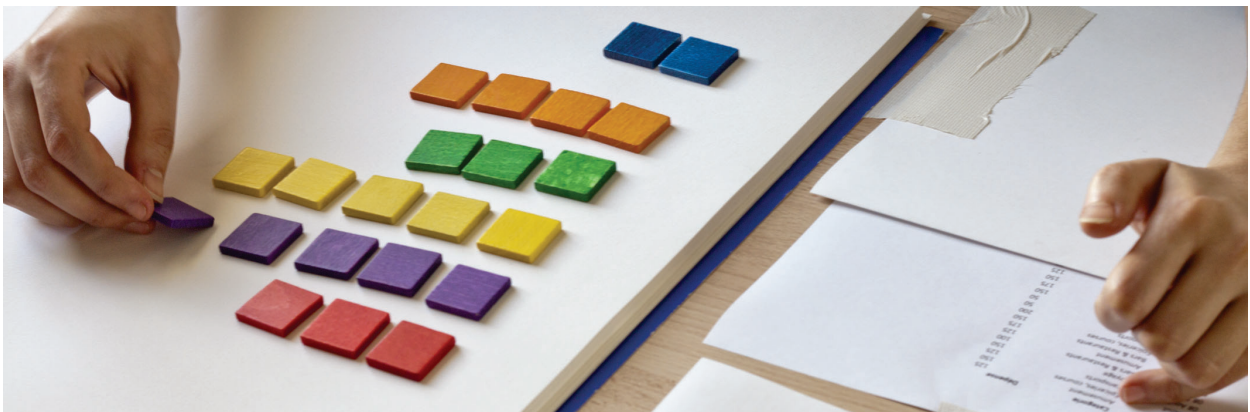


Figure 8.1: Constructing a visualisation with tokens: right hand positions tokens, left hand points to the corresponding data.

In the previous Chapter 7 I defined a new paradigm to author dynamic information visualizations based on constructive assembly of tokens. In this chapter I assess this paradigm. However, currently, little is known about the way people process visual mappings. To address this, we ¹ design an observational study based on constructive assembly of tokens to track this process, as well as to evaluate the *constructive visualization* paradigm.

This study enables us to evaluate if people can author, update and discuss visual representation through the *constructive visualization* paradigm. It is performed with a population that was educated but at the same time non-expert with respect to information visualization.

Analysis of the results allowed us to extend the model presented in the previous chapter and derive a model of the visual mapping process, something that to our knowledge has not been done before. In this chapter, I will describe the study design as well as *how* non-experts were able to create, update and discuss their own visual mappings.

¹ Portions of this chapter will be published in [Huron et al., 2014b]. Thus, any use of “we” in this chapter refers to Samuel Huron, Yvonne Jansen, Sheelagh Carpendale.



Figure 8.2: Additional resources: A dedicated website was made for this experiment, it contains videos, photos, figures, and others: <http://constructive.gforge.inria.fr/>

8.1 Introduction

The use of information visualisation (InfoVis) is becoming increasingly widespread, with the result that InfoVis can now be encountered in everyday life: online, in newspapers, or on TV shows. In response, the research community started to consider InfoVis for purposes other than strictly analytical ones [Pousman et al., 2007, Lau and Vande Moere, 2007] and to explore questions such as the democratization of visualization [Viégas et al., 2007]. However, this democratization requires that the general public, not just experts, be able to design, publish, and discuss their own visualizations with their own data.

As Victor [2013] illustrates, and we discuss in previous chapter, the available software tools either offer only a limited set of pre-defined visualisation templates or require effort and skills, such as coding, to create more adapted or customized results. By comparing different approaches to creating visualizations, e.g. spreadsheet software, programming languages, and computer assisted drawing, he derives three relevant design challenges that may help the community reach the goal of creating accessible yet powerful visualisation tools: simplicity, expressivity, and dynamicity. We can find some of these properties in existing tools that people spontaneously use to help them think visually. Examples include creating visualizations with manual encoding [Alex Bigelow, 2014], on napkins [Chao et al., 2010], on whiteboards [Walny et al., 2011], with paper and scissors [Douglas et al., 1995], or building tangible visualizations [Jansen and Dragicovic, 2013].

In previous chapter (Chapter 7: page 147), we defined *constructive visualization* as a theoretical basis for a new visualisation authoring paradigm based on Papert, Piaget and Froebel's ideas [Froebel, 1887]. Constructive Visualization is motivated by benefits that may arise from constructing a visualisation out of tokens. Since tokens can be added and removed as needed, such constructions offer possibilities for both expressive freedom and dynamic adjustment. We present in this article a study to examine and refine this paradigm. Our goal is to investigate: *if* non-experts people can construct their own visualizations using tokens, *how* they construct their visualizations, and *what* type of visualizations they create. Finally we are interested what *limitations* people encounter with this approach. In particular, we focus on the visual mapping process – the process by which people use tokens to create a visual arrangement that represents their data. Our deconstruction of this process reveals eleven logical tasks that can be grouped according to their main purpose of construction, computation and explanation.

8.2 Motivation & Background

Here we motivate our research question from open research challenges and provide a rationale for the design of our approach.

8.2.1 InfoVis Democratization

The benefits of visual thinking are currently well established [Arnheim, 1969, Card et al., 1999]. Beyond the consumption of prepared visual representations, studies show that the creation and manipulation of visual representations can improve the learning and understanding of students [Gabel and Sherwood, 1980, Gobert and Clement, 1999, Schönborn and Anderson, 2006]. Similarly, even for abstract problems such as Bayesian reasoning [Micallef et al., 2012], the active construction of a visual representation seems promising [Cosmides and Tooby, 1996]. These studies focus on simple diagrams or only require participants to draw on pre-existing visual representations. There is still a lack of studies investigating self-defined visual representations.

Indeed, as we already discuss in Chapter 1 the creation of information visualizations by a wide audience has been identified as a major challenge by several researchers. The *NIH/NSF visualisation research challenges report* states: “A big challenge is to create [...] a system that, while clearly not comprehensive and all-powerful, does help to enable non-experts to perform tasks [...] in any reasonable time frame. [...] The goal is to make visualisation a ubiquitous tool that enables ordinary folks to think visually in everyday activities” [Johnson et al., 2006]. Similarly, Heer and Shneiderman point out the need to create new interfaces for visualisation specifications: “Novel interfaces for visualisation specification are still needed. [...] New tools requiring little to no programming might place custom visualisation design in the hands of a broader audience” [Heer and Shneiderman, 2012].

Tools such as ManyEyes [Viégas et al., 2007] and Tableau Public [Tableau, 2014b] attempt to make the creation of visualizations accessible to a wider audience. These web-based tools allow one to *create*, *publish* and *discuss* visualizations. These tools support the creation of visualizations by providing sets of pre-defined templates which can be populated with one’s own data. Despite the benefits of this work, potential users are limited to the templates provided by the respective websites and have no means of developing their own visual mappings.

8.2.2 The Challenge of Visual Mapping

The development of a visual mapping from data dimensions to visual features is a key task of the visualisation authoring process. The common reference model for this process is shown in Figure 7.7. The core of this model, the *visual mapping* transformation, defines the mapping from a dataset to a visual representation, or more specifically the mapping of data dimensions to the variables of the visual marks that compose the *visual representation* [Bertin, 1973].

While much work has been done on finding perceptually efficient visual representations [Cleveland and McGill, 1984, Ware, 2004], we know comparatively little about how humans perform the step of visual mapping themselves [Grammel et al., 2010]. Consequently, to improve the accessibility of visualisation authoring tools, we first need a better understanding of *how* a visualisation author performs a representational mapping transformation.

8.2.3 *Choosing a Methodology*

Several different approaches exist that can inform the design of new visualizations tools. One is to study currently existing tools to derive possible improvements for these and future tools. For example, Grammel et al. [Grammel et al., 2010] conducted a study to understand how non-experts design visualizations. In order to avoid confounds possibly introduced by the use of a specific software tool, participants were asked to specify visualizations verbally while an operator created and displayed the resulting visualization to the participant. This study identified three barriers that are related to the visual mapping process: *i)* selecting which dimensions to map to visual variables, *ii)* selecting which visual marks to use, and *iii)* decoding and interpreting the visual result. While this study provides valuable insights, it is based on the participants choosing among pre-defined templates and thus leaves many unanswered questions about the details of the visual mapping process.

Another approach is to design a new technique, to develop a prototype, and to compare it empirically to currently existing tools. While this may provide insight into the tools studied, this approach does not allow to unpack the visual mapping process.

Yet another approach is to study human behaviour independently from the design of specific tools. Our work falls into this category. Observing how people construct their own visual mappings may help us to develop a better understanding of basic processes of visualization authoring. This understanding may be valuable for informing future tool design. Observing people developing visualizations without a software authoring tool removes possible tool bias and allows us to directly observe behaviour that is commonly encompassed within a software tool.

8.3 *Study Design*

In this study we sought by the following questions: *i)* to learn more about the visual mapping process, *ii)* to gain some understanding about what makes the visual mapping process sometimes rather difficult [Grammel et al., 2010] and sometimes quite easy [Chao et al., 2010], and *iii)* to explore the suitability of constructive authoring of InfoVis as an approach to the creation of visualizations. This section provides the details of the study design.

8.3.1 *Participants*

We announced our study by mail, social network, mailing list, and randomly approached people on the street (the study took place in a downtown center close to transportation facilities). We recruited 12 participants from a variety of disciplinary and educational backgrounds, with an effort not to disproportionately select those from visualization, human computer interaction, and computer science in general. Participants were between 22 to 43 years old with a median age of 28. They predominantly had a high school diploma, with 2 to 8 years (median 5 years) of further study in a variety of fields such as art, humanities, and science. More demographic background information is summarized in Table 8.1.

For our participants, the average self-reported level of expertise in computer programming is 1.8/5, and their average expertise in drawing is 2/5. The frequency with which they saw visualisation varied from daily to monthly, 6 of 12 participants reported seeing visualizations every day. The frequency with which they author a visualisation varies also from daily (1/12), weekly (4/12), monthly (4/12), yearly (2/12) to never (1/12). When asked how many visualisation they created during the last 12 months, they declared 10 in average (min:0,max:20). For the same period, they declared having seen 57 visualizations in average (min:2,max:200). When asked which tools they use or would use to create a visualization, a variety of tools were named: Excel (4/12), Adobe Illustrator (2/12), R (1/12), but the participant who reply R, commented that he would have to learn it first, Processing (1/12), HTML (1/12), Google Doc (1/12), post-its (1/12).

All participants were rewarded with a 10 Euros gift coupon from a well-known online webstore. All but one participant agreed to be videotaped during the experiment.

		Participant	1	2	3	4	5	6	7	8	9	10	11	12
Demographic Information		Education Domain (Art, Humanities, Science)	H	A	H	S	A	A	A	H	S	H	A	A
		Gender (Male: <input type="checkbox"/> , Female: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		Age (Min: 22, Average: 28, Max:43)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		Higher Education (1year: <input type="checkbox"/> , 8years: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Self rated on 5-point likert scale		Proficiency programming (1: <input type="checkbox"/> , 5: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Proficiency drawing (1: <input type="checkbox"/> , 5: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	over a year		Proficiency create infovis (1: <input type="checkbox"/> , 5: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Proficiency read infovis (1: <input type="checkbox"/> , 5: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Frequently work with data (1: <input type="checkbox"/> , 5: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Time Spent	Minutes	Total time (Min: 40, Average: 70, Max:103)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Task 1 (Min: 5, Average: 10, Max:28)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Task 2 (Min: 2, Average: 5, Max:14)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Logical Tasks Perormed		Task 3 (Min: 3, Average: 7, Max:17)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Number of operations (97: <input type="checkbox"/> , 295: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Computed new values (yes: <input type="checkbox"/> , no: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Created Meta-category (yes: <input type="checkbox"/> , no: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Agregated (yes: <input type="checkbox"/> , no: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Unitize (yes: <input type="checkbox"/> , no: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Changed mapping btw task A & B (yes: <input type="checkbox"/> , no: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Interview	Internalization of token mapping	Data transform. btw. task A & B (yes: <input type="checkbox"/> , no: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Would use a similar technique (yes: <input type="checkbox"/> , no: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Rethought infovis during update (yes: <input type="checkbox"/> , no: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Considered updating (easy: <input type="checkbox"/> , or complex: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Did you manipulate object (yes: <input type="checkbox"/> , no: <input type="checkbox"/> , null: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Questionnaire answers	Self rated on 5-point likert scale Compared with other techniques this was	Did you manipulate data (yes: <input type="checkbox"/> , no: <input type="checkbox"/> , null: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Enjoyed building a infovis (Strongly Agree: <input type="checkbox"/> , S. Disagree: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Would do it again (Strongly Agree: <input type="checkbox"/> , S. Disagree: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Would do it with others data (S. Agree: <input type="checkbox"/> , S. Disagree: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Difficulty of infovis creation (Easy: <input type="checkbox"/> , Hard: <input type="checkbox"/> , Neutral: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Customisability of infovis (Easy: <input type="checkbox"/> , Hard: <input type="checkbox"/> , Neutral: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Modifiabilty of the results (Easy: <input type="checkbox"/> , Hard: <input type="checkbox"/> , Neutral: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Having fun (More: <input type="checkbox"/> , Less: <input type="checkbox"/> , Neutral: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Expressive (More: <input type="checkbox"/> , Less: <input type="checkbox"/> , Neutral: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		Easy to manipulate (More: <input type="checkbox"/> , Less: <input type="checkbox"/> , Neutral: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Easy to update (More: <input type="checkbox"/> , Less: <input type="checkbox"/> , Neutral: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
Effective to explore data (More: <input type="checkbox"/> , Less: <input type="checkbox"/> , Neutral: <input type="checkbox"/>)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			

Table 8.1: Demographic information and questionnaire answers visualized with Bertifier technique from [Perin et al., 2014].

8.3.2 Setup

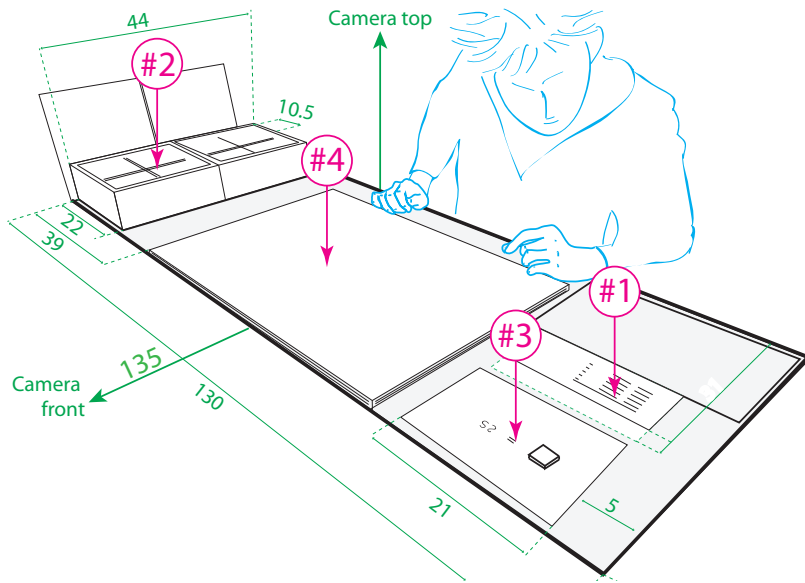


Figure 8.3: Experimental setup (all measurements are in centimetres).

After participants finished filling the consent form and the demographic questionnaire they were invited to sit on a chair at the desk and the setup (Figures 8.3 to 8.5) was described to them. Each element was introduced and indicated by the facilitator in the following order:

- #1 **The printed dataset.** We use an aggregated version of a bank account statement as experimental dataset. The participant sees three months of expenses on an A4 paper fixed to the table. All expenses are grouped into categories: “entertainment”, “bar and restaurants”, “groceries”, “transportation”, and “travel”. To simplify the participants’ data processing all values are rounded to 25. The dataset is ordered first chronologically (August, September, October, November) and then by the category of expenses where the order of categories varies across months.
- #2 **The token box.**² The tokens were contained in two boxes with four compartments taped together and to the table. Six of the eight compartments contain tokens. Each compartment contains 36 colored tokens (red, orange, yellow, green, blue, violet). Participants are not explicitly informed of the total number of tokens. The tokens are 25 millimetres wooden tiles taken from a learning toys kit designed by Froebel for Kindergarten education³. For future use we modelled the piece, and the token box are reproducible through models available as open source on Github⁴.
- #3 **The token mapping.**⁵ Since we rounded all data values to the nearest 25, we suggested to participants a mapping of 1 token \sim 25 and indicated it on an A5 paper sheet with a 3D printed version of a token.
- #4 **An A2 paper canvas.**⁶ The working area was a fixed A2 paper canvas, which was placed in the center of the table.

² This element correspond to component `Co1:Token` in *constructive visualization* paradigm.

³ see Section 3.2.7: page 45; Section 7.4.1: page 145

⁴ <http://constructive.gforge.inria.fr/>

⁵ This element is inspire by the `Co2a:Token Vocabulary` component.

⁶ This element is inspire by the `Co3:Environment` component.



Figure 8.4: Photograph of the experiment room.



Figure 8.5: Photograph of the table for the experiment.

8.3.3 Procedure

After filling out a consent form and a demographic questionnaire, participants were introduced to a specific scenario that provided them with a goal for their task: help a friend improve his budget. The facilitator then explained the study setup and the tools available to help them reach their goal. We structured the study such that participants were first asked to *create* a visualisation based on the available data. When finished, in a semi-structured interview, the facilitator asked participants to explain their visualizations. Then, participants were given new data and asked to *update* their visualization. Afterwards, a second interview was conducted. Next, the participants were asked to annotate their visualisation such that their friend, whose budget they visualized, would be able to later understand what they had done. Finally, participants filled a post-session questionnaire and received a gift card in compensation. The study lasted on average 70 minutes (minimum of 40 minutes; maximum of 103 minutes).

8.3.4 Tasks

The three tasks that we asked participants to perform were:

- A: **Create a visualization.** To increase the ecological validity of our study, we provided participants with a scenario they could encounter in their real life. We told them that a friend has asked for their help with his financial situation. The friend admits to having trouble managing his expenses and to being unsure how to resolve the situation. Then, the facilitator points to the dataset and comments that currently changes over time are hard to see, and a visual representation might be helpful. The available tools for creating such a representation are the tokens placed on the table, and since all values in the dataset are divisible by 25, a mapping of 1 token \sim 25 seems reasonable. Participants are then invited to take all the time they need and to inform the facilitator once they are finished.
- B: **Update a visualization.** The facilitator tells the participants that the friend provides them with one more month of data. The facilitator slides up the data printout (Figure 8.3 #1) to reveal a month previously hidden by a paper mask. Participants are asked to update the visualisation to include the new data, using all the time they need.
- C: **Annotate a visualization.** The facilitator informed the participants that their friend was not currently able to receive their advice. The facilitator provided participants with 4 pens of different colours and a variety of post-its and asked them to annotate their visualisation explaining what they did and what they discovered. Participants were explicitly asked to provide sufficient information such that their friend could (1) read the visualization, (2) understand it, and (3) be able to re-create it if needed. Additionally, participants were asked to indicate possible budget improvements. Again, participants had all the time that they wanted and were asked to declare when they had finished.

8.3.5 Data Collection

To analyse the process of the experimental tasks we gathered four types of data.

Video. We recorded the entire process for the participants who gave their agreement. We used two cameras with different viewing angles: a bird's-eye view, and a view from the front as noted in Figure 8.3.

Questionnaires. Through questionnaires we gathered demographic information, the participants' opinions on the techniques they used, and the task they had performed.

Interviews. Between task A and task B we conducted a semi-structured interview. During this interview, we asked participants to explain the visualisation they created, how they made it, what they manipulated, if they found the task difficult and their thoughts about the activity in general. The goal of these interviews was to obtain more information about their process and the problems they encountered.

Photographs. After each task, we took pictures of the state of the visualisation on the canvas. Sometimes, during the task, we also took pictures of unexpected behaviour.

8.3.6 Data analysis method

We collected approximately 1540 minutes of videos (2 cameras \times 11 participants \times 70 min per session, in average). We analysed the video using a qualitative data analysis approach as described by Creswell [2012]. The coding of the video was performed in several passes in an iterative process. During the different passes we developed and refined the coding. During the first two passes we described and analysed the mapping of the final result and what was directly apparent in the process to the coder e.g. "[the participant] counted and placed the tokens in a heap". During the second pass we tried to identify the regularities between participants in the process. For the third pass we re-defined our coding to express actions of participants in terms of transformations within the InfoVis pipeline. Then we applied this selective coding during two passes.

Our approach is limited to what we could code and observe. Because participants sometimes created their own mapping in an iterative fashion, we could not always identify whether a token was mapped to a data point or not, and if so, to which point. From an observer point of view many transformations (such as data transformations) are implicit mental processes that people do not necessarily make explicit. Where possible, we used information gathered from the interviews to resolve such uncertainties.

