



HAL
open science

Formulating the design rationale of visual representation

Stéphane Conversy, Christophe Hurter, Stéphane Chatty

► **To cite this version:**

Stéphane Conversy, Christophe Hurter, Stéphane Chatty. Formulating the design rationale of visual representation. 2009. hal-00426040v2

HAL Id: hal-00426040

<https://hal.science/hal-00426040v2>

Preprint submitted on 24 Apr 2010

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.

Formulating the design rationale of visual representation

Stéphane Conversy, Christophe Hurter, Stéphane Chatty

Abstract—When designing a representation, a designer implicitly formulates a method required to understand and use the representation effectively. This paper aims at making the method explicit, in order to help designers elicit their design choices. In particular, we present a set of concepts to systematically analyze what a user must theoretically do visually to find information. The analysis consists in a decomposition of the activity of *scanning* into elementary visualization operations. We show how the analysis applies on various existing representation, and how expected benefits can be expressed in terms of elementary operations. The decomposition highlights the challenges encountered by a user when figuring out a representation, and helps designer to exhibit possible flaws in their design. The set of elementary operations form the basis of a shared, common language for representation designers.

Index Terms—Visualization, Infovis, Design Rationale, Visual design.



1 INTRODUCTION

Designing representation is often considered as a craft that requires a lot of iteration, experience and testing before considered achieved. During iteration, designers have to choose between particular designs. They choose with a mix of ad-hoc testing, discussing with users, controlled experiments, and personal or user preferences. The outcome is a set of fixed representations (often only one), which is a compromise between all the considered, prototyped representations. These ways of designing is either costly (control experiment), or error-prone or leading to non-optimal results (ad-hoc testing, personal preference). Though a number of theoretical works help to explain the strengths or weaknesses of visualization [2][3][5][8][22][23], no method exists that helps comparing similar representations in order to choose one of them, based on a systematic analysis.

In fact, when designing a representation, a designer implicitly formulates a method required to understand and use the representation effectively. For example, reading a city map requires scanning it, find noteworthy locations (metro stations, squares...), devise a path to go from one point to another etc. As we will see, resolving a problem with a representation may not be immediately apparent. As a representation is a compromise, it makes some visualization tasks “evident”, with solutions that “pops out”, but makes other tasks more complicated, sometimes tricky. For this last kind of visualization tasks, it’s like interacting with the eyes only: though not interacting with the visualization, user has to figure out a solution to the problem by scanning the picture, seeking graphics, memorizing things etc.

This paper presents a method and related concepts to analyze how a user deciphers a representation. The goal of the paper is not to show better designs for a particular problem. Rather, the goal of the paper is to present a method that exhibits the required steps to figure out a particular representation. The method consists in decomposing visualization task in elementary visual operation (*a la* Keystroke [7]). We argue that most explanation given by designers about their design can be expressed in terms of elementary operations, and we show how a particular design improves the accomplishment of operations.

2 RELATED WORK

We based our work on previous works that can be roughly divided in two groups. The first one is about theories and design guidelines for visualization. The second one relates various models, whose primary objective is to modelize human perception. A third part is devoted to Semiology of Graphics [5], as it is used extensively in the paper.

2.1 Theories and design

Bertin ([5] p148) pointed the problem of scanning what he considers “figuration” (i.e. bad design). He depicts how the eye scans a graphic. During scanning, the eye jumps from one mark to the other, while experiencing perturbation by other marks. The eye then focuses on particular marks to gather visual information. Bertin also introduces the Semiology of Graphics, which is described in another subsection. Furnas devised the requirements for effective view navigation (small views, limited number of actions to move around, discoverable route to any target) [12].

Tufte offers guidelines and principles to attain “graphical excellence and integrity” [22]. Principles include using quantitative visual variables, or using text to disambiguate wherever needed.

Card and al. proposed a taxonomy of visualization. Transformation from data to graphics are classified with Stevens and Bertin data-type scale (N, O, Q) [6]. MacKinlay designed an automatic system based on the properties of visual variables [17] [18].

Green identified cognitive dimensions of notation, which helps designers share a common language when discussing about design [14]. They help making explicit what a notation is supposed to improve, or fail to support. Cognitive dimensions are based on activities typical of the use of interactive systems such as incrementation or transcription. However, they are high-level descriptions, and do not detail visualization tasks. Our work has means and goals (description, and production of a shared language) than cognitive dimensions, but specialized to visualization.

2.2 Models of perception and action

The keystroke-level model helps to compute the time needed to perform an interaction [7]. The CIS model is a model that takes into account the context in which the interaction takes place [4]. Both keystroke and CIS are based on a descriptive model of the interaction, by decomposing it into elementary operations. They are also predictive models, i.e. they can help compute a measurement of expected effectiveness, and enable quantitative comparisons between interaction techniques. These tools have proved to be accurate and efficient when designing new interfaces.

The Act-R model aims at providing tools that simulate human perception and reasoning [2][3]. Act-R uses features of representation

→ Roy G. Biv is with Starbucks Research, E-Mail: roy.g.biv@aol.com.

→ Ed Grimley is with Grimley Widgets, Inc., E-Mail: ed.grimley@aol.com.

→ Martha Stewart is with Martha Stewart Enterprises at Microsoft Research, E-Mail: Martha.stewart@marthastewart.com.

Manuscript received 31 March 2009; accepted 27 July 2009; posted online 11 October 2009; mailed on 5 October 2009.

