

# Selecting the Best Graph Based on Data, Tasks, and User Roles

**Jonathan I. Helfman**  
Principal Research Scientist  
Oracle USA, Inc.  
jon.helfman@oracle.com

**Joseph H. Goldberg**  
Principal Research Scientist  
Oracle USA, Inc.  
joe.goldberg@oracle.com

A peer-reviewed paper from:

**UPA 2007 Conference**  
**Patterns: Blueprints for Usability**  
June 11-15, 2007  
Austin, Texas, USA  
<http://www.usabilityprofessionals.org>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright 2007, UPA and the authors.

## **Abstract**

Enterprise software developers and product managers often have difficulty selecting appropriate graph types for users of business applications. Existing solutions for organizing graph types are not adequately focused on end users and their tasks. We have developed a design pattern that serves as a decision support tool for selecting appropriate graph visualizations based upon characteristics of the data, task, and end users. A series of decision matrices allow the user to rapidly filter acceptable graph types. Images and descriptions of each graph type are shown in the tables along with questions such as whether the data are categorical or quantitative, whether the tasks comprise comparison or identification of trend or totals, and whether the end users are experts or casual users of graphs. While already part of the Oracle usability design pattern framework, the graph selection pattern continues to improve as the result of ongoing usability evaluation.

## **Keywords**

chart types, data interpretation, data set types, information graphics, visual representation

## Introduction

As our business activities continue to be automated and measured, larger and larger quantities of data become available, and the need for quickly making sense of data becomes increasingly urgent. People use graphs to see large amounts of data at once, gain insight, and use that insight to take action. But selection of an appropriate graph type is difficult. The choice of which graph to use depends on the types of data to be displayed and the types of insight people are expected to be able to gain. There are hundreds of graph types to choose from, and each has a different effect on the types of insight that may be gained from the underlying data.

**Table 1.** Functional Graph Type Categorizations

<b>Categorization:</b>	<b>Organized by:</b>
Data	Data dimensions and types
Form	Bar, pie, line, area
Complexity	Primary, secondary
Geometry	Radial, rectangular, multiple

Prior work by Bertin (Bertin, 1983), Tufte (Tufte, 1983), and others exposes the complexity of data visualization issues without providing a step-by-step approach to determining the most appropriate graph type for a particular situation (Few, 2006; Harris, 1999). Many researchers in the field of information visualization take a partial approach to helping people identify appropriate graphs by organizing graphs. A similar approach is taken by popular spreadsheet programs (such as Excel) and other data analysis systems (such as SPSS). These attempts to organize graphs cluster them together based on a variety of similarity metrics. Table 1 shows some of the typical categorizations.

The problem with these approaches is that each categorization restricts the way that people might locate appropriate graphs. For example, a data categorization may be appropriate for analysts with a clear understanding of the form of the data they need to display. A data categorization may be less appropriate for developers and designers who know more about the tasks and roles of the users who must interpret the graphs.

A related problem is that most popular computer systems (such as Excel) allow users to map their data to a variety of inappropriate graph types, with no regard to the user's need to gain insight from the data. Graph type choices are further complicated by stylistic decisions such as whether to use gradients, shadows, or 3D edges, which have no relation to the data or associated task. Clearly there is a need for a task-based framework for helping people identify appropriate graph types. This paper describes our progress toward creating a design pattern that helps people find graph types that are matched to their tasks, data, and user type.

## Design Pattern Preliminary Development

As part of our background work, we used the following approach to limit graph types, based upon data, task, and representation constraints.

### *Data Attributes*

Jacques Bertin's, "Semiology of Graphics," (Bertin, 1983) provided a basis for partitioning graph types based upon properties of data. Of particular interest is the concept of a data *dimension* — a set of related data points of the same type and units such as product names, years, costs, profits, etc. A given dataset contains a finite number of dimensions, which are partitioned into two types: Quantitative (Q) and Categorical (C). (Bertin and other researchers also define a third category, Ordered, Ordinal, or Aggregate, which we consider to be the same as Quantitative because, at a fundamental level, every quantity must be measured over a finite, aggregate amount of time or space.) The Q dimensions are numerical (e.g., product sales) and ordered (e.g., months), whereas C dimensions have individual values that may be reordered (e.g., geographic regions, names, or products). Once dataset dimension types are identified, they may then be described by a string, such as "CCQ" or "QQQ". We focused on datasets of up to three dimensions in this

work; higher order datasets may also be mapped to visualizations, in some cases by splitting the dataset into smaller datasets.