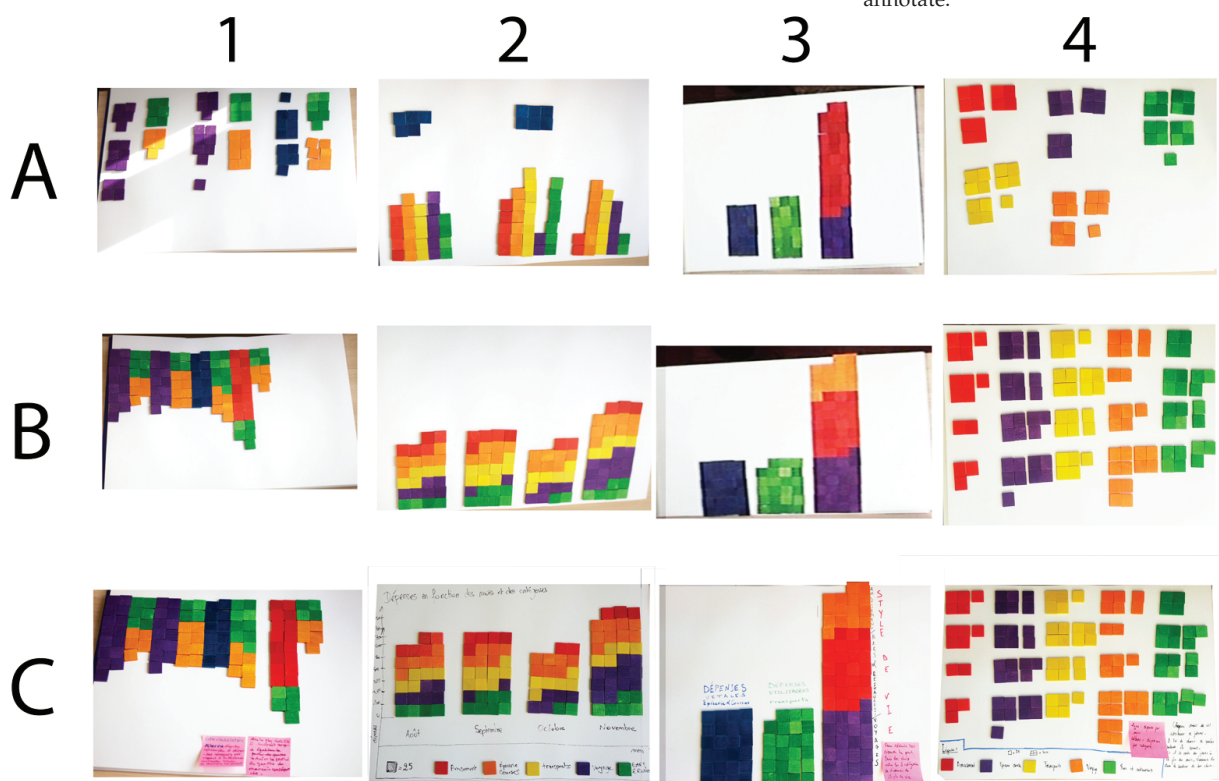


Figure 8.6: Constructed visualizations of participant 1 to 4. Columns represent participants, rows represent tasks: (A) create a visualization, (B) update, (C) annotate.

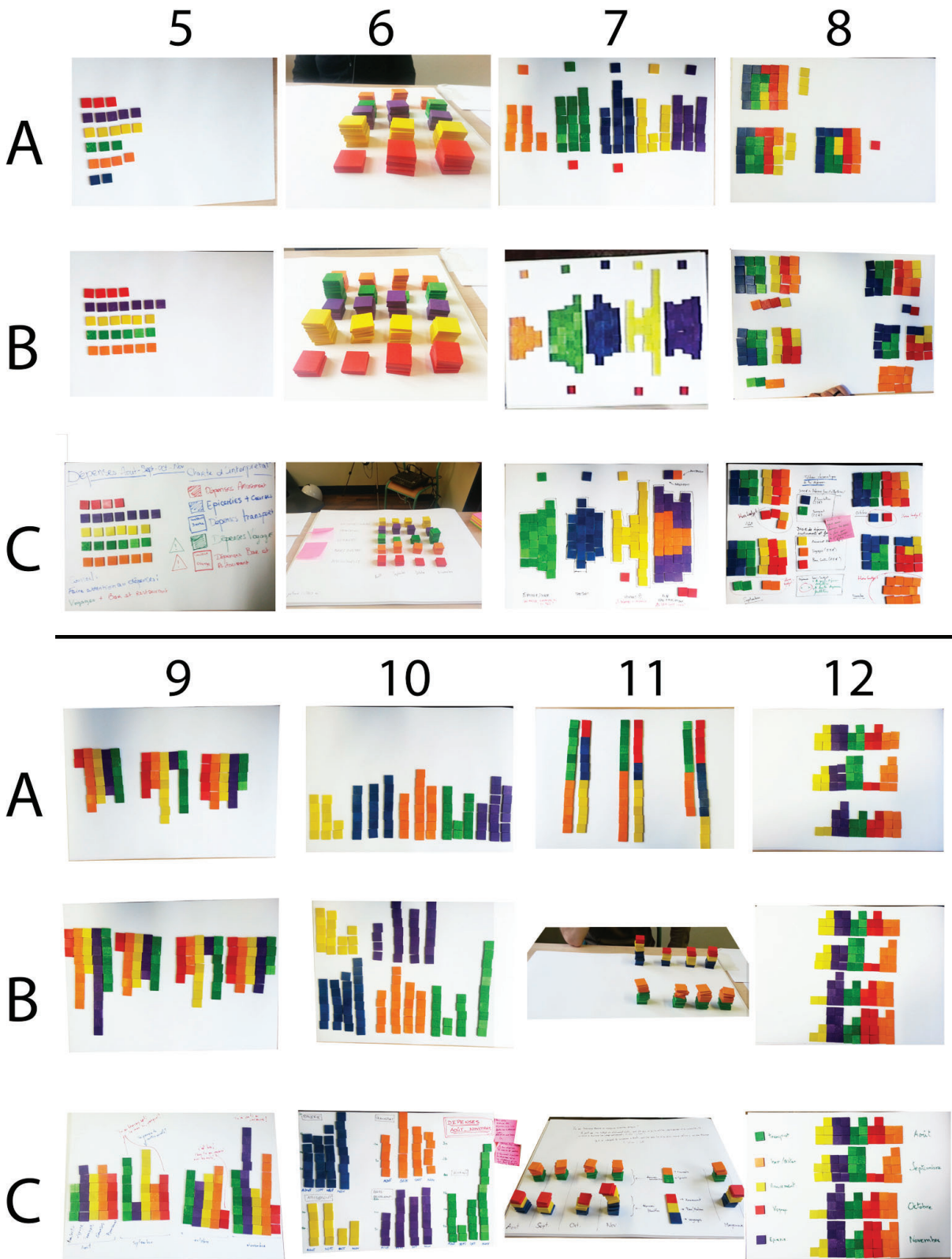


Figure 8.7: Constructed visualizations of participant 5 to 12. Columns represent participants, rows represent tasks: (A) create a visualization, (B) update, (C) annotate.

8.4 Results

All participants were able to complete the three tasks in a relatively short period of time (on average they spent: 11 minutes to *create*, 6 to *update*, and 7.5 to *annotate*). However, the time varied considerably across participants and tasks (Table 8.1). To present our results, we deconstruct the study tasks into their *why*, *what*, and *how* components [Jansen and Dragicevic, 2013] (summary in Table 8.2).

Table 8.2: Summary of identified goals, tasks, and actions.

Why	What (<i>logical task</i>)	How (<i>mental and physical actions</i>)
Construction	1. Load data	READ, COMPUTE, SELECT COLOUR, GRASP, CREATE
	2. Build constructs	ORGANIZE, MOVE
	3. Combine constructs	ARRANGE, ALIGN
	4. Extend	READ, COMPUTE, SELECT COLOUR, GRASP, CREATE, ORGANIZE, MOVE, ARRANGE, ALIGN
	5. Correct	INCREASE, DECREASE, REMOVE
Computation	6. Categorize	SELECT COLOUR, ARRANGE, MERGE, SPLIT
	7. Aggregate	MOVE, MERGE
	8. Compute New Value	SPLIT, COMPUTE + LOAD
	9. Unitize	ORGANIZE, ARRANGE, SPLIT, MERGE
Storytelling	10. Highlighting	SPLIT (temporarily)
	11. Marking	CREATE, SELECT COLOUR

We identified 11 different subtasks, named after their *what* component, i.e., the logical task, and grouped by their *why* component, i.e., their underlying goals, into *construction*, *computation*, and *storytelling*. Each of these 11 tasks can require several actions in different combinations and in different orders of execution. While we cannot directly observe mental operations, we noted three such actions from observable physical actions. These are **READING** (where the participant is looking at the data and sometimes tracing with a finger), **SELECTING** (where the participant makes a selection physically), and **COMPUTING** (where the participant calculates additional values, such as averages, in an observable or verbally declared manner). Figure 8.8 illustrates the relationship between these actions while arrows highlight the most common successions of actions. We present our findings for each of the *logical tasks*. Since actions can be part of different tasks with different purposes, we present **ACTIONS** here in the context of *logical tasks*.

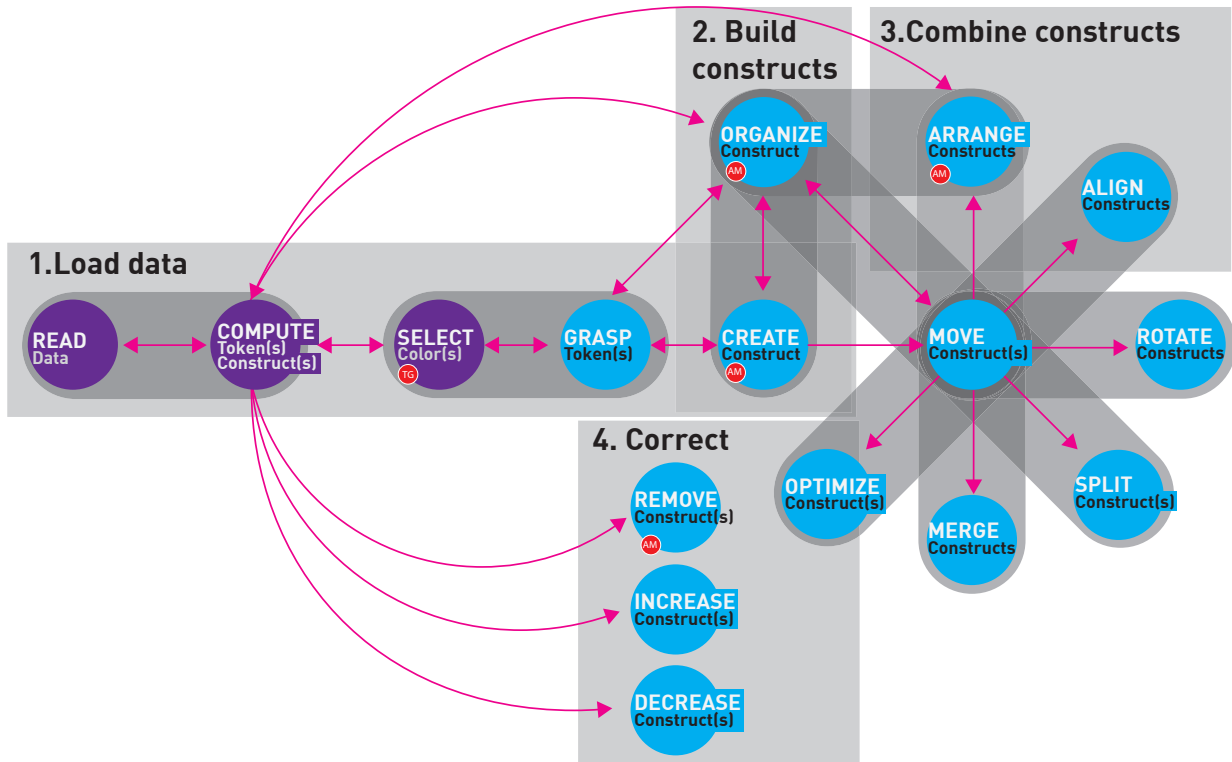


Figure 8.8: A flow diagram showing the most common paths. In purple: the mental tasks, in blue: the physical tasks. The gray background rectangles illustrate the logical tasks. The gray oblongs linking two spheres represent possibly concurrent actions.

8.4.1 Construction

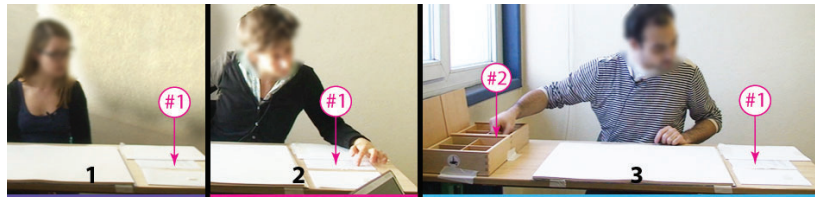
We define a visualisation as a nested construct of tokens forming a representation of data. As a representation, each of the visualizations follows a formal system – the mapping between data dimensions and available physical variables. We call this formal system an *assembly model*. Assembly models vary across participants and either develop or become apparent to an observer during the construction process. Figures 8.6 and 8.7 shows the resulting visualizations.



Figure 8.9: A participant for the pilot study achieving task C.

1. Load data:

Figure 8.10: View from the front camera on three participants (1) reading, (2) reading and pointing, (3) reading and grasping tokens



Loading a dataset involves both mental and physical actions. First, the participant needs to READ the data. As Figure 8.10 illustrates, they could perform this action in different ways: (a) by just looking at the data, (b) by assisting the reading through pointing, (c) by multiplexing reading with some other action such as GRASPING tokens at the same time. Participants then needed to COMPUTE the required number of tokens for the data value, e.g., $125 \sim 5$ tokens. This step was the only action that required computations and which occasionally led to mistakes. Computation actions were commonly not physical, though some people used physical assistance such as accumulating tokens in their hand, counting them up until they reached the data value.

Similar to COMPUTE, the SELECT COLOUR action is also a process which only became apparent once a participant GRASPED tokens of a specific colour. All participants used colour to structure and emphasize single or groups of categories. By choosing a colour, a participant expands her assembly model or makes it explicit.

Tokens were GRASPED individually or in groups (Figure 8.11–1/2) depending on whether participants (a) counted tokens while grasping them, (b) grasped several tokens and counted them into their other hand, (c) replaced tokens already in their hand, or (d) put tokens back.

Once participants placed tokens on the canvas (Figure 8.11–3/4/5) to form a group, they *built* a construct. At this point it was not yet apparent whether participants were already following an assembly

Figure 8.11: A participant GRASPS tokens (1,2,3) and BUILDS a construct (4,5,6). She is: (1) grasping some tokens from the box, (2) manipulating and transporting the tokens from the box to the canvas area (3) positioning the tokens,

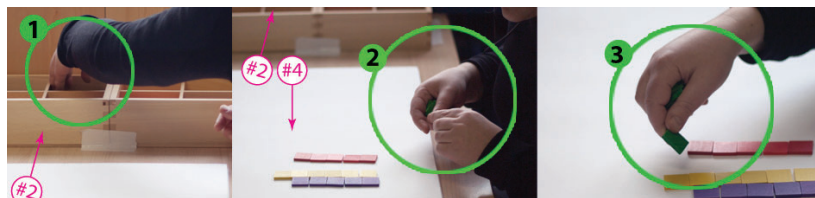
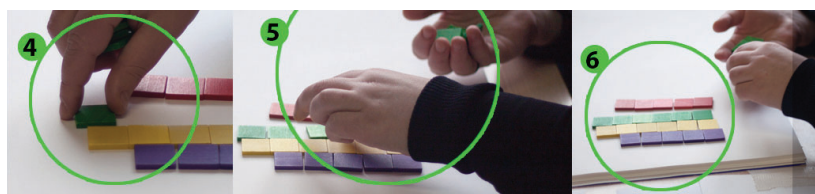


Figure 8.12: (4,5) starting to create a new green construct by placing tokens one by one on the canvas, (5) augmenting the construct in an organized way following her previous assembly (6) done. The pink annotation describes: #2 point to the token box. #4 point to the canvas.



model in their mind or whether that model only emerged during the course of the task. However, we observed that all constructs built by participants were visually dense, and often already highly structured.

2. Build constructs:

Initial constructs were not necessarily organized following a formal system but sometimes just consisted of unorganized heaps of tokens (e.g., Figure 8.15–1). In the next step, participants ORGANIZED such amorphous constructs into meaningful and identifiable shapes such as lines or rectangles (Figure 8.13). By doing so they defined rules and visual parameters formalizing (a) the spatial relationship between tokens *within* the construct they were creating, and (b) the spatial relation of the newly created construct to the canvas.

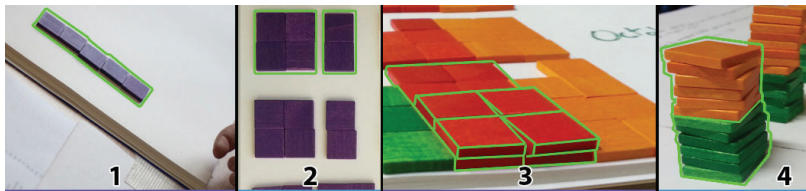


Figure 8.13: Examples of constructs: (1) a line composed of 5 tokens, (2) square and rectangles (3) a construct resembling a horizon graph [Reijner and Software, 2008]: the tokens are layered to optimize canvas real estate. (4) a 3D bar chart column with two nested constructs in green and orange.

3. Combine constructs:

When participants created a second construct for a second data case, they had to consider how to ARRANGE constructs relative to each other (Figure 8.14). We observed that spatial relationships between constructs were often used to encode hierarchical information such as groupings by expense categories then by month or the other way around. Such specifications could be made explicit, for instance, by defining an axis (observed for 11 out of 12 participants, written (11/12) from here on) or other custom configurations (Figure 8.7–4|A/8|A). Since the study dataset contained two dimensions for each value (month, expense category), participants had to decide at least twice how to encode these dimensions in their arrangement. Some participants recombined their constructs repeatedly before deciding on a final encoding.

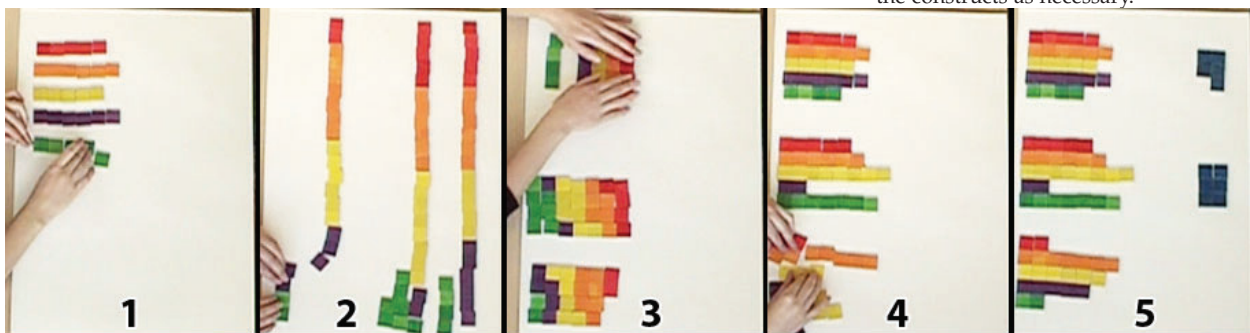


Figure 8.14: Participant 2 is testing four different combinations with the same visual constructs. During each rearrangement she adapted the organization of the constructs as necessary.



Figure 8.15: Participant 4, (1) after **LOADING** all the tokens, (2) **ORGANIZING** the red construct, (3) **EXTENDING** to other constructs.

Some participants tried several different structural combinations before settling on one, sometimes eventually going back to one tried earlier. For example, in Figure 8.14 we can see a participant change from a vertical barchart aligned along the x -axis (1) to a horizontal stacked barchart aligned to the y -axis of her canvas (2). Changing the arrangement of constructs can also impact their internal spatial organization. For example, in Figure 8.14 (3) the participant switched her representation from one where a single colour encoded the data to one where the data value is read from the area of the construct.

4. *Extend*:

This refers to the task of applying rules of an assembly model developed during the initial *building* and *combining* of constructs to the rest of the data. For example, after **ARRANGING** all data cases for the first month, and *combining the constructs* for the month dimension, participants could just repeat their previous actions to add the data for the other months. Often this replication was a linear (Figure 8.16) application of the assembly model but sometimes *extend* led to further changes of the representation, the spatial placement of constructs (without changing the assembly model) or the overall structure. Extension can be applied to any sequence of actions. In Figure 8.15 we can see that participant 4 has repeated a process of extension on *loading* tokens without organizing them (1), then he **ORGANIZES** a first construct and starts **ARRANGING** a second construct to form a hierarchy (2), and then *extends* again to all constructs (3).

Figure 8.16: Participant 12, (1) defines an assembly of construct for the first month (2) then replicates it for the next month (3), shows the result (4).



5. Correct:

COMPUTING the required amount of tokens occasionally led to mistakes. Participants commonly recovered later from such mistakes once the visual representation of the data facilitated the comparison of values. At this point the task of checking the data had become a visual task instead of a mental one thereby reducing the cognitive effort [Hegarty, 2011]. If participants noticed such mistakes, they INCREASED or DECREASED the number of tokens in a construct accordingly. A few times we also observed participants REMOVING an entire construct due to a previous error or a colour re-attribution.

