Understanding Visualization:
A Formal Approach using Category Theory and Semiotics

Paul Vickers, *Member, IET* Joe Faith, and Nick Rossiter.

Abstract—This article combines the vocabulary of semiotics and category theory to provide a formal analysis of visualization. It shows how familiar processes of visualization fit the semiotic frameworks of both Saussure and Peirce, and extends these structures using the tools of category theory to provide a general framework for understanding visualization in practice, including: relationships between systems, data collected from those systems, renderings of those data in the form of representations, the reading of those representations to create visualizations, and the use of those visualizations to create knowledge and understanding of the system under inspection. The resulting framework is validated by demonstrating how familiar information visualization concepts (such as literalness, sensitivity, redundancy, ambiguity, generalizability, and chart junk) arise naturally from it and can be defined formally and precisely. This article generalizes previous work on the formal characterization of visualization by, *inter alia*, Ziemkiewicz and Kosara and allows us to formally distinguish properties of the visualization process that previous work does not.

Index Terms—I.6.9.c Information visualization, G Mathematics of Computing, category theory, semiotics

✦

1 INTRODUCTION

VISUALIZATION is a catch-all term that embraces a wide range of activities concerned with representing, or making visible aspects or features of a given set of data or system, from the graphical analysis of scientific data, through the 'infographics' used to communicate in the popular media, to data art. It has recently grown in scale, popular currency, and theoretical discussion due to a combination of factors including the growth in the importance of data mining and processing in industry and science, and the availability of popular and powerful computer visualization tools, such as Processing (see [1]).

Visualization practice combines a range of skills and disciplines, including statistics, aesthetics, HCI, graphic design, and computer science. However, perhaps because of this diversity, there has been relatively little discussion of its theoretical basis. Purchase et al. [22] remarked that visualization "suffers from not being based on a clearly defined underlying theory", and that "formal foundations are at a nascent stage". The danger of neglecting the theoretical foundations is that the discipline will fragment into isolated communities of practice that fail to learn from one another and replicate work unnecessarily. There have been a number of efforts to map visualization's theoretical foundations, the principal ones being Mackinlay [19], Card and Mackinlay [4], Chi [7], Tory and Möller [25], van Wijk [28] and

Ziemkiewicz and Kosara [30] (see also [16]). We discuss the relationship between these treatments and ours in section 6 and show how using a more powerful formal framework allows us to formally distinguish properties of the visualization process that previous work does not.

This article contributes to this development of theoretical foundations by generalizing previous work and providing a higher-level framework for understanding the visualization process. It does this by employing two existing tools. The first is semiotics, the study of signs. As devised by Saussure and Peirce, semiotics has developed into a powerful theoretical framework for understanding the relationships between signs, sign systems, the consumers of those signs, and the systems they represent. Information visualizations are signs *par excellence*, and thus seem obvious candidates for semiotic analysis. In section 2 we provide a brief introduction to the analysis frameworks used by semiotics and discuss how this applies to visualization. The result is to show how visualization can be understood using a series of relationships, or mappings, from one domain to another, summarized in a semiotic triad.

The second tool is category theory, the mathematical study of systems of structures and their mappings. This is introduced in section 3, and applied to visualization in section 4. The most powerful concept in category theory is the notion of commutativity, which forces one to try to extend and construct structures in such a way as to reach algebraic closure by considering the consequences and implications of a structural description of a system. The end result, or closure, is the general description of visualization given in section 4.3 culminating in Fig. 10 and Table 2. By applying the criteria of commutativity and closure to Peirce's semiotic triad in sections 4 and

• P. Vickers and N. Rossiter are with the School of Computing, Engineering, and Information Sciences, Northumbria University, Newcastle-upon-Tyne, NE2 1XE, United Kingdom.
E-mail: paul.vickers@northumbria.ac.uk
• Joe Faith is with Google Corp. Inc., Mountain View, CA 94043, USA.

5 we use our framework in a natural way to cover (or uncover) some aspects which are already familiar to practitioners of information visualization, and some other aspects which are less obvious. Specifically, the theory lets us differentiate between chart junk that is part of the schema, that which is arbitrary, and redundancy in the layout. We also show how the derived property of 'intensionality' can be used to discriminate between true visualizations and data-driven representations.

2 SEMIOTICS AND VISUALIZATION

Semiotics is the study of the creation and interpretation of signs. Signs are words, images, sounds, smells, objects, etc. that have no intrinsic meaning and which become signs when we attribute meaning to them [6]. Signs stand for or represent something beyond themselves. Modern semiotics is based upon the work of two principal thinkers, the Swiss linguist Ferdinand de Saussure and the American philosopher and logician Charles Sanders Peirce. In Saussure's linguistic system a sign is a link between the *signified* (a concept) and the *signifier* (a sound pattern) both of which are psychological constructs having non-material form rather than material substance [6]. For example, /tree/ is a signifier for the concept of the thing we know as a tree. The sign thus formed is a link between the sound pattern and the concept. Modern applications admit material form for the signifier (e.g., road signs and printed words). Saussure's scheme explicitly excludes reference to objects existing in the real world; the signified is not directly associated with an object but with a mental concept.

Peirce's semiotics is based upon a triadic relationship that comprises [29]:

- The *object*: the thing to be represented (note, this need not take a material form);
- The *representamen*: the form that the sign takes (word, sound, image, etc.,); the representamen represents the object;
- The *interpretant*: the sense we make of the sign.

Peirce thus admits the referent that Saussure brackets. Fig. 1 (adapted from Vickers [29]) shows a Peircean triad drawn as a 'meaning triangle'.[1] It should be noted that the Saussurean signifier and signified correspond only approximately to Peirce's representamen and interpretant; unlike Saussure's signified, Peirce's interpretant itself becomes a sign vehicle in the mind of the interpreter. The triangle shows the sign formed by the name Agamemnon which signifies an individual cat of that name.

To see how this relates to visualization consider Fig. 2 (adapted from Vickers [29]) which shows a semiotic relationship between a set of student marks and an external representation. Using a spreadsheet program we take a data set collected from the real world system of a
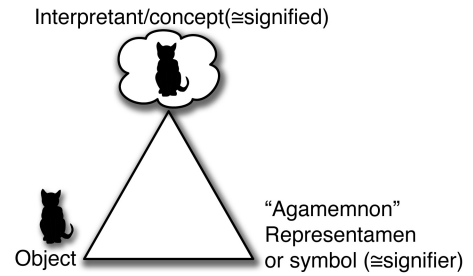
1. In 1923 Ogden and Richards [20] gave this visual interpretation of Peirce's triad since when it has become the conventional representation (see also Sowa [24]).

Fig. 1. Semiotic 'meaning triangle' showing a Peircean semiotic triad (notation after Sowa [24]). Approximations to Saussure's terms are given in parentheses: A real-world cat is the object. It is represented by the symbol "Agamemnon" which evokes the concept of Agamemnon the cat in our mind.

cohort of students studying a course. These data are then presented to the user as a chart. Note, the chart is not the data set but a representation of it. So, we have a Peircean sign in which the data set is the referent object, the chart serves as its representamen, and the interpretant is the sense we make of the data by studying the chart.
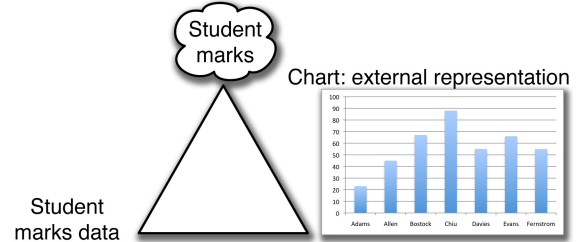

Fig. 2. A chart is a representation of the data. The interpretant is the concept formed in our mind when we read the chart representation.

It is important to note that, contrary to Saussure's original structuralist view, sign systems exist within a social and cultural context which, the post-structuralists would argue, needs to be accounted for. Peirce's semiotics, through the notion of a ground, admits such a context. It is important because visualization requires the producer (the addresser, in semiotic terminology) and the consumer (the addressee) to share contextual knowledge for successful meaning making to take place (see also van Wijk [28]). In Peirce's semiotics meaning is mediated such that the "meaning of a sign is not contained within it, but arises in its interpretation" [6]. Hjelmslev [13] recognized that no sign can properly be interpreted without first contextualizing it so that in addition to a sign's denotative (literal) meaning its context also lends it connotative meaning. For instance, in the example of our student system the ground would include knowledge about what constitutes a pass mark and where the grade boundaries lie (see also Hullman and Diakopoulos [14]).