Determining the *length* of each dimension is important for further limiting recommended graph types. Although not clearly defined by Bertin, length may be considered as the total number of categories (C dimensions) or the total number of unique values (Q dimensions). Very large lengths may require a data transformation such as clustering or re-binning. In general, we selected lengths of seven-to-ten as the transition point between small and large lengths. Longer length Q dimensions are almost always plotted along the horizontal and vertical dimensions of a rectangular graph. Shorter length Q dimensions may be plotted using ordered graphical attributes of marks such as gray value or bar length. Long C dimensions are good candidates for hierarchical clustering. Short C dimensions are typically plotted as bars or pie slices, but may also be represented using unordered graphical attributes of marks such as shape or color.

Other data-bound attributes were identified to help separate graph types:

- **Aggregate data** is formed when quantitative data are aggregated into uniformly-sized bins or ranges of a sequential scale, such as when salary data are aggregated into salary ranges. Although aggregate data are quantitative, aggregated bins or ranges may also be considered a form of category or group. Aggregate data may, therefore, be shown using different bars or pie slices for each aggregate bin or range. Unlike categorical data, however, aggregate data have a well-defined quantitative order and should therefore be shown with gray values instead of colors in Scatter and Bubble Graphs. (It is this particular behavior of aggregate data that causes Bertin and other to refer to it as Ordered or Ordinal).
- **Series and Pairwise.** A dimension with distinct (non-duplicated), sequential values forms a Series. The other dimensions in the dataset may contain duplicate values. A dimension is Pairwise if it has a value for every possible distinct combination of two Series dimensions. Values in a Pairwise dimension may contain duplicates.
- **Duplicate Values.** Some graph types allow duplicate values in a single dimension to be distinguished from one another, without obscuration.
- **Positive and Negative Values.** Though most graph types can show both positive and negative values, pie graphs and color matrices cannot.
- **Null Values.** Matrices and bar graphs can clearly indicate when both zero-value and null data values are present, but most other graph types cannot.
- **Fractional Values.** Certain graph types, such as binary matrices, can't display fractional data.
- **Hierarchy.** When a hierarchy is defined in a dimension, further restrictions are placed on allowable graph types (e.g., stacked or clustered bars).

#### *Task Attributes*

Several task-based attributes were included to further partition graph types.

- **Task Type.** Basic task categories included Identification, Comparison, and Compare Trends. These also coincide with depth of analysis; identification tasks require a simple lookup, comparison tasks require a user to discover differences between two dimensions, whereas trend analysis requires the discovery of larger trends and inflections in a dataset. These task types were further broken down to include specific dimension combinations, such as "given a C dimension, lookup a Q", or "compare two Q dimensions".
- **Show Percentages.** Some graph types (e.g., pie graphs) are better than others (e.g., bar graphs) at emphasizing that the displayed values are percentages of a larger whole.
- **Show Totals.** Some graph types that show totals (e.g., stacked bars) may display both data and its aggregated values in one view.

#### *Representation Attributes*

A number of attributes address the way that dimensions are represented in graphs.

- **Minimize Obscuration.** Graph types that don't allow overlapping (e.g., Matrices, Bar graphs).
- **Maximize Precision.** Graph types that support a high level of reading precision (e.g. bar length vs. area).

- Simplify Interpretation. Graph types that are easily used by untrained users (e.g., bar or pie)
- Minimize Screen Space. Graph types that are particularly space efficient.
- Maximize accessibility. Graph types that use shape or texture to indicate categories rather than assuming that users can distinguish between colors.

*Dataset Type Decision Tables*

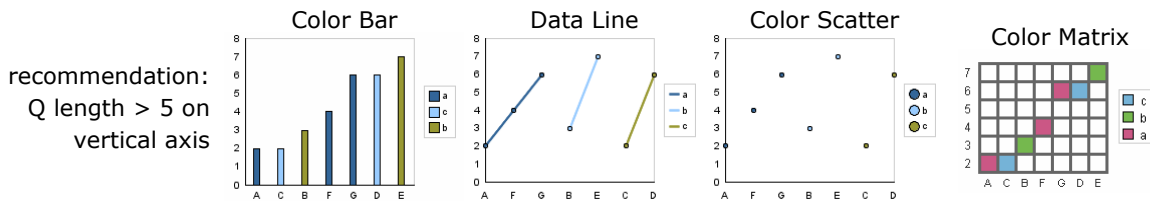
A table was developed for each combination of dimension types (e.g., "CCQ"). Columns were defined by each of 43 possible graph types and rows were defined by data, task, and representation attributes. The fewest possible attributes were selected to efficiently separate the various graph types. Graph types that were recommended in response to a particular attribute were coded as closed circles (●), and types that were acceptable, but less recommended, were coded as open squares (□). Example decision tables with sample datasets are shown in Tables 2 and 3.

