Proceedings of the Design Society, Volume 5: ICED25
https://doi.org/10.1017/pds.2025.10323

The twelve ways to convey engineering information visually

Bernhard P. Bettig[1,✉], Winnie H. Fu[1] and David A. Lamb[2]

[1] *West Virginia University Institute of Technology, USA,* [2] *U.S. Army DEVCOM-GVSC, USA*

✉ bernhard.bettig@mail.wvu.edu

ABSTRACT A systematic process was used to develop a complete taxonomy of visual representation mechanisms applicable to the display of any kind of engineering information. The resulting twelve categories are broadly divided into eight related to graphical elements treated individually and four related to the arrangement of two or more graphical elements treated in conjunction with each other. The taxonomy is oriented to inform the further development of user interface software frameworks supporting the automated display of interactive engineering information in any form.

KEYWORDS: communication, visualisation, ontologies

1. Introduction

Design_shell is a software framework to simplify the development of highly interactive mechanical engineering design and analysis software. It has been used in a variety of applications including, for example, feature modelling, tethered spacecraft dynamics visualization, Multiphysics hydro generator vibration simulation, micro-electro-mechanical system topology optimization, and GPS track visualization. To facilitate seamless interaction with engineering information, Design_shell avoids using traditional dialog boxes in favour of direct user manipulation of displayed elements. This is accomplished by maintaining a mapping from all displayed elements – including textual words and numbers – to the underlying model object or attribute being represented. The motivation for the research presented here is ensuring that Design_shell is structured in a way that it is generally able to display engineering information in all the ways that an application developer could need.

The objective of the research presented here is to converge on a complete taxonomy of ways of displaying engineering information by systematically examining and organizing all the information-bearing visual elements in a large number and variety of engineering documents and software. The scope of the research is the visual elements themselves and the generic mechanisms they employ to convey information. The research does not address the semantics of the conveyed information and does not address how the user might interact with the information, other than looking at it. The research only examines the completeness of the proposed taxonomy and does not evaluate it with respect to existing taxonomies. The taxonomy should be helpful to inform the development of user interface software frameworks that can lead to richer, less sterile user interface designs, such as for context-sensitive and adaptable digital assistance systems for engineers, especially in virtual and augmented reality environments.

2. Related work

Numerous sources address the mapping from data (or information or a model) to visualization by the user. Model-view-controller software architectures, developed early in the history of graphical user

