Working Paper Series
ISSN 1170-487X

Reconstructing Minard's Graphic with the Relational Visualisation Notation

by: Matthew C Humphrey

Working Paper 96/30
December 1996

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

Reconstructing Minard's Graphic with the Relational Visualisation Notation

Matthew C. Humphrey

Department of Computer Science,

University of Waikato, Hamilton, New Zealand

matth@cs.waikato.ac.nz

ABSTRACT

Richly expressive information visualisations are difficult to design and rarely found. Few software tools can generate multi-dimensional visualisations at all, let alone incorporate artistic detail. The Relational Visualisation Toolkit is a new system for specifying highly expressive graphical representations of data without traditional programming. We seek to discover the accessible power of this notation—both its graphical expressiveness and its ease of use. Towards this end we have used the system to design and reconstruct Minard's visualisation of Napoleon's Russian campaign of 1812. The resulting image is very similar to the original, and the design is straightforward to construct. Furthermore, the design is sufficiently general to be able to visualise Hitler's WWII defeat before Moscow.

KEYWORDS: Visualisation, graphic design, visual design

94]. Nevertheless, a couple of systems have reconstructed components of Minard's graphic. Sage integrates 10 data attributes into a map-like display that shows troop movement, troop size, temperature, important battles and dates [Roth 94]. It uses colour to show temperature and foregoes the line graph. Much of the geography is missing and the troop path is blocky and discontinuous. It carries little of the artistic brilliance of the original. Mathematica(tm) has been used to recreate the troop path and rivers, although temperatures, troop numbers, city labels and other important details were omitted [Shaw 94].

The Relational Visualisation Notation is a new graphical notation that allows users to specify visualisation designs without the use of programming [Humphrey 96]. Programming here means the expression of procedural or control-oriented constructs, either textually or graphically. We seek to discover the accessible power of the Relational Visualisation Notation. To what extent can complex, graphically expressive visualisations be easily specified?

Towards this end we have used the notation to specify a design for temperature-linked troop movements, and we have used that design to automatically generate a reconstruction of Minard's Napoleonic visualisation. Like the original, it contains a continuous troop path that is shaded for advancing and retreating troops. Cities and troop numbers appear on a map of Russia, and a line graph showing temperature is linked to the cities. We also show the steps that a user might take to create such a visualisation in order to gives us a feel for the usability of the tool. Furthermore, we find that the constructed design is sufficiently general to be able to visualise Hitler's WWII defeat before Moscow.

2. RELATIONAL VISUALISATION NOTATION OVERVIEW

The Relational Visualisation Notation (RVN) is composed of three main parts: semantic data models, graphic relations and design diagrams. The semantic data modelling component provides facilities for describing, storing and retrieving information according to the relational model. It is not unlike a relational database. Each graphic relation visually represents one information relation. They define the information and graphical models of the visualisation as well as the transformation between them. Design diagrams combine multiple information and graphic relations into a visualisation design specification. All of the components are supported with an integrated set of direct manipulation design tools, called the Relational Visualisation Toolkit (RVT).

Graphic relations are defined by an information schema, a graphic schema and a set of bindings. The information schema describes, in a single relation, the information to be presented by the graphic relation. The graphic schema is a template made of boxes, circles, lines and other graphic figures including novel elements, such as graphic iteration and graphic selection. It defines the visualisation's visual structure very much like a parameterised icon [Draper 90] [Gray 90]. The bindings are algebraic expressions, not unlike those found in spreadsheets, that encode the relationships between the information schema and the graphical schema. Algebraic expressions are a successful technique for enabling people without programming skills to specify computational functionality [Nardi 93].

Relational Visualisation Notation design diagrams are directed, acyclic graphs that combine several source relations to produce several output graphic relations. The nodes represent source data, relational operators (such as project, select and join), and reusable, hierarchically nested designs. Design diagrams may be used independently to produce a visualisation, or used as a component of more complex visualisations.