8.4.2 Computation

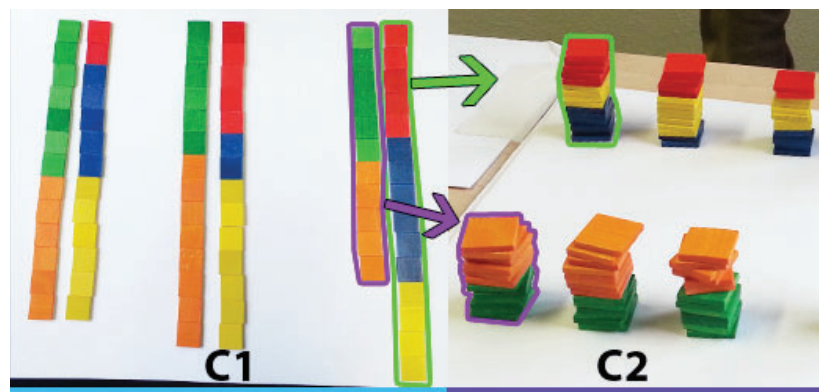
The overarching goal we set participants – helping a friend to improve his budget – inspired some participants to compute additional values such as total amount of expenses, aggregation of categories into mandatory and discretionary expenses, or explicit deltas between high-expense and low-expense months. Hence, among all the participants' visualizations, we observed several constructs which are not directly mapped to data, but are a result of a computation by the participant. We observed such behavior during all three tasks (A, B, C). In this section we describe four different tasks we observed: 6) Categorize, 7) Aggregate, 8) Compute new value, 9) Unitize.



6. Categorize:

A common data transformation task was the creation of meta-categories (5/12) by developing groupings that participants considered meaningful, e.g., grouping expense categories into essential and non-essential. Meta-categories were encoded in different ways, such as by *spatial arrangement* (4/12), by *colour semantics* (e.g., warm vs. cold colours) (1/12), *colour attribution* (2/12), and *spatial merges* (3/12). For example, participant 11 (Figure 8.18–C1/C2) used spatial arrangements to indicate two categories: “reducible expenses” (red, blue and yellow) and “irreducible expenses” (orange and green). These categories persisted throughout the study: first as stacked 2D lines (Figure 8.18–C1) then as stacked 3D bars (C2).

Figure 8.18: Computations: C1: aggregate by spatial merge, categorize by proximity; C2: same categories as in C1 but different arrangement of constructs.



Participant 8 (Figure 8.19) used colour semantics to express a similar grouping. She decided to SELECT warm COLOURS for “leisure expenses”, and cool colours for “necessary expenses”. This encoding allowed her to keep the original categories identifiable and comparable.

Figure 8.19: Participant 8, task C, Categorize: Colour semantic.



7. Aggregate:



Figure 8.20: Computing: information preserving during *aggregation*;

Aggregation was the most commonly performed computation (7/12). When participants aggregated data, they usually did so within one dimension, i.e., aggregating all expenses per month or all expenses per category. Aggregation could be done in the data domain, by mentally adding the data, or in the visual domain, by **MERGING** constructs, or by **INCREASING** constructs during initial *loading*.

We also observed aggregation by **MERGING** without explicit awareness. For instance, participant 1 (Figure 8.21–A) *categorized* by **SELECTING** the same colour for a subset of categories during task A. Then during task B, she wanted to free space on the canvas and **MERGED** constructs representing different data categories (Figure 8.21–A2/A3). With this action she effectively *aggregated* the values for same-coloured categories. When asked during the interview what she did, she stated: “I just added the new month. I didn’t change anything else.”

Participant 2 performed an aggregation that preserved the underlying data. As shown in Figure 8.20–B, she **MERGED** all categories within a month similar to participant 1. However, she had assigned different colours to categories so that the separate categories were still visible. Additionally, her arrangement into one bar per month represented the aggregated value for total expense per month.

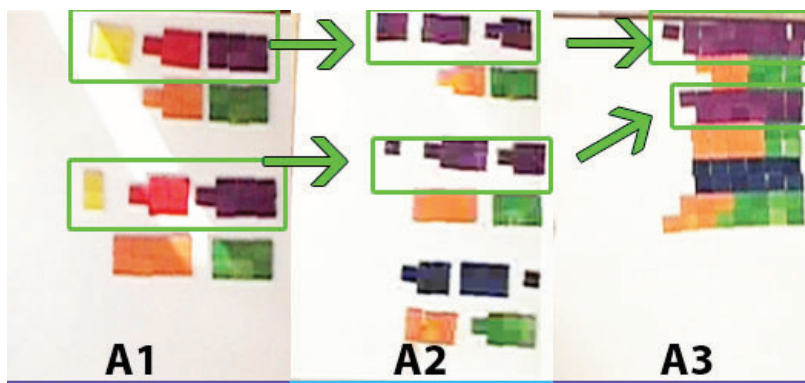


Figure 8.21: Computations: A1/2: categorize by colour assignment; A2/3: aggregation by merging constructs;

8. Compute new value:

Some participants (3/12) computed new values to provide additional goal related information (to help their friend with his budget). In Figure 8.6–A|2 and Figure 8.22, we can observe that participant 2 computes the monthly difference against the month with the lowest expenses (represented by blue tokens above each month). She explained in the interview that she did this because her visualisation previously did not allow her to see the difference between months. Similarly, participant 11 computed in task C the average of each expense category across months and built two piles in a column to the right to present this extra information (Figure 8.7–C|11).

Figure 8.22: computation of delta with another month in blue (highlighted);



9. Unitize:

Some of the participants (4/12) subdivided their constructs into meaningful units (e.g., $2 \times 2 \sim 100$). We observed two ways to perform this task: ORGANIZE into subconstructs, or ORGANIZE such that the width or height of a construct represented a meaningful unit, e.g., a line of 4 tokens within a bar represents 100 (Figure 8.23–E & Figure 8.7–B|8 and Figure 8.6B|2). Figure 8.15–2 shows the first approach where the participant ORGANIZES his constructs into squared subconstructs of four tokens. The second approach, changing the width of a construct, can be seen with participant 3 (Figure 8.6–B|3 to C|3). During the interview he explained that columns with a basis of 4 tokens facilitate the visual estimation of variables. Participant 8 (Figure 8.23–F & Figure 8.19 & Figure 8.7–8|A/B/C) employed an unexpected computation technique. After receiving the instructions for the study she stated it would be difficult to help the friend without knowing his overall budget (which was not provided). To compensate, she decided to impose a hypothetical budget based on the given expenses. To do so, she ARRANGED the constructs for all categories of

a month into a square, and she considered all tokens which did not fit the square as excess expenses. Priority inside the square was given to “needed expenses” the remaining space was filled with tokens from the “leisure” meta-category.

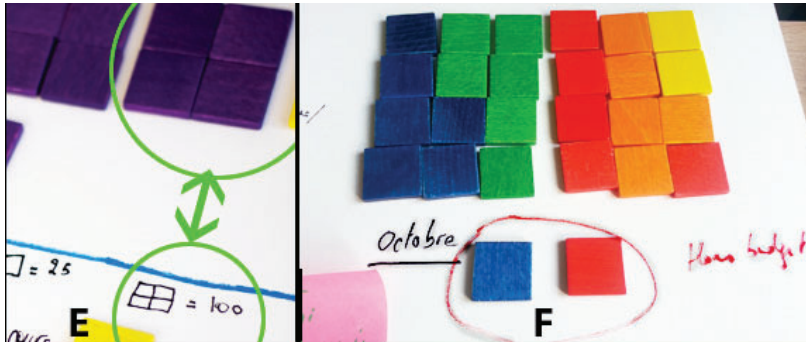


Figure 8.23: Computations: E: *unitized* construct ~ 100 ; F: *unitized and merged* construct to illustrate a month's budget.

8.4.3 Storytelling

During the interviews, we observe that some of the participants (5/12) spontaneously manipulated their visualizations to support their arguments. Here we list two tasks, highlighting and marking.

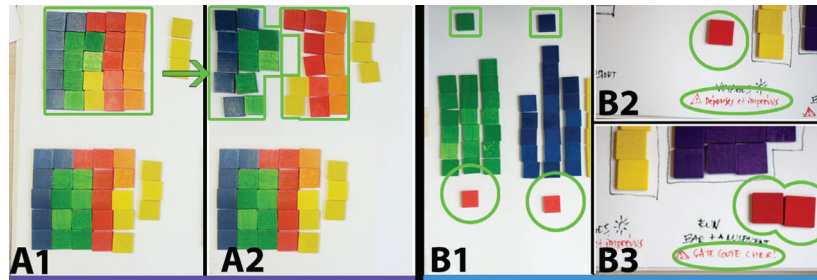
10. Highlight:

Some of the participants (4/12) supported their explanations during the interview with temporary modifications such as pointing, MOVING, SPLITTING and MERGING. For instance, participant 8 performed a modification that could be observed as the separation between cool coloured constructs (“mandatory expenses”) and warm coloured constructs (“fun expenses”) in Figure 8.24–A1 and A2. This modification supported what she was saying: “If we want to explain to our friend his expenses, we have to produce a synopsis to show what is the financial flexibility” (at this point she splits the construct to highlight this, see Figure 8.24–A2).

11. Marking:

One participant used tokens as markers, indicating points of interest (e.g., possible expenses to optimize). In Figure 8.24–B1, he placed red tokens at the bottom edge of two columns to *mark* problematic expenses in these categories. After the update for task B the point of interest changed. The participant termed it “an indicator” and declared that “It’s an indicator to [...] alert. It’s more an analytic pointer than just the data.”.

Figure 8.24: Storytelling: **A1**: unitized construct; **A2**: temporary SPLIT to explain the categories; **B1**: marking of categories exceeding budget; **B2/B3** marking and annotation how to optimize budget.



8.5 Discussion

In this section we discuss our findings in light of our initial questions about better understanding what the visual mapping process entails and how non-experts cope with the challenge of both developing and explaining their visualizations.

8.5.1 Unpacking the Visual Mapping Process

This study has revealed many details about the complexity and variability of the visual mapping process. From these we identified 11 elementary tasks people performed when constructing and communicating a visual representation of data (illustrated in Figure 8.8). We observed that people had similar general goals (construction, computation and storytelling) which transferred into similar logical tasks and led to performing the same type of physical actions. Thus these appear to be a representative set of basic goals, tasks, and actions which possibly generalize to other visual mapping processes and different data types. Further studies are necessary to determine generalizability. In this section we discuss our findings in light of our initial questions about better understanding what the visual mapping process entails and how non-experts cope with the challenge of both developing and explaining their visualizations.

8.5.2 Bottom-up and Top-down Procedures

Performed actions combined and recombined differently across participants. As illustrated by Figure 8.25, no specific pattern for the exact sequence of actions can be observed among the participants. All have their individual process with participants going back and forth between different types of actions throughout the creation process. Still, we did observe two distinct types of procedures. The most common one (10/12) consisted of starting from a single data case, then building progressively towards defining higher level structures, e.g., an axis. In the other case, people started by defining the dimensions and axis first and then populated them with data. We call the former a *bottom-up procedure* and the latter a *top-down procedure* (2/12).

In contrast to our observations, a large part of currently available commercial software tools enforce some form of workflow to guide

people through the visualisation authoring process. For example, many existing tools for non-experts are based on the use of pre-existing templates. People have to decide early on which template to pick. If their data is compatible, they might be able to switch between different templates, but usually no fluent transitions exist between the different template, and users need to re-orient themselves again after a change of visualisation technique / template.

With the constructive approach as used in this study, people were free to rearrange constructs as they wished. We hypothesize that any rearrangement action leaves their interpretation and understanding intact – at least as long as their constructs are not destroyed during the rearrangement. However, further studies are also needed.

8.5.3 The Ease of Constructing Visual Mapping

Previous work seems dichotomized on the ease of defining visual mappings – some emphasized their difficulty [Grammel et al., 2010] while others show counter examples [Chao et al., 2010]. A relevant question for the accessibility of InfoVis authoring tools in general is what makes one situation easier to master than another. Taking inspiration from the way mathematics is taught to pre-school children in kindergarten [Brosterman et al., 1997, Tarr, 2013], we wondered whether the same tools would also facilitate non-expert adults in creation of visualizations and in understanding of the underlying data.

Overall, none of the participants showed difficulty in creating their first visual representation. It took them less than 30 seconds to build a first construct. Most of the participants (10/12) declared that they would use a similar technique in the future. One of them said that she had already used a similar approach with real coins to plan future expenses. She explained: “A coin of 1€ equals 100¢ [...] I do that when I have big expenses to plan. I make a summary of the last 6 months [...] It helps me to make abstract things concrete [...] When I see the number 125 I absolutely don’t know what it represents. Here we can see the [...] relation between expenses”.

We believe that the tangible properties of our constructive authoring method afforded a more explicit visual mapping process that could emerge as a result of manipulating visual constructs. As op-

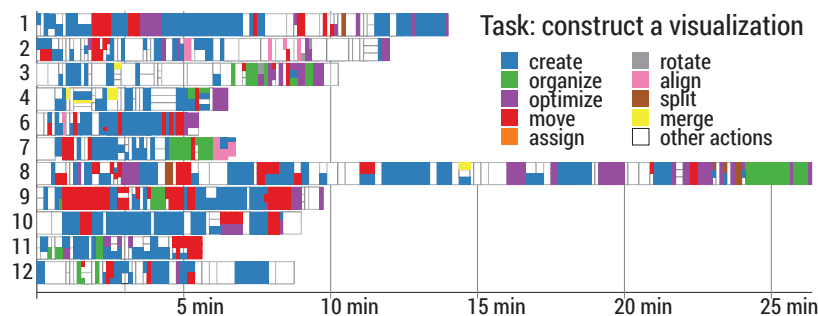


Figure 8.25: Sequences of actions during task A for all but participant 5.

posed to common authoring tools, no early decisions on visualisation techniques were necessary to get started. However, the pre-defined initial token value definition could have played a role here, and participants might have had more problems developing their visual mapping had they been completely free in their choice of token value.

As mentioned previously, the only barrier we observed and on which participants commented was the initial transformation from the data domain to the tangible and visual domain. We deliberately chose a data value to token mapping that was non-trivial (as opposed to simple mappings such as 1 token \sim 100) and required a little computation. Our interest here was to observe whether participants would transfer such computations as early as possible into the tangible/visual domain, e.g. by counting up in steps of 25 while accumulating tokens, or whether they would perform this transformation mentally (divide data value by 25) and then grasp the corresponding amount of tokens. This study demonstrates the use of complementary strategies [Kirsh, 1995] as part of constructive processes for visualisation purposes.

8.5.4 *The Richness of Visual Mappings*

While some participants simply recreated well-known visualizations such as bar charts, others developed unexpected diverse visual mappings. We observed that some participants – after starting with a standard design – subsequently customized their visual mappings to better fit their particular question, e.g., see the total spending per month, or to perceptually optimize, e.g., use cool colors for essential expenses and warm colors for life style related ones. Others strove for more visually compact representations which even led one participant to create a horizon graph representation [Reijner and Software, 2008] (Figure 8.13–3).

Overall we find that even though most of the visualizations resemble common chart types, they were customized by their creators to fit their specific interest. Furthermore, many people made use of the constructive properties – effortless manipulation of constructs – for storytelling purposes to support their argument during the interviews.

8.5.5 *Internalization of Data to Token Mappings*

We were interested in how far participants internalized the data to token mapping. During the interview, we observed participants speaking interchangeably about tokens and data while making enclosing or delimiting gestures, and while pointing to tokens. To inquire further, we systematically asked them two questions: (Q1) “What did you manipulate during your construction process?”, and, depending on their answer: (Q2) “What was the value (or meaning) of [the declared object]?”, by replacing “[declared object]” with their reply to Q1.

Most participants were surprised by this question and sometimes asked the facilitator to refine the question. Then, most of them replied

to Q₁ by referencing first the object and then the data (6/12 P.). Two replied with only the data or first the data and then the object. Four other participants (4/12 P.) spoke only about the object for Q₁, and then spoke about the data for Q₂. Some of them also used other words than “data” such as “numbers”, “unit”, or “expenses”, and for tokens, some used “wooden tiles” or “little colored squares”.

Even though the data collected came from a small population sample, and more research is needed to generalize our findings, we identify a clear awareness of participants for a tight coupling between the data and their tangible proxy, the wooden tiles. This coupling is so tight that the majority stated they manipulated “an object coupled to data” and used words such as “represent”.

8.6 Limitations

This study is a first attempt to unpack how people create visual representations. However, a single study cannot answer all open questions about this process and is necessarily limited by several factors, which require further investigation. We discuss limitations due to (i) properties of the tangible tokens, (ii) our study protocol, and (iii) our method.

8.6.1 Tangible tokens.

While the aim of this study was to remove possible tool biases introduced by specific authoring tools, our use of tangible tokens possibly introduced biases of its own. First, the physical properties of the tiles we provided as tokens limited the range of possible visual representations (Figure 8.26). We chose these wooden tiles since their use is already established as a suitable tool to teach simple mathematics to kindergarten children. Their flat design certainly limited the range of possible encodings and possibly suggested the creation of 2D representations. Still, for this first study we chose fluency of handling over expressiveness of tools. Furthermore, most screen-based visualizations are two-dimensional, hence we aimed to avoid introducing biases through construction with tools that suggest the creation of 3D structures. The variety of visual representations (see Figures 8.6 and 8.7) illustrates that the available tools were sufficiently expressive to support a range of different visual representations.

8.6.2 Study protocol.

Our study protocol could also have introduced some limitations. The initial instructions primed participants that “with numbers, changes are difficult to see over time”. Still, this motivational remark did not uniformly lead to equivalent visual mappings (category to colour and time spatially encoded). For instance, participants 3 & 5 chose not to represent time and aggregated over time throughout the entire study, while participant 4 did so initially but then changed for task B to encode time spatially. Concerning the category to colour mapping, we



Figure 8.26: Tokens of different shapes, colors, sizes.

also observed aggregations, i.e., assigning the same colour to different categories, as described in Figure 8.21. Our study design did not allow us to determine whether our instructions or previous exposure to visualizations had a larger influence on the created designs. Another limitation for the possible mappings was due to the pre-defined token-to-data unit mapping (1 token \sim 25). To avoid awkwardness in data manipulation, we rounded expenses to be multiples of 25. Such rounding of data values is not generalizable. More accurate mappings such as 1 token \sim 1 would increase the accuracy of the visualisation but also require much larger numbers of tokens.

8.6.3 *Method.*

While we report on rich data about the visual mapping process, where people transform data to visual representations, our understanding stops short of their internal processing. Our data provided us with verbal reports and videos of their actions. Thus we have evidence of the result of their thinking process and the resulting intermediate steps. It is from the combination of evidence of these individual steps with participants' self-reports that we have assembled the visual representational mapping process described in this article.

8.6.4 *Generalizability.*

This study is a qualitative observation of token-based visualisation construction by non-experts and offers detailed observations that reveal phenomena and behaviors within this process. This first step in the unpacking of visual mapping can be extended with other studies such as examining: (i) different mapping processes, (ii) other types of authoring tools, and (iii) different token type and materials that may well support other types of actions. However this study does show that non InfoVis experts did create valuable visual mappings without encountering the same barriers that they did with software authoring tools [Grammel et al., 2010]. While this study does not prove generalizability, it does suggest that creating environments where people can assemble their visualizations from data-linked tokens may be beneficial.

8.7 *Implications for Research and Design*

This study's main contribution to visualisation is an initial opening of the black box generally termed "visual representation" mapping in the InfoVis pipeline. This study reveals many processes that are internal to this step and invites further explorations that will continue to unpack these processes. It also demonstrates that, with this particular tangible environment our participants were readily capable of constructing visual representations. Our study results also suggest several design implications to further the democratization of InfoVis authoring tools with associated implications for on-going research.

8.7.1 *Supporting different construction procedures:*

Most visualisation authoring software requires first choosing a visualisation technique and then the data dimensions to visualize. This can be seen as a top-down procedure: the visualisation model needs to be selected before any data can be seen visually. In contrast, our study environment provided the opportunity to build visualizations bottom-up by manipulating a small part of the dataset and gradually defining an assembly model. With such an approach, data is visible from the very beginning, and the visualisation model develops iteratively during construction.

Open research questions: How do different visualisation authoring procedures impact the understanding of data? How does a bottom-up procedure transfer to different data types and tasks? What are the limits of a bottom-up procedure? How can we assure correctness of the result. Does this affect people's confidence in their constructions?

8.7.2 *Exploiting processing fluency:*

Processing fluency is the "the subjective experience of ease with which people process information" [Alter and Oppenheimer, 2009]. The method we used was originally designed 200 years ago to teach mathematics to non-literate kindergarten children [Brosterman et al., 1997]. The original intent for such a method was to simplify the way children can cognitively process abstract and complex ideas such as mathematics. Our work suggests now that people without skills in InfoVis can construct useful visual representations for themselves when offered a method with for which they already possess fluency. However, our study does not separate between the constructive and tangible aspects of the tool.