interfaces, provide mappings that are completely general. (Reenskaug, 2003) (Kay, 1993) (Apple Computer, Inc., 2001) (Fowler, 2003) (Holovaty & Kaplan-Moss, 2009) (Smith, 2009) These architectures are the basis for the windows code running in modern web browsers, mobile devices, and desktop computers, which can theoretically display any kind of information in any way. From a practical point of view, the windows code is useful for simple ways of conveying information but provides limited helpfulness for conveying information in more nuanced ways. Simple display elements such as static text, text input boxes, check boxes, sliders, and calendar date pickers can be used to directly display and edit model information, (Smart, et al., 2020) (Meta Open Source, 2023) however, more sophisticated displays of information must be specially programmed or must use special software libraries. The libraries can be useful for specific types of visualizations, but the windows framework forces displayed elements to remain inside their own visualization windows and forces the windows to remain outside of each other. This rigidness makes for very sterile displays of information. On the other hand, software that is not model based, such as word processor or presentation software, allows for great freedom to create rich expressions of information. Their underlying schemas explicitly allow for text, images, shapes, diagrams, charts, and annotation in almost any combination. (ISO/IEC, 2006) (Ecma International, 2021) Interestingly, new features in some office software also allow displaying information automatically retrieved from a source, for example reading and displaying stock prices. In fact, the schemas theoretically enable mapping displayed elements to models through smart tags and embedded XML.
The discipline of Information Visualization (Spence, 2014) provides more specific studies on domain-independent mapping from data to visualization. Here the focus is mostly on charts and diagrams, and the mapping is treated more like a directed pipeline, however, Information Visualization literature contains the taxonomies that are the most applicable here. For some Information Visualization researchers, the pipeline from data to visualization is the basis for the taxonomy, while others focus on the type of data or the characteristics of the chart or diagram. One source (Spence, 2014) distinguishes the elements of visualizations based on the pipeline: data → representation → presentation → visualization, in which *representation* refers to the various ways in charts and diagrams can show information, *presentation* refers to the viewing and layout of the charts and diagrams, and *visualization* refers to what happens in the mind of the user. Another source (Chi, 2000) is based on the pipeline: data → analytical abstraction → visualization abstraction → view in which *analytical abstraction* refers data that has been extracted and filtered, *visualization abstraction* refers to the visual elements, and *view* provides for zooming, etc. The type of processing at and between the states is considered as the basis for the taxonomy. Both sources provide a high-level classification and detailed examples within the classification, but a systematically derived lower-level classification is not really provided. One source (Shneiderman, 1996) provides a lower-level classification based on the type of visual data: 1-dimensional (e.g., lines of text), 2-dimensional (e.g., geographical map), 3-dimensional (e.g., human body), temporal (e.g., project management timeline), multi-dimensional (e.g., scatter plot), Tree, and Network. Another source (Tory & Möller, 2002) base their taxonomy on the type of data model: continuous (with different independent and dependent variable dimensions, e.g., scalar data that is a continuous function of two-dimensional coordinates) and discrete (with connected data (e.g., tree) and unconnected data (e.g., scatter plot)). Another taxonomy (Culler & Schepers, 2014) is based on chart type: Cartesian (e.g., line plot, scatter plot, histogram), non-cartesian (e.g., table, box plot, radar plot, pie chart, tree, Venn diagram), Combination (e.g., bubble plot), Unusual (e.g., polar plot), and Highly specialized (e.g., digital timing diagram). Commercially, a distinction is made between charts and diagrams, where *charts* are designed for numbers (e.g., line plot, histogram, pie chart, color coded map) and *diagrams* are for text and relationships (e.g., tree, flow chart). (Microsoft, 2023) (Microsoft, 2023) (Microsoft, 2023) (Ecma International, 2021). A good summary of Information Visualization taxonomies is available, (Chengzhi, Chenghu, & Tao, 2003) however, our scope encompasses whole documents and therefore also includes, for example, how documents are arranged, how elements are arranged on a title page, and how figures are captioned. The authors are not aware of any taxonomies or classification systems, including in user interface design research, that encompass our scope.

3. Investigation methodology

3.1. Investigation process

The investigation examined approximately 80 English language documents and software applications, including engineering reports, specifications, patents, journal articles, lecture slides, worked textbook problem solutions, operating and instruction manuals, schematics, books and dissertations, online information sources, resumes, office software, mathematical software, and engineering software. Starting with an initial ad-hoc taxonomy (Bettig B. , 2012), a working taxonomy was continuously modified after parsing one or more documents until it converged to a final taxonomy with about 450 lines of detail. To speed the parsing of documents, a working document analysis form was created with checkboxes to quickly eliminate common situations that had already been catalogued and blanks to record examples indicative of a new category or subcategory.

The selection of documents was somewhat arbitrary and based on convenience, but it was attempted to find as diverse and representative types of documents as possible. The initial document was an industry engineering report written by the first author. (Bristol Aerospace Limited, 1991) After examining a dissertation, (Green, 1999) a patent, (United States Patent No. 11,194,043 B2, 2021) and a design specification (NHTSA, 2013, March) it was decided to examine a book on Information Visualization (Spence, 2014) as a way of covering a lot of variety at once. This done, the taxonomy became quite large, and the other types of documents and software were examined more quickly, but no less carefully. Towards the end, undergraduate engineering students also helped with examining documents.

3.2. Terminology used in the investigation

As the investigation progressed, these definitions and terminology were established to remove ambiguity from the taxonomy.

A *meaning item* is defined as an explicit entity, attribute, relationship, or collection that is contained in or intended to be conveyed by a document or model. The union of all meaning items is the information contained in a document or model. The classification of meaning items is not in the scope of this paper, but the meaning items of a document or model can be imagined, for example, to be the vertices and edges of a semantic network.