The notation is implemented in Smalltalk/V on a Mac 520 as an integrated environment that supports the design, development, evaluation and production of visualisations. The Relational Visualisation Toolkit takes a design and a set of source relations as input and produces a visualisation as output. The visualisation consists of an ordered series of primitive graphical objects, such as rectangles, lines and text. Rendering the objects produces an image. The toolkit and examples are available through URL *http://www.cs.waikato.ac.nz/~matth/dphil.html.*

2. RELATIONAL MODEL

Minard's graphic is based on data that describe the movement of troops between cities—data that can be described by two relations: The *Napoleon* relation describes the movements of troops between cities and the *Temperature* relation describes the temperature on different days. Each tuple describes a battalion that travelled from one place to another on a particular date. In the schema of Table 1, *troops* is the number of soldiers, *fromPlace* and *toPlace* the origin and destination city names, *fromDate* date the troops left the city, *advanced* is true if the troops were advancing and false if they were retreating, and *losing* is the number of soldiers that perished along the way. The data for the relation are taken from Minard's hand-drawn picture—Table 1 shows three tuples derived from it.

troops integer	*fromPlace* string	*fromDate* string	*advanced* Boolean	*losing* integer	*toPlace* string
422000	Kawno	?	true	0	Kawno East
98000	Tarantino	18 Oct	false	2000	Malejaies
50000	Botr	21 Nov	false	22000	Studienska

Table 1 Schema and sample data of troop movement data

Data have also been extracted from Minard's graphic to show the temperature on particular days. In the schema of Table 2, *date* gives the date on which the temperature was recorded, while *temperature* holds the temperature in degrees of Réaumur's thermometer.

date	*temp*
string	float
14 Nov	–21.0
6 Dec	–30.0

Table 2 Temperature schema and sample data

3. VISUALISATION DESIGN

The RVN design diagram for Minard's graphic, shown in Figure 2, contains three relational information sources across the top. The graphic source *Russia* contains a background map and generates the names and positions of the cities from an associated image. The *Napoleon* and *Temperature* icons refer to source data relations. The remainder of the design is divided roughly into three parts identified by the three vertical columns of operators. The leftmost column contains the *movement* and *smartlabel* graphic relations that reproduce the troop movement path. The middle column smooths the troop path discontinuities by adding filled circles around the cities. The third column produces the temperature graph and links it to the troop movement path.

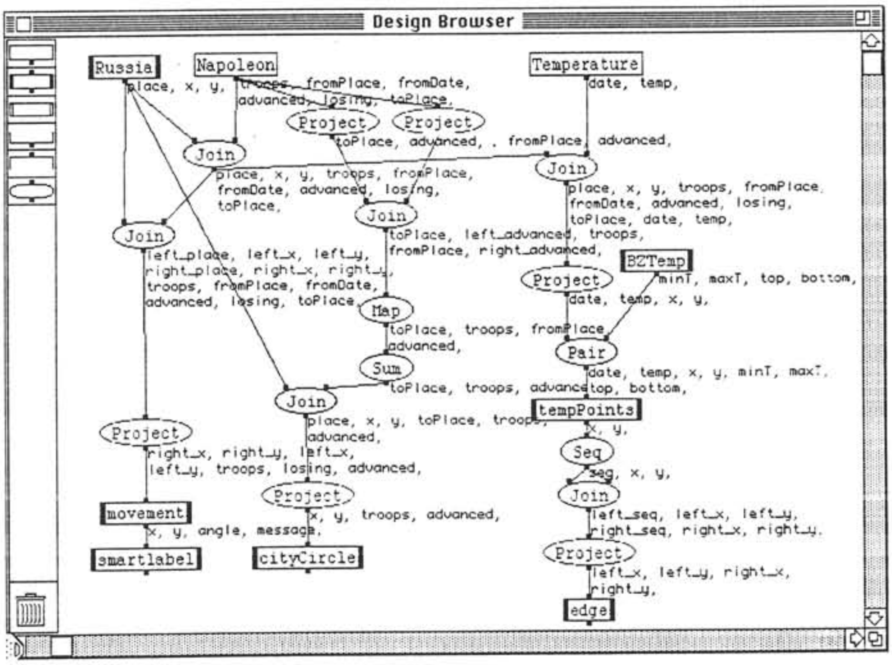

Figure 2 RVN Design of Minard's graphic

Troop movement path

The first step in producing the troop movement path is to convert the city names of the *Napoleon* relation into map coordinates. The first join of *Russia* and *Napoleon* adds the coordinates of the *fromPlace* and the second adds the coordinates of the *toPlace*. Join automatically prefixes replicated attribute names with *left_* and *right_*, so that *left_x* is the *x*-value of the destination city, and so forth. The result is projected to just the coordinates of the origin and destination cities, the number of departing troops, the number perished and whether they were advancing or retreating.