Note that this semiotic framework excludes some examples of what is popularly regarded as information visualization such as Radiohead's "House of Cards" video [23]. The video was shot without any cameras,

being derived solely from data obtained from 3D images produced by Geometric Informatics for close proximity objects and Velodyne Lidar for landscapes. The data sets used to make the video can be downloaded from the project's web site which encourages us "to create your own visualizations" [23]. According to Lima [17], the video, while unarguably data-driven, ought not be considered a visualization as it provides no insight into the data — it is pure spectacle [2]. As Card, Mackinlay, and Shneiderman put it: "The purpose of visualization is insight, not pictures" [5]. Visualization, then, is a process that begins with the real world, or more narrowly, a system in the real world about which we are interested (e.g., a mechanical system or a cohort of students on a degree course). From the system we gather data which are then mapped via transformation rules to a representation (a graph or chart, an interactive 3D model, statistical box plots, etc.) and this representation is then 'read' by the person wishing to gain insight into the system. Note that the data being visualized always have a context (a ground), a scenario in which they were intended for use.

Reading the representation evokes concepts and ideas in the mind and inferences are drawn leading to understanding of the system and, as we shall see later, knowledge of the truth as it pertains to that system. This process is encapsulated in Fig. 3.
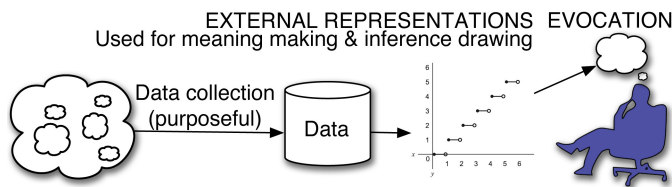

Fig. 3. Understanding the real world through visualization: Data are purposefully collected from the real world and, via mappings, representations are produced. These are used in turn for meaning making and drawing inferences about the data. The visualization process encompasses cognition in the observer's mind.

3 CATEGORY THEORY

In this section we introduce the main concepts of category theory [18], a branch of mathematics developed to analyze systems of structures, and mappings between those structures, in their most general form. This level of generality means that category theory is capable of demonstrating similarities between disparate fields of mathematical enquiry — from set theory to theoretical computer science — in such a way that allows insights from one to be translated to another. It can thus act as a Grand Unified Theory in mathematics.

In sections 3.1–3.3 we introduce the basic elements and concepts of category theory, and in sections 4.1–4.3 we show how these elements and concepts apply to visualization processes. In section 5 we derive some results of analyzing information visualization in this way.

3.1 Morphisms, Objects, and Triangles

Here we introduce the central concepts of the object and the morphism and how they may be arranged into diagrams, the fundamental one being the triangle. In section 4 we will see that the elements of a visualization process (such as data sets and representations) are objects and the processes that map between them (such as measurement and rendering) are morphisms.

Category theory is built from just two classes of entity: objects and morphisms (we follow the convention that Objects are Capitalized and *morphisms* are *italicized*). Almost anything can be considered as an object: physical objects, abstract objects, or entire systems. Indeed, much of category theory's power comes from its recursive ability to treat ever more complicated systems as building blocks in the next level of abstraction. All that is required for an entity to qualify as an object is that it can be individuated, that is, we have some method for determining whether two objects are identical.[2] This method is represented as a mapping from the object to itself, known as the identity morphism (see section 3.2).

Morphisms are mappings between objects. They are represented diagrammatically by (and often called) arrows. For a mapping to qualify as a morphism there must be a unique target for each domain object, i.e., one object each at the base and the head of the arrow. Ontologically, a morphism may be understood as a generalization of a mathematical function, i.e., as an association between its source and target. Morphisms may represent physical, causal, or temporal processes, or purely formal relationships.

Objects and morphisms can be combined into diagrams, the simplest of which, and the most basic tool in category theory, is a triangle, as in Fig. 4.
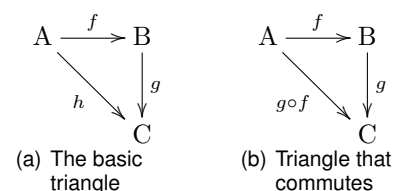

Fig. 4. Category theory triangles. If the triangle in (b) commutes then $h = g \circ f$.

3.2 Commutativity, Identity, and Associativity

Here we introduce the notions required to ensure that a category is well defined. In section 4.1 we see that these notions enforce some very basic conceptual hygiene on visualizations (such as the necessity of determining whether two versions of a visualization are 'the same').

Given objects A, B, and C and morphisms $f : A \longrightarrow B$ and $g : B \longrightarrow C$, the first question a category theorist

2. In category theory it is strictly not possible to show that things are the same or identical. The strongest statement possible is that two sets are naturally isomorphic (unique up to natural isomorphism), that is, indistinguishable.

would ask about Fig. 4(a) is whether there is another morphism, $h : A \longrightarrow C$, that completes the triangle such that the result of applying f then g is the same as would be achieved by h. This morphism is the *composition* of f and g, or $g \circ f$. (Note the ordering: compositions are read right to left as the morphism g is applied to the result of f.) If there is such a morphism h then the triangle is said to commute, the objects and morphisms together form a category, and we can write the equation $h = g \circ f$.

Consider an informal example, the category of familial relations. Suppose A, B, and C are persons, f is the mapping of 'motherhood', and g is 'sisterhood'; then $g \circ f$ corresponds to 'aunthood', the diagram commutes, and we have a category. But now suppose that g is 'friendship'. We do not have a well defined mapping for $g \circ f$ in this case (other than the tautological 'the relationship I have with a friend of my mother'). Hence we cannot form a commuting triangle or create a category. To form a category encompassing both kinship and friendship we must enrich our vocabulary of morphisms to describe those relationships. Societies that are based on communal kin-groups rather than atomic families will tend to develop richer vocabularies to capture these composite relationships, hence their set of morphisms will tend to form well-defined categories. This is an example of how category theory can be used in practice: no one would suggest that kin-groups form vocabularies because of their category-theoretic properties, but the category-theoretic perspective suggests a set of questions that could be asked of that vocabulary, and a conceptual framework for analyzing it.

To form a category it is also necessary that all objects X each have a single identity morphism: $1_X : X \longrightarrow X$ such that for every morphism $f : A \longrightarrow B$ we have $1_B \circ f = f = f \circ 1_A$ (i.e., mapping from A to B with f has the same outcome before and after 1_A and 1_B are applied). We can view this diagrammatically:

$$1_A \circlearrowleft A \xrightarrow{\ f\ } B \circlearrowright 1_B$$

Thus, this notion of 'sameness' is defined with respect to the morphisms in a category, rather than the identity of the object itself. The implications of this definition are explored in section 4.1.

The final requirement for a category to be valid is that morphism composition is associative, i.e., $(h \circ g) \circ f = h \circ (g \circ f)$. In section 4 we apply this definition of a category to the semiotic triad.

3.3 Types of Morphism

Here we introduce types of morphisms, an idea which lies at the heart of this work. Section 5.1 shows how these types correspond to, and have a direct impact upon, basic visualization concepts such as ambiguity, literalness, redundancy, and chart junk.

Morphisms are of several types, the main four which concern us here being monomorphic, epimorphic, endomorphic, and isomorphic. These are defined as follows.

- $f : A \longrightarrow B$ is monomorphic (monic) iff $\forall g_1, g_2 : C \longrightarrow A, g_1 \neq g_2 \implies f \circ g_1 \neq f \circ g_2$. See Fig. 5(a).
- $f : A \longrightarrow B$ is epimorphic (epic) iff $\forall g_1, g_2 : B \longrightarrow C, g_1 \neq g_2 \implies g_1 \circ f \neq g_2 \circ f$. See Fig. 5(b).[3]
- $f : A \longrightarrow A$ is endomorphic (endic) if it maps an object to itself. See Fig. 5(c).
- $f : A \longrightarrow B$ is isomorphic (isic) if there is an inverse mapping $f^{-1} : B \longrightarrow A$ such that $f \circ f^{-1}$ and $f^{-1} \circ f$ are both identity morphisms. That is, $f^{-1} \circ f = 1_A$ and $f \circ f^{-1} = 1_B$. See Fig. 5(d).

$$C \underset{g_2}{\overset{g_1}{\rightrightarrows}} A \xrightarrow{\ f\ } B \qquad\qquad A \xrightarrow{\ f\ } B \underset{g_2}{\overset{g_1}{\rightrightarrows}} C$$
      (a) Monomorphism   (b) Epimorphism

$$A \xrightarrow{\ f\ } A \qquad\qquad A \underset{f^{-1}}{\overset{f}{\rightleftarrows}} B$$
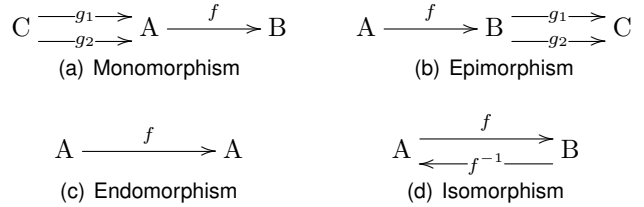   (c) Endomorphism   (d) Isomorphism

Fig. 5. Diagrams illustrating monomorphism, epimorphism, endomorphism, and isomorphism.