The graph type decision tables shown in Tables 2 and 3 organize graphs by the type of their associated dataset. An important finding is that datasets themselves may be organized by their structure as determined not only by the types and numbers of their dimensions (C or Q), but also by the patterns of possible duplications that they allow (e.g. C-series or C-pairwise). Organizing the graph types by associated dataset type reduces the complexity of the problem for more than half of the dataset types (two dataset types have a single possible graph type, seven have only two possible graph types, and three have only three possible graph types). Because these dataset types may be determined automatically, this approach may become the basis for a semi-automated graph type selection system.

**Table 2.** "CCQ - C series" with Sample Dataset

	C <sub>1</sub>	A	B	C	D	E	F	G
	C <sub>2</sub>	a	b	c	c	b	a	a
	Q	2	3	2	6	7	4	6

The "CCQ - C series" dataset type has unique values in C<sub>1</sub> with duplicate values possible only in Q and C<sub>2</sub>. It is possible to show a single level of hierarchy in the C dimensions for the Data Line. C<sub>1</sub> should be plotted on the horizontal axis.

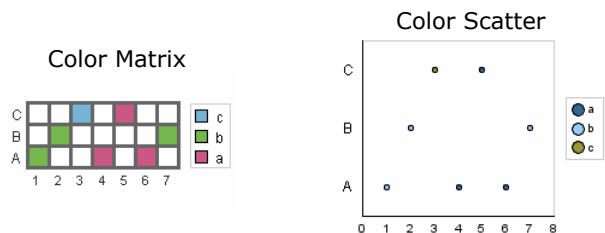


Length constraints	C <sub>1</sub> ≤ ~7, C <sub>2</sub> ≤ ~7	C <sub>2</sub> ≤ ~7	C <sub>2</sub> ≤ ~5	C <sub>2</sub> ≤ ~7
Simplify interpretation	●			
Show +/- values	●	●	●	□
Abbreviate Y scale		●	●	●
Minimize obscuration	●	□	●	●
Compare Qs	●	□	□	●
Show null values	●			●
Compare trends	□	●	□	□
Categorize trends		□		

**Table 3.** "CCQ - Q series" with Sample Dataset

The "CCQ - Q series" dataset type has unique values in Q with duplicate values possible only in C<sub>1</sub> and C<sub>2</sub>. Q should be plotted on the horizontal axis.

Q	1	2	3	4	5	6	7
C <sub>1</sub>	A	B	C	A	C	A	B
C <sub>2</sub>	b	b	c	a	a	a	b



	C <sub>2</sub> ≤ ~7	C <sub>2</sub> ≤ ~5
Length constraints		
Simplify interpretation	●	
Show null values	●	
Maximize precise interpretation	●	□
Show +/- values	□	●

**Graph Selection Design Pattern**

The dataset type decision tables provided a foundation for a graph type selection methodology, but its instantiation was too complex for use by UI designers, product managers, and developers. A 2-step set of smaller decision tables was developed, rather than the larger, more extensive decision tables shown above. The user first selects a graph family, and then selects a graph type from a restricted set of choices.

*Selection of Graph Family*

The scope of the design pattern covers a total of 42 common graph types, organized into 6 families. It does not presently cover less-frequently used families, such as Polar/Radar graphs, Stock Hi-Lo graphs, and other more exotic and less frequently used graphs. The covered graph families are listed below; the first three families are especially appropriate for casual, infrequent users, such as executives. The remaining families are intended for more experienced and frequent users, such as business or financial analysts.

- **Single-Quantity Graphs** (6 types) represent a single quantity or KPI (Key Performance Indicator). Single-Quantity Graphs may also display target values and one or more thresholds to provide context for interpreting the quantity or KPI. Values may be positive or negative. Single-Quantity Graphs are appropriate for casual users.
- **Category Graphs** (4 types) show and compare simple positive or negative quantities that are associated with one or more categorical or aggregate data dimensions.
- **Percentage Graphs** (4 types) are used to compare the percentages of values in a quantitative data dimension that are associated with one or more categorical or aggregate data dimensions. A Percentage Graph should not be used to show both positive and negative percentages.
- **Total Graphs** (4 types) are used to compare the totals of values in a quantitative data dimension that are associated with two categorical or aggregate data dimensions. A Total Graph should not be used to show totals of both positive and negative values. Total Graphs are appropriate for experienced users.
- **Multi-Quantity Graphs** (6 types) show the relationship between two or more quantitative dimensions. Data dimensions are plotted by positioning marks along the horizontal and vertical axis. Data values may be positive or negative. The shape, size, and color of the marks may indicate additional categorical or quantitative data dimensions. The marks may be connected to emphasize trends.