Open research questions: How do our results transfer to systems which are constructive but lack the tangible aspect? Would a constructive authoring tool implemented on an interactive touch surface provide similar benefits? How can we transform more complex InfoVis techniques into more fluent ones?

8.7.3 *Tangible design:*

The physical affordances of tokens likely affected people's actions and their visual designs. While the wooden tiles used in our study could be piled, participants found them slippery and mostly created 2D designs. Constructs were also easily destroyed during moving actions and required optimization actions afterwards. However, participants frequently moved constructs around using multifinger and bimanual gestures (cf. Figure 8.25). Such actions could be further simplified through "sticky edges" of tokens. Other materials such as Lego bricks would facilitate stacking and might have led to more 3D encodings. Materials with programmable properties [Gilpin et al., 2010, CMU & Intel Research, 2006] also seem promising for *constructive visualization* purposes.

Open research questions: Which material properties are most salient in their effect on constructive strategies? Which properties are most effective? How would the increased complexity of programmable properties affect people's proficiency with physical objects?

Chapter Summary

In this chapter, I presented an observational study designed to help us learn more about the visual mapping process and to explore the suitability of constructive authoring of visualizations for non-experts. More specifically, we provided our participants with a simple environment in which they could build visualizations out of coloured wooden tiles. We asked them, in the context of helping a friend with his budget, to create a visualization, to update it with new data and to explain their visualisation to their friend using notes and annotations.

Through observing their activities, we deconstructed the visual mapping process into three high level activities: construction, computation and storytelling. These were composed of several logical tasks: load data, build constructs, combine constructs, extend, correct, aggregate, categorize, compute new values, and unitize. The storytelling component was composed of highlighting and marking. These logical tasks were achieved by performing lower level actions as detailed in Section 8.4.

From this study our main contributions are:

- The details of the logical tasks and actions previously simply considered to be a "black box" that was called the visual mapping process.
- A model of the visual representation mapping process that describes the actions, their purposes and the interplay between them.
- The observation of highly diverse process action sequences, indicating that while people used the same actions they did not adhere to the same sequencing.
- The finding that in this tangible constructive environment non-experts were readily able to construct, update, and explain their

own visual representations in a short period of time.

The implications for the design of non-expert digital and tangible authoring tools are that more effort directed at supporting both bottom-up and top-down procedures to visualisation authoring would be beneficial. We suggest that the ease with which non-experts created visualizations in this tangible environment might be, at least in part, due to the possibility of seeing and performing the visual constructs develop during the assembly process. This study opens new research perspectives such as: (i) providing better tangible visualisation construction environments; (ii) creating digital visualisation construction environments; (iii) further investigating the visual mapping processes themselves.

Part Tree Summary

In the first part of this dissertation, I introduced a short history of token-based visual representations from antiquity to contemporary periods. Ullmer et al. [Ullmer et al., 2005] were the first to explore the possibility of using tokens as tangible devices to manipulate computers. They identified three manifestations, “interactive surfaces”, “token+constraint”, and “creative assembly.”

In the previous part of the dissertation, I focused on adapting token-based approaches to dynamic data visualisation for use by non-experts on screen-based devices. This resulted in a definition and generalization of token-based techniques for dynamic visualisation as “token+constraint”. These techniques were well-received by the general public when used successfully in different applications and deployed ‘in the wild’. However, even with a toolkit, they would not provide a means for people with no code proficiency to create their own dynamic visualizations. Given our goal of democratizing dynamic visualization, we also want to democratize the visual mapping of dynamic visualization.

In this third part of the dissertation, I would like to find a mean for non-expert to author dynamic visualization. I first present *constructive visualization* (Chapter 7), a token-based paradigm which has great potential in helping to realize the democratization of dynamic visualisation authoring. This paradigm is motivated by the benefits likely to emerge from the incremental process of construction. As tokens can be simply added and removed as needed, this type of environment provides expressive and dynamic adjustment freedom.

Constructive visualisation is a new paradigm (Chapter 7) which has not yet been formalized or studied (Table 7.1: page 144) in information visualization. This paradigm offers the possibility of providing people with the means to construct their own visualizations in a simple, expressive and dynamic way.

In the second chapter (Chapter 8) I chose to use physical tokens to explore this paradigm’s potential, as well as to unpack the visual mapping process. I study how a non-expert population creates, updates

and discusses their own token-based visualizations. Observation of their activities enabled us to deconstruct the visual mapping process with tangible tokens into three high-level activities. These activities were composed of different logical tasks as well as manipulative tasks. From these observations, I built a model (Section 8.4) that integrated these activities and tasks. In the chapter, I also discuss the implications of our findings for research on non-expert authoring tools and their design, as well as on our understanding of the visual mapping process.

Part IV

Conclusion and Perspectives

9

Conclusion and Perspectives

“The world is full of magical things patiently waiting for our wits to grow sharper.” - Bertrand Russell

I have proposed and studied a new paradigm to support the democratization of simple dynamic information visualization. This paradigm is motivated by the idea that, from a cognitive perspective, dynamic visualisation may benefit from constructive processes.

This paradigm is the result of multiple iterations of design, observation and the identification of research questions. Most of the research questions were addressed by designing, deploying and studying one or more projects, each of which enabled novel contributions. However, these contributions were also specific to particular contexts; some were completely restricted to a particular application domain, while others could be extended and/or generalized. Nevertheless, even where the project is focused on a particular application, this work contributes to the understanding of different aspects of dynamic visualisation for non-experts, an area of research that has been largely neglected.

The structure of this chapter is as follows. First I describe and summarize the approaches used to address the different research questions. Then I describe the contributions of the research. Finally, I conclude with a discussion of the perspectives and the broader implications of this research.

9.1 Progress on research problems

During the initial phases of this research, I considered the broad question: *“Can we empower non-experts in their use of visualisation by enabling this wider audience to (i) contribute to data stream analysis, and (ii) create their own visualizations”*. From this broad question I derived three research problems, as described in the introduction: **PROBLEM1** engaging non-experts in dynamic data analysis, and **PROBLEM1.2** visually encoding dynamic data for non-experts, and **PROBLEM2** allowing non-experts to author dynamic data visualizations. In this

section I provide a brief summary of how I addressed each of these problems.

9.1.1 PROBLEM 1: *Is it possible to engage a public into a useful dynamic data analysis process?*

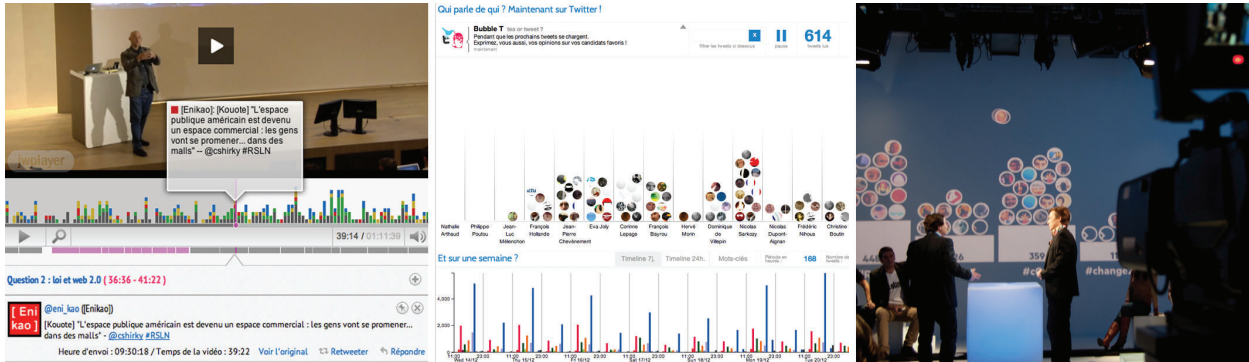


Figure 9.1: From Left to right: PolemicTweet meta-data player (Chapter 4), Bubble-T web-page, Bubble-TV photography of the TV show stage (Chapter 5).

I addressed this question by designing, deploying, and studying two different systems (PolemicTweet Chapter 4, and Bubble-TV Chapter 5) dedicated to engaging non-experts in dynamic data analysis. Both systems were deployed “in the wild”. But addressed different audience sizes—ranging from 20 to 300 for PolemicTweet, and from 10 000 to 700 000 for Bubble-T and Bubble-TV. Both systems consisted of a “social protocol” and dynamic visualisation to engage the audience in a dynamic tagging activity. The live deployment of these systems allowed me to observe and experiment with the design of the tools while they were operating.

I assessed the merits of these systems and the factors influencing adoption, participation, and use in a short-term deployment (for Bubble-TV) and a long-term deployment (for PolemicTweet). With the help of my colleagues, I deployed PolemicTweet for a wide range of different events (more than 30). Despite the multiple varying factors, the system was shown to be effective for (i) producing useful structured annotation and tagging, and (ii) engaging the audience of an event in a dynamic activity of analysis and tagging. The system provides a *proof of concept* that illustrates how audiences of non-experts can perform analysis tasks on dynamic data. Bubble-TV was an extension of PolemicTweet adapted to answer polls on Television. The deployment of Bubble-TV demonstrated the feasibility of scaling this approach to larger numbers of people and greater volumes of information in a shorter time period.

In summary, these two systems and their respective studies demonstrate the feasibility of engaging audiences of different sizes, in different contexts, and with different media, in useful dynamic data analysis.

9.1.2 PROBLEM 1.2: *How to visually represent dynamic data for non-experts?*



Figure 9.2: The Visual Sedimentation metaphor applied to a bar chart (left), a pie chart (center), and a bubble chart (right).

While designing PolemicTweet (Chapter 4), I saw the possibility of creating visual representations from discrete visual elements, called tokens. I identified several important properties of this approach with respect to dynamic visualization—such as the homothetic mapping, which retains the identity of each data chunk over time, and the ability to visually express incremental processes. These encouraged me to explore a token-based approach to construct dynamic data visualizations for non-expert users. For the visual design of the two subsequent systems, Bubble-T, and Bubble-TV (Chapter 5), I combined token-based visual encodings with other representation systems. In Bubble-T, I combined tokens with physical simulation to animate dynamic data updates in a predictable way. Then in Bubble-TV, I combined a visual token system with an area chart so that the token-based visual representation could be scaled to larger amounts of data. The positive outcomes of these combinations encouraged me to pursue this direction further.

To unify these different visual representation systems, I introduced a new generalizable design metaphor, *Visual Sedimentation* (Chapter 6), inspired by the physical process of sedimentation. I then demonstrated how this metaphor could be used to generate a wide variety of visualizations from data streams such as Twitter topics, websites like StackOverflow and Wikipedia, and concurrent versioning systems. I also described a variety of design variables that the metaphor implies and showed how varying them can produce a wide range of different visualizations. The metaphor provided guidelines for developing a toolkit and for defining a design space with vast expressive potential. The open-source toolkit enables designers and coders to implement novel visualizations easily and efficiently using this metaphor. Moreover, this toolkit can be used to explore the design space defined by Visual Sedimentation.

In summary, Visual Sedimentation provides a visual encoding system to represent dynamic data to non-experts. This visual encoding

system is based on a model combining (i) token-based encoding driven by a constrained physical simulation with (ii) area-chart visual encodings. This combination constitutes a new family of visualisation techniques all based on an easily comprehensible metaphor. I contribute a toolkit for implementing visualizations that use Visual Sedimentation and a design space that explores alternative ways of applying the metaphor.

9.1.3 PROBLEM2 How can we democratize dynamic visualisation authoring tools?

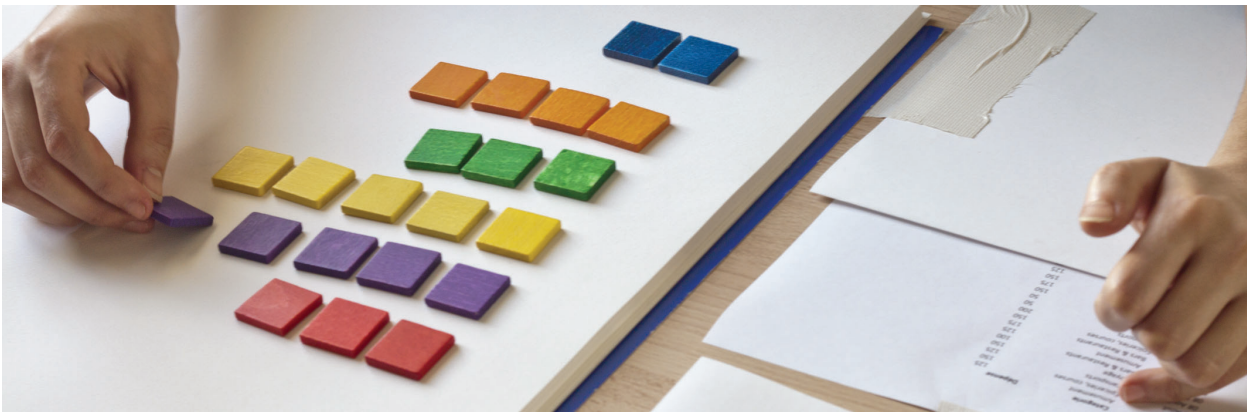


Figure 9.3: Constructing a visualisation with tokens: right hand positions tokens, left hand points to the corresponding data.

Although I provided a toolkit that makes it easier for developers to build visualizations that use Visual Sedimentation, it remains difficult for non-experts without coding proficiency to create their own dynamic visual representations. Previous research [Grammel et al., 2010] provides insights into the different barriers that non-experts encounter when using existing tools for this activity: (i) choosing the right data, (ii) choosing the visual mapping, (iii) interpreting.

To address these problems I decided to rethink non-expert authoring of dynamic information visualisation and the supporting tools by taking a physical token assembly perspective (Chapter 7). First I identified design challenges to democratizing the authoring of dynamic visualizations. Second, I defined a visualisation paradigm based on the analysis of existing real-world scenarios and inspired by constructivist theory. Third, I set up an observational study with tangible tokens (Chapter 8) to assess the applicability of this paradigm to non-expert authoring activities. Finally, I used this study to gain a more intricate understanding of the visual mapping processes that non-experts used.

This *constructive visualization* design paradigm (Chapter 7) is a first attempt to help democratize the authoring of dynamic visualizations. Constructive visualisation provides people with the means to construct visualizations on their own in a simple, expressive and flexible way. This approach is inspired by educational theory ([Brosterman

et al., 1997, Piaget and et Niestlé, 1948, Papert and Harel, 1991)) and also exemplified by people who created visualizations for their own needs with their own means ([Hunger, 2008, Wilson, 2012]). Based on these examples I defined the basic standard components and processes of a constructive approach to authoring visualizations. This first model provided me with foundations for an observational study of authoring behaviour.

Finally I performed a study (Chapter 8) to evaluate this paradigm and extend our understanding of this phenomenon. I gather empirical evidence, that non expert user could produce, update, discuss and annotate a visual mapping in a short periode of time with a simple, tangible constructive environment. Moreover, participants did not encounter the same barriers observed in previous work [Grammel et al., 2010]. By observing and analysing their activities, we learned about the visual mapping process and derived the first descriptive model of this activity. This model provides the foundations for further investigations of constructive visualization, and may also inform the design of new visualisation techniques.

In summary, I define a new paradigm to democratize dynamic visualisation authoring tools. While this paradigm has not been explored yet in the domain of information visualization, my study provides evidence that non-experts can overcome previously observed barriers with this approach. Our observations show that they can create, dynamically update, and discuss visual mappings if they use a constructive tool.

9.2 Contributions

This research has contributed to the design and understanding of token-based dynamic visual representation artefacts and phenomena related to them. Below I summarize the main contributions.

9.2.1 Major Contributions

- *A new design paradigm* for non-experts to author dynamic data visualizations. I defined the novel paradigm of constructive visualization, a design paradigm which provides a means for non-experts to author their own dynamic visualizations. The InfoVis community has already recognized that creating a simple tool for non-expert users is important, but the application of the constructive principle to dynamic information visualisation authoring and its application have never been examined before.
- *An operational description of constructive visualization*. I created an operational description of the *constructive visualization* authoring paradigm that can be used to design, build, and study new information visualisation systems. I built this conceptual framework based on observations of how people construct visualizations out of tokens. This description defines the major components and pro-

cesses required to author a *constructive visualization* environment. I also described how these components and processes can be applied in four real life scenarios.

- *Empirical evidences of constructive visualization application.* I gathered empirical evidence to support the claim that people can create, update, annotate and discuss visualizations that they have constructed by assembling tokens that have been mapped to data. I examined the visual mapping process in detail, delving into the logical tasks and activities involved. From these observations I extracted a model for describing the activities that constitute the visual mapping process. I also studied the sequence of activities adopted by the participants and observed high diversity.
- *A new family of dynamic visualisation techniques* supported by a metaphor, a toolkit and design space, and case studies. I used a metaphor to design a model for a new family of visualisation techniques that provides a visual encoding of dynamic data updates at multiple time scales. This metaphor is generative and allowed me to generate new classes of visualizations.
- *Experimental evidence of the feasibility of engaging non-expert users in dynamic data analysis.* I gathered experimental evidence to support the following: (i) that during an event, it is possible to engage and empower people to conduct analysis tasks on dynamic data, (ii) that these types of analyses on dynamic data can be effective, and (iii) that the data produced can be useful afterwards.

9.2.2 *Minor Contributions*

- *An initial description of the visual mapping process with tangible tokens.* I conducted a detailed exploration of the visual mapping process—a set of tasks and actions during the design of visualizations that had previously been treated as a “black box”. From my observations I reported on the diversity in the sequences of actions, showing that although people used the same actions, they did not adhere to the same orderings. From these observed actions I defined a model of the visual mapping process in a tangible constructive environment. This model describes the actions, their purposes, and the interplay between them when building visual representations.
- *The design and deployment in the wild of three specialised systems.* I designed three different (PolemicTweet, Bubble-T, Bubble-TV) information visualisation systems that used real-time crowd-sourcing and public participation. All three of these systems were composed of a token-based visual representation. These systems were deployed in several situations, ranging from medium-sized groups (20 people) to large-sized groups (300 people) to very large groups (700,000 people).

9.3 Reflection

The increasing amount of digital data and the associated complexity for processing it, both for individuals and organisations, is a major challenge for society. It is likely that this will result in cultural change as transformative as the invention of printing press¹. Recent events (e.g. Quantify self movement², Privacy³, Access⁴, and property⁵) testify to the multitude of overarching issues related to our data use in society.

The way we as a society, will organize this data processing is one of the major political issues for the twenty-first century. This challenge contains many facets, scientific, social, educational, economical, ecological, political, technical, among others. In this thesis we focus only on one tiny aspect, namely how to provide a generalizable and simple approach for non-experts to visualize and manipulate dynamic data by authoring dynamic visualizations.

In this dissertation we focus on this aspect of democratizing reading and authoring techniques. The motivation for selecting this particular aspect is our belief in a society where everyone can become data literate by accessing tools that are so simple to use that they can be understood by kindergarten children but still extremely powerful so that they can be flexibly applied to sophisticated problems. Providing such tools could be as powerful as the democratization of computing with the personal computer in the twenty century.

Our short history of token based tools for processing information (Chapter 3) revealed that the use of tokens provides an effective and fluent representation of data for people. This might be due to different properties such as countability, incrementability, manipulability, and possibly constructibility. However we do not yet have a comprehensive model of the cognitive mechanisms underlying this fluency, leaving us with an open avenue for future research. However we can identify that tokens provide more simplicity, freedom and dynamicity than previous visualization authoring paradigms.

The design space resulting from the combination of (i) digital technology and automation, (ii) a token-based representation system and (iii) a constructive environment, is extremely large. This design space has the potential to inspire powerful technologies that enable non-expert individuals and groups to encode and decode their own visual representations such as tools to communicate, analyse, categorize, and manipulate information.

¹ <http://www.ted.com/talks/clay-shirky-how-cellphones-twitter-facebook-can-make-history>

² <http://ec.europa.eu/digital-agenda/futurium/en/content/emergence-quantified-self>

³ <http://www.theguardian.com/world/2013/sep/05/nsa-gchq-encryption-codes-security>

⁴ <http://www.slate.com/articles/technology/technology/2013/01/aaron-swartz-jstor-mit-can-honor-the-internet-activist-by-fighting-to-make.html>

⁵ <http://www.econlib.org/library/Columns/y2003/Lessigcopyright.html>

9.4 Perspectives

This research is a first attempt to explore a constructive approach to dynamic visualisation aimed at non-experts. The paradigm I define raises more interesting open research questions than it answers. In the following section, I list some of them.

9.4.1 Unpacking Visual mapping

Visual mapping procedure is a process that can be found in nearly every InfoVis reference model [Card et al., 1999, Chi, 2000, Carpendale, 1999, Jansen and Dragicevic, 2013, Fry, 2000, Cottam et al., 2012]. This process is an important part of information visualisation phenomena. However, this process is not well understood, and has typically been treated as a “black box”. To my knowledge, prior to my work, no model of the actions underlying the process existed. In Chapter 8 I made a first attempt to explore the activities in this “black box”, although my inquiry was limited to a specific instrument (tangible tokens), a population with particular characteristics (non-experts), specific data (expense reports), and a defined setup. I believe further studies should be carried out to better understand how we, as humans, perform visual mapping activities. This knowledge would provide insights that might help us create more appropriate tools for authoring and reading visual artefacts. The following are some of the open research questions:

- How do visual mapping processes vary with different populations, procedures, data types, instruments, contexts, and tasks?
- Do people share common cognitive schemas for visual mapping?
- Is visual mapping an instrument-dependent process?
- How do instruments affect people’s ability to perform a visual mapping?