A *graphical element* is defined as a clump of pixels or blob of ink that conveys meaning or represents information from the document or model. A graphical element represents and maps to a meaning item. Aside from its existence, a graphical element may also have a separate meaning (and map to a different meaning item) due to how it appears (e.g., its color) or where it appears (e.g., position on a chart). *Graphical elements* are defined as one graphical element or a collection of zero or more graphical elements that convey meaning or represent information. Graphical elements are therefore hierarchically structured and can map to more than one meaning item.

A *piece of information* is defined as a mapping of graphical elements to a meaning item. A piece of information is the result of applying a visual representation mechanism. The classification of *visual representation mechanisms* is the topic of this paper. For the purposes of identifying them in documents, pieces of information must be:

1. Intentional – the pixels or ink blobs that make up a graphical element must not look accidental or like artifacts from rendering.
2. Isolatable – the pixels or ink blobs that make up a graphical element must be at a level of granularity that has meaning, given the type and context of the information being conveyed.
3. Information-bearing – if the visual representation mechanism was not employed in a particular instance (if the piece of information did not exist), some information or fact (a meaning item) would not be explicitly relayed to the reader.
4. Labelable – a term or phrase can be used to state the type of information represented based on the mechanism (e.g., a number is a "page number" because it is at the corner of the page.)
5. Superficial – the conveyed meaning should be generalizable to a meaning relating to visual elements. (E.g., a schematic shows a pipe between a compressor and a condenser. Superficially, a line connects one graphical element to another one.)
6. Consistent – the conveyed meaning generally should not change between instances of similar situations. That is, meaning items of the same type, should have the same visual representation mechanism(s).
7. Not necessarily persistent – the set of visual representation items should also include pieces of information that are temporary (e.g., from highlighting during mouse hover).

4. The results: The twelve ways of conveying engineering information visually

The resulting taxonomy has two broad categories: “Graphical Elements,” which applies to graphical elements treated individually, and “Arrangements,” which applies to two or more graphical elements treated in conjunction with each other.

Graphical Elements

A. Textually

Here the conveyed meaning depends on the existence of a sequence of characters or symbols, which is assumed to follow a syntax and be human-readable or computer-interpretable. This category can be further subdivided into the categories “paragraph,” “sentence,” “clause” or “phrase”, and “word,” each of which can appear individually. Words also include the limiting case of a sequence of one character or symbol, which can be distinguished from Category B in that here the combining of characters or symbols will “erase” the initial meaning (e.g., combining “I” and “n” erases the meaning of “I” and results in a new meaning “In”). Included in this category is the layout of the text itself, for example in paragraphs. In engineering, text is pervasive in bodies of writing, headings and captions, and annotation.

B. As symbols or graphics

Here an image or graphic by itself represents a concept. The existence of the image or graphic denotes or implies the existence of a corresponding meaning item. Unlike Category A, combining symbols does not “erase” the meaning of a symbol (e.g., combining and, means that the folder has been selected, and the individual symbols still represent the meanings of “selected” and “folder”). Symbols may be mixed with text. Symbols may be simple geometric shapes, detailed logos, photographs (that are treated as a whole), or a composition of graphical elements (that is treated as a whole). Subcategories include:

- Image identifies an instance from the model (e.g., corporate logo, profile photo)
- Shape or image identifies a type of information or members of a collection (e.g., blocks in a diagram, markers in a chart, software icon, clipart)
- A composition or collection of graphical elements correspond to an instance from the model (e.g., chart containing axes, titles, markers, annotation, legend, etc.)

Note that a static screen capture of a chart would be categorized here, as would a dynamic composition of all the elements of the chart. In engineering, figures as whole will fall under this category, as will individual lines and blocks within the figures.

C. Diagrammatically

Here a connecting shape, adjacency, or containment shows the existence of a relationship between meaning items. Unlike Categories D and E, the exact position of the elements does not have anything to do with the meaning. The style of the related elements is depicted consistently to represent the type of related entity, and the style of the joining shape is depicted consistently to represent the type of relationship. If the relationship is directional, an arrowhead is used. Subcategories include, for example:

- Flowchart, tree diagram, network diagram
- Venn diagram
- Molecular structure diagram

With respect to the mapping, Category C maps a relationship from the model to a graphical element that pins together the related graphical elements, as shown in Figure 1. Note that Category C does not apply to the diagram itself, which is thought of as a composition and is classified under B. Engineering examples include wiring diagrams and piping schematics.