- **Multi-Scale Graphs** (18 types) show relationships between two or more quantitative data dimensions. The values of these dimensions are of different scales within the same units, or are measured in different units.

A decision matrix, allowing rapid selection of an appropriate graph family based upon characteristics of the user and intended tasks, is shown in Table 4. Decision attributes include both User Type and Task Characteristics. Casual users include executives or others who use graphs to see high-level trends, on a relatively infrequent basis. Experienced users such as financial analysts who are not intimidated by complex graphs might seek deeper insight from graphs, such as trend comparison or inflection points. Task Characteristics are high-level filters; graph types are further limited in the second step.

**Table 4.** Decision Matrix for Selection of Graph Family.

<b>Graph Family</b>	<b>User Type</b>	<b>Task Characteristics</b> what data to show or compare?
Single Quantity Graphs	Casual	single quantities with optional thresholds
Category Graphs	Casual	categories associated with simple quantities
Percentage Graphs	Casual	categories associated with percentages
Total Graphs	Experienced	categories associated with totals
Multi-Quantity Graphs	Experienced - frequent user	relationships between quantitative dimensions
Multi-Scale Graphs	Experienced - frequent user	relationships between quantities at different scales or units

*Selection of Graph Type and Detailed Description*

Once the appropriate graph family has been selected, a second decision matrix is provided. This matrix includes a list of Task and Data-oriented characteristics by which the user may select the graph type. Table 5 presents an example decision matrix for the Category graph family. The combination of responses across the questions provides guidance to make the appropriate selection.

**Table 5.** Example Graph Type Decision Matrix for the *Category* Graph Family

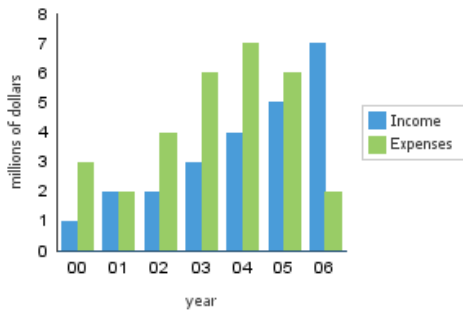
<b>Task or Data Characteristics</b>	<b>Bar Graph</b>	<b>Horizontal Bar Graph</b>	<b>Clustered Bar Graph</b>	<b>Horizontal Clustered Bar Graph</b>
Show or compare quantities associated with one set of categories	yes	yes	no	no
Show quantities associated with one categorical or aggregate data dimension — categories have long labels or display area is short and wide	no	yes	no	no
Show or compare quantities associated with two or more sets of categories	no	no	yes	yes
Show quantities associated with two or more categorical or aggregate data dimensions — categories have long labels or display area is short and wide	no	no	no	yes

The Design Pattern provides a representative image for each graph type, with use case information. High-level guidelines for dimension-axis assignments, ordering, color usage, clustering, and hierarchy representation may also be provided at this level.

**Example Scenarios**

Two example scenario descriptions and graph type solutions are provided to illustrate the design pattern process. While these descriptions are both from the point of view of the end user, the Graph Selection Design Pattern is intended for a designer or product manager within an enterprise software company.

Selecting the Best Graph based on Data, Tasks, and User Roles



**Figure 1.** Recommended Clustered Bar Graph for Scenario 1.

*Scenario 1: Corporate Supply-Chain Executive*

Sam Johnson is a CFO of a manufacturing company, who needs to visualize revenue fluctuations within a specific vertical market on a yearly basis. He would like an appropriate graph visualization that shows the line items on his balance sheet, which consists of two categories: yearly income and expenses. Specifically, Sam needs to know the end-of-year values for income and expenses for one of his company’s product families over the past ten years. He primarily needs to compare income to expenses in any given year. As a secondary consideration, he would like to be able to tell how income and expenses are related

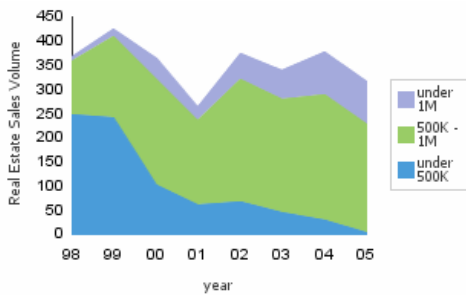
over time.