Monomorphism, epimorphism, and isomorphism are generalizations of the more familiar set theoretic terms injection, surjection, and bijection, respectively. For example, let us define three set objects A, B, and C representing people and a morphism $f : A \longrightarrow B$ representing motherhood, such that f maps each person in A to their respective mother in B. If g_1 and g_2 map people in one object to their friends in another then:

- f is **not** monic since two different friends of mine can have the same mother; that is, though we might have a pair of mappings such that $g_1 \neq g_2$, the compositions $f \circ g_1$ and $f \circ g_2$ could be equivalent.
- f **is** epic since two different friends of my mother must be different people (they cannot be the same person).
- f is **not** endic since one cannot be one's own mother.
- f is **not** an isomorphism since if f^{-1} were 'child' (the obvious candidate for an inverse map) then $f \circ f^{-1}$ could be my spouse (a person in B who is not my mother) and $f^{-1} \circ f$ could be my sibling (a person in A with whom I share a mother).

These simple examples illustrate how sensitive the properties of monomorphism and epimorphism are to the definitions employed for the underlying types. In section 5 we apply each of these definitions to the visualization category generated in section 4.

4 CATEGORY THEORY APPLIED TO VISUALIZATION

This section applies category theory to the semiotic triad. The end result can be seen in the commutative diagram

---

3. These definitions of monic and epic, through the use of the $\neq$ relational operator, rely on the closed world assumption. This is satisfactory for the classical logic of the category of sets but not for the intuitionistic logic of categories in general.

in Fig. 10 but, as is usual with category theory, we proceed by constructing the diagram in stages, checking for commutativity at each step.

Peircean semiotics is based on a triadic relationship between object, representamen, and interpretant. We can draw our semiotic triad as the commutative diagram in Fig. 6(a) (where O, R, and I stand for object, representamen, and interpretant respectively).

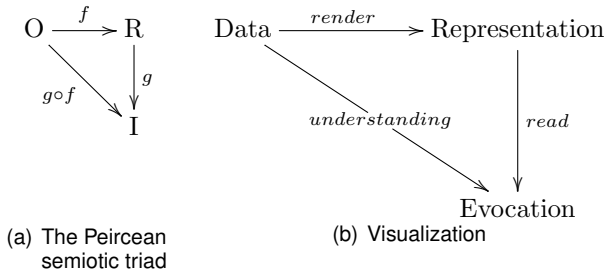(a) The Peircean semiotic triad

(b) Visualization

Fig. 6. The Peircean semiotic triad shown as commuting triangles

In visualization the object is the data collected from a given system, the representamen is the representation, and the interpretant is the mental state evoked by the representation in the mind of the interpreter. Thus, we name the objects in our commutative diagram Data, Representation, and Evocation respectively. The morphisms between them we define as follows:

- The transformation from Data to Representation is named *render* because the data are rendered in a given way so as to represent the Data while maintaining structure and content.
- The morphism between Representation and Evocation is called *read* because the interpreter reads the Representation.
- To the composition $read \circ render$ we assign the name $understanding$ as a proper Representation leads to the reader understanding some aspect of the Data.

Thus, Fig. 6(b) shows the Peircean semiotic triad presented as a commutative diagram which forms the core of the visualization process.

In visualization the object can be further decomposed into a set of data and a system which that data measures. Thus, the starting point for the visualization process is not data but the system from which the data were gathered. Furthermore, the interpretation of the representation leads not only to understanding the data but also to beliefs and inferences about the system, the truth of which can be tested. Thus, if we combine Figs 3 and 6(b) we get Fig. 7. A prototypical visualization process consists, then, of the following entities and processes, which we will describe using the category theoretic terms of objects and morphisms:

- **System**: a real world system, object, or phenomenon, such as a class of students.
- **Data**: a set of data that describes some aspect of that System, produced by a *measure*, such as test scores for those students.

- **Representation**: some visual, aural, haptic, or literal artefact of that Data, produced by a process of *render*ing, such as a bar chart of their performance.
- **Evocation**: what the Representation evokes in the mind of the user through the user's reading of it, such as the teacher's viewing of the bar chart.
- And that Evocation is thus an understanding of the original data ($understanding = read \circ render$).

Fig. 7 shows this expansion, though it should be noted that this diagram is incomplete and is to be read only as a stepping stone on the way to Fig. 10.
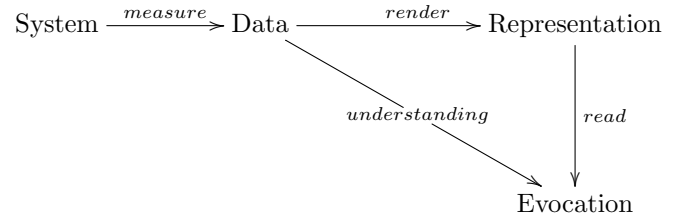

Fig. 7. The objects System, Data, Representation, Evocation and the morphisms between them.

4.1 The Visualization Process is a Category

For these objects and morphisms to form a category certain conditions must apply.

1) **Object Identity**: each of {System, Data, Representation, Evocation} must have an identity operation defined, and the kernel of this identity function will, in turn, identify an equivalence class of objects that are considered identical under this mapping. In particular this requires that we are able to decide unequivocally if two instances of the same System, Data, Representation, and Evocation are identical. This simple requirement forces a great deal of conceptual hygiene. Unless we can answer the following questions we are vulnerable to the accusation that the visualization process is not well-defined.

   a) 1_{System}: The problem of identifying systems is a common and urgent one in most empirical science; we cannot talk about replicating results unless we are doing the same things to the same systems. Moreover, unlike a data set which captures a snapshot of a system and is static, a System may experience change over time. For example, in a System of a class of students the individual students age and mature, change their clothes, and some may even drop out of the course. How may we know when two System objects have the same denotation? Agreed criteria are needed for establishing System identity which will allow a System to experience change over time while still being considered to be the same System.

   b) 1_{Data}: When do we say that two sets of data, such as test scores, are the same? Do the absolute scores matter or is it the same set of scores

when expressed as a percentage? What degree of precision is required? If we had a very large set of scores (such as when studying the changes in national exam performance) then identity might be defined in terms of there being no statistically significant difference between samples or aggregate distributions.

     c) $1_{\text{Representation}}$: When are two representations the same? Is a printed version the same representation as an on-screen version, or an auditory display equivalent to a visual one?

     d) $1_{\text{Evocation}}$: What does it mean to say that two users form the same mental picture of the data? The obvious problems in determining internal psychological states mean that this issue is usually operationalized in terms of the ability to answer questions about the data ('which student did best?', 'has average performance gone up or down?'), where the same answer implies the same understanding.

2) **Morphisms must be Maps**: recall there can be only one object at each end of an arrow. In the case of *render* this requires that a single set of Data (to within 1_{Data}) generates a single Representation. This is not obviously true. For example, visualization tools that allow interactive data exploration seem capable of generating many representations of a single data set. But in this case it is the tool, bound to that Data, that is considered to be the Representation rather than any particular state or view that it produces.

3) **Commutativity of Morphisms**: As section 3.2 showed, commutativity is required. For example, in Fig. 7 the result of *read*ing the Representation produced by *render*ing the Data is an *understanding* of the Data. If the Data is *render*ed and then *read*, but as a result the reader does not understand the data then the process of visualization has failed.

4) **Morphism Associativity**: As per section 3.2, the composition of morphisms must be associative.

Visualization processes for which conditions 1–4 are satisfied can be considered as valid categories satisfying the axioms of category theory.

4.2 The Intension of the Visualization Process

Fig. 7 shows the visualization process as a category, but this represents a single concrete instance of this process. It could represent, for example, the production and consumption of a single particular scatter plot from a single particular set of data. We now need to generalize this notion to describe, for example, the properties of scatter plots in general. In this section we generalize each of the objects of the visualization category.