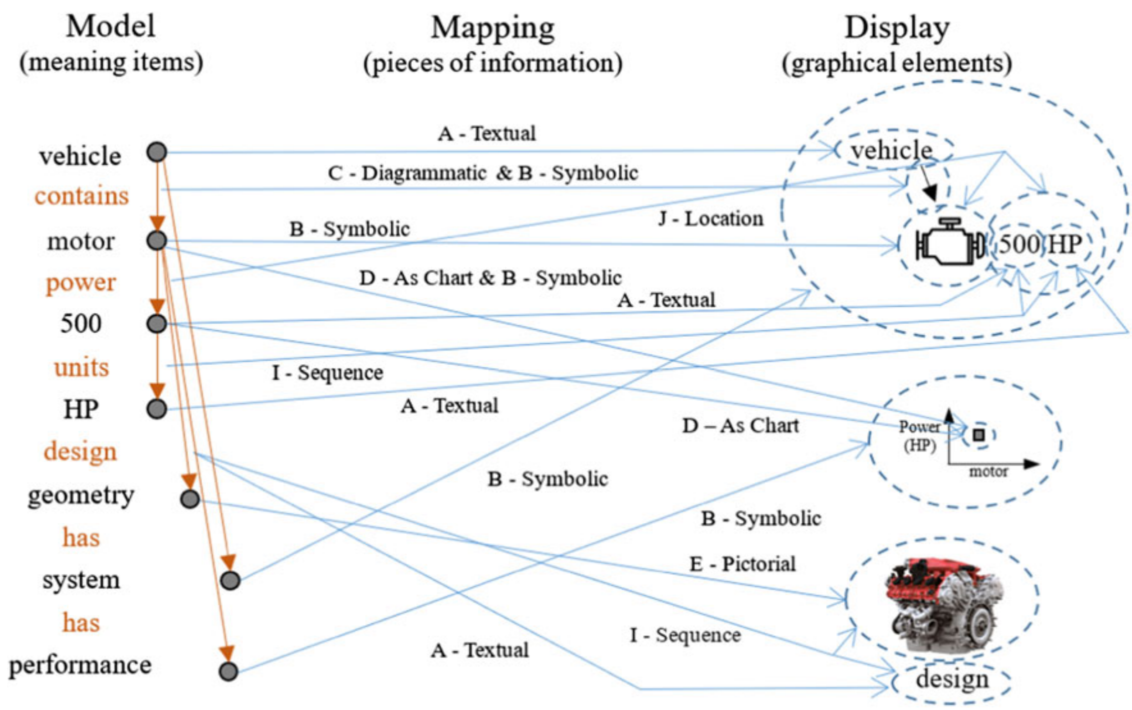

Figure 1. Mapping simplified model meaning items to display graphical elements

D. As charts (or controls)

Here the position, size, color, or other quantitative property of graphical elements indicate a quantifiable property of the meaning item, according to a scale. Subcategories include:

- Position conveys value (e.g., dial, line plot, surface plot)
- Size conveys value (e.g., pie chart, scrollbar (size), line width)
- Color conveys value
- Number of graphical elements conveys value

The graphical elements are typically mapped shapes from B. Category D additionally maps each plotted graphical element to one or more meaning items and, as with Category C, the mapping is to individual shapes in the chart and not to the chart as a whole. In engineering, experimental test data is often plotted using line graphs.

E. Pictorially

Here geometric or locational information is presented in two or three spatial dimensions. Typical examples include photographs, maps, and engineering drawings. Length units are used in each direction. The subcategories are defined by the choices, for example:

- Raster versus vector data
- 2D versus 3D data
- Realistic versus symbolic geometry

With respect to the mapping, Category E applies to any object or concept that has location information, is explicitly represented by a graphical element, and is mappable to a meaning item (e.g., a line representing a wall location, a person identified in a photograph, a region identified on a map). An image as a whole is classified under B however, individual pixels (also B) are located using E. As with C, a figure comprising a composition of located lines, markers, etc., is classified as B. In engineering, Computer-Aided Design models and Graphical Information System data falls under this category.