The movement graphic relation, shown in Figure 3, draws lines of varying thickness between the cities. Input attributes *x*, *y*, *tx*, and *ty* stand for the coordinates of the origin and destination cities. *Troops*, *losing* and *advancing* have the same meaning as described above. The graphic schema contains three guidelines that define the shape of the connecting line segment, and a polygon that becomes the segment itself. A polygon is used because the width of an ordinary line segment is constant throughout itself. The longest line segment is named *axis* and directly connects the origin and destination cities. The two others are *fromLine* and *toLine*, and they represent the path segment width at the origin and destination, respectively.

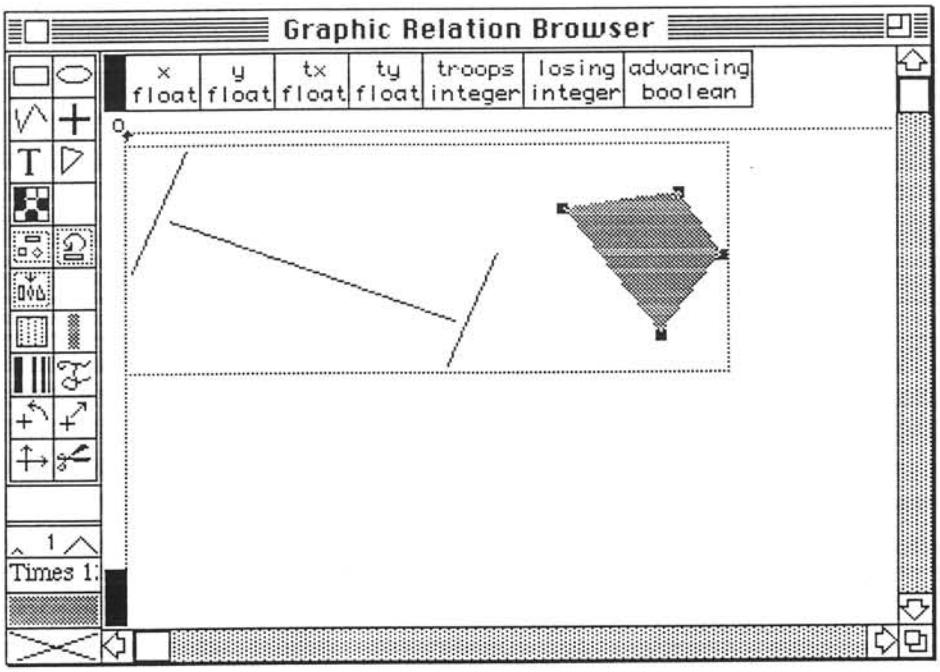

Figure 3 Movement graphic transformation

The *fromLine* and *toLine* are perpendicular to direction of troop movement, are proportional in length to the origin and destination populations, and are centred on the axis itself. The graphic relation preserves these constraints with the algebraic expressions shown in Figure 4. They are presented as assignment statements, although the user need only state the expression body within the toolkit.

```
fromLine.length := 75 * troops / 422000
fromLine.angle := axis.angle - pi / 2.0
fromLine.putcx := x
fromLine.putcy := y
toLine.length := 75 * (troops - losing) / 422000
toLine.angle := axis.angle - pi / 2.0
toLine.putcx := tx
toLine.putcy := ty
```

Figure 4 Bindings onto movement

The Relational Visualisation Notation allows graphic objects to have many different sets of attributes—line segments attributes include *x*- and *y*-end points as well as length and angle. The length of *fromLine* and *toLine* is bound to linear expressions of the number of troops, and their angles are 90° to the axis. Assigning the *putcx* and *putcy* attributes moves the midpoint to a specific location while preserving the length and angle.