For information on obtaining reprints of this article, please send email to: tvcg@computer.org .

similar to Bertin. However, the tool is not targeted towards designer, as its purpose is to modelize human behavior so as to anticipate real-world usage. It does not take into account some arrangement such as ordered, or quantitative layout, nor a description of how a representation is supposed to be used. Act-R has tentatively been used to carry out autonomous navigation of graphical interface, together with the SegMan perception/action substrate [21]. However the interfaces used as testbed are target toward WIMP application, which do not exhibit high-level properties such as Bertin's ones.

Eye tracking enables researchers to analyze what users look at when solving a problem. However, a large part of the literature is devoted to how to process tracking data so as to be able to analyze them [10] [19] [20].

The previous models are incomplete with respect to high-level properties on which designers rely. The next subsection discusses those properties, which stem from Bertin's Semiology of Graphics.

2.3 Semiology of graphics

Semiology of graphics is a theory of abstract graphical representation such as maps or bar charts [5]. It describes and explains the perceptual phenomenon and properties underlying the act of reading abstract graphics. We use this theory as the basis for the analysis of representation scanning.

Semiology of graphics relies on the characterization of data to be represented (the data type), and the perceptual properties of the visual variables used in a representation, such as color or shape. The characterization of data distinguishes three types:

- Nominal: are only equal or different to other values (e.g. a name of a metro station)
- Ordered: obey a < relation (e.g. a rank in a list),
- Quantitative: can be manipulated by arithmetic (e.g. a weight, a cardinal, a price).

Nominal: are only equal or different to other values (e.g. a name of a metro station),

Ordered: obey a < relation (e.g. a rank in a list),

Quantitative: can be manipulated by arithmetic (e.g. a weight, a cardinal, a price).

Abstract representations are a set of marks (be it a point, a line, or a zone) on a blank sheet. Marks use visual variables such as shape, color, luminosity, size, position etc. Visual variables can also be characterized by their perceptual properties. They can help a reader to:

Selective: ...filter marks, in a glance (e.g. all red marks among other colored marks with the same values for all visual variables but color).

Associative: ...ignore differences (eg. all red marks among other colored marks whatever their shape)

Ordered: ...order marks (e.g from light to dark)

Quantitative: ...quantify differences between marks (e.g. twice as large)

A faithful representation uses an adequate mapping between data and visual variables (types must match). For example, colour cannot convey ordered or quantitative information. And even though "quantitative" implies "ordered", a quantitative visual variable makes it hard to order information.

Bertin also defines three levels of reading a representation: the elementary level, which enables the reader to "unpack" visual variables of a single mark, the middle level, which enables the reader to perceive a size-limited pattern or regularity, and the global level, which enables the reader to grasp the representation as a whole, and sees at a glance emergent visual information.

3 THEORETICAL SCANNING OF REPRESENTATION

As said before, when designing a representation, a designer implicitly formulates a method required to understand and use the

representation effectively. The work presented here is an analysis of this method, a way to make it explicit.

When trying to solve a problem with a representation, a user fulfils a visualization *task*, i.e. a set of visual and memory operations. A visualization task can be decomposed into a sequence of *steps*. Each step requires a sequence of elementary visualization *operations* to be accomplished. Operations include *entering the representation*, *memorizing information*, *seeking a subset of marks*, *unpack a mark and verifying a predicate*, *seeking and navigating among a subset of marks*, *exiting from the representation*. As we will see in the remaining, operations are facilitated by the use of (possibly) adequate visual cues (Bertin's *selection*: colour, size, alignment).

In the following, we analyze *theoretical* scanning of representation. We qualify scanning as theoretical because we analyze an idealized way of navigating through the representation. The scanning is ideal when the user knows exactly what she is looking for, how to use the representation so as to step through with the minimum, but necessary, amount of steps, and act accordingly. Thus, we do not take into account other phenomena such as learning, understanding, error, chance, or personal perceptual disabilities (like color blindness). This is similar to the approach taken with the Keystroke-level model: when applying a decomposition, the designer analyzes an idealized interaction, as if the user knows exactly what she has to do to, acts accordingly, and without any errors nor hesitation.

The next section shows with an example how to perform an analysis of representation scanning. Based on this, we detailed further the steps and operations required, and what factors affect users efficiency at achieving them.

4 A FIRST GLIMPSE: BUS SCHEDULE REPRESENTATION

There is no such thing as an absolute effective representation: to be effective, a representation must reduce the amount of work required to fulfil a task. Thus, we need an analysis of the activity to identify the tasks, before assessing if a representation is adequate. The problem to be solved by a user is to answer the following question: "I am at the *IUT Ranguueil* station, it is 14h18. How long should I wait for the next bus to station *Université Paul Sabatier*?". The user knows that two bus lines go to the destination (bus #68 and #108).

Fig. 1 is an excerpt of a typical representation of a bus schedule. The display is a physical panel at station booth, on which lay paper sheets with a table containing the time of passage at stations for a line during one day. The drawing overlaid over the representation shows the ideal visualization tasks a user has to perform when trying to answer the question. A circle depicts an eye reading, an arrow an eye movement. A red shape depicts an intermediate operation inside a step, while a green shape depicts the last operation of a step. Some circles are both red and green because of back steps over a previously visited mark.

Fig. 1 shows multiple strategies to answer the question. The following details the second one (corresponding to line 108), but the same reasoning applies to the other one (line 68). The step number are in the form $x.y.z$, which means that step y is the y th substep of step x , and step z , the z th step of step y .

Step 0: the user should memorize two bus line numbers, and the current time

Step 1: the user should find a correct bus line. The number of the line is represented with a text, with large typeset and a bold face, placed at the top-right corner of a paper sheet.