9.4.2 Constructive visualisation explorations, and evaluation

The *constructive visualization* paradigm and study was the first attempt to study visual representation generation from a constructivist point of view. My study suggests that this approach may provide strong benefits for non-experts. However, this work, in its current form, is not reliably generalizable. For it to be generalized, more research is required to understand the phenomenon and assess its strengths, weaknesses, and possible applications. The following is a list of some of the limitations of my work, which also provide directions for future work:

- Can *constructive visualization* be useful for increasing visual literacy proficiency?
- How do construction procedures impact understanding of data?

- Does the benefit of actively engaging in the construction process transfer to simply watching it?
- How can the constructive procedure be transferred to different visualizations types such as graph, matrices and others?
- Does the construction process affect people's confidence in the visual representation produced?
- Do my results transfer to systems, which are constructive but not tangible or physical?
- Would a constructive authoring tool implemented on an interactive desktop provide similar benefits?
- Is it possible to automate the tedious task of *constructive visualization* while retaining the cognitive benefits?

9.4.3 *Contributive visualisation*

All three initial systems (PolemicTweet, Bubble-T, and Bubble-TV) were driven by data from Twitter feeds. However, those particular streams did not exist when the visualizations were being designed. Indeed, for PolemicTweet and Bubble-TV, the visualizations were themselves used to engage users and encourage them to produce the data. Most existing tools that use the InfoVis reference model [Card et al., 1999, Chi, 2000, Carpendale, 1999, Jansen and Dragicevic, 2013] assume that the data exists prior to the visualisation process. Only Fry [Fry, 2000] and Cottam et al. [Cottam et al., 2012], who considered introducing data-stream in the pipeline, recognized the acquisition of data as part of the process. However, as my three initial systems (as well as other previous work [Huang and Pilhofer, 2012, Bird and Rogers, 2010]) testify, in many cases visualisation itself can be used as an engagement and input tool for gathering data that does not yet exist. These data are gathered as small contributions from the community. Due to constraints in time, I was not able to explore this aspect of visualizations more thoroughly. However, I see much potential for research and a number of questions still remain to be answered, including:

- How does this type of visualisation impact the data input?
- Can we use this process for more complex data?
- What is the design space of this type of system?

9.5 *Conclusion*

Advances in design and science will require progresses in visualization. The community of people authoring as well as interpreting visualizations will become more and more diverse, and the datasets will grow in complexity, velocity, and diversity. This dissertation is a first step towards addressing these future challenges. At the

heart of my contribution is a novel visualisation authoring paradigm—*constructive visualization*. This paradigm reveals new perspectives on the visualisation design process. Most previous visualisation creation paradigms separated the process of creating the data representation from the development of interactions to suit particular data needs and tasks, with the former taking place before the latter. The paradigm introduced in this dissertation is different. Instead, the focus is on creating an interactive environment where people can assemble and disassemble, from modular visually-linked-data units: visualizations that directly fit their needs. The design space is large and still unexplored. However I have begun to scratch the surface by designing useful applications based on these ideas. The toolkit, models, and examples I supply allow further designs and techniques to be generated. They also raise further questions to be addressed by future experiments and studies. I am deeply committed to the idea of empowering non-experts to analyse and construct dynamic information visualizations. As data becomes a major part of our daily lives, I believe we can create fluent tools to empower all members of the population to make sense of it. In summary, I hope this thesis makes a significant contribution to the design and understanding of dynamic visual representations for people without information visualisation expertise.

A

Additional Materials

A.1 Lexicon

Assembly model : An assembly model is the process by which a construct is created. This process can involve different types of activities such as construction and deconstruction.

This notion is discussed in: Chapter 7 page 139 and Chapter 8 page 161.

Authoring : From the verb author. In the sense “be the author of”.

This notion is discussed in: Part III.

Construction : Is the process of assembling elements to generate a new whole. The etymological roots of the term ‘construction’ are particularly interesting in the context of this dissertation. It comes from the latin verb “constrūo”, meaning “pile up by layers”¹, and oxford dictionaries translated it into “heap together”².

This notion is discussed in: Part III page 137

Construct : A construct is the result of the assembly of two or more tokens, as well as the nested assembly of two or more sub-constructs of tokens.

This notion is discussed in: Chapter 8 page 161.

Datastream : are data that change over time and are transmitted by network connections.

This notion is discussed in: Chapter 2 page 21 and Part II.

Dynamic : The Oxford dictionaries’³ definition of dynamic: “(Of a process or system) characterized by constant change, activity, or progress”. In this dissertation we regularly use the term dynamic in talking about data and visualization. For additional clarity we propose to define these different word combinations.

Dynamic data : are data that change over time. Most of the time these changes concern new elements, but sometimes the structure of the data might also change. It is also

¹ Gaffiot Dictionnaire : <http://www.lexilogos.com/latin/gaffiot.php?q=constructus>

² Oxford dictionaries: <http://www.oxforddictionaries.com/definition/english/construction?q=construction>

³ <http://www.oxforddictionaries.com/definition/english/dynamic?q=dynamic>

important not to conflate changes in the data themselves (which make them dynamic) and the transmission of these changes.

Dynamic analysis : analysis of dynamic data.

Dynamic visualization : “visualization that changes over time”. This definition is provide by [Cottam et al. \[2012\]](#).

Presentation : A presentation is the act of displaying a representation, emphasizing and organizing areas of interest.

Representation : We use the definition of Marr [[Marr, 1982](#)],“ A representation is a formal system for making explicit certain entities or types of information . ”

non-experts : People (casual users, novices, data experts) who have not been trained in information visualization. This definition is inspired by [Grammel \[2012\]](#) previous work.

Token : A token is an atomic unit to which information or data has been mapped.

History of this artefact is presented in Chapter 3 page 33. Artefacts using this notion are discussed, as a visual element in Part II, as a tangible element in Part III.

A.2 Studying Constructive Visualization

This section contains additional material for facilitating the analysis, and replication of the study presented Chapter 8.

A.2.1 Dataset

Listing A.1: Translation of the observer script in Markdown format

1	Month;	Category;	Amount
2	08 August;	Amusement;	125
3	08 August;	Bars & Restaurants;	100
4	08 August;	Epicerries, courses;	150
5	08 August;	Transports;	125
6	08 August;	Voyage;	150
7	09 September;	Amusement;	125
8	09 September;	Bars & Restaurants;	175
9	09 September;	Epicerries, courses;	150
10	09 September;	Transports;	200
11	09 September;	Voyage;	50
12	10 October;	Amusement;	50
13	10 October;	Bars & Restaurants;	150
14	10 October;	Epicerries, courses;	175
15	10 October;	Transports;	150
16	10 October;	Voyage;	125
17	11 November;	Amusement;	75
18	11 November;	Bars & Restaurants;	175
19	11 November;	Epicerries, courses;	225
20	11 November;	Transports;	125
21	11 November;	Voyage;	300

A.2.2 Potocol

Listing A.2: English translation of the observer script

```

1 # Welcome,
2
3 We are studying how visualizations can be constructed
4 from simple building blocks and similar elements combined,
5 for this, we will ask you to create a visualisation from block.
6
7
8
9 ### 1. AGREEMENT
10
11 * Before the experiment, I have to ask you to read and sign this consent form.
12 This form said and formalised that we agree on:
13     * 1, the uses of the data produced during the experience
14     (video, photos, visualisation...),
15     * 2, on the conditions of the experiment.
16 * Thank you.
17 * Operator: verify if all the field are filled.
18 * Operator: Sign and give one version to the participant.
19 * Operator: Create an ID for the participant in the ID file.
20
21
22
23 ### 2. DEMOGRAPHIC form
24 * Your ID for this experimentation is (give him a post it with the ID)
25 * Also, Could you fill this demographic form, on this computer.
26 [ DEMOGRAPHIC QUESTIONNAIRE ]
27 * Don't hesitate to ask me if something is not clear.
28 * Thank you.
29
30
31


```

32 ### 3. TASK A
33
34 * Now let's start the first phases of the experiment.
35 * Record the time and start the cameras.
36 * Let's go to this table if you want.
37 * Could you please attach this mic to your collar ?
38 * Thank you
39 * On this table you could find:
40 * A - some expenses data from a bank account
41 * B - some tokens
42 * C - a correspondence table between the unit of the expenses and the tokens.
43
44 * A friend came to you, and he give you this expenses
45 data sheet on several month. Your friend has some trouble to manage his expenses.
46 He came, to ask your help?
47
48 * Here (point the data sheet) it's difficult to see the change over time,
49 it's more simple if we can see it.
50 * In order to understand the expenses of your friend
51 and to be able to give him some advices,
52 you have decided to create a visualisation from these tokens.
53 * Could you please create this visualization?
54 And let me know when you have finish.
55 * Start the stopwatch, and stop it when the participant
56 declare to complete the task.
57 * Write the time in the appropriate datasheet.
58 * Take pictures form the top and the side
59
60
61
62 ### 4. INTERVIEW 1
63
64 The participant declared is done.
65 Let's make a quick interview discuss what you have done.
66
67 * take a picture form the top and the side
68 * Start the interview (DOC INTERVIEW 1).
69 * Take pictures form the top and the side
70
71
72
73 ### 5. TASK B
74
75 * Take a picture form the top and the side
76 * Your friend came back quickly,
77 he bring you the expenses of the next month (November). (show it)
78 * Could you update the visualisation according to this new data?
79 Please let me know when you are done.
80 * Start the stopwatch.
81 * Take pictures form the top and the side
82
83
84
85 ### 6. INTERVIEW 2
86
87 The participant declared to be done.
88
89 * Write the time in the appropriate datasheet.
90 * Take a picture form the top and the side
91 * Let's make an interview to discuss the results.
92 * Start the after update interview (DOC INTERVIEW 2).
93 * Take pictures form the top and the side
94
95
96
97 ### 7. TASK C
98
99 So your friend didn't comeback, and you must leave.

100 So you decide to give him a note to explain what you did, and what you discovered.
101
102 Here some felt pen and some post it,
103 you can write on the paper under your visualization, annotate it,
104 modify the arrangement, do what you want to do.
105
106 The goals of this note is when your friend will be back, to allow him to:
107 * read the visualisation
108 * understand it
109 * and remake it if he needs to
110
111 You can also explain him what you think about his expenses problems.
112 Let me know when you are done.
113
114
115
116 ### 8. POST-EXPERIMENT FORM
117
118 The participant declared to be done.
119
120 * Take pictures form the top and the side
121 * Thank for your participation, to finalize the experience
122 you have to fill up a last questionnaire.
123
124 [END QUESTIONNAIRE]
125
126
127 ### 9. REWARD
128
129 Thank, if you want to reward you for the time you spent
130 on this task, I can give you a Amazon gift ticket.
131
132 * If you want it, please can you fill this questionnaire that I could be reimburse.
133 [REWARD QUESTIONNAIRE]

A.2.3 Consent form, Questionnaires, and Interview Guidelines

Figure A.1: Consent form page 1, french version.



Ethics application – REB13-0755
Formulaire de consentement
 Institut de recherche et d'innovation du Centre Pompidou
 4 rue Aubry le Boucher,
 75004 Paris

Bonjour

Nom des chercheurs, Université, Département, Téléphone & Email:
 Samuel Huron, Responsable du design – IRI – Institut de recherche et d'innovation du centre pompidou, et doctorant – INRIA équipe AVIZ. Samuel.huron@cybunk.com

Titre du projet:
 Constructing Visualization

Ce formulaire de consentement, dont une copie vous a été remise, n'est qu'une partie du processus de consentement éclairé. Si vous voulez plus de détails sur quelque chose mentionné ici ou informations ne figurent pas ici, vous ne devez vous sentir libre de la demander. S'il vous plaît, prenez le temps de lire attentivement et de comprendre toute les informations contenu dans ce document.

Sujet de l'étude :

Nous étudions dans quelle mesure les visualisations peuvent être construits à partir de blocs de construction simples et d'autres éléments combinables similaires.

Qu'est ce qui vous sera demandé de faire ?

Au début de l'étude, vous serez invité à fournir des contacts et des informations démographiques. Puis un ensemble de données vous seront présentée, ces données seront associé à un ensemble de blocs de construction physiques (ceux-ci peuvent être des choses telles que des blocs de bois, des carreaux colorés, ou des blocs lego). La correspondance entre les données et les blocs physiques vous sera expliquée. Après cela, vous avez la possibilité de construire une visualisation de blocs de construction donnés. Cette visualisation représentera l'ensemble des données fictives. Vous pouvez travailler avec les blocs de construction jusqu'à ce que vous soyez satisfait de la visualisation que vous avez créée.


Durant l'étude, vous pouvez être invité à créer votre propre visualisation seul ou avec un petit groupe d'autres personnes. L'ensemble du processus sera filmé. Des photographies des visualisations que vous avez faites seront présent.

Après que vous ayez créé une visualisation qui vous convient, nous procéderons à une interview semi-structurée avec vous, ou avec le groupe s'il s'agissait d'un projet de groupe. A la fin de l'étude, vous serez invité à remplir un bref questionnaire sur votre expérience.

Votre participation est entièrement volontaire. Vous pouvez refuser de participer ou pouvez décidé de vous retirer de l'étude à tout moment sans pénalité en indiquant votre souhait aux chercheurs.

Protocol title: Constructing Visualization
Ethics application – REB13-0755| PI: Samuel Huron | 13/08/2013 | V 0.02

Figure A.2: Consent form page 2, french version.



Ethics application – REB13-0755
Formulaire de consentement
 Institut de recherche et d'innovation du Centre Pompidou
 4 rue Aubry le Boucher,
 75004 Paris

Quelle genre d'informations personnel vont être collecter ?

Si vous acceptez d'y participer, vous serez invité à fournir certaines coordonnées personnelles et les données démographiques, telles que le nom, l'adresse e-mail et votre âge.

Vous devez choisir entre plusieurs options si vous décidez de prendre part à cette recherche. Vous pouvez toute les choisir, ou juste certaines ou aucun d'entre eux. S'il vous plaît mettez une croix pour accorder votre autorisation aux chercheurs sur les lignes correspondante (s):

Je donne ma permission pour l'enregistrement audio	Yes: <input type="checkbox"/> No: <input type="checkbox"/>
Je donne ma permission pour être filmer	Yes: <input type="checkbox"/> No: <input type="checkbox"/>
Je donne ma permission afin que les visualisations que j'ai créées soient photographiées.	Yes: <input type="checkbox"/> No: <input type="checkbox"/>
Je donne ma permission pour que des citations anonymes de ce que j'ai dis durant l'étude sois publié dans la présentation des résultats de recherche.	Yes: <input type="checkbox"/> No: <input type="checkbox"/>
Je donne ma permission pour que des extraits cours de vidéo enregistré puisse être publié durant la présentation des résultats de recherche.	Yes: <input type="checkbox"/> No: <input type="checkbox"/>

Quels sont les risques et des bénéfiques si je participe à cette recherche?

Il n'y a pas de préjudices connus associés à votre participation à cette recherche. L'avantage pour vous en tant que participant est l'occasion de créer de nouvelles visualisations en développant une représentation des données vous-même. Vous serez rémunéré pour votre temps. Vous pouvez demander aux chercheurs des questions au sujet de cette étude à tout moment avant, pendant ou après l'étude.

Avec votre permission, nous pouvons utiliser des extraits vidéo ou des images fixes de la vidéo dans des présentations et des publications, mais cela ne peut se produire si vous avez donné votre accord ci-dessus. Tous les clips ou images fixes de la vidéo ne seront pas associées à votre nom ou a vos coordonnées. Si le consentement est donné aux présentes clips vidéo et / ou photographies, aucune anonymat peut être garanti et vous pourriez être clairement reconnaissables en tant que participant à cette étude. Veuillez noter que les images une fois photographiés ou filmés sont affichés dans un forum public, les chercheurs n'ont aucun contrôle sur l'utilisation future par d'autres qui peuvent copier ces images et reposter dans d'autres formats ou les contextes , y compris peut-être sur l'internet.

Qu'advient il des informations que je fournis ?

Protocol title: Constructing Visualization
Ethics application – REB13-0755| PI: Samuel Huron | 13/08/2013 | V 0.02

k1 - Questionnaire de début d'expérience
**Obligatoire*

1. **Id :** *

.....

2. **Date de naissance ***
Exemple : 15 décembre 2012

.....

3. **Sexe :** *
Une seule réponse possible.

Homme
 Femme

Education

4. **Quel est votre niveau d'éducation (nombre d'années avant ou après le bac)? ***

.....

5. **Quel est votre niveau en programmation informatique ? ***
Une seule réponse possible.

1 2 3 4 5
 Débutant Spécialiste

6. **Quel est votre profession actuel ? ***

.....

7. **Quel est votre niveau en dessin ? ***
Une seule réponse possible.

1 2 3 4 5
 Débutant Spécialiste

8. **Quel est le niveau de votre dernier diplôme obtenu ? (licence, ingénieur, master, doctorat) ***

.....

9. **Avez vous l'habitude de manipuler des données informatiques quantitatives ? ***
Une seule réponse possible.

1 2 3 4 5
 Débutant Spécialiste

10. **Dans quelle domaine avez vous fait vos études ? ***

.....

Visualisation 1

11. **Comment définissez vous une visualisation de donnée ? ***

.....

12. **Sélectionnez l'un des endroits où vous avez vu des visualisations de données dernièrement ***
Plusieurs réponses possibles.

dans une application (iphone, tablette tactile, PC)
 dans une page web
 dans les journaux
 à la télévision
 dans un livre
 Autre :

13. **En général, combien de fois voyez-vous des visualisations de données ? ***
*(Par exemple, dans une application, sur le web ou dans un journal, télévision, autres ...)?
 Plusieurs réponses possibles.*

jamais
 une fois par ans
 une fois par mois
 une fois par semaine
 tout les jours

14. **En général, à quelle fréquence faites-vous des visualisations (par exemple, comme une esquisse, comme un histogramme, un camembert ou un graphique sous Excel ou d'autres outils ...) ? ***
Plusieurs réponses possibles.

Jamais
 une fois par ans
 une fois par mois
 une fois par semaine
 tout les jours

15. **Quel est l'outil que vous utilisez / utiliserez si vous devez créer une visualisation ? ***

.....

16. **Utilisez vous un de ces outils pour faire une visualisation ? ***
Plusieurs réponses possibles.

Papier et crayon (dessin, esquisse)
 Logiciel de tableur comme Microsoft excel ou similaire
 Programmation informatique
 Autre :

17. **Pouvez-vous évaluer la facilité ou la difficulté relative à la construction d'une visualisation avec l'outil dont vous êtes le plus familier ? ***
Une seule réponse possible.

Très facile
 Facile
 Normale
 Difficile
 Très Difficile
 Je n'ai jamais utilisé d'outils pour cela

18. **Pouvez-vous évaluer la facilité / difficulté pour faire une visualisation "comme vous le désirez" avec l'outil avec lequel vous êtes le plus familier ? (Au choix) ***
Une seule réponse possible.

Très facile
 Facile
 Normale
 Difficile
 Très Difficile
 Je n'ai jamais utilisé d'outils pour cela

19. **Pouvez-vous nous indiquer dans quelle mesure l'outil avec lequel vous êtes familier vous permet de "modifier la visualisation" que vous avez faite ? (Au choix) ***
Une seule réponse possible.

Très facile
 Facile
 Normale
 Difficile
 Très Difficile
 Je n'ai jamais utilisé d'outils pour cela

20. **Combien de visualisation avez vous vue durant les 12 derniers mois ? ***

.....

21. **Quel est votre niveau en création de visualisation ? ***
Une seule réponse possible.

1 2 3 4 5
 Débutant Spécialiste

22. **Combien de visualisation avez vous créé durant les 12 derniers mois ? ***

.....

23. **Quel est votre habitude de lire une visualisation ? ***
Une seule réponse possible.

1 2 3 4 5
 Débutant Spécialiste



Créé par


Figure A.3: Pre experiment questionnaire, french version.

Figure A.4: First interview guideline, french version.



Ethics application – REB13-0755
Interview

Institut de recherche et d'innovation du Centre Pompidou
4 rue Aubry le Boucher,
75004 Paris

EXEMPLE DE QUESTIONS
Pour l'entretien (1) semi structuré AVANT mise à jour.

Titre du projet : Etude de la construction de visualisation


Exemple de questions

1. Pouvez vous m'expliquer ce que vous avez fait ?
2. Pouvez vous m'expliquer comment lire cette visualisation ?
3. Avez vous apprécié créer votre visualisation ? Pouvez vous expliquer ?
4. Souhaitez vous utiliser cette technique dans le future? Pouvez vous expliquer ?
5. Pour vous, qu'elles sont les bénéfices de créer votre propre visualisation?
6. Que manipulez vous durant votre processus de construction?
 - a. Qu'elle était la valeur (signification) de [objet déclaré pour manipulation]?
7. Avez vous appris quelque chose à propos des données durant le processus ? Pouvez vous expliquer?
8. Pouvez vous, vous imaginez utiliser une technique similaire avec d'autres données? Pourquoi?
9. Y a t-il des choses que vous n'avez pas aimé? Vous aimé?
10. Y at-il des choses qui vous ont limité dans ce que vous vouliez faire?
11. Pouvez-vous imaginer des façons dont ces questions pourraient être minimisés?
12. Que va tu dire a ton amis quand il va revenir ?
13. Souhaitez vous ajouter quelque chose ?