City circles

When two troop movement segments meet at a city, they leave an angular gap because their ends are perpendicular to the direction of movement, and possibly of different widths. These gaps can be made smooth by placing circles over them. A circle is needed whenever two or more segments meet at a city—terminal cities retain their perpendicular edges. The circle must cover the larger of either the troops that arrive or depart. (More troops leave Smorgoni than arrive, so we cannot always assume that arrivals are greater than departures.) To shade the circle correctly, the troops at a city should be considered advancing if either they arrived by advancing or they are departing by advancing.

To compute these circles, troop movement data is projected into two distinct relations, one containing the departure cities, and the other containing arrivals. These are joined to find all of the cities which have troops both arriving and departing. The map operator assigns the correct advancing value, and the summarisation computes the maximum number of troops travelling into or out of each city.

The new advancing and troop size data are joined to *Russia* to determine the city positions. The result is then projected to the coordinates of the circle, its width and whether the troops are advancing or not. The *cityCircle* graphic relation uses this information to produce one circle of the appropriate diameter and shading for each city.

Temperature

The last component of Minard's graphic is the temperature scale that shows the freezing weather that the soldiers endured. The scale plots temperature on a vertical axis, while the horizontal axis shows both space and time. The temperature points are positioned horizontally according to the location of the city where the measurement was taken.

Because the troops consistently retreated westward, the horizontal axis also plots time, from right to left, although the temporal divisions are not uniform.

An vertical axis for plotting temperature, shown in Figure 5, is created in graphic source *BZtemp*. It outputs the minimum and maximum data values by binding the outputs to the text values of the end points. The axis source also outputs the physical positions of the end points. These four values are used later to plot the temperature by linearly interpolating the values between the axis endpoints. Although the horizontal lines could have been produced automatically, in this case they have been drawn by hand. Notice, also, that the graphic relation has been scrolled; the source is created relative to the map of Russia so that they will overlap.

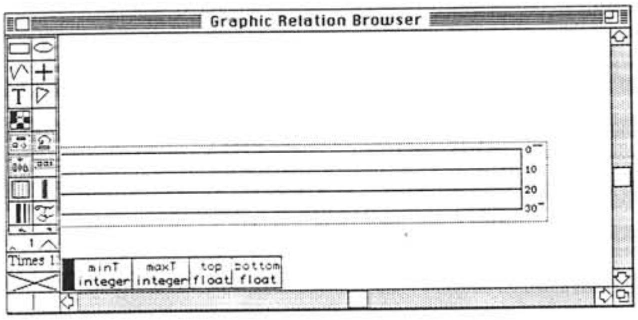

Figure 5 Temperature scale graphic source

As shown in the main design, the temperature data is joined by date with the troop movement data where the coordinates of the departure cities are known. This computation attaches the temperatures to the cities where the troops were at on a particular day. The result is projected to date, temperature and city location. The vertical axis data is paired with this and the result is the visualised as distinct data points. The data points are plotted with the *tempPoint* graphic relation, shown in Figure 6. It accepts data that describes each data point, that is, the temperature, date, coordinates of the city, and the four values produced by the temperature axis. The graphic relation contains three objects: a small ellipse named *point*, a text object called *dateline*, and a vertical line called *column*. These represent the data point, the date and the connection to the city, respectively.

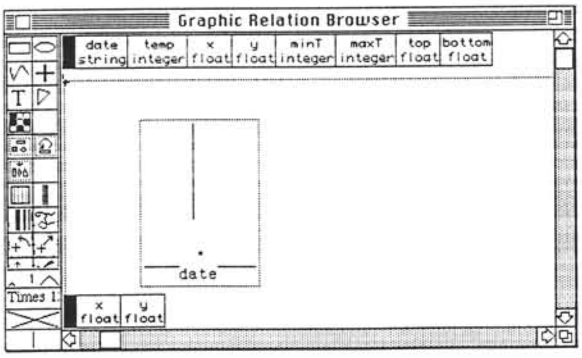

Figure 6 Temperature data point graphic relation

The center of ellipse *point* is placed according to the *x* position of the city and the linear interpolation of the temperature against the axis. The date string is taken directly from the date value. The top of the text object is positioned below the ellipse. The horizontal midpoint of the text is taken from the midpoint of the ellipse. The endpoints of the vertical line are taken from the coordinates of the city and the data point. The line will be vertical because both *x*-coordinates are the same. The output attributes are bound to the center of the ellipse. Figure 7 shows these arithmetic relationships.

```
point.putcx := x
point.putcy := interp(maxT, minT, top, bottom, temp)
dateline.contents := date
dateline.atop := point.ably + 1
dateline.amidx := floor ((point.ablx + abrx) / 2)
column.xfrom := x
column.yfrom := y
column.xto := x
column.yto := point.putcy
```