Step 1.2: the user should find the correct destination (here "IUT Ranguueil") among a list of stations. The list is a subset of marks of kind "text", aligned vertically, with no marks in-between. The station are ordered according to the rank of station along the line.

Step 1.3: the user must find a compatible time of passage. He has to navigate through a row of texts that displays hour and minute for each bus passage. As the X dimension is multiplexed over Y, the user may not find a compatible time in this row: he has to start Step 1.2 over by moving to the next row (Step 1.4). Finally, the user finds a

compatible time when he finds the first time of passage that is superior to the current time.

Step 2: the user has to perform mental computation (a difference between two times) to find the duration before the next bus.

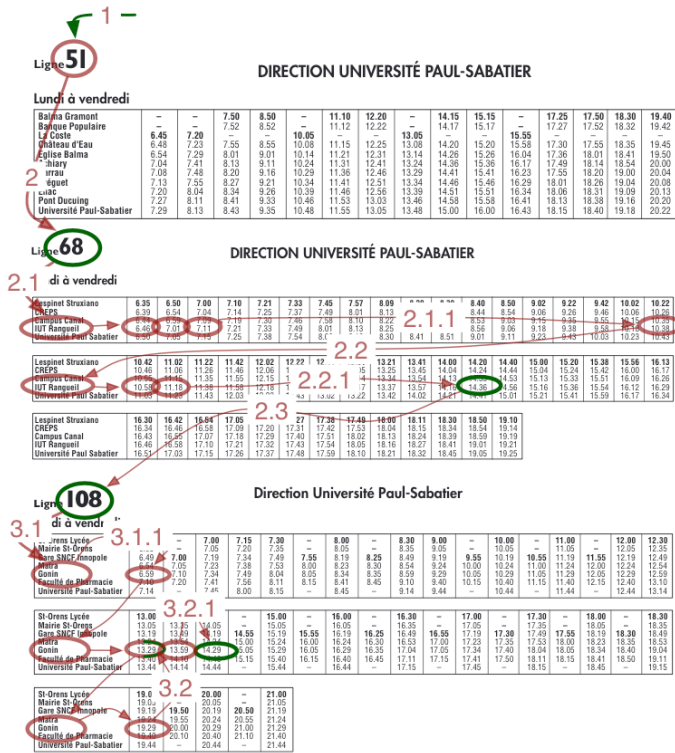


Fig. 1 : A bus schedule representation with the required steps to find particular information.

5 ELEMENTARY OPERATIONS

This section details the various elementary operations required to implement the steps. For each operation, we detail it, and give elements that aid or hinder operation achievement.

5.1 Entering the representation

A representation is rarely used in isolation. Users are surrounded by different representation from various systems. When they solve a problem, users may have to transform the input they have access to, into the language of the considered representation. In the bus schedule example, they may translate the representation of a time they see on a watch, into numbers in the form hh:mm so as to comply with the ordered-by-time menu-like vertical representation. Taking into account this step is important when a switch of representation does not require translation, and makes the second representation easier to apprehend.

5.2 Memorizing information

Users have to know which information to seek. They have to memorize these information, so as to compare them to the information that arise from the representation. As we have seen in the examples, different representations require different memory capacity, which may hinder performances. For example, in the tabular bus scheduling view, users have to use three cells of information: current time, 68 and 108. Memory requirements are overlooked when comparing visualization: memory fades with time, and considering the number of steps required with tabular representation, users may have forgotten important information before the end of the scanning.

5.3 Seeking a subset of marks

When users search for line bus information, they have to search for a subset of all the marks in the representation. In order to find the correct line, the user has to navigate from line number to line number. Perceiving a subset is made easier with selective visual variables: marks can be extracted from the soup of all marks at one glance, which narrows down the number of marks to consider. For example, the number of the bus line is represented with a text, with a large typeset, a bold face, and placed at the top-right corner of the sheet. The size and position of bus line number make the marks selectable. Furthermore, when elements in a subset are closed enough, no other in-between element perturbs the navigation from mark to mark. The list is even easier to navigate in, as marks are aligned horizontally or vertically list (or in other words, marks differ by only one dimension (X or Y)).

On the opposite, perceiving a subset can be harder in presence of similar marks that do not belong to the considered subset. All time information has similar properties, except for the start time of each bus, which is bold. If it were regular, it would be harder to find in a glimpse the starting time of bus.

5.4 Unpack a mark and verifying a predicate

When the user sees a candidate mark, she has to assess it against a predicate. In the tabular bus schedule example, the user has to find a line number that matches one of the correct bus. Assessing a predicate may require extracting or unpacking [1] visual dimension from a mark. This is what Bertin calls "elementary reading". Assessing a predicate may require cognitive comparison to memorized information (is that bus number a memorized one?), or visual comparison with another mark (example in the following).

5.5 Seeking and navigating among a subset of marks

Among a subset, a user may search for a particular mark. If marks are displayed in a random position, finding a mark requires a linear, one-by-one scanning of marks, with a predicate verification for each. The time needed is $O(n)$. If marks are ordered (as in the ordered-by-time schedule), a user can find this regularity to speed up navigation, for example, by using a dichotomy approach, which leads to a $O(\log(n))$. If marks are displayed at quantitative position, we can hope for $O(1)$. However, it may require secondary marks, such as a scale ticks and legend: in this case, scanning is split into two phases: navigating into the scale first, then into primary marks.