F. As artwork

Here the graphical elements steer psychologically to the topic or to fulfilling some other communication intent without providing any explicit information (e.g., cover and background art). Subcategories include, for example, background graphical element (e.g., graduated, patterned, or textured) and shape element (e.g., geometric, organic, impressionistic, realistic). As with C, a composition of F is classified as B. In engineering, slide presentations are now typically given with some kind of abstract background image.

G. As arrangement markings

Here graphical elements reinforce arrangement information from I-L and the subcategories correspond to I-L categories:

- For expected sequences (I) (e.g., end of document symbol)
- For expected locations (J) (e.g., leaders (line with arrow))
- For mutual alignment or grouping (K) (e.g., table grid lines, border)
- To support readability (L) (e.g., separating line)

With respect to the mapping, Category G graphical elements mirror the mapping of the I-L piece of information. In engineering, tables typically have grid lines and borders.

H. Through stylization

Here variations in appearance have meaning. The stylized graphical elements already provide a piece of information through their existence, but through Category H an additional piece of information is added. Subcategories include:

- One or more of: static (e.g., font, line or region color, size, or style), animated (e.g., flashing), or user interactive (e.g., pan or zoom setting)
- One of: to identify the level in a hierarchy (e.g., heading level), to identify the type or collection of information (e.g., direct quotation), to identify as having a particular attribute, property, tag, or state (e.g., hyperlink), to distinguish objects from each other (e.g., countries on map), for artistic or readability purposes.

With respect to the mapping, graphical elements that are already mapped to a meaning item are further mapped to another meaning item. Obviously, the same graphical elements could be the target of multiple conflicting stylizations, with results that are undefined. Note that when the stylization expresses a scaled value (e.g., font color represents temperature), this falls under D. In engineering, document styles are one example; line styles in orthographic drawings are another.

Element Arrangements

I. Using expected sequences

Here the position of an element in a sequence conveys meaning. More specifically, the reader knows a priori, following a convention, what type of information a graphical element provides based on its position in a sequence of graphical elements. (E.g., in a mailing address, the reader expects to find the city name after the street address.) Category I also applies to ordered information and sequences (e.g., alphabetical list or sequence of steps). Subcategories include:

- Meaning item is in expected sequence (e.g., book or report sections)
- Attributes of meaning item are sequenced (e.g., mailing address, literature reference)
- Descriptor precedes (or follows) content (e.g., titles and headings of sections, figure and table captions, title as first element of row or column of table)
- Descriptor immediately precedes (or follows) alternative depiction (e.g., position of logo after name)
- Item precedes list of subordinate items (e.g., chapters, sections, and subsections in a table of contents, menu in a computer program)

With respect to the mapping, Category I maps relationships in the model to n-tuples of related graphical elements. In engineering, document sections will follow an expected sequence, as will headings and captions.

J. Using expected locations

Here the spatial position of an element conveys type or membership information. More specifically, the reader knows a priori, following a convention, what type of information a graphical element provides, or what association it has with other graphical elements, based on its location. (E.g., a number at the top-right of a page is interpreted as being the page number.) The subcategories are:

- Positioning with respect to page, view, or window
- Positioning subordinate information (controlling one direction) (e.g., indenting lower-level information)
- Positioning about or within a graphical element (controlling all directions) (e.g., annotation, text within block)
- Specialized positioning (e.g., in mathematical expressions)

Category J does not convey value or location information like D or E. Category J is different from Category K in that Category J does not represent mutual alignments. In engineering, examples include indented hierarchical bills of material, the locations of title blocks in drawings, and annotation.