Titre de protocol: Constructing Visualization | PI: Samuel Huron | 13/08/2013 | V 0.02

Page 1/1

Figure A.5: Second interview guideline, french version.



Ethics application – REB13-0755
Interview

Institut de recherche et d'innovation du Centre Pompidou
4 rue Aubry le Boucher,
75004 Paris

EXEMPLE DE QUESTIONS
Pour l'entretien (2) semi structuré APRES mise à jour.

Titre du projet : Etude de la construction de visualisation

Exemple de questions

1. Pouvez vous décrire ce que vous avez fait durant cette mise à jour ?
2. Est ce que la mise a jour vous à amener à repenser la composition que vous avez faites précédemment ?
3. Que manipulez vous durant votre processus de mise a jour ?
 - a. Qu'elle était la valeur (signification) de [objet déclaré pour manipulation]?
4. Avez vous appris quelque chose à propos des données durant le processus de mise a jour ? Pouvez vous expliquer?
5. Y a-t-il des choses que vous n'avez pas aimées? Vous avez aimées ?
6. Pouvez vous noter le degré facilité ou complexité de cette mise a jour sur une échelle de 1 a 5, 1 étant le plus complexe 5 le plus simple ?
7. Y a-t-il des choses qui vous ont limité dans ce que vous vouliez faire? Si oui quoi ?
8. Pouvez-vous imaginer des façons dont ces problèmes pourraient être minimisés?
9. Que vas tu dire a ton ami quand il va revenir ?
10. Souhaitez vous ajouter quelque chose ?

Titre de protocol: Constructing Visualization | PI: Samuel Huron | 13/08/2013 | V 0.02

Page 1/1

k1 - Questionnaire de fin d'expérience
**Obligatoire*

1. **Id :**

2. **Avez vous apprécié construire votre visualisation ? ***
Une seule réponse possible.

Pas du tout d'accord
 Pas d'accord
 Ni en désaccord ni d'accord
 D'accord
 Tout a fait d'accord

3. **Est ce qu'il y a des aspects de cette activité qui vous donne envie de la refaire ? ***
Une seule réponse possible.

Pas du tout d'accord
 Pas d'accord
 Ni en désaccord ni d'accord
 D'accord
 Tout a fait d'accord

4. **Pouvez vous vous imaginer construire une visualisation de cette manière mais avec d'autres données ? ***
Une seule réponse possible.

Pas du tout d'accord
 Pas d'accord
 Ni en désaccord ni d'accord
 D'accord
 Tout a fait d'accord

5. **Qu'elle mots proposez vous pour décrire la construction d'une visualisation avec cette technique ? ***
Proposez des mots séparé par une virgule.

.....

6. **Qu'elle mots utiliseriez vous pour décrire la construction d'une visualisation avec cette technique ? ***
Plusieurs réponses possibles.

Facile
 Difficile
 Amusant
 Frustrant
 Excitant
 Créatif
 Epuisant
 Fascinant
 Intrigant
 Autre :

7. **Pouvez vous noter le degré de difficulté que vous avez éprouvé pour construire votre visualisation? ***
(Choisissez une réponse)
Une seule réponse possible.

Facile
 Modéré
 Normal
 Difficile
 Très difficile

8. **Pouvez vous noter le degré de difficulté que vous avez éprouvé pour construire une visualisation **selon votre propre idée**? ***
(Choisissez une réponse)
Une seule réponse possible.

Facile
 Modéré
 Normal
 Difficile
 Très difficile
 Je n'en utilise pas

9. **Pouvez vous noter le degré de difficulté de modification de la visualisation que vous avez créé ? ***
(Choisissez une réponse)
Une seule réponse possible.

Facile
 Modéré
 Normal
 Difficile
 Très difficile
 Je n'en utilise pas

10. **Si vous utilisez déjà un outil pour visualiser des données, est ce que celui ci est plus ou moins **amusant** ? ***
Une seule réponse possible.

Beaucoup plus
 Plus
 Pareille
 Moins
 Beaucoup moins
 Je n'en utilise pas

11. **Si vous utilisez déjà un outil pour visualiser des données, est ce que celui ci est plus ou moins **expressif** ? ***
Une seule réponse possible.

Beaucoup plus
 Plus
 Pareille
 Moins
 Beaucoup moins
 Je n'en utilise pas

12. **Si vous utilisez déjà un outil pour visualiser des données, est ce que celui ci est plus ou moins **facile à manipuler** ? ***
Une seule réponse possible.

Beaucoup plus
 Plus
 Pareille
 Moins
 Beaucoup moins
 Je n'en utilise pas

13. **Si vous utilisez déjà un outil pour visualiser des données, est ce que celui ci est plus ou moins **facile à mettre à jour** ? ***
Une seule réponse possible.

Beaucoup plus
 Plus
 Pareille
 Moins
 Beaucoup moins
 Je n'en utilise pas

14. **Si vous utilisez déjà un outil pour visualiser des données, est ce que celui ci est plus ou moins **efficace pour explorer les données** ? ***
Une seule réponse possible.

Beaucoup plus
 Plus
 Pareille
 Moins
 Beaucoup moins
 Je n'en utilise pas


Formé par


Figure A.6: Post experiment questionnaire, french version.

B

Bibliography

Gregory D. Abowd, Matthias Gauger, and Andreas Lachenmann. The family video archive: an annotation and browsing environment for home movies. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, MIR '03, pages 1–8, New York, NY, USA, 2003. ACM. URL <http://doi.acm.org/10.1145/973264.973266>.

Anand Agarawala and Ravin Balakrishnan. Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 1283–1292, New York, NY, USA, 2006. ACM. URL <http://doi.acm.org/10.1145/1124772.1124965>.

Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. Visualizing time-oriented data—a systematic view. *Comput. Graph.*, 31(3):401–409, June 2007. URL <http://dx.doi.org/10.1016/j.cag.2007.01.030>.

KRISTIN Aleeb-Lunddberg. Kindergarten mathematics laboratory nineteenth-century fashion. *The Arithmetic Teacher*, 17(5):372–386, 1970. URL <http://www.jstor.org/stable/41186220>.

Danyel Fisher Miriah Meyer Alex Bigelow, Steven Drucker. Reflections on how designers design with data. In *Proc. of the ACM International Conference on Advanced Visual Interfaces (AVI)*. ACM, 2014. URL <http://research.microsoft.com/pubs/217732/design-vis.pdf>.

C. Allen. The use of video in organizational studies. *SIGCHI Bull.*, 21(2):115–117, October 1989. URL <http://doi.acm.org/10.1145/70609.70630>.

Adam L Alter and Daniel M Oppenheimer. Uniting the tribes of fluency to form a metacognitive nation. *Personality and Social Psychology Review*, 13(3):219–235, 2009. URL <http://web.princeton.edu/sites/opplab/papers/alteropp09.pdf>.

R.A. Amar and J.T. Stasko. Knowledge precepts for design and evaluation of information visualizations. *IEEE TVCG*, 11(4):432–442, 2005. ISSN 1077-2626. DOI: 10.1109/TVCG.2005.63. URL <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1432689>.

David Anderson, Jonathan S. Yedidia, James L. Frankel, Joe Marks, Aseem Agarwala, Paul Beardsley, Jessica Hodgins, Darren Leigh, Kathy Ryall, and Eddie Sullivan. Tangible interaction + graphical interpretation. In *Proc. of SIGGRAPH*, pages 393–402, NY, USA, 2000. ACM. URL <http://dl.acm.org/citation.cfm?id=344779.344960>.

Rudolf Arnheim. *Visual thinking*. UofC Press, 1969. URL <http://books.google.fr/books/about/?id=rU4gAQAAIAAJ>.

Olivier Aubert and Yannick Prié. Advene: Active reading through hypervideo. In *Proceedings of the Sixteenth ACM Conference on Hypertext and Hypermedia*, HYPERTEXT '05, pages 235–244, New York, NY, USA, 2005. ACM. URL <http://doi.acm.org/10.1145/1083356.1083405>.

Olivier Aubert, Yannick Prié, and Camila Canellas. Leveraging video annotations in video-based e-learning. *arXiv*, 2014. URL <http://arxiv.org/abs/1404.4607>.

Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '02, pages 1–16, New York, NY, USA, 2002. ACM. URL <http://doi.acm.org/10.1145/543613.543615>.

Andrew Barry and Lucy Kimbell. Pindices. *ZKM*, 2005. URL <http://www.lucykimbell.com/LucyKimbell/Pindices.html>.

James R. Beniger and Dorothy L. Robyn. Quantitative graphics in statistics: A brief history. *The American Statistician*, 32(1):1–11, 1978. URL <http://www.jstor.org/stable/2683467>.

M.S. Bernstein, J. Brandt, R.C. Miller, and D.R. Karger. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 33–42. ACM, 2011. URL <http://hci.stanford.edu/publications/2011/realtime/adrenaline-uist2011.pdf>.

Jacques Bertin. *Semiologie graphique*. Mouton; Paris: Gauthier-Villars, 1973. URL <http://books.google.fr/books?id=F4weAAAAMAAJ>.

Jacques Bertin. *La graphique et le traitement graphique de l'information*. Flammarion, 1977. URL http://www.persee.fr/web/revues/home/prescript/article/colan_0336-1500_1977_num_36_1_1162.

Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samuel White, and Tom Yeh. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, UIST '10, pages 333–342, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0271-5. DOI: 10.1145/1866029.1866080. URL <http://doi.acm.org/10.1145/1866029.1866080>.

Jon Bird and Yvonne Rogers. The pulse of tidy street: Measuring and publicly displaying domestic electricity consumption. In *Workshop on Energy Awareness and Conservation through Pervasive Applications (Pervasive 2010)*, 2010. URL https://idc-online.com/technical_references/pdfs/electrical_engineering/The%20pulse%20of%20Tidy%20street.pdf.

Alan F. Blackwell. The reification of metaphor as a design tool. *ACM TOCHI*, 13(4):490–530, 2006. URL <http://portal.acm.org/citation.cfm?doid=1188816.1188820>.

M. Bostock, V. Ogievetsky, and J. Heer. D3 data-driven documents. *Visualization and Computer Graphics, IEEE TVCG*, 17(12):2301–2309, 2011. URL <http://dx.doi.org/10.1109/TVCG.2011.185>.

N Brosterman, K Togashi, and E Himmel. *Inventing kindergarten*. Harry N. Abrams, 1997. URL <http://www.getcited.org/pub/100128705>.

F. Bry, V. Gehlen-Baum, and A. Pohl. Promoting awareness and participation in large class lectures: The digital backchannel backstage. In *Proceedings of the IADIS International Conference on e-Society, Avila, Spain*, pages 27–34, 2011. URL <http://www.en.pms.ifi.lmu.de/publications/PMS-FB/PMS-FB-2011-2/PMS-FB-2011-2-paper.pdf>.

Scott Bultman. *The Froebel Gifts 2000*. Kindergarten Messenger, 2000. URL <http://www.amazon.fr/The-Froebel-Gifts-Scott-Bultman/dp/1930349009>.

B. Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design: Getting the Design Right and the Right Design*. Interactive Technologies. Elsevier Science, 2010. ISBN 9780080552903. URL <http://books.google.fr/books?id=2vfPxocmLh0C>.

Lee Byron and Martin Wattenberg. Stacked graphs & geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252, November 2008. ISSN 1077-2626. DOI: 10.1109/TVCG.2008.166. URL <http://dx.doi.org/10.1109/TVCG.2008.166>.

Nan Cao, Yu-Ru Lin, Xiaohua Sun, D. Lazer, Shixia Liu, and Huamin Qu. Whisper: Tracing the spatiotemporal process of information diffusion in real time. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2649–2658, dec. 2012. ISSN 1077-2626. DOI: 10.1109/TVCG.2012.291. URL <http://dx.doi.org/10.1109/TVCG.2012.291>.

S.K. Card and J. Mackinlay. The structure of the information visualization design space. In *Proc of. Information Visualization, IEEE*, pages 92–99, 1997. DOI: 10.1109/INFVIS.1997.636792. URL http://faculty.utpa.edu/fowler/vis-papers/Card-Mackinlay_1997_Structure-of-IV-Design-Space.pdf.

Stuart K Card, Jock D Mackinlay, and Ben Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999. URL <http://books.google.fr/books/about/?id=wdh2gqWfQmgC>.

Marianne Sheelagh Therese Carpendale. *A framework for elastic presentation space*. PhD thesis, Simon Fraser University, 1999. URL <http://pages.cpsc.ucalgary.ca/~sheelagh/wiki/pmwiki.php?n=Main.Thesis>.

J Carroll, W Kellogg, and R Mack. *Interface metaphors and user interface design*. Research reports // IBM, 1987. URL http://books.google.fr/books/about/?id=_xluYgECAAJ.

Jordi Cat. Otto Neurath, 2010. URL <http://plato.stanford.edu/entries/neurath/>.

Ximena Catepillán and Waclaw Szymanski. Counting and arithmetic of the inca. *Revista Latinoamericana de Etnomatemática*, 5(1):47–65, 2012. URL <http://www.redalyc.org/pdf/2740/274023595003.pdf>.

E. Catto. Box2d, 2010. URL <http://box2d-js.sourceforge.net/>.

William O Chao, Tamara Munzner, and Michiel van de Panne. Rapid pen-centric authoring of improvisational visualizations with napkinvis, 2010. URL <http://www.cs.ubc.ca/~wochao/napkinvis/SummaryNapkinVis.pdf>.

Michael Chapman. *Constructive evolution: Origins and development of Piaget's thought*. Cambridge University Press, 1988. URL <http://books.google.fr/books/about/?id=7WgCnXmdX1MC>.

Ed H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Proc. of InfoVis 2000. IEEE*, pages 69–75, 2000. DOI: 10.1109/INFVIS.2000.885092. URL <http://www-users.cs.umn.edu/~echi/papers/infvis00/Chi-TaxonomyVisualization.pdf>.

Ed H Chi and Stuart K Card. Sensemaking of evolving web sites using visualization spreadsheets. In *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*, pages 18–25. IEEE, 1999. URL <http://www2.parc.com/istl/groups/uir/publications/items/UIR-1999-09-Chi-InfoVis99-SensemakingSpreadsheets.pdf>.

WS Cleveland and R McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical*, 1984. URL <http://www.tandfonline.com/doi/full/10.1080/01621459.1984.10478080>.

CMU & Intel Research. Claytronics video, 2006. URL <http://www.tinyurl.com/claytronics>. [accessed 2013-03-30].

A. Cockburn and T. Dale. Ceva: a tool for collaborative video analysis. In *Proceedings of the international ACM SIGGROUP conference*

on Supporting group work: the integration challenge, pages 47–55. ACM, 1997. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.9008>.

S. Cogdill, T.L. Fanderclai, J. Kilborn, and M.G. Williams. Backchannel: whispering in digital conversation. In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, pages 8–pp. IEEE, 2001. URL http://www.hicss.hawaii.edu/HICSS_34/PDFs/DDPTC03.pdf.

Leda Cosmides and John Tooby. Are humans good intuitive statisticians after all? rethinking some conclusions from the literature on judgment under uncertainty. *cognition*, 58(1):1–73, 1996. URL <http://www.cogsci.ucsd.edu/~coulson/203/cosmides1.pdf>.

Joseph A. Cottam, Andrew Lumsdaine, and Chris Weaver. Watch this: A taxonomy for dynamic data visualization. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 193 –202, oct. 2012. DOI: 10.1109/VAST.2012.watchthis. URL <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6400552>.

Cowdflower. Senti crowdfower, 2012. URL <https://senti.crowdfower.com/senti>. accessed 6 Sep. 2012, <https://senti.crowdfower.com/senti>.

John W Creswell. *Qualitative inquiry and research design: Choosing among five approaches*. Sage, 2012. URL <http://books.google.fr/books/about/?id=0JYEbDtkxq8C>.

Anita Cross. The educational background to the bauhaus. *Design Studies*, 4(1):43 – 52, 1983. ISSN 0142-694X. DOI: [http://dx.doi.org/10.1016/0142-694X\(83\)90007-8](http://dx.doi.org/10.1016/0142-694X(83)90007-8). URL <http://www.sciencedirect.com/science/article/pii/0142694X83900078>.

Nigel Cross. Descriptive models of creative design: application to an example. *Design Studies*, 18(4):427–440, 1997. URL <http://oro.open.ac.uk/39443/8/Descriptive%20Models%20-%20DS.pdf>.

Thomas Crump. *The anthropology of numbers*. Number 70. Cambridge University Press, 1992. URL <http://books.google.fr/?id=uxMEi6kp0JKC>.

G. Daniel and M. Chen. *Video visualization*. IEEE, 2003. URL <http://dl.acm.org/citation.cfm?id=1081499>.

demonocracy.info. A visualization of United States debt., 2012. URL http://demonocracy.info/infographics/usa/us_deficit/us_deficit.html. Last access: March 2013.

N. Diakopoulos, S. Goldenberg, and I. Essa. Videolyzer: quality analysis of online informational video for bloggers and journalists. In *Proceedings of CHI*, pages 799–808, 2009. URL <http://www.nickdiakopoulos.com/projects/videolyzer-information-quality-analysis-for-videos/>.

N. Diakopoulos, M. Naaman, and F. Kivran-Swaine. Diamonds in the rough: Social media visual analytics for journalistic inquiry. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 115–122, oct. 2010. DOI: 10.1109/VAST.2010.5652922. URL <http://www.nickdiakopoulos.com/projects/social-media-visual-analytics/>.

Nicholas A. Diakopoulos and David A. Shamma. Characterizing debate performance via aggregated twitter sentiment. In *Proceedings of the 28th international conference on Human factors in computing systems, CHI '10*, pages 1195–1198, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. DOI: <http://doi.acm.org/10.1145/1753326.1753504>. URL <http://doi.acm.org/10.1145/1753326.1753504>.

Marian Dork, Daniel Gruen, Carey Williamson, and Sheelagh Carpendale. A visual backchannel for large-scale events. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1129–1138, November 2010. ISSN 1077-2626. DOI: 10.1109/TVCG.2010.129. URL <http://dx.doi.org/10.1109/TVCG.2010.129>.

Mark Doughty, Duncan Rowland, and Shaun Lawson. Co-viewing live tv with digital backchannel streams. In *Proceedings of the 9th international interactive conference on Interactive television*, pages 141–144. ACM, 2011. URL <http://dl.acm.org/citation.cfm?id=2000147>.

Sarah Douglas, Christopher Hundhausen, and Donna McKeown. Toward empirically-based software visualization languages. In *Visual Languages, Proceedings., 11th IEEE International*, pages 342–349. IEEE, 1995. URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=520828>.

A. Engström, M. Perry, and O. Juhlin. Amateur vision and recreational orientation:: creating live video together. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 651–660. ACM, 2012. URL <http://dl.acm.org/citation.cfm?id=2145304>.

J. Fekete. The infovis toolkit. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 167–174, 2004. DOI: 10.1109/INFOVIS.2004.64.

Harold N. Fisk. Geological Investigation of the Alluvial Valley of the Lower Mississippi River., 2005. URL <http://pruned.blogspot.fr/2005/06/geological-investigation-of-alluvial.html>. Last access: 10 March 2013.

Angus Graeme Forbes, Tobias Hollerer, and George Legrady. “behaviorism”: a framework for dynamic data visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6): 1164–1171, 2010. URL <http://dx.doi.org/10.1109/TVCG.2010.126>.

Phil Frei, Victor Su, Bakhtiar Mikhak, and Hiroshi Ishii. Curlybot: Designing a new class of computational toys. In *Proc. of CHI*, CHI '00, pages 129–136, NY, USA, 2000. ACM. ISBN 1-58113-216-6. DOI: 10.1145/332040.332416. URL <http://doi.acm.org/10.1145/332040.332416>.

Michael Friendly and J. Daniel Denis. Milestones in the history of thematic cartography, statistical graphics, and data visualization., 2001. URL <http://www.datavis.ca/milestones/>. Accessed: May 21, 2014.

F. W. Froebel. *The education of man*. New York: D. Appleton & Company (Original work published 1826), 1887. URL <http://books.google.com/books?isbn=0486145301>.

Benjamin Jotham Fry. Organic information Design. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2000. URL <http://benfry.com/organic/thesis-0522d.pdf>.

George W Furnas. Generalized fisheye views. In *Human Factors in Computing Systems CHI '86 Conference Proceedings*, volume 17. ACM, 1986. URL <http://furnas.people.si.umich.edu/Papers/FisheyeCHI86.pdf>.

Dorothy Gabel and Robert Sherwood. The effect of student manipulation of molecular models on chemistry achievement according to piagetian level. *Journal of Research in Science Teaching*, 17(1): 75–81, 1980. URL <http://onlinelibrary.wiley.com/doi/10.1002/tea.3660170112/abstract>.