Navigating inside a list of texts is equivalent to reading a menu, for which performance may be known quite accurately [9]. However, some graphical elements may hinder navigation. For example, navigating in a row surrounded by other rows, such as in a matrix, is difficult. It is the equivalent of a visual steering law: it requires that the eye be capable to stay in a tunnel. Some representation are supposed to aid, but has not been proved effective (think of excel sheet that are supposed to facilitate the task with one or two colored row). This depends on the width of the tunnel. Navigating inside a vertical list of text is easier than navigating in a horizontal one, since a horizontal row is as narrow as the height of a glyph.

5.6 Exiting from the representation

As much as users have to translate information into the language of the representation, users have to translate back the information to the problem. In the bus schedule example, the information to get is the waiting time for the next bus. The tabular representation does not give the information directly, as it requires a mental computation. In the city map example, translating map direction to real-world or recognizing street layout is easier if the map is oriented like the terrain (North of map matching the actual North direction).

6 FORMULATING DESIGN RATIONALE

We argue that designers implicitly design a method composed the elementary operations when they invent new representation. We also think that most explanation given by designers can be expressed in terms of elementary operations, and in particular in how a particular design improves operation achievement. In the following, we present various designs for bus schedule and ATC strip papers. We express the expected gains with the concepts presented above. We balance the claims by our own analysis, and possible lost of performance due to non-support of overlooked operation. We based our analysis on existing literature when available, or by directly interviewing designers.

6.1 Ordered-by-time linear representation



Fig. 2 : An ordered-by-time bus schedule representation with the required steps to find particular information.

The bus company proposes the representation in Fig. 2 on its web site. It displays an ordered list of time of passage at the departure station along the X dimension, with the corresponding bus as a cell containing a background color and white text. The required steps are:

- Step 0: memorize current time (*memorization*)
- Step 1: find the first time superior to the current time (*predicate*)
- Step 2: find next compatible bus (*predicate*)
- Step 3: find the associated time (*seeking a mark*)
- Step 4: compute the waiting time before the passage (*exiting*)

The following operation may be eased by the design:
seeking and navigating among a subset of marks: as time of passage are displayed in a ordered manner, it may ease navigation
seeking a subset of marks: the user can select elements at the right of step 2 (selection based on location).

6.2 Spiral representation

SpiraClock is an interactive tool that displays nearby events inside an “analog clock” (according to their authors [11]). Time is mapped to angle, but nothing is mapped to radius, which allows representing more information by multiplexing the angle over the radius (Fig. 3). The watch also displays the current time, and adapts the event occurrences accordingly. The occurrence of the event is actually depicted by the “most recent” limit of a “slice”. A duration is a relative angle, or a curvilinear distance, which makes it represented in a quantitative way (more accurately on the exterior of the spiral (*i.e.* for close events) than in the interior). There is also a scale depicted with black squares along the circle. The designers considered that adding textual information about hours is useless, as the design uses a common/known reference (a watch) and because the visualization is focused on current time. The steps required to answer the question are:

- Step 0: memorize two bus colors (*memorization*)
- Step 0: find the end of minute hand (*seeking a mark*)
- Step 2: find the next matching colored mark (*i.e.* corresponding to line 68 or 108) (*seeking a mark*)
- Step 3: evaluate the distance between the matching mark and the minute hand, and estimate the waiting time (*unpack and exiting*)

The following operation may be eased by the design:
exiting: with SpiraClock, a rough idea of the waiting time is directly visible, as it is proportional to distance, and the design uses a known, shared scale.

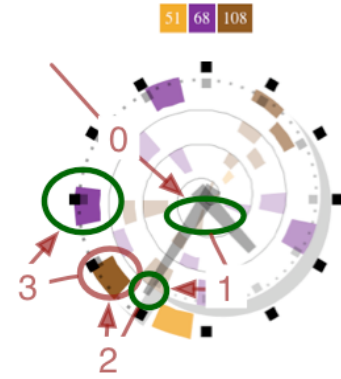


Fig. 3 : SpiraClock

6.3 Quantitative linear representation

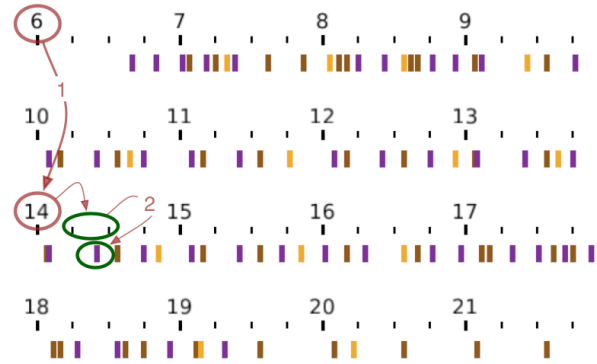


Fig. 4 : A linear, quantitative bus schedule representation with the required steps to find particular information

Fig. 4 shows a representation based on a linear quantitative scale. Each colored rectangle is a passage of a bus at the departure station. The horizontal position of a rectangle corresponds to the time of passage, and is multiplexed along the vertical dimension. To aid navigation, a linear scale is provided, with textual information about hours, and small ticks to mark quarters between hours. The procedure to find the next bus consists in finding the hour, then the quarter corresponding to the minute. Starting from this position, the user must look at the colored marks, and sweep horizontally until the next colored mark. The distance between the mark and the position on the scale gives the time to wait for the bus (no computation is needed). The firsts steps correspond to the act of entering the representation. If users had a watch with a similar linear layout, it may have been easier. This representation is denser than SpiraClock: it displays all the needed information, and is thus suitable for other tasks, such as finding a bus at a particular time in the future. It does not require an interactive system, and can be printed on a sheet of paper. The spiral equivalent is shown Fig. 5. Compared to the other SpiraClock, a scale is necessary to find a time quickly (*seeking and navigating*). However, seeking an event requires a steering law inside a narrow spiral, which is hard to do.