*Scenario 1 Solution*

As a CFO, Sam can be considered a Casual user. There are a total of three dimensions to visualize: Dollars (quantitative), Years (quantitative), and Line Items (categorical). From the Graph Family decision matrix, the best fit is Category Graphs. Looking at the Category Graphs matrix (Table 5), a Clustered Bar Graph is best for comparing two categories, provided the bar labels are not very long (Figure 1).

*Scenario 2: Financial Healthcare Strategist*

Julie Smith is an experienced real-estate analyst for a national mortgage brokerage. She tracks sales of homes in California, enabling her employer to offer reasonable mortgage products. Julie needs a graph visualization that allows her to understand trends in sales totals for three different market segments over a eight year period.



**Figure 2.** Recommended Stacked Area Graph for Scenario 2.

*Scenario 2 Solution*

As an analyst, Julie is adept at using graphs intended for experienced and frequent users. She requires a graph family that shows totals of quantities (Sales) over time-based intervals. The Totals graph family is appropriate in this case, with three dimensions: Sales (quantitative), Years (Quantitative), and Market Segments (categorical). The Totals graph family decision matrix is shown in Table 6. Julie needs to shows trends in costs, so a Stacked Area Graph (Figure 2) is the most appropriate representation, with Sales on the vertical, and Years on the horizontal. Each Market Segment defines one of the stacked areas, with the total height representative of the summation of the three Market Segments.

**Table 6.** Graph Type Decision Matrix for *Totals* Graph Family

Task or Data Characteristics	Stacked Bar	Stacked Area	Horizontal Stacked Bar
Show two categorical or aggregate dimensions, one of which has associated totals	yes	no	yes
Emphasize trends in totals	no	yes	no
Show totals associated with a categorical (or aggregate) and a sequential dimension	no	yes	no
Show two categorical or aggregate dimensions, one of which has associated totals, and categories have long labels	no	no	yes

From the Design Pattern, Stacked Area Graphs represent totals of categories associated with continuous quantities. Because they fill in the areas between different categories with solid colors, they tend to emphasize trends better than Stacked Line Graphs. As with other Stacked Graphs, only the bottom-most category is plotted with respect to the horizontal axis (each additional category is plotted with respect to its previous category) so accurate judgments may only be possible for the first category. To maximize accurate judgments, consider sorting categories so the lower ones have less variability.

### Conclusions

A method was developed to integrate Task, Role, and Data attributes into a design pattern for selecting appropriate graph type visualizations. Intended for UI designers, developers, and product managers, the decision table-based method is designed for rapid use in enterprise software companies. A complex set of large decision matrices was initially developed to represent the problem in a structured way. The design pattern was then abstracted from these, by noting which attributes most efficiently partition different graph families, and specific graph types. Compared to the detailed matrices, the final design pattern provides a much simpler method to select an appropriate graph. Unlike other popular approaches for organizing graph types into functional categories, our design pattern clusters graph types by user task and role; we expect a user-based approach will be more usable. Formal usability evaluation of the design pattern is currently underway.

### Acknowledgements

The present work was motivated by several Oracle product teams who needed an easy methodology to select appropriate graph visualizations for datasets. Early feedback was provided by Nina Gilmore, Michael Remington, and Carmen D'Arlach. Support for integration within the Oracle usability design pattern framework was provided by George Hackman and Katy Massucco.

### References

- [1] Bertin, J. (1983). *Semiology of Graphics: Diagrams Networks Maps*. Madison, WI. University of Wisconsin Press.
- [2] Few, S. (2006). *Information Dashboard Design: The Effective Visual Communication of Data*, Sebastopol, CA. O'Reilly.
- [3] Harris, R.L. (1999). *Information Graphics: A Comprehensive Illustrated Reference, Visual Tools for Analyzing, Managing, and Communicating*. Atlanta, GA. Oxford University Press.
- [4] Tufte, E.R. (1983) *The Visual Display of Quantitative Information*. Cheshire, CT. Graphics Press.

## **Authors**

The authors are both principal usability researchers at one of the world's largest enterprise software companies, and have been investigating issues related to information visualization and design patterns for several years.

Jonathan I. Helfman, Ph.D., has worked at Bell Labs, AT&T Labs, FXPAL, and Oracle USA, researching, inventing, designing, and implementing interaction techniques and interactive multimedia technologies for helping people communicate, access, and organize information. He has been working and publishing in the fields of user interface technology and information visualization for 20+ years.

Joseph H. Goldberg, Ph.D., CPE has studied large/widescreen display issues, information visualization, querying, and other interface issues for over six years, while a Principal Research Scientist at Oracle USA. He was instrumental in building a usability lab containing and recording interactions on several large and widescreen displays, and has been publishing and presenting in the HCI community for 20+ years.