K. Using mutual alignment or grouping

Here the alignment or proximity of items conveys that they are members of the same collection or sequence. More specifically, the reader knows that, because the graphical elements are aligned and/or are separated from other graphical elements, their meaning items belong to the same composition (or list), are properties of the same meaning item, or are the instances of the same type of something. The subcategories are:

- Grouping (using, for example, white space around the group, matching indentation and justification, or matching bullets or numbering) (e.g., a bulleted list)
- Tables (e.g., traditional table, table of contents, calendar)
- Special alignment (e.g., orthographic views in mechanical drawing)
- Special layout (e.g., distributing the nodes of a network diagram in a circle)

With respect to the mapping, a Category K piece of information maps the set of related meaning items to the set of their graphical elements. In engineering, one example is the alignment of data in the rows and columns of tables. Another is the layout of related controls in a control panel.

L. Using positioning to support readability or convenience

Here the arrangement does not actually convey meaning itself but makes the presented information easier to comprehend by making elements more clearly visible and avoiding confusing the reader. (E.g., placing labels where there is white space makes them easier to read; avoiding lines in a diagram from crossing can avoid confusion; overlapping tabs allows more tabs to be displayed.) As with F, there is no mapping to a meaning item. The subcategories are:

- Make graphical elements non-overlapping and not crossing
- Layout for readability

In engineering, a legend will be put in a part of a graph where there is no plotted data. Dimensions in drawings should be placed so that they avoid crossing each other.

5. Discussion

The evolution of the taxonomy is shown in Table 1. Versions 1 through 7 were created during the study, followed by the final version. (Versions 3b-7 differ from the final version only in terms of wording and definitions.) The observations that lead to the twelve categories were:

> Observing that some graphical elements do not map directly to a meaning item resulted in a new category “Embellishments”. Later, it was observed that Embellishments clearly occurred in two distinct situations: (a) as artwork (F) and (b) as association/arrangement markings reinforcing the existence of graphical element arrangements (G).

Observing that "relative positioning" (graphical element arrangements) could be conveniently subdivided using cues from Design_shell classes (Bettig B. , 2012) led to I, J, and K. For example, the layout of most textual information can be controlled using subclasses of Breakdown_with_heading (a subclass of Depiction_with_heading). Breakdown_with_heading objects put headings before breakdowns (I), indent the breakdown as appropriate for the level and structure of the information (J), and allow the breakdown items to share the same alignment (K), as shown in Figure 2.
Observing that some intentional positioning of graphical elements did not convey meaning led to category (L).

Some wording changes can also be seen in Table 1. In most cases these do not reflect definition changes (e.g., "Schematically" to "Diagrammatically," "As graphic charts" to "As charts (or controls)," "Using" to "As" or "Through") however some wording changes did reflect definition changes:

4. Observing some ambiguity between "Through grouping" (K), in which the definition was based on proximity and white-space separation, and "Through locations" (J), whose definition overlapped K, the definition for J (and I) was tightened to include only "expected" locations (and sequences) where the user or reader knows a priori, based on convention, where to look for the graphical elements.

Table 1. Evolution of the taxonomy of visual representation mechanisms

Initial	Ad-hoc	Version1	Version 2	Version 3a	Final Version
(Bettig B. , 2012) Textually Symbolically Pictorially Schematically As graphic charts Using emphasis		Textually Symbolically Pictorially Schematically As graphic charts As embellishments Using stylization	Textually Symbolically Pictorially Schematically As graphic charts As association markings As abstract artwork Using stylization	Textually Symbolically Pictorially Schematically As graphic charts As arrangement markings As artwork Using stylization	Textually As symbols or graphics Diagrammatically As charts Pictorially As artwork As arrangement markings Through stylization
Using relative positioning		Using relative positioning	Using relative positioning	Through sequences Through locations Through grouping	Using expected sequences Using expected locations Using mutual alignment or grouping Using positioning to support readability