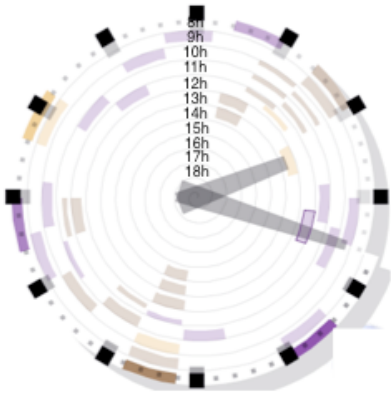


Fig. 5 : SpiraClock, showing more information.

6.4 ATC strip papers

The activity of Air Traffic Controllers (ATCOs) consists in maintaining a safe distance between aircrafts by giving clearances to pilots (heading, speed, or level (altitude) orders). ATCOs must detect potential conflicts in advance. To do so, they use various tools, including a radar view, and flight strips [16]. A flight strip is a paper that shows the route followed by an airplane when flying in a sector (Fig. 6). The route is presented as an ordered sequence of cells, each cell corresponding to a beacon, with its name, and its time of passage. Controllers lay paper strip on a strip board, usually by organizing them in column. The layout of strips on a board, though physical, can be considered as a representation, as much as the radar image.

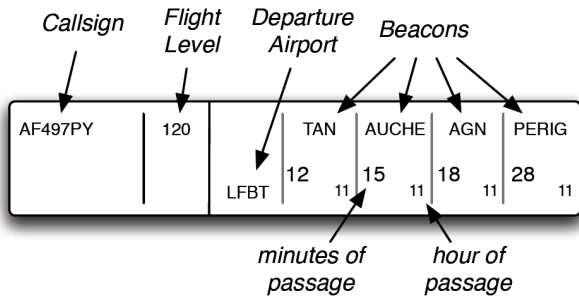


Fig. 6 : An ATC paper strip.

One of the activities of a controller is to integrate the arrival of a flight into the current traffic. To do so, the controller has to check that for each beacon crossed by the new flight, no other flights cross this particular beacon at the same time. Fig. 7 shows the corresponding theoretical scanning, with typical paper strips organized in column. The steps are:

- Step 0: find the beacon texts on the arrival strip, and for each beacon (horizontal text list scanning, with no perturbation), do the following steps (*seeking and navigating*)
- Step 0.1: memorize the beacon text, find the minute information (hour is usually not important), and memorize it (*memorizing*)
- Step 1: for each other strips (vertical rectangular shape list scanning), do the following steps (*seeking and navigating*)
- Step 1.1: find the beacon texts, and for each beacon (horizontal text list scanning, with no perturbation), do the following steps (*seeking and navigating*)
- Step 1.1.1: compare the beacon text to the one memorized in step 0.1. If it is the same, find the minute text, compare it to the one memorized in step 0.1. If the number is about the same (+5 min), find flight level, and check it and compare with memorized level. If it is the same, do something to avoid a conflict. (*predicate*).

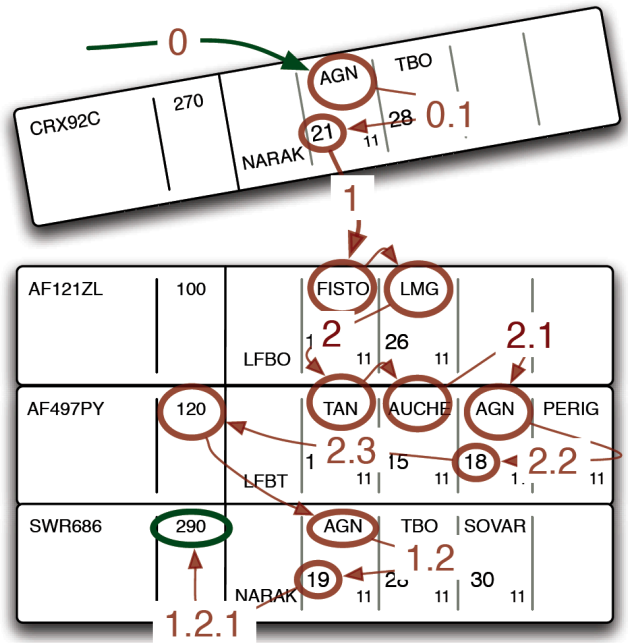


Fig. 7 : scanning on regular ATC paper strip.

Prospective systems aim at replacing papers with entirely digital systems, so as to capture clearances into the system (currently the system is not aware of clearances from the controllers to the pilots). Those systems replicate partly the existing representation, and we show in the following how they compare with respect to representation scanning.