4.2.1 Schema is the Generalization of Data

In the case of the Data we have the familiar notion of a Schema, which refers to the structure rather than any particular values of the data. For example, a relational database of tables and attributes is defined using a database schema, which is then filled with values during its lifetime of use. The Schema is the intension (the constant conditions that capture the set of all possible values for that data set) while the Data is the extension (the variable set of actual values).[4] Not all Data has a corresponding Schema (generally known as unstructured data). Examples include natural language text which can be represented, for example, by pictorial illustrations. However there can be no rules governing the layout of this Representation: there may be a set of Representations of a similar style (for example, several illustrations in a single book) but each is individually inspired by the text it is designed to Represent. Unless the Data can be generalized into a Schema, there can be no corresponding generalization of the Representation into a Layout. This criterion also helps to distinguish between works that are data art or data-driven art (see section 5.3).

4.2.2 Layout is the Generalization of Representation

We can also generalize the notion of a particular Representation to that of a Layout, that is, the way in which Data belonging to a Schema are represented. For example, the two scatter plots in Fig. 12(a) are different Representations, but share a Layout. Layout is a familiar notion from data visualization tools (such as the charting feature in spreadsheet programs) that allow the user to choose which of many options are used to graphically represent selected data. The tool uses a set of *rules* to map from a data Schema to a representational Layout.

4.2.3 Questions are the Generalization of Evocation

Generalization is the process of splitting the extension into an unsaturated (incomplete) and a saturated (complete) part (in Frege's sense of the terms [11]). It is the former that constitutes the intension. Data, for example, may be associated with a Schema, such as a table, and the values that can fill the empty spaces. Representations can be split into a Layout, such as a set of axes, and the marks that are placed in the space defined by the axes. In the case of the contents of the mental states evoked by *reading* a Representation, the equivalent is to split the proposition describing that mental state into a property for which there may be some object of which it can be truthfully predicated. That is, if *reading* a Representation, such as a bar chart of exam results, evokes the thought that 'Alan got the best mark' (or $best\_mark(Alan)$), then the generalization of this is the question 'Who got the best mark?' (or $best\_mark(\_)$). That is, Questions are system predicates at the intensional level while Evocation involves the extension of those predicates for specific

---

4. In philosophy a distinction is drawn between a term's intrinsic meaning or its *intension* and its denotation, or *extension*. Frege gave the example of the *morning star* and *evening star*. The two terms have different meanings (intensions) but both have the same denotation (extension), the planet Venus [24].

cases. Evaluation of these extensional predicates allows truth statements about the System to be tested.

Completing the *generalization* mapping, from extension to intension, provides a salutary reminder of the importance of the underlying Question in visualization; "From Killer Questions Come Powerful Visualizations" as Johnstone puts it [15]. In software development terms, answering a question is the *requirement* of the Representation. This is true even in the case of *exploratory* data analysis (the terminology is due to Tukey [27]), where the purpose is not to answer a specific prior Question or hypothesis, but to discover hypotheses worth subsequent testing using conventional *confirmatory* data analysis. Exploratory data analysis is what happens when you don't know what question you're trying to answer. Confirmatory data analysis starts with Data and a Question and then seeks an answer using a Representation. Exploratory data analysis starts with Data and a Representation, and then seeks a Question worth asking.

Fig. 8 shows the Data, Representation, and Evocation objects from our visualization category at the extensional level and Schema, Layout, and Questions as their respective generalizations, or intensions.
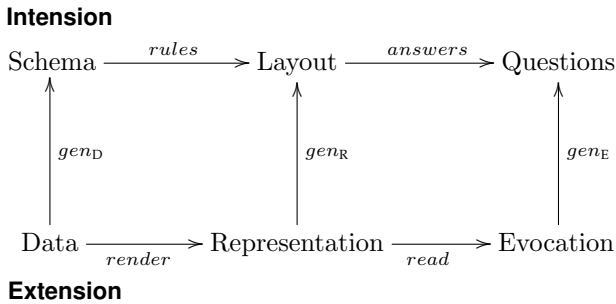

Fig. 8. Introducing the intensional level of the category generalizing the extension with gen_D : Data $\longrightarrow$ Schema, gen_R : Representation $\longrightarrow$ Layout, gen_E : Evocation $\longrightarrow$ Questions

4.3 Completion of the Visualization Category

Combining our partial diagrams (Figs. 7 and 8) we get the diagram in Fig. 9. From a category-theoretic point of view, that is, by considering its formal structural properties, this diagram is now complete as it is closed in the mathematical sense. Seeking closure requires us to find a terminal object, an object at which all paths of possible morphism compositions terminate. Although Fig. 9 does have such an object (Questions), we would not say that this is the terminal object of the visualization process. Therefore, we need to identify a new terminal object that reflects the true end of this process.

4.3.1 *Knowledge is the Terminal Object of Visualization*

Data is not the start of the visualization process, and nor is the ability to answer Questions about that Data the end. What we are seeking is Knowledge of the System. Visualization starts with a System that we measure in various ways to generate Data. That Data will always
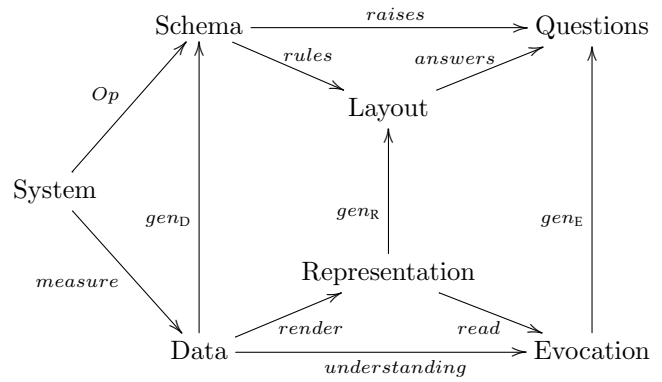

Fig. 9. Intermediate mathematically closed visualization category with terminal object Questions.

be partial (in both senses of the word) but it is all we have. Similarly, the only Knowledge we can gain is that which can be deduced from the evidence presented in the Representation, and the Questions define what Knowledge we can gain from the visualization process. That is, Knowledge operationalizes the Questions by allowing the abstract nature of the Questions to be practically measured or assessed. It is by answering questions that one gains knowledge. This relationship is also analytic: Knowledge is not well-defined unless it is capable of answering questions, and those questions are epistemologically prior to the knowledge of the answers. (One can imagine a question to which there is no answer, but not an answer for which there is no question.)

The ultimate goal of the visualization process is to gain Knowledge of the original System. When this succeeds (when the diagram commutes) the result is a *truth* relationship between the Knowledge and the System. When this process breaks down and we fail to deduce correct conclusions then the diagram does not commute.[5]
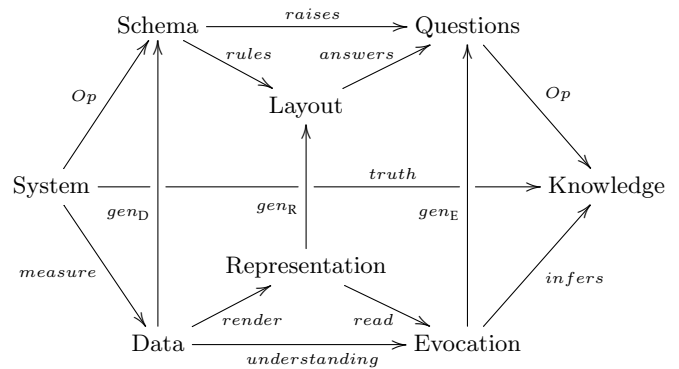
4.3.2 *The Completed Category*

Fig. 10. The visualization process as a category with terminal object Knowledge and initial object System.

5. Note, that although this is a strongly realist and representational use of terms such as truth and knowledge it does not necessarily imply a commitment to objectivism about the status of that knowledge. See, for example, Faith [10].

The full diagram describing the general visualization process is given as Fig. 10 and a specific example is shown in Table 1. In this example we start with students in a mathematics class (System). We want to know how well they are performing, overall and individually (Knowledge), so we determine that we must gather data about their performance in a test (Schema). This Schema, as well as being a generalization of the Data is also a time-invariant *descriptive* abstraction of the System.[6] Having gathered the results (Data) we choose a bar chart (Layout) that will be capable of showing the distribution of overall and individual marks (Questions). The Schema acts as a frame and determines, or *raises*, what Questions can be answered by the Layout. For example, from a table with two columns labelled 'Student' and 'Grade' we can tell that the Data denoted by this Schema will let us answer questions like "What was the average mark?", but not questions like "Who was the tallest?".

The Layout, in turn, is used to *answer* the Questions, but only some Schema-derived Layouts are capable of doing this. For example, a bar chart will enable us to spot the best student but a pie chart would not.

Showing the data (Representation) in this way evokes an understanding of the data that the cohort achieved an average mark of more than 70, that Alan got the highest mark, etc. So now we have Knowledge about our System (the Maths class): we know who performed best of all, that nobody failed, that the average mark was satisfactory (this requires prior context), etc.