Mohamed Medhat Gaber, Arkady Zaslavsky, and Shonali Krishnaswamy. Mining data streams: a review. *SIGMOD Rec.*, 34(2):18–26, June 2005. ISSN 0163-5808. DOI: 10.1145/1083784.1083789. URL <http://doi.acm.org/10.1145/1083784.1083789>.

Reebee Garfalo. Genealogy of Pop Rock Music., 1979. URL <http://www.historyshots.com/rockmusic/>. Last access: 18 Jan. 2014.

Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155 – 170, 1983. ISSN 0364-0213. DOI: 10.1016/S0364-0213(83)80009-3. URL <http://www.sciencedirect.com/science/article/pii/S0364021383800093>.

Knud Molenbach George Fitzmaurice. LegoWall Prototype Last access: 17 July 2013., 2009. URL <http://www.autodeskresearch.com/people/george/videos>.

Kyle Gilpin, Ara Knaian, and Daniela Rus. Robot pebbles: One centimeter modules for programmable matter through self-disassembly. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2485–2492. IEEE, 2010. URL <https://groups.csail.mit.edu/drl/wiki/images/3/3f/RobotPebbles-GilpinKnaianRus-ICRA10.pdf>.

Luc Girardin and Dominique Brodbeck. A visual approach for monitoring logs. In *LISA*, volume 98, pages 299–308, 1998. URL <http://dl.acm.org/citation.cfm?id=665071>.

Audrey Girouard, Erin Treacy Solovey, Leanne M. Hirshfield, Stacey Ecott, Orit Shaer, and Robert J. K. Jacob. Smart blocks: A tangible mathematical manipulative. In *Proc. of TEI*, pages 183–186, NY, USA, 2007. ACM. ISBN 978-1-59593-619-6. DOI: 10.1145/1226969.1227007. URL <http://doi.acm.org/10.1145/1226969.1227007>.

globalspec.com. Geotechnical Services Information., 2013. URL http://www.globalspec.com/learnmore/engineering_services/geotechnical_services. Last access: 10 March 2013.

Janice D Gobert and John J Clement. Effects of student-generated diagrams versus student-generated summaries on conceptual understanding of causal and dynamic knowledge in plate tectonics. *Journal of research in science teaching*, 36(1):39–53, 1999. URL [http://onlinelibrary.wiley.com/doi/10.1002/\(SICI\)1098-2736\(199901\)36:1%3C39::AID-TEA4%3E3.0.CO;2-I/abstract](http://onlinelibrary.wiley.com/doi/10.1002/(SICI)1098-2736(199901)36:1%3C39::AID-TEA4%3E3.0.CO;2-I/abstract).

Sarah Goodwin, Jason Dykes, Sara Jones, Iain Dillingham, Graham Dove, Alison Duffy, Alexander Kachkaev, Aidan Slingsby, and Jo Wood. Creative user-centered visualization design for energy analysts and modelers. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2516–2525, 2013. URL <http://openaccess.city.ac.uk/2618/>.

Google. Google Chart Editor., 2013. URL <https://support.google.com/drive/answer/63824>. Last access: 10 Dec. 2013.

Lars Grammel. *User interfaces supporting information visualization novices in visualization construction*. PhD thesis, University of Victoria, 2012. URL <http://dspace.library.uvic.ca:8080/handle/1828/4359>.

Lars Grammel, Melanie Tory, and M Storey. How information visualization novices construct visualizations. *IEEE TVCG*, 16(6):943–952, 2010. URL http://webhome.cs.uvic.ca/~mtory/publications/infvis10_grammel.pdf.

Lars Grammel, Chris Bennett, Melanie Tory, and Margaret-anne Storey. A Survey of Visualization Construction User Interfaces. In *EuroVis-Short Papers*, pages 19–23. The Eurographics Association, 2013. URL <http://webhome.cs.uvic.ca/~mtory/publications/VisToolSurvey-eurovis13-short.pdf>.

Saul Greenberg, Sheelagh Carpendale, Nicolai Marquardt, and Bill Buxton. *Sketching user experiences: The workbook*. Access Online via Elsevier, 2011. URL <http://sketchbook.cpsc.ucalgary.ca/>.

J. Hannon, K. McCarthy, J. Lynch, and B. Smyth. Personalized and automatic social summarization of events in video. In *Proceedings*

of the 16th international conference on Intelligent user interfaces, pages 335–338. ACM, 2011. URL <http://irserver.ucd.ie/bitstream/handle/10197/2952/p335-hannon.pdf?sequence=1>.

Larry Hardesty. Making crowdsourcing easier, 2012. URL <http://web.mit.edu/newsoffice/2012/making-crowdsourcing-easier.html>. accessed on 7. Sep. 2012, <http://web.mit.edu/newsoffice/2012/making-crowdsourcing-easier.html>.

Johnatan Harris. We feel fine., 2012. URL <http://www.wefeelfine.org/>. Last access: March 2012.

D. Harry, J. Green, and J. Donath. Backchan. nl: integrating backchannels in physical space. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 1361–1370. ACM, 2009. URL http://web.media.mit.edu/~dharry/backchannl_names.pdf.

Alexander G. Hauptmann. Lessons for the future from a decade of informedia video analysis research. pages 1–10, 2005. URL http://dx.doi.org/10.1007/11526346_1.

Christopher G Healey. Perception in visualization. *IEEE Transactions on Visualization and Computer Graphics*, 10:2008, 2007. URL <http://www.csc.ncsu.edu/faculty/healey/PP/>.

Jeffrey Heer and Ben Shneiderman. Interactive dynamics for visual analysis. *Queue*, 10(2):30:30–30:55, 2012. ISSN 1542-7730. DOI: 10.1145/2133416.2146416. URL <http://doi.acm.org/10.1145/2133416.2146416>.

Jeffrey Heer, Stuart K. Card, and James A. Landay. Prefuse: A toolkit for interactive information visualization. In *Proc. of the SIGCHI, CHI '05*, pages 421–430, NY, USA, 2005. ACM. ISBN 1-58113-998-5. DOI: 10.1145/1054972.1055031. URL <http://doi.acm.org/10.1145/1054972.1055031>.

Mary Hegarty. The cognitive science of visual-spatial displays: Implications for design. *Topics in Cognitive Science*, 3(3):446–474, 2011. ISSN 1756-8765. DOI: 10.1111/j.1756-8765.2011.01150.x. URL <http://dx.doi.org/10.1111/j.1756-8765.2011.01150.x>.

Scarlett R Herring, Chia-Chen Chang, Jesse Krantzler, and Brian P Bailey. Getting inspired!: understanding how and why examples are used in creative design practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 87–96. ACM, 2009. URL <http://dl.acm.org/citation.cfm?id=1518717>.

Elizabeth G Hetzler, Vernon L Crow, Deborah A Payne, and Alan E Turner. Turning the bucket of text into a pipe. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 89–94. IEEE, 2005. URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1532133>.

Thomas J Howard, Stephen J Culley, and Elies Dekoninck. Describing the creative design process by the integration of engineering design and cognitive psychology literature. *Design studies*, 29(2):160–180, 2008. URL <http://www.sciencedirect.com/science/article/pii/S0142694X08000173>.

Hui-Kuang Hsieh. Chinese tally mark. *The American Statistician*, 35(3):174–174, 1981. URL <http://www.tandfonline.com/doi/abs/10.1080/00031305.1981.10479346>.

Wynne Hsu and Irene MY Woon. Current research in the conceptual design of mechanical products. *Computer-Aided Design*, 30(5):377–389, 1998. URL <http://www.sciencedirect.com/science/article/pii/S0010448597001012>.

Jon Huang and Aron Pilhofer. The Death of a Terrorist: A Turning Point?, 2012. URL <http://www.nytimes.com/interactive/2011/05/03/us/20110503-osama-response.html>. Last access: March 2013.

J. Hullman and N. Diakopoulos. Visualization rhetoric: Framing effects in narrative visualization. *IEEE TVCG*, 17(12):2231–2240, 2011. ISSN 1077-2626. DOI: 10.1109/TVCG.2011.255.

Michael Hunger. On LEGO Powered Time-Tracking; My Daily Column | Better Software Development, 2008. URL <http://jexp.de/blog/2008/08/on-lego-powered-time-tracking-my-daily-column/>.

Samuel Huron and Jean-Daniel Fekete. User-Supplied Sentiments in Tweets. In *IEEE Visweek : 2nd Workshop on Interactive Visual Text Analysis*, Seattle, Washington, USA, 2012. IEEE. URL <http://hal.inria.fr/hal-00734407>.

Samuel Huron, Romain Vuillemot, Raphael Velt, and Yves-Marie Haussonne. Bubble T : Et si twitter votait ?, 2012. URL <http://dev.fabelier.org/bubble-t/>. March 2012.

Samuel Huron, Petra Isenberg, and Jean-Daniel Fekete. PolemicTweet: Video Annotation and Analysis through Tagged Tweets. In *Proceedings of the IFIP TC13 Conference on Human-Computer Interaction (INTERACT)*, volume 8118, pages 135–152, Heidelberg, Allemagne, September 2013a. IFIP, Springer. DOI: 10.1007/978-3-642-40480-1_9. URL <http://hal.inria.fr/hal-00817591>.

Samuel Huron, Romain Vuillemot, and Jean Daniel Fekete. Bubble-TV: Live Visual Feedback for Social TV Broadcast. In *CHI 2013 Workshop : Exploring and enhancing the user experience for television*, Paris, France, 2013b. URL <http://hal.inria.fr/hal-00796354>.

Samuel Huron, Romain Vuillemot, and Jean-Daniel Fekete. Visual Sedimentation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2446–2455, December 2013c. DOI: 10.1109/TVCG.2013.227. URL <http://hal.inria.fr/hal-00846260>. CIFRE ANRT N 2010/1534, CINEGIFT DGCIS project.

Samuel Huron, Sheelagh Carpendale, Alice Thudt, Anthony Tang, and Michael Mauerer. Constructive Visualization. In ACM, editor, *ACM conference on Designing Interactive Systems in 2014*, Vancouver, Canada, June 2014a. ACM, ACM. URL <http://hal.inria.fr/hal-00978437>.

Samuel Huron, Yvonne Jansen, and Sheelagh Carpendale. Constructing Visual Representations: Investigating the Use of Tangible Tokens. *IEEE Transactions on Visualization and Computer Graphics*, 2014b. URL <http://hal.inria.fr/hal-01024053>. To appear.

Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. Direct manipulation interfaces. *Human Computer Interaction*, 1(4):311–338, 1985. URL <http://cleo.ics.uci.edu/teaching/Winter10/231/readings/1-HutchinsHollanNorman-DirectManipulation-HCI.pdf>.

James Hutton. Theory of the earth. *Transactions of the Royal Society of Edinburgh*, I(II):209–304, 1788. URL <http://sacred-texts.com/earth/toe/index.htm>.

Georges Ifrah. *The Universal History of Computing: From the Abacus to Quantum Computing*. John Wiley & Sons, Inc., New York, NY, USA, 2001. URL <http://books.google.fr/books/?id=tnhQAAAAAAAJ>.

P.G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM, 2010. URL <https://www.misrc.umn.edu/workshops/2012/fall/Ipeirotis.pdf>.

Hiroshi Ishii and Brygg Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '97*, pages 234–241, New York, NY, USA, 1997. ACM. ISBN 0-89791-802-9. URL <http://doi.acm.org/10.1145/258549.258715>.

Yvonne Jansen and Pierre Dragicevic. An Interaction Model for Visualizations Beyond The Desktop. *IEEE TVCG*, 19(12):2396 – 2405, 2013. DOI: 10.1109/TVCG.2013.134. URL <http://hal.inria.fr/hal-00847218>.

Yvonne Jansen, Pierre Dragicevic, and Jean-Daniel Fekete. Evaluating the efficiency of physical visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2593–2602, Paris, France, 2013. ACM. URL <http://hal.inria.fr/hal-00781831>.

Hans-Christian Jetter, Jens Gerken, Michael Zöllner, Harald Reiterer, and Natasa Milic-Frayling. Materializing the query with facet-streams: a hybrid surface for collaborative search on tabletops. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3013–3022. ACM, 2011. URL http://kops.ub.uni-konstanz.de/bitstream/handle/urn:nbn:de:bsz:352-186490/Jetter_Materializing.pdf?sequence=3.

L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 151–160, 2011. URL <http://www.aclweb.org/anthology/P11-1016.pdf>.

Chris Johnson, Robert Moorhead, Tamara Munzner, Hanspeter Pfister, Penny Rheingans, and Terry S. Yoo. Nih/nsf visualization research challenges report. *LOS ALAMITOS, CA: IEEE COMPUTING SOCIETY*, 2006. URL <http://www.ncbi.nlm.nih.gov/pubmed/16548457>.

Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 139–146. ACM, 2007. URL <http://dl.acm.org/citation.cfm?id=1226998>.

Chris Jordan. Running the numbers: An american self-portrait, 2006. URL <http://www.chrisjordan.com/gallery/rtn/#plastic-cups>. Last access: 12 July 2013.

Chris Jordan. Turning powerful stats into art., 2008. URL <http://www.youtube.com/watch?v=f09lQ8Q1iKE>.

George Gheverghese Joseph. *The crest of the peacock: Non-European roots of mathematics*. Princeton University Press, 2011. URL <http://press.princeton.edu/titles/9308.html>.

Alan Kay. User interface: A personal view. *The art of human-computer interface design*, pages 191–207, 1989. URL http://ui.korea.ac.kr/Board/Upload/a%20personal%20view_n.pdf.

Daniel Keim, Tobias Schreck, Milos Krstajić, and Christian Rohrdantz. Real-time visual analytics of text data streams. *Computer*, 99(1):1, 2013. URL <http://dx.doi.org/10.1109/MC.2013.152>.

David Kirsh. Complementary strategies: Why we use our hands when we think. In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*, pages 212–217. Lawrence Erlbaum Associates Mahwah, NJ, 1995. URL <http://adrenaline.ucsd.edu/kirsh/Articles/Cogsci95/cogsci95.html>.

R. Kosara. Visualization criticism - the missing link between information visualization and art. In *Information Visualization, 2007*, pages 631–636, 2007. DOI: 10.1109/IV.2007.130.

Wally J. Kowalski. Roman board games, 1998. URL <http://ablemedia.com/ctcweb/showcase/boardgames5.html>.

Mrs Maria Kraus-Boelte and John Kraus. *The kindergarten guide: An illustrated hand-book, designed for the self-instruction of kindergartners, mothers, and nurses*, volume 1. E. Steiger & Co., 1892. URL <https://archive.org/details/kindergartengui02kraugoog>.

Milos Krstajić, Enrico Bertini, and Daniel Keim. Cloud-lines: Compact display of event episodes in multiple time-series. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2432–2439, December 2011. URL <https://kops.ub.uni-konstanz.de/xmlui/bitstream/handle/urn:nbn:de:bsz:352-174833/Keim.pdf?sequence=3>.

Miloš Krstajić, Enrico Bertini, Florian Mansmann, and Daniel A. Keim. Visual analysis of news streams with article threads. In *Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques, StreamKDD '10*, pages 39–46, New York, NY, USA, 2010. ACM. URL https://kops.ub.uni-konstanz.de/xmlui/bitstream/handle/urn:nbn:de:bsz:352-127061/krstajic_visual.pdf?sequence=2.

George Lakoff and Mark Johnson. *Metaphors we live by*. University of Chicago press, 2008. URL <http://press.uchicago.edu/ucp/books/book/chicago/M/bo3637992.html>.

Paul Laseau. *Graphic thinking for architects & designers*. John Wiley & Sons, 2001. URL <http://books.google.fr/books/?id=KRCyqDjYSLAC>.

Andrea Lau and Andrew Vande Moere. Towards a model of information aesthetics in information visualization. In *Information Visualization, 2007. IV'07. 11th International Conference*, pages 87–92. IEEE, 2007. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4271966.

Eric le Boulout. La politique sur Twitter en temps reel avec Bubble-T, 2012. URL <http://www.01net.com/editorial/561450/la-politique-sur-twitter-en-temps-reel-avec-bubble-t/>.

LeMonde. DATA - De quel candidat a la présidentielle parle Twitter maintenant, tout de suite ?, 2012. URL <http://bigbrowser.blog.lemonde.fr/2012/03/14/data-de-quel-candidat-a-la-presidentielle-parle-twitter-maintenant-tout-de-suite/>.

Golan Levin. The dumpster., 2006. URL <http://www.flong.com/projects/dumpster/>. Last access: March 2013.

RC MacCormac. Froebel's kindergarten gifts and the early work of Frank Lloyd Wright. *Environment and Planning B*, 1(1):29–50, 1974. URL <http://www.envplan.com/abstract.cgi?id=b010029>.

Wendy E. Mackay and Michel Beaudouin-Lafon. Diva: exploratory data analysis with multimedia streams. In *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '98*, pages 416–423, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co. URL <https://www.lri.fr/~mbl/DIVA/chi98-paper/introduction.html>.

Wendy E Mackay and Anne-Laure Fayard. Hci, natural science and design: a framework for triangulation across disciplines. In *Proceedings of the 2nd conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 223–234. ACM, 1997. URL <https://www.lri.fr/~mackay/pdf/FILES/DIS97.Triangulate.pdf>.

John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond. The scratch programming language and environment. *ACM TOCE*, 10(4):16, 2010. URL <http://web.media.mit.edu/~jmaloney/papers/ScratchLangAndEnvironment.pdf>.

Richard Mander, Gitta Salomon, and Yin Yin Wong. A “pile” metaphor for supporting casual organization of information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, pages 627–634, New York, NY, USA, 1992. ACM. URL <http://monet.cs.columbia.edu/courses/csw4170/resources/p627-mander.pdf>.

JP Manning. Rediscovering Froebel: A call to re-examine his life & gifts. *Early Childhood Education Journal*, 2005. URL <http://link.springer.com/article/10.1007/s10643-005-0004-8>.

Adam Marcus, Michael S. Bernstein, Osama Badar, David R. Karger, Samuel Madden, and Robert C. Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 227–236, New York, NY, USA, 2011. ACM. URL <http://marcua.net/papers/twitinfo-chi2011.pdf>.

David Marr. *Vision: A computational approach*. MIT Press, 1982. URL <http://mitpress.mit.edu/books/vision-0>.

Alexander Marshack. *The roots of civilization: The cognitive beginnings of man's first art, symbol and notation*. Weidenfeld and Nicolson London, UK, 1972. URL <http://books.google.fr/?id=TbU9AAAAIAAJ>.

Jean-Christophe Martin. Les muscles numériques de la web-campagne, 2012. URL <http://www.franceinfo.fr/emission/hyper-revue-de-presse/2011-2012/les-muscles-numeriques-de-la-web-campagne-03-15-2012-11-15>.

J.F. McCarthy et al. Digital backchannels in shared physical spaces: experiences at an academic conference. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 1641–1644. ACM, 2005. URL <http://www.danah.org/papers/CHI2005Backchannels.pdf>.

JP McCormack, J Cagan, and CM Vogel. Speaking the Buick language: capturing, understanding, and exploring brand identity with shape grammars. *Design studies*, 2004. URL <http://www.sciencedirect.com/science/article/pii/S0142694X03000231>.

E Mcgrath. Methodology matters: Doing research in the behavioral and social sciences. In *Readings in Human-Computer Interaction: Toward the Year 2000 (2nd ed.* Citeseer, 1995. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.361.1234>.

Robert H McKim. Experiences in visual thinking. 1972. URL <http://eric.ed.gov/?id=ED073690>.

Brian McNely. Backchannel persistence and collaborative meaning-making. In *Proceedings of the 27th ACM international conference on Design of communication*, SIGDOC '09, pages 297–304, New York, NY, USA, 2009. ACM. URL dl.acm.org/ft_gateway.cfm?id=1622053.

Mediametrie. La fréquentation des sites Internet français Février 2012 Les vacances profitent aux sites des chaînes de télévision, 2012. URL <http://www.mediametrie-estat.com/la-frequentation-des-sites-internet-francais-en-fevrier-2012/>.

Luana Micallef, Pierre Dragicevic, and Jean-Daniel Fekete. Assessing the Effect of Visualizations on Bayesian Reasoning Through Crowdsourcing. *IEEE TVCG*, 18(12):2536 – 2545, 2012. DOI: 10.1109/TVCG.2012.199. URL <http://hal.inria.fr/hal-00717503>. Best Paper Honorable Mention.

Charles Minard. La carte figurative des pertes successives en hommes de l’armée française dans la campagne de Russie 1812-1813. , 1869. URL http://fr.wikipedia.org/wiki/Carte_figurative. Last access: 18 Jan. 2014.

Andrés Monroy-Hernández. *Designing for remixing: Supporting an online community of amateur creators*. PhD thesis, Massachusetts Institute of Technology, 2012. URL <http://llk.media.mit.edu/papers/andres-dissertation.pdf>.

Robert Moorhead, Chris Johnson, Tamara Munzner, Hanspeter Pfister, Penny Rheingans, and Terry S. Yoo. Visualization research challenges: A report summary. *Computing in Science and Engineering*, 8(4):66–73, 2006. URL <http://www.csee.umbc.edu/~rheingan/pubs/cga06.pdf>.