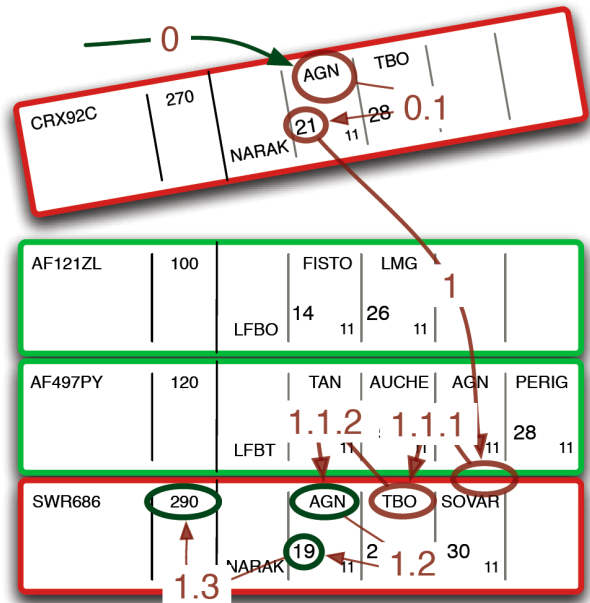


Fig. 8 : Scanning with paper strips in colored holders.

6.5 Strips in colored holders

The previous strip look and layout is typical to the En-Route control centre at Bordeaux, France. In other En-Route control centres, people use rigid, colored holders in which they slide a paper strip. The look of strips is different, since each strip is surrounded by the colored frame of the holder. Fig. 8 shows a theoretical scanning with coloured strip holders : red is for North-South flights (odd flight level), while green is for South-North flights (even flight level). Because of the different parity, controllers are sure that red and green

flights never enter into conflict. Red holders can quickly be extracted from green ones (selection based on color). Hence, colored strip holders enable controllers to narrow the set of flights to compare with the new one, and lower the number of required steps accordingly (*seeking and navigating*). Holder color can also ease *predicate verification*: holder color of the arriving strip can be matched easily to holder color of other strips, without requiring the user to determine if it's a North-South or a South-North flight.

AF497PY	120		TAN	AUCHE	AGN	PERIG
		LFBT	12 11	15 11	18 11	28 11
AF121ZL	100		FISTO	LMG		
		LFBT	14 11	26 11		
CRX92C	270		AGN	TBO		
		NARAK	21 11	28 11		
SWR686	290		AGN	TBO	SOVAR	
		NARAK	19 11	28 11	30 11	

Fig. 9 : A refinement of paper strips, in which beacons are highlighted.

Fig. 9 shows a refinement of paper strips in which beacons are in Bold Face. This is supposed to facilitate beacons navigation (*seeking and navigating*), as they are selectable from other marks. For example, the font of beacons on strip is similar to the font of other text (such as hour, or departure airport): users may look at those other marks and spend time to discover that they do not belong to the subset they seek in.

6.6 Pen-based digital stripping system

Fig. 10 shows a digital, pen-based system, that adds an interaction allowing the controller to press a beacon cell, so as to highlight in red the minutes of passage over that beacon on other strips. This facilitates *seeking and navigating*, as it reduces the subset of marks to consider when comparing times of passage.

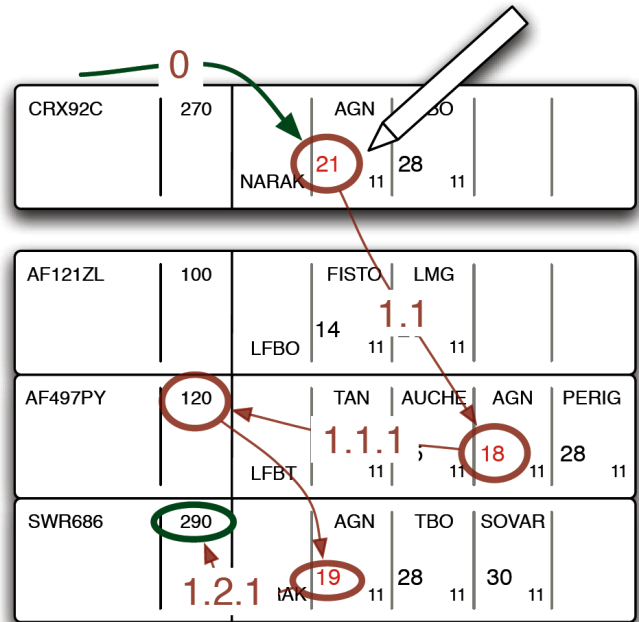


Fig. 10 : A pen-based digital stripping system, that enable highlighting of information.

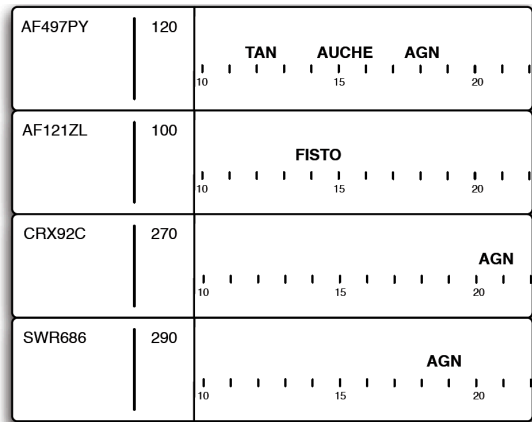


Fig. 11 : Dynastrip.

6.7 Dynastrip

Dynastrip displays beacons in a quantitative way, since time is mapped to X (Fig. 11) [13]. All time scales are aligned across strips. The main goal of Dynastrip is to display position relative to route in the strip, which adds information. Dynastrip designers also hoped that it would enable controllers to catch conflicts: if beacons with the same text are vertically aligned, it means that multiple flights pass over the same beacon at the same time. On Fig. 11, three flights pass over AGN at roughly the same time. Integrating a strip thus requires to step through all beacons of the arriving flight, and for each beacon do a visual vertical steering to see if a beacon is in the “tunnel”, and compare the beacon with the considered one (*seeking and navigating*).