5   RESULTS: APPLYING THE CATEGORY

With this category corresponding to the visualization process we can use the concepts of category theory to consider its properties, and show how common intuitions about visualization can be defined more formally.

5.1   Properties of the *render* Morphism

To illustrate how this may be done, we will consider what it means in visualization terms, for the *render* morphism to be monomorphic, epimorphic, isomorphic, and endomorphic. In each case we take a standard category theoretic definition and apply it to the visualization process category and we find that this yields a property or issue that is important for visualization.
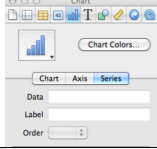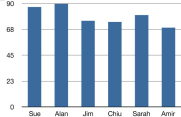
5.1.1   Monomorphism Corresponds to Sensitivity

A *render* morphism is monic iff $\forall measure_1, measure_2$: System $\longrightarrow$ Data : $render \circ measure_1 \neq render \circ measure_2 \implies measure_1 \neq measure_2$.

That is, if the System is measured in two different ways the resulting Representations will necessarily be different. For example, suppose we assess student performance with two different tests the results of which are captured by $measure_1$ and $measure_2$ and represent them

6. Formally, a Schema has "the structure of a continuant which does not specify time or timelike relationships" [24, p. 73].

TABLE 1
Example of how the visualization category applies to objects in practice

Category Object	Example
System	A cohort of students on a course
Schema	
Data	
Layout	
Representation	
Questions	$best\_mark(\_)$ $average\_mark(\_)$
Evocation	$best\_mark(Alan)$ $average\_mark(> 70)$
Knowledge	"Alan performed the best", "Nobody failed", "The average mark was satisfactory"

in two ways: a simple textual description (e.g., "John was the best student"), and a bar chart. The differences in the test would not make any difference to the textual description but they would to the bar chart (assuming that identity morphisms on the Representations and Data are well-defined). We would normally describe this in terms of the sensitivity of the visualization. Sensitivity is usually assumed to be a desirable property of a representation (and this assumption may often be valid) but the point here is to show that this important property may be defined formally and precisely.

5.1.2   Epimorphism Corresponds to Non-Redundancy

A *render* morphism is epimorphic iff $\forall read_1, read_2$: Representation $\longrightarrow$ Evocation : $read_1 \circ render \neq read_2 \circ render \implies read_1 \neq read_2$.

That is, if two people *read* the same Representation in different ways they will reach different *understandings* of the Data. Although this may seem tautological, there are important cases when it is not true. Consider a set of data in which three attributes are measured for each sample (e.g., student performance on three different tests). This data could be represented using a conventional scatter plot in which the x-coordinate corresponds to test 1, y to test 2, and both the size and shade of each point correspond to test 3 (see Fig. 11). One individual may notice the position and size of each point, whereas another may notice the shade. They would draw identical conclusions about the data, but they have read the representation in different ways. The representation in

this case is redundant as test 3 is represented in two different ways; hence there is more than one way of gaining an understanding of that information from a single representation.
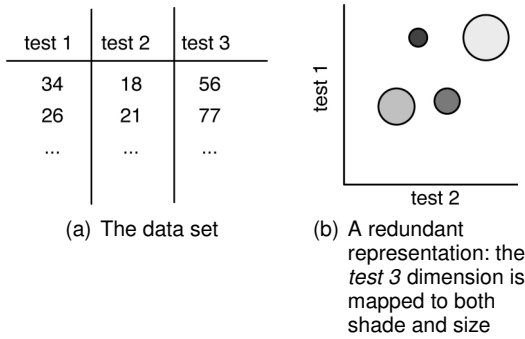

test 1	test 2	test 3
34	18	56
26	21	77
...	...	...

(a) The data set

(b) A redundant representation: the *test 3* dimension is mapped to both shade and size

Fig. 11. Redundancy in representations

5.1.3 Endomorphism Corresponds to Literalness

A *render* morphism is endomorphic (a mapping of an object back onto itself) iff $Data = Representation$. That is, some Representations consist in simply presenting the Data (for example, in a spreadsheet or printed table). These, too, are valid visualizations, and the concept they demonstrate is *literalness*.[7]

Contrast this with Cox's notion that visualizations are necessarily metaphorical [8], that there is a direct relationship between visualization and the mapping process (cognitive and creative) in metaphor theory. She says (p. 89):

> Linguistic and visual metaphors are defined as mappings from one domain of information (the source) into another domain (the target). Likewise, data-viz maps numbers into pictures, resulting in visaphors, digital visual metaphors.

However, we can see that endomorphic *render*ings are valid visualizations that are not metaphorical.

5.1.4 Isomorphism Corresponds to Non-Ambiguity

A *render* morphism is isomorphic iff there is an inverse morphism (which we will call *decode*) such that $decode \circ render = 1_{\text{Data}}$ and $render \circ decode = 1_{\text{Representation}}$. Thus, the *decode* morphism lets us recover the original Data from a Representation *render*ed from it (to within the degree of accuracy determined by 1_{Data}). Non-isomorphic Renderings are thus ambiguous in the sense that two different data sets may generate the same Representation. Sometimes this is a valuable property, for example in representations that aggregate or filter large or complex data sets into simpler forms. In other situations it is less desirable. Further, as isomorphism implies both monomorphism and epimorphism, a rendering that is isomorphic, in addition to being non-ambiguous, will also exhibit sensitivity and an absence of redundancy.

7. N.b., we are bracketing the discussion over the metaphysical status of logical data compared with that of a physical representation.

These properties are summarized in Table 2 which shows what combinations of morphism properties determine what characteristics of a particular *render*ing (e.g., only endic *render*ings are literal, monic ones exhibit sensitivity, non-isomorphism leads to ambiguity, etc).

TABLE 2
Summary of morphism properties as applied to the *render* morphism.

	Non-endomorphic			Endomorphic		
	Monic	**Epic**	**Isic**	**Monic**	**Epic**	**Isic**
Sensitive	✓		✓	✓		✓
Non-redundant		✓	✓		✓	✓
Non-ambiguous			✓			✓
Literal				✓	✓	✓

5.2 Non-Epimorphic Layout Implies Chart Junk

In this section we show how one common contentious issue in information visualization (chart junk) can be characterized using category theory. In the scatter plots

(a) Plain charts

(b) Arbitrary chart junk
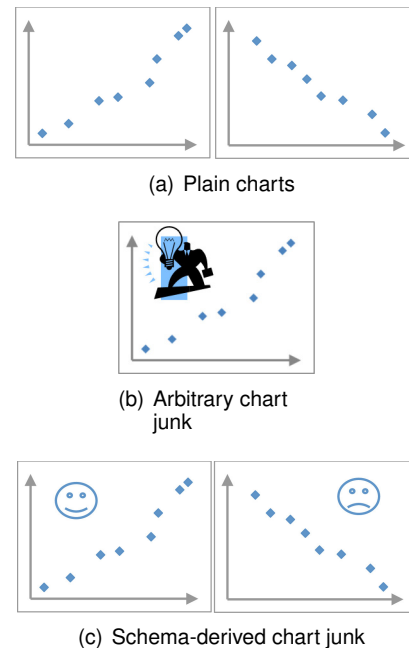
(c) Schema-derived chart junk

Fig. 12. Chart Junk

in Fig. 12(a) every element of the Layout was derived from some element of the Schema. If we write these objects as sets, then we can say that the morphism between them is a surjection:

$$\begin{aligned} \text{Schema} &= \{Instances, Attributes\} \\ \text{Layout} &= \{Points, Axes\} \\ rules(Instances) &= Points \\ rules(Attributes) &= Axes \end{aligned}$$

However, this is not always the case. For example, suppose we decorate one of our scatter plots with a figure as in Fig. 12(b). What is the Layout in this case? If the

Layout is a *generalization* of this particular Representation then it will include the decoration. But this decoration is not derived epimorphically using a *rule* from any part of the Schema — it is an arbitrary addition (perhaps inspired from some property of the System not captured in the Data). It is 'chart junk'.

The following aspects of this definition of chart junk should be noted.

1) This use of the term is not intended to be pejorative: chart junk can be useful in aiding understanding of the system [3]. The purpose of this category-theoretic definition is to highlight the difference between decorative elements in Layouts that communicate Data (those that are derived from a Schema) and those that do not.

2) It is a much narrower use of the term than Tufte's original definition [26] which defined chart junk as all unnecessary, redundant, or non-data ink. Consider the example in Fig. 12(c) in which scatter plots are decorated with faces indicating the data trend. The decoration is redundant chart junk in Tufte's sense, but not in ours since it communicates something about the data. There is a rule for deriving this element of the Layout from the Schema which thus ensures that Fig. 10 commutes.