Figure 7 Data point bindings

The *tempPoints* graphic relation generates the physical locations of the data points as output. These points are sorted in right-to-left order along the axis, with each point receiving a unique sequence number. The points are then paired each to its adjacent neighbour by joining the data to itself over the sequence values—the pairs are the end points of the line graph segments. The data are projected to pairs of coordinates, which are visualised as line segments. The connected segments form the line graph.

5. RESULT

Using source data describing Napoleon's movements and the design outlined in Section 3, the Relational Visualisation Toolkit produces the screen image shown in Figure 8. The result closely replicates Minard's original. The troop path is very similar—it follows the same course with the same deflections at cities. It has the same shading for advancing and retreating movements. The troop path branches similarly to the original, and rejoins it correctly. It also smoothly connects the cities.

The temperature graph mimics the original, including the axis labels, and the dates. The troop population labels are visible and have the same values as the original. The figure titles and geography are also the same, but these are specified directly by the user. The user has complete control over their appearance. Even the cities are placed directly by the user; there is no need to determine or specify their coordinates.

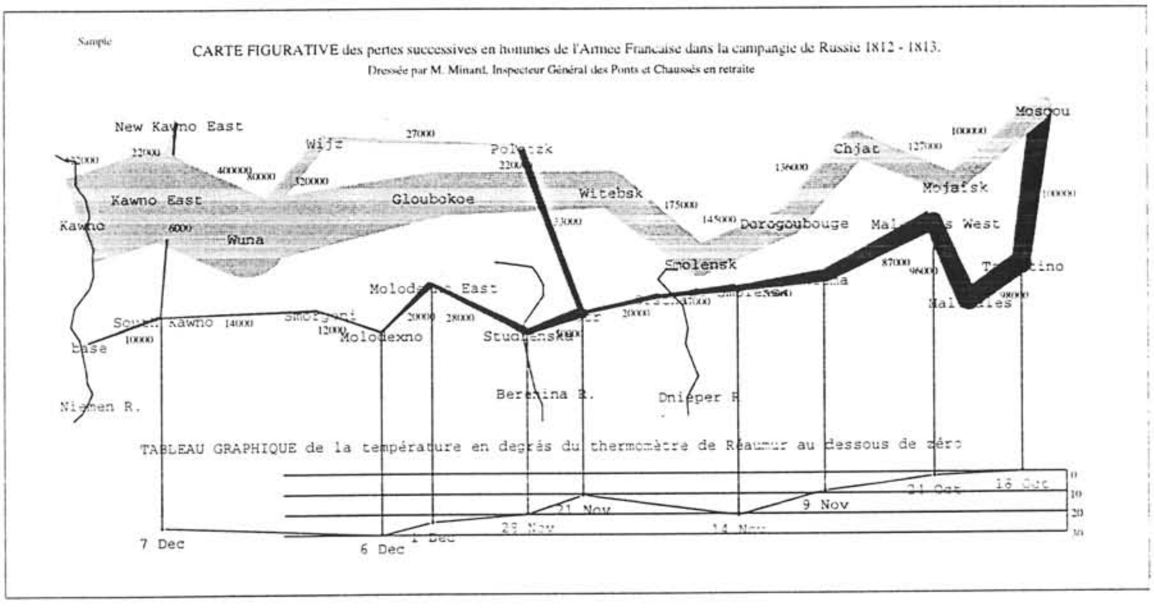

Figure 8 Reconstructed image

Several differences to the original are apparent. The reconstruction contains additional cities that mark where the army changed course—cities that do not appear on the original. The original situates the retreat from Moscow directly south of Moscow itself, whereas the reconstruction layers the assault and retreat on top of each other. The reconstruction also has an extra temperature value for the 24th of October. The extra value is needed because the original temperature graph deflects at that point, but there is no corresponding city.