6.8 Relative position of flight between beacons

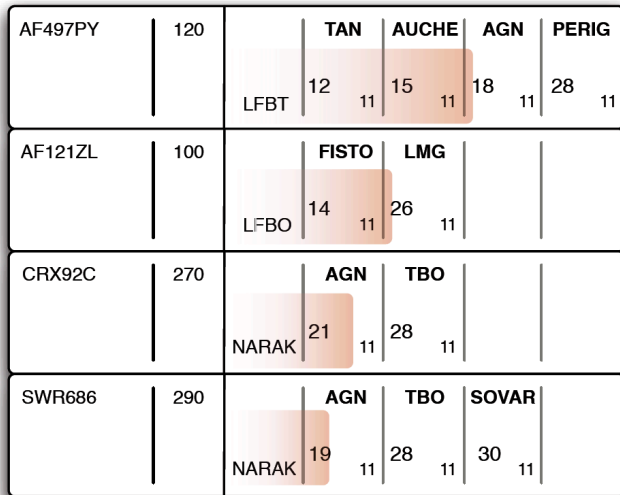


Fig. 12 : digital strips with dynamic gradients.

The designer of the gradient in Fig. 12 hoped that users would benefit from knowing where a plane is compared to the prevision. However, it requires a not-so-easy translation back to the conceptual problem (*exiting*). The gradient position is quantitative, and reflects the position of the plane between two beacons. Since beacon distances are arbitrary, the (invisible) scale inside each cell is different. The user has to compute this distance by using knowledge of the airspace configuration.

The designer also states that it enables the user to assess quickly if a flight is going to get out of the sector: if the gradient is in the last beacon, then the controller should perform a “shoot request” to the next sector. However this requires that the user first assess which beacon is the last one, then assesses if the gradient is in the corresponding cell. A further design that aligns the cell to the right (i.e. to the last flown over beacon) eases this task (Fig. 13), as it only requires checking if a gradient reached a right-most cell by scanning through the corresponding column (in other words, do a selection according to the X visual variable) (*seeking and navigating*).

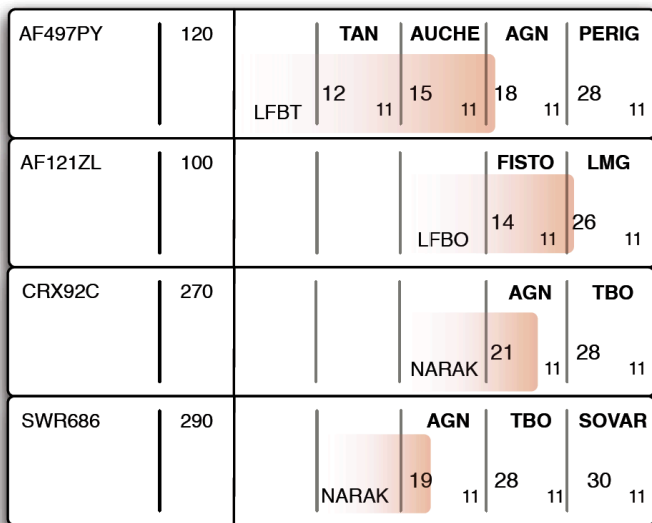


Fig. 13 : digital strips with dynamic gradients aligned to left

7 VALIDITY AND LIMITATION

Theoretical scanning is, as its name implies, only theoretical. We do not know if it holds in real world, which questions the validity of the

work presented here. However, we think that designers implicitly rely on theoretical scanning (though designers' expectations do not always stand against reality [15]). A deeper understanding of the phenomena is thus necessary, to explicit design choices and expected benefits, and to get a reasonable confidence in the design.

Bertin's semiology of graphics has not been assessed, neither Furnas' Effective View Navigation. Nevertheless their concepts permeate a large number of visualization designs. They allow identifying relevant concepts and dimensions when analyzing or designing new visualizations. We think that the elementary operations that we identified in this paper will serve as a similar framework for representation rationale. In the same way, we do not know if navigation in an ordered set is easier than in a random set, and if navigation in a quantitative one is easier than in an ordered subset. Again, a number of visualizations rely on it: making it explicit helps designers think about the effectiveness of their design.

The absence of distinction between “beginners” and “experts” seems doubtful as well. This is clearly the case with the ATC example: we know that ATC controllers do not scan the strips the way we theorized it. They heavily used their knowledge of the sector, recurrent problems and recurrent aircraft, to detect conflicts. Again, our description aimed at explicating what the visualization enables for a naïve reader. However, during a course of action, ATC controllers regularly do what they call a “tour of the radar image”, or a “tour of the strip board”, so as to check “everything”. In this case, they are supposed to heavily scan both representations, and may exhibit some of the theorized behaviour. Furthermore, we observed that ATC controllers make errors when training on a new sector, because some representations have flaws. Flaws are compensated by expertise, which is somewhat related to knowledge in the head and memory (in some cases, an ATC controller is considered as an expert on a sector after 2 years of training). However, in high load situations, with lots of aircraft, or with particular problematic conditions such as unexpected storms, the representation is becoming more important, and controllers may exhibit the theorized behaviour.

8 CONCLUSION AND RESEARCH AGENDA

In this paper, we presented a method to analyze theoretical scanning of graphical representations. We defined a set of elementary operations, and argue that rationale for design can be expressed in terms of elementary operations. We showed on various examples how such an analysis can be achieved, and how gains and losses can be explained with elementary operations. The set of elementary operations forms the basis of a shared, common language for representation designers.