3) There is a difference between redundancy at the extensional level of the Data and Representation introduced in Fig. 8 and arbitrary chart junk at the intensional level of the Schema and Layout. A Representation may include some redundancy even though there is no chart junk. In the example of the scatter plot in which a single Data attribute is represented using two retinal attributes such as in Fig. 11(b) — or, indeed, the example in Fig. 12(c) of a scatter plot decorated with a face representing the polarity of the correlation — we have redundancy in the Representation, but the Layout *rules* are surjective (epic): every element of the Layout is the product of some aspect of the Schema. Redundancy at the level of Data and Representation may not be arbitrary at the level of Schema and Layout. Conversely, chart junk is not (necessarily) redundant, it is arbitrary.

5.3 Intensionality and Infographics

For an example of the power of intensional generalization, compare the two infographics in Figs. 13 and 14. In Fig. 14(b) both sides of the diagram follow the same pattern; this pattern is a schema. The existence of this intensional generalization means the representation can support two operations that Fig. 13 cannot. The first operation is that it can generate questions that it can then answer (for example: "I see that 54% of left-leaning people support gay rights; so what proportion of right-leaning individuals do?" — Fig. 14(a)). The second operation that Fig. 14(b) can support is that the layout can be generalized to more examples (e.g., how would
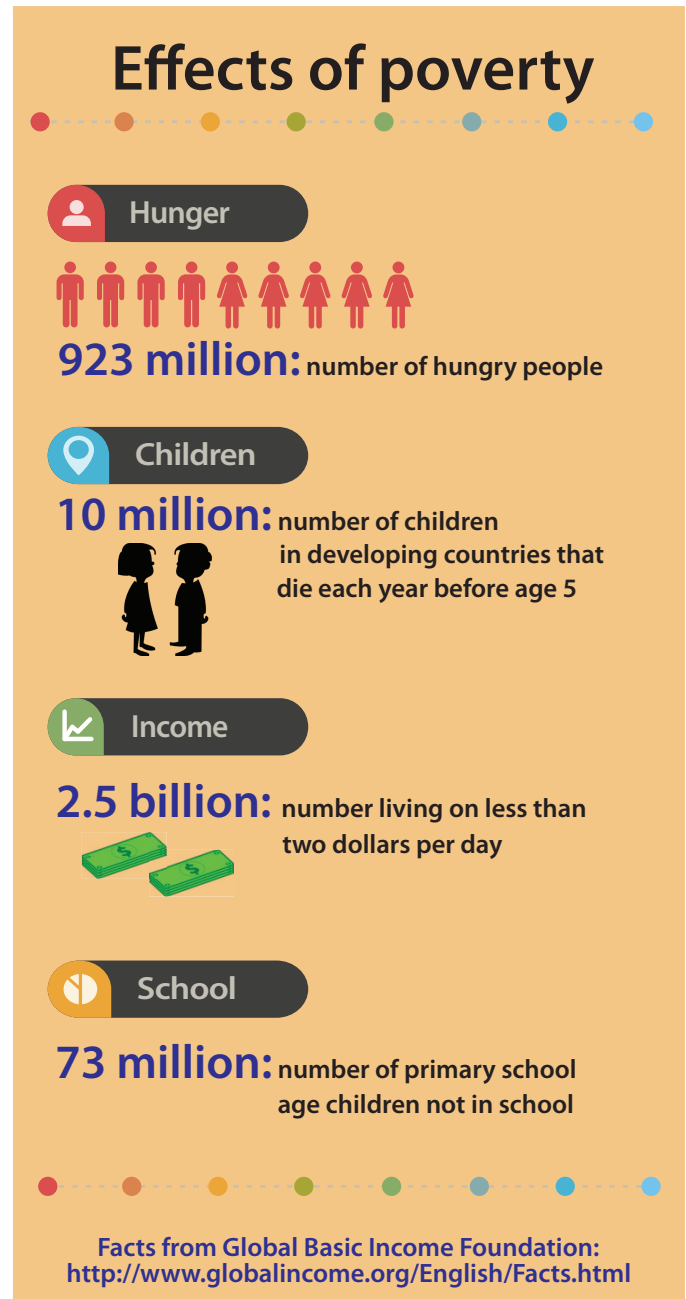

Fig. 13. Infographic showing facts about world poverty. Facts taken from the Global Basic Income Foundation at http://www.globalincome.org/English/Facts.html

this diagram apply to European socially progressive and economically libertarian political parties?). Having no intensional level, the example in Fig. 13 can support neither of these operations. It communicates information that evokes understanding, but does not empower the user to interrogate or reuse the information it presents.

6 RELATED THEORY

6.1 Algebraic Semiotics

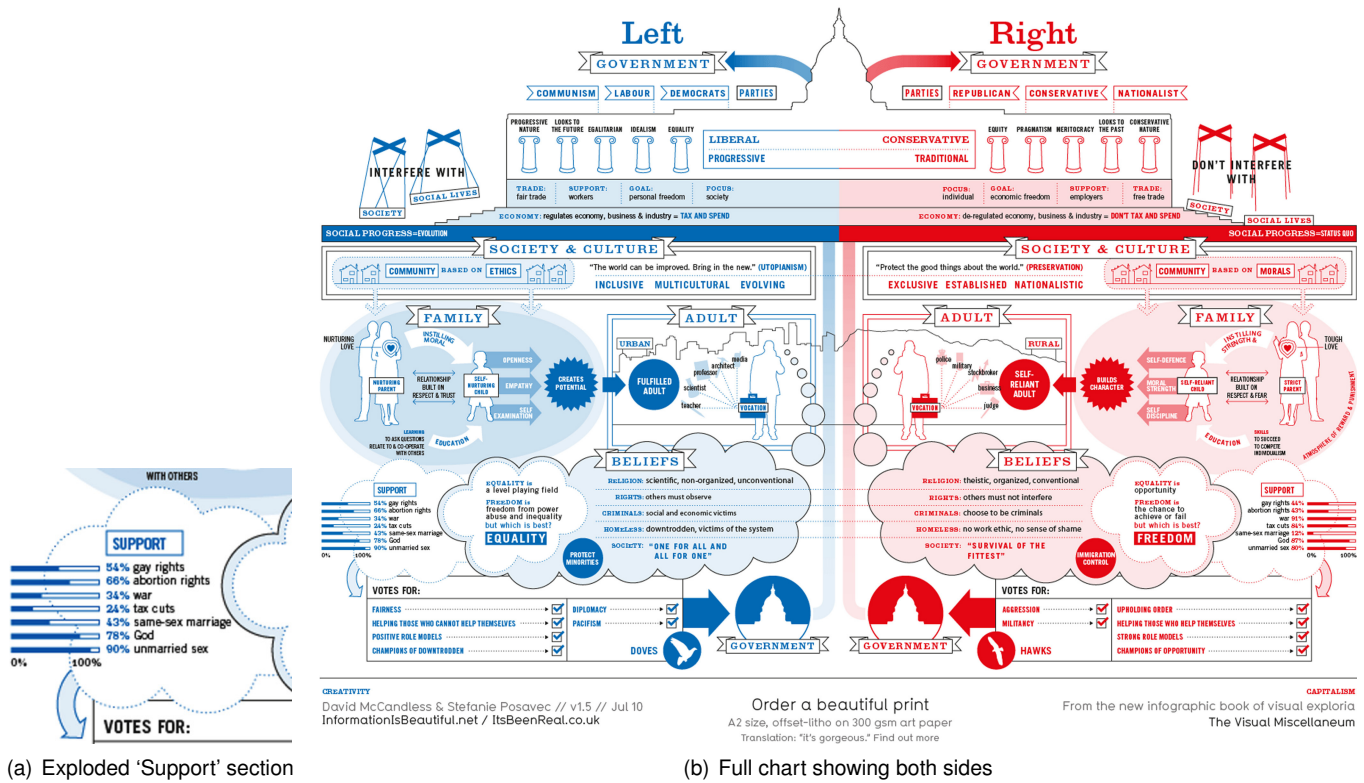Both Peirce and Saussure understand signs in terms of relationships and mappings between signs and sign

(a) Exploded 'Support' section

(b) Full chart showing both sides

Fig. 14. Infographic illustrating political differences. David McCandless 2009 — http://www.informationisbeautiful.net/visualizations/left-vs-right-us/

systems. Thus, they seem natural candidates for the category theory treatment. The first effort to apply category theory to semiotics was that of Goguen and Harrel [12] who attempted a formalist treatment of the semiotics of information visualization and user interface design, which they described as 'algebraic semiotics'.