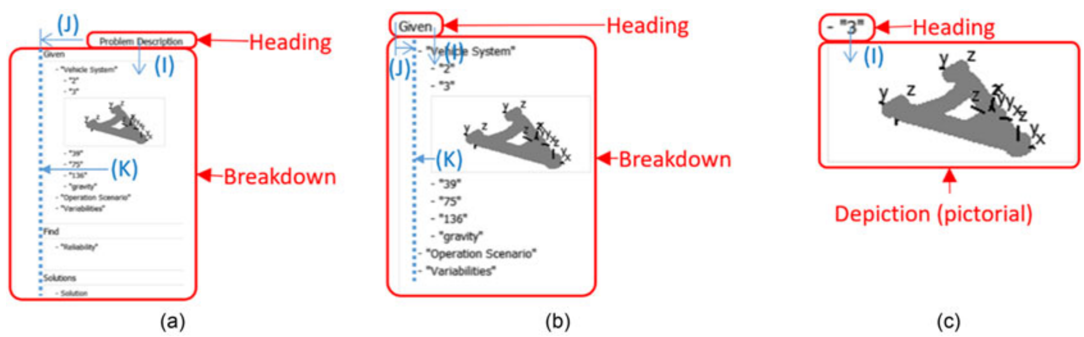

Figure 2. Three fundamental arrangement mechanisms implemented by Breakdown/Depiction_with_heading for (a) page of information, (b) structured textual information, (c) labeled figure

5. Observing that grouped graphical elements also tended to be aligned (e.g., items in a list) led to renaming "Through grouping" (K) to "Through grouping or alignment" (Version 4). This was helpful because it now included, for example, related items in a chart that were aligned, but not in proximity. However, this introduced overlap with "Using locations" (J), for example with captions being aligned by virtue of having the same justification. The issue was resolved by restricting K to apply to mutually related meaning items and J to those that are not mutually related. An interesting

outcome of these restricted definitions was that Annotation, which had been classified under K based on the proximity of the annotation to the annotated item, no longer fell into the definition for K because annotation and annotated items have directed, not mutual relationships. To avoid creating a new category, it was rationalized that annotation is "expected" to be in a particular location relative to the annotated item, for example, be it just above and to the right as with a label or in a margin directly to the left or right as with a comment.

6. Investigating the consolidation of categories .that introduce graphical elements led to renaming B from "Symbolically" to "As symbols and graphics." With this change, only A and B relate to the existence of graphical elements and all other categories use graphical elements from A and B. As part of this change, graphical elements from A and B were not required to map to a meaning item. The restructured taxonomy is shown below. Pragmatically, for purposes of learning the taxonomy and relating it to documents and user interfaces, the original flat structure of the categories was maintained as the "official" taxonomy of this research.

1. Graphical Elements (graphical elements treated individually)
    - 1.1. Introduction of Graphical Elements (related to the existence of graphical elements)
        - 1.1.1. Text (A)
        - 1.1.2. As symbols and graphics (B)
    - 1.2. Control Over the Display of Graphical Elements
        - 1.2.1. Through stylization (H)
        - 1.2.2. As charts (or controls) (D)
        - 1.2.3. Pictorially (E)
    - 1.3. Usage of Graphical Elements (no new functionality required)
        - 1.3.1. Diagrammatically (C)
        - 1.3.2. As artwork (F)
        - 1.3.3. As arrangement markings (G)
2. Arrangements (graphical elements treated in conjunction with each other)
    - 2.1. Representing Relationships (in the model)
        - 2.1.1. Using expected sequences (I)
        - 2.1.2. Using expected locations (J)
        - 2.1.3. Using mutual alignment or grouping (K)
    - 2.2. Not Representing Relationships
        - 2.2.1. Using positioning to support readability and convenience (L)

6. Conclusions

A systematic process was used to develop a complete taxonomy of visual representation mechanisms applicable to the display of any kind of engineering information. The resulting twelve categories are broadly divided into eight related to graphical elements treated individually and four related to the arrangement of two or more graphical elements treated in conjunction with each other. The taxonomy is oriented to inform the further development of user interface software frameworks supporting the automated display of interactive engineering information in any form.

References

Apple Computer, Inc. (2001, May). WebObjects Overview. *Retrieved June 22, 2023, from* https://developer.apple.com/library/archive/documentation/LegacyTechnologies/WebObjects/WebObjects_5/WebObjectsOverview/WebObjectsOverview.pdf

Bettig, B. (2012). *Design_shell Programmer's Manual.* Beckley, WV, USA: West Virginia University Institute of Technology.