Pascal Müller, Peter Wonka, Simon Haegler, Andreas Ulmer, and Luc Van Gool. *Procedural modeling of buildings*, volume 25. ACM, 2006. URL <http://dl.acm.org/citation.cfm?id=1141931>.

Tamara Munzner. A nested model for visualization design and validation. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):921–928, 2009. URL <https://www.cs.ubc.ca/labs/imager/tr/2009/NestedModel/NestedModel.pdf>.

BA Myers, J Goldstein, and MA Goldberg. Creating charts by demonstration. *Proc. of the SIGCHI*, 1994. URL <http://dl.acm.org/citation.cfm?id=191715>.

Brad Myers, Scott E. Hudson, and Randy Pausch. Past, present, and future of user interface software tools. *ACM Trans. Comput.-Hum. Interact.*, 7(1):3–28, March 2000. ISSN 1073-0516. DOI: 10.1145/344949.344959. URL <http://doi.acm.org/10.1145/344949.344959>.

Brad A. Myers. A brief history of human-computer interaction technology. *interactions*, 5(2):44–54, March 1998. ISSN 1072-5520. DOI: 10.1145/274430.274436. URL <http://doi.acm.org/10.1145/274430.274436>.

Marie Neurath. *The Transformer: Principles of Making Isotype Charts*. Princeton Architectural Press, 2009. URL <http://books.google.com/books?id=k6lmNAAACAAJ&pgis=1>.

O. Neurath. *Modern Man in the Making*. Knopf, 1939. URL <http://books.google.ca/books?id=PxZmAAAAMAAJ>.

Andrew A. Norton, Matthew A. Rubin, and Leland Wilkinson. Streaming graphics. *Statistical Computing and Graphics Newsletter*, 12(1):11–14, 2001. URL <http://www.cs.uic.edu/~wilkinson/Publications/dancer.pdf>.

J J O'Connor and E F Robertson. Inca mathematics, 2001. URL http://www-history.mcs.st-and.ac.uk/HistTopics/Inca_mathematics.html.

The Editors of Encyclopædia Britannica. Sedimentation (Geology). Britannica., 2013. URL <http://global.britannica.com/EBchecked/topic/532291/sedimentation>. Last access: March 2013.

Alex F. Osborn. *Applied Imagination, Principles and Procedures of Creative Thinking*. Charles Scribner's Sons, 1953. URL http://www.amazon.com/Imagination-Principles-Procedures-Creative-Thinking/dp/B000H5J5WW/ref=pd_bbs_sr_1/102-8461719-9389763?ie=UTF8&s=books&qid=1181614703&sr=1-1.

B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008. URL <http://books.google.ca/?id=XQsqsqLLKrEC>.

Seymour Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, Inc., NY, USA, 1980. ISBN 0-465-04627-4. URL <http://books.google.fr/books/?id=HhIEAgUfGHwC>.

Seymour Papert. What is logo? who need it? *Logo philosophy and implementation*, 1999. URL <http://www.microworlds.com/company/philosophy.pdf>.

Seymour Papert and Idit Harel. Situating constructionism. *Constructionism*, pages 1–11, 1991. URL http://web.media.mit.edu/~calla/web_comunidad/Reading-En/situating_constructionism.pdf.

Amanda Parkes, Vincent LeClerc, and Hiroshi Ishii. Glume: Exploring materiality in a soft augmented modular modeling system. In *Proc. of CHI*, pages 1211–1216, NY, USA, 2006. ACM. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.230.6069>.

M.L. Parry, P.A. Legg, D.H.S. Chung, I.W. Griffiths, and M. Chen. Hierarchical event selection for video storyboards with a case study on snooker video visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):1747–1756, 2011. URL <http://www.ncbi.nlm.nih.gov/pubmed/22034291>.

Charles Perin, Pierre Dragicevic, and J Fekete. Revisiting bertin matrices: New interactions for crafting tabular visualizations. 2014.

Piaget. Piaget on Piaget, Part 3., 1977. URL http://www.youtube.com/watch?v=x9nSC_Xgabc. Last access: 17 July 2013.

Jean Piaget. *Six études de psychologie*. Ed. Denoël, 1989. URL <http://www.fondationjeanpiaget.ch/fjp/site/presentation/index.php?PRESMODE=1&DOCID=1195>.

Jean Piaget and Delachaux et Niestlé. *La naissance de l'intelligence chez l'enfant*. 1948. URL <http://books.google.fr/books?id=MOu5nQEACAAJ>.

Jean Piaget and Bärbel Inhelder. *Le développement des quantités physiques chez l'enfant: conservation et atomisme*. Delachaux et Niestlé, 1962. URL <http://books.google.fr/?id=NEN-JIhQnv8C>.

Studio Pinata. LEGO CALENDAR REVISITED Last access: 17 July 2013., 2009. URL <http://www.pinata.fi/blog/lego-calendar-revisited/>.

Z. Pousman, J.T. Stasko, and M. Mateas. Casual information visualization: Depictions of data in everyday life. *IEEE TVCG*, 13(6): 1145–1152, 2007. ISSN 1077-2626. DOI: 10.1109/TVCG.2007.70541.

Stuart Pugh. *Total design: integrated methods for successful product engineering*. Addison-Wesley Wokingham, 1991. URL <http://books.google.fr/books/?id=RKIQAQAAMAAJ>.

Vincent Puig, Johann Holland, Thibaut Cavalié, Cora Benjamin, Johan Mathé, Yves-Marie Haussonne, and Sébastien Liévin. *Atelier ihm 2007 lignes de temps, une plateforme collaborative pour l'annotation de films et d'objets temporels.*, 2007. URL <http://www.iri.centrepompidou.fr/res/media/IHM07Workshop-IRI.pdf>.

quantifiedself.com. Quantified Self data visualization archives., 2013. URL <http://quantifiedself.com/data-visualization/>. Last access: 09 Dec. 2013.

Alexander J. Quinn and Benjamin B. Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11*, pages 1403–1412, New York, NY, USA, 2011. ACM. URL http://www.cs.umd.edu/~bederson/images/pubs_pdfs/p1403-quinn.pdf.

Hayes Solos Raffle, Amanda J. Parkes, and Hiroshi Ishii. Topobo: A constructive assembly system with kinetic memory. In *Proc. of*

CHI, CHI '04, pages 647–654, NY, USA, 2004. ACM. URL <http://tangible.media.mit.edu/project/topobo/>.

Casey Reas and Ben Fry. *Processing: a programming handbook for visual designers and artists*, volume 6812. Mit Press, 2007. URL <http://mitpress.mit.edu/books/processing>.

Hannes Reijner and Panopticon Software. The development of the horizon graph, 2008. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.363.5396>.

Jun Rekimoto, Yuji Ayatsuka, Hitoraka Uoi, and Toshifumi Arai. Adding another communication channel to reality: an experience with a chat-augmented conference. In *CHI 98 conference summary on Human factors in computing systems*, CHI '98, pages 271–272, New York, NY, USA, 1998. ACM. URL <http://www.sonycs1.co.jp/person/rekimoto/papers/chi98chat.PDF>.

Gábor Renner and Anikó Ekárt. Genetic algorithms in computer aided design. *Computer-Aided Design*, 35(8):709–726, 2003. URL <http://www.sciencedirect.com/science/article/pii/S0010448503000034>.

M. Resnick, F. Martin, R. Sargent, and B. Silverman. Programmable bricks: Toys to think with. *IBM Systems Journal*, 35(3.4):443–452, 1996. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5387219.

Mitchel Resnick. All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. In *Proc. of ACM SIGCHI C&C*, pages 1–6, USA, 2007. ACM Press. URL <http://web.media.mit.edu/~mres/papers/kindergarten-learning-approach.pdf>.

Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. Scratch: Programming for all. *Commun. ACM*, 52(11):60–67, 2009. URL <http://doi.acm.org/10.1145/1592761.1592779>.

Rafael Roberto, Daniel Freitas, João Paulo Lima, Veronica Teichrieb, and Judith Kelner. ARBlocks: A Concept for a Dynamic Blocks Platform for Educational Activities. *2011 XIII Symposium on Virtual Reality*, pages 28–37, 2011. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5951832>.

George Robertson, Mary Czerwinski, Kevin Larson, Daniel C Robins, David Thiel, and Maarten Van Dantzich. Data mountain: using spatial memory for document management. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 153–162. ACM, 1998. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.283>.

Eleanor Robson. Tables and tabular formatting in sumer, babylonia and assyria, 2500-50 bce. pages 108–144, 2003. URL <http://www.hps.cam.ac.uk/people/robson/tables.pdf>.

J. Roschelle and S. Goldman. Videonoter: A productivity tool for video data analysis. *Behavior Research Methods*, 23(2):219–224, 1991. URL <http://download.springer.com/static/pdf/508/art%253A10.3758%252FBF03203368.pdf>.

Philip J Runkel and Joseph Edward MacGrath. *Research on human behavior*. Holt, Rinehart & Winston New York, 1972. URL <http://books.google.fr/books?id=wAZ9AAAAMAAJ>.

T. Ruppert and J. Kohlhammer. A radial visualization tool for depicting hierarchically structured video content. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 251–252. IEEE, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5650177&tag=1.

Jeffrey M. Rzeszotarski and Aniket Kittur. Kinetica: Naturalistic multi-touch data visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '14*, pages 897–906, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. URL <http://doi.acm.org/10.1145/2556288.2557231>.

P. Saraiya, C. North, and K. Duca. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE TVCG*, 11(4):443–456, 2005. ISSN 1077-2626. DOI: 10.1109/TVCG.2005.53.

Kai Schäfer, Volker Brauer, and Willi Bruns. A new approach to human-computer interaction-synchronous modelling in real and virtual spaces. In *Proceedings of the 2nd conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 335–344. ACM, 1997. URL <http://dl.acm.org/citation.cfm?id=263638>.

Denise Schmandt-Besserat. *How writing came about*. University of Texas Press Austin, 1996. URL <http://utpress.utexas.edu/index.php/books/schhop>.

Konrad J Schönborn and Trevor R Anderson. The importance of visual literacy in the education of biochemists. *Biochemistry and Molecular Biology Education*, 34(2):94–102, 2006. URL <http://www.ncbi.nlm.nih.gov/pubmed/21638645>.

R. Schroeter, J. Hunter, and D. Kosovic. Vannotea: A collaborative video indexing, annotation and discussion system for broadband networks. In *Knowledge capture*, pages 1–8. ACM Press (Association for Computing Machinery), 2011. URL <http://www.citeulike.org/user/casam/article/3043271>.

Eric Schweikardt and Mark D. Gross. roblocks: A robotic construction kit for mathematics and science education. In *Proc. of ICMI*, pages 72–75, NY, USA, 2006. ACM. URL <http://dl.acm.org/citation.cfm?id=1181010>.

Andrew Sears and Julie A Jacko. *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*. CRC press, 2007. URL <http://books.google.fr/books?id=b3rg6wmjc5QC>.

Michael Sedlmair, Miriah Meyer, and Tamara Munzner. Design study methodology: Reflections from the trenches and the stacks. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2431–2440, 2012. URL <http://www.cs.ubc.ca/labs/imager/tr/2012/dsm/>.

E. Segel and J. Heer. Narrative visualization: Telling stories with data. *IEEE TVCG*, 16(6):1139–1148, 2010. ISSN 1077-2626. DOI: 10.1109/TVCG.2010.179. URL <http://vis.stanford.edu/files/2010-Narrative-InfoVis.pdf>.

B. Shneiderman and C. Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, 2010. ISBN 9780321537355. URL <http://books.google.fr/books?id=2CfR0gAACAAJ>.

Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=545307.

Herbert Alexander Simon. *The sciences of the artificial*, volume 136. MIT press, 1969. URL <http://books.google.fr/?id=k5Sr0nFw7psC>.

David Canfield Smith. Pygmalion: a creative programming environment. Technical report, DTIC Document, 1975. URL <http://acypher.com/wwid/Chapters/01Pygmalion.html>. Accessed: May 21, 2014.

David Canfield Smith, Charles Irby, Ralph Kimball, WL Verplank, and Erik Harslem. Designing the star user interface. In *Human-computer interaction*, pages 653–661. Morgan Kaufmann Publishers Inc., 1987. URL <http://dl.acm.org/citation.cfm?id=98408>.

R. Snow, B. O’Connor, D. Jurafsky, and A.Y. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics, 2008. URL <http://web.stanford.edu/~jurafsky/amt.pdf>.

Rey P. Butler B. Shneiderman B. Sopan, A. Monitoring academic conferences: Real-time visualization and retrospective analysis of backchannel conversations, 2012. URL <http://www.w3.org/TR/media-frag/>.

Robert Spence and A Press. *Information visualization*. {Addison Wesley}, 2000. URL <http://books.google.fr/books/about/?id=gYQoAQAAAJ>.

- M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*, pages 53–63. Association for Computational Linguistics, 2011. URL <http://www.aclweb.org/anthology/W/W11/W11-2207.pdf>.
- SpotFire. SpotFire., 1990. URL <http://www.spotfire.com>. Last access: 10 Dec. 2013.
- Stackoverflow. Stackoverflow Tags., 2013. URL <http://stackoverflow.com/tags>. Last access: 10 March 2013.
- M. Stefik, G. Foster, D.G. Bobrow, K. Kahn, S. Lanning, and L. Suchman. Beyond the chalkboard: computer support for collaboration and problem solving in meetings. *Communications of the ACM*, 30(1): 32–47, 1987. URL <http://dl.acm.org/citation.cfm?id=7887>.
- Stephenson Stephen K. Ancient scientific calculators, 2005. URL <http://sks23cu.net/TSA/CB10/sksAncientSciCalc.pdf>. Web document, Accessed: May 21, 2014.
- Danyel Fisher Irina Spiridonova Steven Drucker, Roland Fernandez. Sandance webpage, 2013. URL <http://research.microsoft.com/en-us/projects/sanddance/>. Accessed 2013-07-21.
- George Stiny. Kindergarten grammars: designing with froebel's building gifts. *Environment and Planning B*, 7(4):409–462, 1980. URL http://home.fa.utl.pt/~lromao/2008_09/sg/aula_2/kinder_2.pdf.
- Ivan E Sutherland. *Sketch pad a man-machine graphical communication system*. PhD thesis, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 1963. URL <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-574.pdf>.
- Tableau. Tableau., 2014a. URL <http://www.tableausoftware.com>. Last access: 12 July 2013.
- Tableau. Tableau Public. , 2014b. URL <http://www.tableausoftware.com/public>. Last access: 23 March 2014.
- Patricia Tarr. Pestalozzian and Froebelian Influences on Contemporary Elementary School Art. *Studies in Art Education*, 30(2):115–121, 2013. URL <http://www.jstor.org/discover/10.2307/1320778>.
- TED. TED: Ideas worth spreading, 2013. URL <http://www.ted.com/>. Last access : 19/01/2013.
- William Irwin Thompson. *The time falling bodies take to light: mythology, sexuality and the origins of culture*. Palgrave Macmillan, 1996. URL <http://books.google.fr/books/about/?id=u3nB0NpAX60C>.

Matthew Tobiasz, Petra Isenberg, and Sheelagh Carpendale. Lark: Coordinating co-located collaboration with information visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1065–1072, 2009. URL <http://innovis.cpsc.ucalgary.ca/Research/Lark>.

Y. Tonomura, A. Akutsu, K. Otsuji, and T. Sadakata. Videomap and videospaceicon: tools for anatomizing video content. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 131–136. ACM, 1993. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.133.8108>.

Anne Treisman. Preattentive processing in vision. *Computer vision, graphics, and image processing*, 31(2):156–177, 1985. URL <http://dl.acm.org/citation.cfm?id=5091>.

Edward R Tufte and PR Graves-Morris. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 1983. URL http://www.edwardtufte.com/tufte/books_vdqi.

Brygg Ullmer, Hiroshi Ishii, and Robert J. K. Jacob. Token+constraint systems for tangible interaction with digital information. *ACM Trans. Comput.-Hum. Interact.*, 12(1):81–118, March 2005. URL <http://doi.acm.org/10.1145/1057237.1057242>.

Andrew Vande Moere. Infoticles: Information modeling in immersive environments. pages 457–461. IEEE, 2002. URL <http://infoscape.org/publications/spie04.pdf>.

Andrew Vande Moere. Time-varying data visualization using information flocking boids. In *IEEE Symposium on Information Visualization (InfoVis 2004)*, pages 97–104, 2004. DOI: 10.1109/INFVIS.2004.65. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1382896&tag=1.

Andrew Vande Moere. Beyond the tyranny of the pixel: Exploring the physicality of information visualization. In *Information Visualization, 2008. IV'08. 12th International Conference*, pages 469–474. IEEE, 2008. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.208.6726>.

Andrew Vande Moere and Helen Purchase. On the role of design in information visualization. *Information Visualization*, 10(4):356–371, 2011. URL <http://infoscape.org/publications/ivs12.pdf>.

B. Victor. Drawing Dynamic Visualizations Talk., 2013. URL <http://worrydream.com/#!/DrawingDynamicVisualizationsTalk>. Last access: 12 July 2013.

F.B. Viégas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. Manyeyes: a site for visualization at internet scale. *IEEE TVCG*, 13(6): 1121–1128, 2007. URL <http://www.bewitched.com/manyeyes.html>.

F.B. Viégas, M. Wattenberg, M. McKeon, F. van Ham, and J. Kriss. Harry potter and the meat-filled freezer: A case study of spontaneous usage of visualization tools. In *Proc. of HICSS*, pages 159–159, 2008. URL <http://dl.acm.org/citation.cfm?id=1334515.1334980>.

Fernanda Viégas and Martin Wattenberg. Artistic data visualization: Beyond visual analytics. In Douglas Schuler, editor, *Online Communities and Social Computing*, volume 4564 of *Lecture Notes in Computer Science*, pages 182–191. Springer Berlin Heidelberg, 2007. URL http://dx.doi.org/10.1007/978-3-540-73257-0_21.

Fernanda Viégas, Ethan Perry, Judith Donath, and Ethan Howe. Artifacts of the presence era: visualizing presence for posterity. In *ACM SIGGRAPH 2004 Sketches*, SIGGRAPH '04, pages 19–, New York, NY, USA, 2004. ACM. URL <http://doi.acm.org/10.1145/1186223.1186247>.

Studio Vitamin. Lego Calendar Last access: 17 July 2013., 2013. URL <http://vitaminsdesign.com/projects/lego-calendar/>.

J. Walny, S. Carpendale, N.H. Riche, G. Venolia, and P. Fawcett. Visual thinking in action: Visualizations as used on whiteboards. *IEEE TVCG*, 17(12):2508–2517, 2011. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.225.9403>.

J. Walny, Bongshin Lee, P. Johns, N.H. Riche, and S. Carpendale. Understanding pen and touch interaction for data exploration on interactive whiteboards. *IEEE TVCG*, 18(12):2779–2788, 2012. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.260.427>.

Colin Ware. *Information visualization: perception for design*. Elsevier, 2004. URL <http://books.google.fr/books?id=qFmS95vf6H8C>.

W. Weber and H. Rall. Data visualization in online journalism and its implications for the production process. In *Information Visualisation (IV)*, pages 349–356, 2012. URL <http://dx.doi.org/10.1109/IV.2012.65>.

Wikipedia. Sedimentation. Wikipedia., 2013. URL <http://en.wikipedia.org/wiki/Sedimentation>. Last access: March 2013.

Leland Wilkinson, D Wills, D Rope, A Norton, and R Dubbs. *The grammar of graphics*. Springer, 2006. URL <http://www.springer.com/statistics/computational+statistics/book/978-0-387-24544-7>.

Wesley Willett, Jeffrey Heer, Joseph Hellerstein, and Maneesh Agrawala. Commentspace: structured support for collaborative visual analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3131–3140. ACM, 2011. URL <http://vis.stanford.edu/papers/commentspace>.

Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. Strategies for crowdsourcing social data analysis. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems, CHI '12*, pages 227–236, New York, NY, USA, 2012. ACM. URL <http://doi.acm.org/10.1145/2207676.2207709>.

Mark Wilson. How GM Is Saving Cash Using Legos As A Data Viz Tool Last access: 17 July 2013. , 2012. URL <http://www.fastcodesign.com/1669468>.

Pak Chung Wong, H. Foote, D. Adams, W. Cowley, and J. Thomas. Dynamic visualization of transient data streams. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 97–104, oct. 2003. URL <http://dl.acm.org/citation.cfm?id=1947389>.

S.Y. Wu, R. Thawonmas, and K.T. Chen. Video summarization via crowdsourcing. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, pages 1531–1536. ACM, 2011. URL http://mmnet.iis.sinica.edu.tw/pub/wu11_crowdsourcing.pdf.

P Wyeth and G Wyeth. Electronic blocks: Tangible programming elements for preschoolers. *Proc. of the Eighth IFIP TC13*, 2001. URL http://espace.library.uq.edu.au/eserv/UQ:96423/Electronic_blocks.pdf.

Ji Soo Yi, Rachel Melton, John Stasko, and Julie A Jacko. Dust & magnet: multivariate information visualization using a magnet metaphor. *Information Visualization*, 4(4):239–256, 2005. URL <http://www.cc.gatech.edu/gvu/ii/dnm/>.