The objects in Goguen's algebraic semiotics are sign systems. The actual definition is complex, involving theoretical apparatus taken from mathematical algebra but, the key elements are signs, constructors, and axioms. The signs form the vocabulary, or set of all possible signs. The constructors provide a systematic way of generating those signs. And the axioms constrain those signs. Consider Goguen's example of a simple time of day system which shows the number of minutes since midnight. The signs are the set of natural numbers generated using two constructors: the constant 0 (representing midnight) and a successor operation, s, where for a time t, $s(t)$ is the next minute. A single axiom, $s(1439) = 0$, constrains the set of generated signs to a 24 hour day. Goguen thus defines signs in terms of generalized structures from which the validity of any particular sign is derived. In terms of of first-order logic, the sign system is a theory of which any particular sign is a model.

Goguen then considers the mappings between sign systems, which he calls semiotic morphisms. Consider a slow, regular sand glass containing 1440 grains of sand in which one grain of sand falls every minute, and which is turned when the last grain falls. We can define a mapping from the time of day system by respectively mapping its elements (constant 0, the s constructor, the set of numbers, the $s(1439) = 0$ axiom) to our new elements (empty lower glass, the falling of a grain of sand, the possible piles of sand, turning the glass).

Goguen's is a strongly structuralist theory in two senses. First, signs are defined as such in virtue of their membership of, and role within, a sign system. It is the structure of the sign system that defines its constituents as signs. Second, the only relationships considered — the semiotic morphisms — are between sign systems rather than between sign systems and either external or mental states. It all happens within the 'third order' [9]. In particular, there is no distinction in this framework between a set of data and a visualization of that data. Data is just another sign system.

We find this structuralism problematic. For example, how can this framework be used to discuss the quality of a visualization? Goguen attempts an answer to this 'narrowly' by characterizing how well his semiotic morphisms preserve the structure of a sign system. We argue that the solution, as presented in this article, is a post-structuralist scheme that expands the use of category theory to explicitly incorporate other elements of the visualization process, including the visualization's context, and how the visualization is used in practice.

6.2 Non-Category Theoretic Approaches

Our framework is method-blind in that it does not specify *how* the various objects (Schema, Data, Questions, etc.) and morphisms (*render*, *raises*, etc.) should be defined. Other formal approaches to visualization have sought more to explore the *hows* by understanding the biology, as it were, of different visualization techniques. They focus variously on classification (taxonomy), on describing the differences and commonalities between visualization techniques, on codifying graphic design and layout rules, and on understanding the different representational processes employed in visualization of all kinds.

Tory and Möller [25] show how theoretical models and taxonomies focus on understanding the design space and design choices of the visualization process. Their approach seeks to understand and clarify the differences between different types of visualization to show how "traditional divisions (e.g., information and scientific visualization) relate and overlap" [25]. The focus is on the process level letting us see both the relationships between different task types and what types of task each design model enables users to perform.

Continuing in the taxonomic vein Chi [7] aimed to understand how visualization techniques work and how they fit into different design spaces. Understanding the different operating steps and how they are re-used in different visualization techniques enables, it is claimed, more rapid implementation of visualizations.

We see from Fig. 10 that the taxonomic techniques above deal with, and could thus be used in the specification of the Schema and Layout objects and the *rules*, and *render*ing morphisms. Consider Tory and Möller's taxonomy which suggests suitable schemata for different data types. Two-dimensional data with one dependent and one independent variable, for instance, can be represented using scatter plots and bar charts whilst data with any number of dependent and independent variables can be mapped to charts with colour, multiple views, glyphs, and parallel co-ordinate plots. Chi's taxonomy gives a very detailed analysis of 36 visualization techniques (Representation + Layout) showing what kinds of data they are well-suited to. Thus, taxonomy-based treatments help the visualization designer with specifying the intensional level of the category for a given System and Data.

Mackinlay [19] analyzed information graphics and combined first-order logic with a 'composition algebra' to codify graphic design criteria to allow construction of a tool that automatically designs graphical presentations of relational information. The composition algebra's rules enable multiple data sets with common axis dimensions to be displayed on a single combined plot. It is very much concerned with layout rules and representational techniques. Thus, we can see that this technique might help in determining the *rules* morphism used in the specification of Layout. Mackinlay offers 'expressiveness

criteria' that indicate the encoding technique that should work for each data mapping. For example, a Layout may use size as a display attribute effectively for ordinal or quantitative data while colour, orientation, and shape will all work well for nominal data. Card and Macklinay [4] took this foundation further and organized the visualization design space as a framework that can be used to design new or augment existing visualizations. Principally, their work is concerned with the intensional decisions of how best to map a given Schema to a Layout given the information requirements (Questions) of the task.

Perhaps closest in scope to our work, Ziemkiewicz and Kosara [30] used set theory to explore the mapping between data and representation. They asserted that for a visualization to be readable the rendering must be both injective and surjective, that is, bijective (isomorphic in our category). Non-injective mappings, they say, can lead to representations in which more than one data item is mapped to the same visual element leading to ambiguity (see section 5.1.4) while non-surjective mappings lead to representations containing marks not derived from the data (which would include chart junk, see section 5.2). Their analysis seems to preclude chart junk as a valid element in a visualization, while our scheme makes no assessment of the desirability of chart junk. Ziemkiewicz and Kosara add a further stipulation that all information visualizations must involve non-trivial interactivity and be syntactically notational. Using these three criteria they then classify and distinguish between different types of visualization (e.g., scientific visualization, information graphics, etc.). Within information visualization they identify three classification axes: linear vs non-linear mappings, how information loss is treated in the rendering, and the semantic notationality of the system. This theoretical framework enables taxonomic analysis within information visualization and between different non-information visualization techniques. With the focus being on the properties of the representational mapping their work sits, again, in the intensional part of our category centered on the *rules* morphism (the intension of *render*).

Ziemkiewicz and Kosara's work is very helpful in understanding different visualization processes in a taxonomic sense (though they tend to exclude narrative visualizations for their lack of interactivity). The set theoretic approach allows sensitivity, redundancy, and ambiguity to be defined in a similar way to our approach but category theory enables us to go further and differentiate between:

1) Types of chart junk (arbitrary and schema junk);
2) The individuation of attributes and the individuation of instances.

van Wijk's [28] model of visualization treats knowledge acquisition as the accumulation of knowledge elements over quantized intervals of time. The knowledge gained depends (a) on the existing knowledge of the

user (a basic tenet of information theory) and (b) the perceptual and cognitive abilities of the user which will be affected by their prior experience (an acceptance of the post-structuralist view of context). The model also permits changing of the visualization specification (the layout *rules* in our category) to facilitate further exploration of the data. Pinker [21] can also be brought to bear here as his study of how people read graphs can be used to inform the design of layouts and data-to-representation mappings. A particular strength of van Wijk's work [28] is that it provides insight into why some types of visualization are successful and others not; it gives us a tool for understanding why a categorical description does not commute, that is, it helps us to understand how the *read* transformation takes place to form Evocations in the mind of the user.

Hullman and Diakopoulos [14] used the tools of rhetoric to offer techniques for understanding the selection of data (Schema and *measure*ment design), the design of representations (Layout), and the role of annotations and interactivity. They also deal explicitly with denotative and connotative meaning in understanding how visualizations are viewed. This aspect of their work finds its place on the intensional and extensional sides of our category.

6.3 Summary

The theoretical development in section 4 gave rise to the visualization process category in Fig. 10 and the summary of how different types of mapping affect a visualization through the *render* morphism in Table 2. Fig. 10 provides an overview of a well-formed visualization process. As far as a visualization designer is concerned, this category diagram will not tell them *how* to create a visualization, rather it tells them all the major components that must be considered for correct knowledge creation and sense making to take place. Visualization design works principally at the intensional level while user interaction with an instance of the visualization process takes place at the extensional level. Fig. 10 shows the objects and morphisms but does not specify their types. Table 2 summarizes the main morphism types available and, as we showed in section 5.1, these types materially affect the properties of the visualization. For example, when specifying a *render* morphism to transform Data into a Representation, we saw that if the Representation was required to contain no ambiguity then it must be isomorphic (bijective for sets). Fig. 10 also tells us that a complete visualization process needs to be considered from both the extensional (real world of instances) and intensional (definitional, schematic) levels.

The category theory approach in this article takes a higher level view of visualization than the examples of previous work described above. These earlier theoretical and formal treatments, which focus either on taxonomic descriptions or on process to show how certain effects can be achieved using particular visualization techniques, can be seen to contribute to the development of one or more objects or morphisms within the visualization category in Fig. 10.