We have successfully applied the analysis method presented here on other representations, such as calendars (month, week and day view), items rating by customers in online stores, widgets, or radar images. Further research is clearly needed, for which we formulate the following research agenda. First, we need to conduct experiments to compare real world representation scanning to theory. We are currently doing this using an eye-tracking system. Results should validate and invalidate the arguments presented here (but again, arguments come from designer expectancy). Based on the results, and with more experience, we should refine and extend the set of elementary operations. The second step would be to measure the elementary operations under various conditions, so as to turn the descriptive model into a predictive model. This would complement the Act-R model, which does not take into account some of the expected phenomena presented here.

ACKNOWLEDGEMENTS

The authors wish to thank A, B, C. This work was supported in part by a grant from XYZ.

REFERENCES

- [1] Amar, R., Stasko, J., A Knowledge Task-Based Framework for Design and Evaluation of Information Visualizations, Proceedings of IEEE InfoVis '04, Austin, TX, October 2004, pp. 143-149.
- [2] Anderson, J. R., Matessa, M., & Lebiere, C. (1997). ACT-R: A theory of higher level cognition and its relation to visual attention. *Human Computer Interaction*, 12(4), 439-462.
- [3] Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review* 111, (4). 1036-1060.
- [4] Appert, C., Beaudouin-Lafon, M., Mackay, W. (2004) Context Matters: Evaluating Interaction Techniques with the CIS Model. In *People and Computers XVIII - Design for Life - Proceedings of HCI 2004*, Leeds, UK. Springer Verlag, pages 279-295.
- [5] Bertin, J. (1967) *Sémiologie Graphique - Les diagrammes - les réseaux - les cartes*. Gauthier-Villars et Mouton & Cie, Paris. Réédition de 1997, EHESS.
- [6] Card, S.K., Mackinlay, J.D., Shneiderman, B., *Readings in Information Visualization: Using Vision to Think*. San Francisco, California: Morgan-Kaufmann, (1999)
- [7] Card, S. K., Moran, T. P., and Newell, A. 1980. The keystroke-level model for user performance time with interactive systems. *Commun. ACM* 23, 7 (Jul. 1980), 396-410.
- [8] Cleveland, W., McGill, R., *Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods*. *Journal of the American Statistical Association* 79, (1984)
- [9] Cockburn, A., Gutwin, C., and Greenberg, S. 2007. A predictive model of menu performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (San Jose, California, USA, April 28 - May 03, 2007)*. CHI '07. ACM, New York, NY, 627-636.
- [10] Crowe, E. C. and Narayanan, N. H. 2000. Comparing interfaces based on what users watch and do. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications (Palm Beach Gardens, Florida, United States, November 06 - 08, 2000)*. ETRA '00. ACM, New York, NY, 29-36.
- [11] Dragicevic, P. and Huot, S. 2002. SpiraClock: a continuous and non-intrusive display for upcoming events. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems (Minneapolis, Minnesota, USA, April 20 - 25, 2002)*. CHI '02. ACM, New York, NY, 604-605.
- [12] Furnas, G. W. 1997. Effective view navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Atlanta, Georgia, United States, March 22 - 27, 1997)*. S. Pemberton, Ed. CHI '97. ACM, New York, NY, 367-374.
- [13] Grau, J.Y.; Nobel, J.; Guichard, L.; Gawinoski, G. "Dynastrip": a time-line approach for improving the air traffic picture of ATCOS. *Digital Avionics Systems Conference, 2003. DASC'03. The 22nd Volume 1, Issue , 12-16 Oct. 2003* Page(s): 5.E.1 - 51-11 vol.1, IEEE.
- [14] Green, T.R.G (1989). Cognitive dimensions of notation. In *People and computers V*, Cambridge University Press, pp443-460.
- [15] Johansen, S. A. and Hansen, J. P. 2006. Do we need eye trackers to tell where people look?. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (Montréal, Québec, Canada, April 22 - 27, 2006)*. CHI '06. ACM, New York, NY, 923-928.
- [16] MacKay, W. E. 1999. Is paper safer? The role of paper flight strips in air traffic control. *ACM Trans. Comput.-Hum. Interact.* 6, 4 (Dec. 1999), 311-340.
- [17] Mackinlay, J. 1999. Automating the design of graphical presentations of relational information. In *Readings in information Visualization: Using Vision To Think*, S. K. Card, J. D. Mackinlay, and B. Shneiderman, Eds. Morgan Kaufmann Publishers, San Francisco, CA, 66-82.
- [18] Mackinlay, J., Hanrahan, P., and Stolte, C. 2007. Show Me: Automatic Presentation for Visual Analysis. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov. 2007), 1137-1144.
- [19] Riedl, M. O. and St. Amant, R. 2002. Toward automated exploration of interactive systems. In *Proceedings of the 7th international Conference on intelligent User interfaces (San Francisco, California, USA, January 13 - 16, 2002)*. IUI '02. ACM, New York, NY, 135-142.
- [20] Salvucci, D. D. and Anderson, J. R. 2001. Automated eye-movement protocol analysis. *Hum.-Comput. Interact.* 16, 1 (Mar. 2001), 39-86.
- [21] St. Amant, R. & Riedl, M.O. A perception/action substrate for cognitive modeling in HCI. *International Journal of Human-Computer Studies*, 55, 1 (2001), 15-39.
- [22] Tufte, E. R. *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, CT, 2nd ed., 2001.
- [23] Wilkinson, L., *The grammar of Graphics*. New York: Springer Verlag (1999).