Bristol Aerospace Limited. (1991). RAR-609-72-165 CF116 Avionics Update Tactical Secure Voice Installation Feasibility Study. Winnipeg: Bristol Aerospace Limited.

Chengzhi, Q., Chenghu, Z., & Tao, P. (2003). Taxonomy of Visualization Techniques and Systems - Concerns between Users and Developers are Different. *Asia GIS 2003.* Wuhan, China.
Chi, E. H. (2000). A Taxonomy of Visualization Techniques Using the Data State Reference Model. *InfoVis '00: IEEE Symposium on Information Visualization, October 9-10.* Salt Lake City, UT.
Culler, M., & Schepers, D. (2014, Dec.). Data Visualization Taxonomy. *(World Wide Web Consortium (W3C)) Retrieved June 26, 2023*, from https://lists.w3.org/Archives/Public/public-svg-a11y/2014Dec/att-0008/dataviz-taxonomy.html
Ecma International. (2021, Dec.). *ECMA-376: Office Open XML file formats 5th edition.* Retrieved June 27, 2023, from https://www.ecma-international.org/publications-and-standards/standards/ecma-376/
Fowler, M. (2003). *Patterns of Enterprise Application Architecture.* Addison-Wesley.
Green, T. (1999). *Research Issues of Geometry-Based Visual Languages and Some Solutions (Ph.D. Dissertation).* Tempe: Arizona State University.
Holovaty, A., & Kaplan-Moss, J. (2009). *The Definitive Guide to Django.* Apress.
ISO/IEC. (2006). 26300:2006 Information Technology - Open Document Format for Office Applications (OpenDocument) v1.0. *International Standards Organization*.
Kay, A. (1993). *The Early History of Smalltalk*. Retrieved June 22, 2023, from http://gagne.homedns.org/~tgagne/earlyhistoryst.html
Meehan, P., O'Mara, B., Voiola Roque, P., Duong, T., Wang, W., Lambe, M. J., . . . Holas, J. (2021, December 7). *United States Patent* No. 11,194,043 B2.
Meta Open Source. (2023). React DOM Components. Retrieved July 14, 2023, from https://react.dev/reference/react-dom/components
Microsoft. (2023). Available chart types in Office - Microsoft Support. *Retrieved June 27, 2023, from* https://support.microsoft.com/en-us/office/available-chart-types-in-office-a6187218-807e-4103-9e0a-27cdb19afb90
Microsoft. (2023). Descriptions of SmartArt graphics - Microsoft Support. *Retrieved June 27, 2023, from* https://support.microsoft.com/en-us/office/descriptions-of-smartart-graphics-cf1a453b-de4a-4217-8da0-1aff97bb32cd#__toc296443662
Microsoft. (2023). When should I use a SmartArt graphic and when should I use a chart? - Microsoft Support. Retrieved June 27, 2023, from https://support.microsoft.com/en-us/topic/when-should-i-use-a-smartart-graphic-and-when-should-i-use-a-chart-a75a5489-1fb7-4462-b47e-ab2effab6efc
NHTSA. (2013, March). LIDAR speed-measuring device performance specifications. Report No. DOT HS 809 811. Washington, DC: National Highway Traffic Safety Administration.
Reenskaug, T. (2003). *Trygve/MVC*. Retrieved June 22, 2023, from https://folk.universitetetioslo.no/trygver/themes/mvc/mvc-index.html
Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *Proceedings 1996 IEEE Symposium on Visual Languages*. Boulder, CO, USA.
Smart, J., Zeitlin, V., Dunn, R., Csomor, S., Petty, B., Montorsi, F., & Roebling, R. (2020, Apr 25). *wxWidgets: Documentation*. Retrieved July 14, 2023, from https://docs.wxwidgets.org/3.0/
Smith, J. (2009, February). Patterns - WPF Apps With the Model-View-ViewModel Design Pattern. *MSDN Magazine*, 24(2).
Spence, R. (2014). *Information visualization : an introduction (Third edition).* London: Springer.
Tory, M., & Möller, T. (2002, July). A Model-Based Visualization Taxonomy. *Retrieved June 26, 2023, from* https://www.researchgate.net/publication/2560256_A_Model-Based_Visualization_Taxonomy