It is worthwhile to consider the general ease or difficulty of producing such an image. A reasonable estimate of complexity is to count up the number of elements the user must specify to produce the design. Such counts appear in Table 3. The item type refers to a constructional element of the Relational Visualisation Notation. Relations refers only to the schema of the *Napoleon* and *Temperature* relations; the data are not part of the design. The graphic sources are the hand-drawn map of Russia and the temperature axis. Although the map of Russia generates information, the user did not have to specify the actual city coordinates. The relational operators refer to the specific project, select, and join operators (and others) used in the design. The graphic relations are the design components that generate output. The graphic objects are the rectangles, lines and text that appear in graphic relations. Binding expressions refer to the algebraic statements that relate schema attributes to graphic object attributes.

Quantity	*Item*	*Description*
2	Relations	Source data independent to the visualisation
2	Graphic sources	Map of Russia and the temperature axis
16	Relational operators	Project, select, join, etc.
5	Graphic relations	Troop movements, city circles, labels
20	Graphic objects	Rectangles, text, lines, circles
35	Binding expressions	Relating troops to sizes, etc.

Table 3 Design elements

Some specific skills are required to produce an RVN design. The user must be able to specify algebraic expressions, a skill that the widespread use of spreadsheets is making universal. The user must be able to design and specify relational data models. Such a skill is more rare, but nevertheless justified because it is reasonable to expect that people who wish to visualise multi-relational data will have an understanding of the structure of that data. A user who attempts to visualise (or statistically analyse or otherwise manipulate) data without understanding the data model is asking for trouble.

The ability to construct relational queries by assembling primitive operators is also needed. Although this skill is common among information specialists and people who work with relational databases, many others do not have it. Nevertheless, this skill is strongly related to that of knowing the structure of the data. For example, a "join" makes sense only when there is a logical connection among relations. People who understand their data, but have limited querying skills should be more willing to extend their knowledge and have an easier time doing so because such skills are task-specific [Nardi 93].

The utility of a visualisation is measured not only in its effectiveness for a particular design, but also its ability to be used for previously unseen data. In 1941, Hitler attempted "Operation Barbarossa," a full-scale assault on the Northern, Central and Southern fronts of the Soviet Union. Some data are available that describe the movement of German armies through the Central region during Hitler's 1941-1942 Russian campaign [Lucas 79] [Parrish 89]. Although the data were not in a relational form, extensive interpretation has transformed them into a relation with the same schema as Napoleon's.

Applying this data to the Minard design in place of the *Napoleon* relation yields the image of Figure 9. The movement of troops is much more complicated here, with several outbranchings and rejoinders. Some troops, like those going to Kiev, left the central Russian theatre and did not return. Nevertheless, the new visualisation clearly shows the encirclements of Minsk and Smolensk and the rapid successes that the

Wehrmacht enjoyed until it reached Moscow. After a long siege on Moscow and its environs, the German forces eventually withdrew, taking considerable losses.