7 CONCLUSIONS

This article is concerned with the foundational concepts necessary for developing from first principles a formal description of visualization. We have:

1) Shown how the process of information visualization fits within the semiotic theory of Peirce and Saussure;

2) Shown how the semiotic frameworks can be formally characterized using the language of category theory;

3) Extended the Peircean semiotic triad into a mathematically closed category (see Fig. 10) that incorporates the context of the visualization, including the relationship between the triad and the original system under study, the relationship between the data, representation, schema, and layout, and the questions and human insight and knowledge that result from the visualization process;

4) Used this formal framework to

a) formally define visualization properties (literalness, redundancy, sensitivity, generalizability, chart junk) that were previously defined intuitively, if at all, and to

b) expand on definitions provided in earlier work (see Table 2) including differentiating between redundancy and arbitrary chart junk. These insights arise out of, and were discovered using, the category theoretic treatment of the visualization process;

5) Used the category to derive the property of 'intensionality' as applied to visualizations, and provided examples of how that property empowers users.

Prior to this work many of the visualization concepts discussed above were understood intuitively or heuristically, but now we have verifiable formal theoretical descriptions that can be used to make principled judgements about the visualization process in general and about individual specific instances in particular. The category in Fig. 10 is a mathematically formal diagram that defines a well-formed visualization process; if all the morphisms and objects are well defined, the diagram will commute, and we can be confident that the knowledge gained is reliable and reflective of the system. The diagram can be used for reasoning and understanding. We can also envisage a tool that encapsulates the category's formalisms which, through some form filling or other exercise allows the user to specify details of the system, schema, and proposed layout, together with candidate questions. The tool could then validate the design and also show the consequences of specific design decisions. For example, a proposed layout would

result in redundancy while another would exhibit non-ambiguity, sensitivity, etc.

We have identified the principal objects implicated in visualization, the 14 morphisms (relationships) that map between the structures in these objects, and four properties that each of those relationships may possess (monomorphism, epimorphism, isomorphism, and endomorphism). We have considered the implications to visualization by applying these properties to just one of those relationships (the *render* morphism).

In future work we will consider the implications of these properties for the remaining morphisms and apply higher-order category theory tools to compare different visualizations in terms of their ability to generate the same knowledge. This would be particularly appropriate when considering visualization across modalities allowing us to compare, say, a graphical representation with an auditory display.

REFERENCES

[1]   http://www.processing.org.
[2]   S. Barrass and P. Vickers, "Sonification design and aesthetics," in *The Sonification Handbook*, T. Hermann, A. D. Hunt, and J. Neuhoff, Eds.   Berlin: Logos Verlag, 2011, pp. 145–172.
[3]   S. Bateman, R. L. Mandryk, C. Gutwin, A. Genest, D. McDine, and C. Brooks, "Useful junk?: The effects of visual embellishment on comprehension and memorability of charts," in *Proceedings of the 28th International Conference on Human Factors in Computing Systems*, ser. CHI '10.   New York, NY, USA: ACM, 2010, pp. 2573–2582. [Online]. Available: http://doi.acm.org/10.1145/1753326.1753716
[4]   S. Card and J. Mackinlay, "The structure of the information visualization design space," in *Information Visualization, 1997. Proceedings., IEEE Symposium on*, Oct. 1997, pp. 92–99.
[5]   S. K. Card, J. Mackinlay, and B. Shneiderman, "Information visualization," in *Readings in Information Visualization: Using Vision to Think*, K. Card, Stuart, J. Mackinlay, and B. Shneiderman, Eds.   Morgan Kaufman, 1999, pp. 1–34.
[6]   D. Chandler, *Semiotics: The Basics*, 2nd ed.   Routledge, 2007.
[7]   E. H. Chi, "A taxonomy of visualization techniques using the data state reference model," in *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*, 2000, pp. 69–75.
[8]   D. Cox, "Metaphoric mappings: The art of visualization," in *Aesthetic Computing*, ser. LEONARDO, P. A. Fishwick, Ed.   Cambridge, MA: MIT Press, 2006, pp. 89–114.
[9]   G. Deleuze, "How do we recognise structuralism?" in *Desert Islands and Other Texts (1953-1974)*, ser. Foreign Agents, M. Taormina, Ed.   Semiotext(e), 2002, translated by David Lapoujade.
[10] J. Faith, "Emergent representations: Dialectical materialism and the philosophy of mind," D.Phil thesis, University of Sussex, 2000.
[11] G. Frege, "Function and concept," in *Translations from the Philosophical Writings of Gottlob Frege*, P. Geach and M. Black, Eds.   Blackwell Publishers Ltd, 1960.
[12] J. A. Goguen and D. F. Harrell, "Information visualization and semiotic morphisms," in *Multidisciplinary Approaches to Visual Representations and Interpretations*, ser. Studies in Multidisciplinarity, G. Malcolm, Ed.   Elsevier, 2005, pp. 83–97.
[13] L. Hjelmslev, *Prolegomena to a Theory of Language*.   Madison: University of Wisconsin Press, 1961 (1943).
[14] J. Hullman and N. Diakopoulos, "Visualization rhetoric: Framing effects in narrative visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2231–2240, Dec. 2011.
[15] M. Johnstone, "From killer questions come powerful visualisations," Fell In Love With Data blog at http://fellinlovewithdata.com/reflections/killer-questions, 6 Jun. 2011.
[16] R. Kosara, "EagerEyes blog," http://eagereyes.org.
[17] M. Lima, "Visual complexity blog: Information visualization manifesto," Aug. 2009. [Online]. Available: http://www.visualcomplexity.com/vc/blog/?p=644
[18] S. Mac Lane, *Categories for the Working Mathematician*.   Springer-Verlag New York, Inc., 1971.
[19] J. Mackinlay, "Automating the design of graphical presentations of relational information," *ACM Trans. Graph.*, vol. 5, pp. 110–141, Apr. 1986. [Online]. Available: http://doi.acm.org/10.1145/22949.22950
[20] C. K. Ogden and I. A. Richards, *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*, 8th ed.   New York: Routledge & Kegan Paul, 1923.
[21] S. Pinker, "A theory of graph comprehension," in *Artificial Intelligence and the Future of Testing*, R. Freedle, Ed.   Lawrence Erlbaum Associates, 1990, pp. 73–126.
[22] H. Purchase, N. Andrienko, T. Jankun-Kelly, and M. Ward, "Theoretical foundations of information visualization," in *Information Visualization*, ser. Lecture Notes in Computer Science, A. Kerren, J. Stasko, J.-D. Fekete, and C. North, Eds.   Springer Berlin / Heidelberg, 2008, vol. 4950, pp. 46–64. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-70956-5_3
[23] Radiohead. (2007) House of cards. [Online]. Available: http://code.google.com/creative/radiohead/
[24] J. F. Sowa, *Knowledge Representation: Logical, Philosophical, and Computational Foundations*.   Brooks/Cole, 2000.
[25] M. Tory and T. Moller, "Rethinking visualization: A high-level taxonomy," in *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*.   IEEE Computer Society, 2004, pp. 151–158.
[26] E. R. Tufte, *The Visual Display of Quantitative Information*.   Cheshire, CT: Graphics Press, 1983.
[27] J. W. Tukey, *Exploratory Data Analysis*.   Addison-Wesley, 1977.
[28] J. J. van Wijk, "Views on visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 421–432, Jul.–Aug. 2006.
[29] P. Vickers, "Sonification for process monitoring," in *The Sonification Handbook*, T. Hermann, A. D. Hunt, and J. Neuhoff, Eds.   Berlin: Logos Verlag, 2011, pp. 455–492.
[30] C. Ziemkiewicz and R. Kosara, "Embedding information visualization within visual representation," *Advances in Information and Intelligent Systems*, vol. 251, no. 307–326, 2009.

Paul Vickers is a UK Chartered Engineer, holds a BSc degree in Computer Studies from Liverpool Polytechnic, and a PhD in Software Engineering & HCI from Loughborough University. He is currently Reader in Computer Science at Northumbria University. His research is in the computing domain where it intersects with creative digital media with a particular emphasis on auditory display. He is a board member of the International Community for Auditory Display.

Joe Faith has a BSc in Mathematics from Bristol University and a DPhil in Philosophy from Sussex University. He has combined academic research in visualization and bioinformatics with work in industry including as a founder of an eLearning company. The research in this article was undertaken while he was a Senior Lecturer at Northumbria University, and he is currently with Google Inc.

Nick Rossiter has a BSc in Chemistry with Mathematics and a PhD in Physical Chemistry from Hull University. He is currently Visiting Research Fellow in Northumbria University's School of Computing, Engineering and Information Sciences, where he was formerly a Reader in Databases. His research interests include theoretical aspects of information systems and modelling in databases.