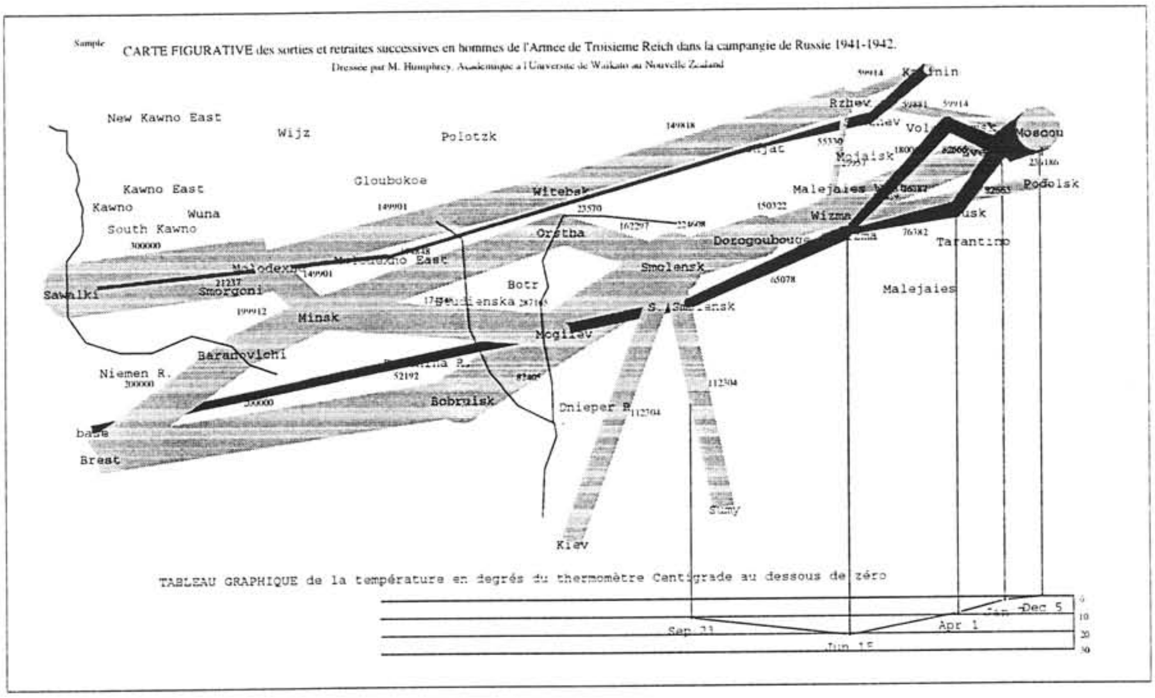

Figure 9 New visualisation showing Hitler's WW II defeat before Moscow

The functionality of the Minard design was not modified in order to produce this diagram. The relational design is the same as for Napoleon's data, and all of the graphic relations are the same. In fact the visualisation is produced from the same specific design. Several circumstantial changes have been made however. New cities needed to be added to the map of Russia, to account for those appearing in the data. These were created by copying and renaming existing cities; no new bindings or functional units were created. The Russian geography was updated to correspond to contemporary 1942 maps and the titles were changed. In effect, only non-formal "decorative" information components were updated.

6. CONCLUSION

The Relational Visualisation Notation is sufficiently expressive to reconstruct Minard's graphic. The expressiveness integrates both formal elements, such as the troop path and temperature graph, and informal elements, such as the titles and background map. The richness of expression arrives without traditional programming structures. Rather, the user creates the visual elements directly and connects them with algebraic and relational expressions. Ultimately, the goal is not to reconstruct existing visualisations, but to develop new reusable ones. The visualisation of Hitler's Central Russian campaign demonstrates that rich, expressive and reusable visualisations can be easily created.

REFERENCES

Bertin, J., *Semiology of Graphics*, W.J. Berg, Tr. University of Wisconsin Press, Milwaukee, Wis., 1983.

Draper, S.W., Waite, K.W., Gray, P.D., "Alternative Bases for Comprehensibility and Competition for Expression in an Icon Generation Tool", in Human-Computer Interaction – INTERACT '90, D. Diaper, D Gilmore, G Cockton, & B Shackel, eds., Elsevier Science (North Holland), pp. 473-477.

Gray, P.D., Waite, K.W., Draper, S.W., "Do-It-Yourself Iconic Displays: Reconfigurable Iconic Representations of Application Objects", in Human-Computer Interaction – INTERACT '90, D Diaper, D Gilmore, G Cockton, & B Shackel, eds., Elsevier

Humphrey, M. C., *A Graphical Notation for the Design of Data Visualisations*, Ph.D. Thesis, University of Waikato, submitted December 1995.

Lucas, J., *War on the Eastern Front: 1941-1945, the German Soldier in Russia*, Janes Publishing Co., London, 1979.

Nardi, B., *A Small Matter of Programming, perspectives on end user computing*, MIT Press, Cambridge, Massachusetts, 1993.

Parrish, M., *Battle of Moscow, The 1942 Soviet General Staff Study*, Pergammon-Brassey's, Washington, D.C., 1989.

Roth, S., Interactive Graphic Design using Automatic Presentation Knowledge, in Human-Factors in Computing Systems, CHI 94, pp. 112-117.

Tufte, E.R., *The Visual Display of Quantitative Information*, Graphics Press, 1983.

Tufte, E.R., *Envisioning Information*, Graphics Press, 1991.

Shaw, MathUser: The Wolfram Research Newsletter for Mathematica Users, 1994.