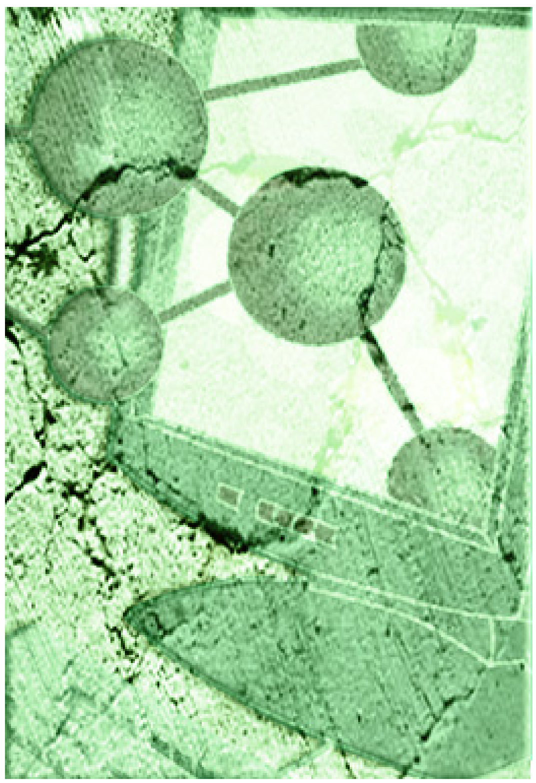
the Rational edge
e-zine for the rational community

The art of modeling
Part III: Visual composition

by Ben Lieberman, Ph.D.
Principal Architect
BioLogic Software Consulting, LLC

download pdf
(xx K)



Clutter and confusion are failures of design,
not attributes of information.
--Edward Tufte, *Envisioning Information*, p.51

In Part I and Part II of this series we explored the theory and practice of model creation. In Part III we will focus entirely on the principles of model presentation, with special attention to the most common form of presentation: two-dimensional visual diagrams. Software developers use these diagrams extensively to facilitate visualization of the inherently invisible aspects of software programs.

Since it is impossible to comprehend all of the electron flows in a running computer system, we must rely on indirect indications of a program's functions: displays on a screen, status lights on a control board, hard copy from a printer, and so forth. To make these complex, interacting systems understandable to humans, a group of very experienced software development theorists created a descriptive language and collection of standard diagram notations called the Unified Modeling Language, or UML.[1] Although this modeling language has been in general usage only since about 1997, it has become the de facto standard for the visual display of software construction and deployment and is supported by a wide selection of computer-aided modeling tools. The majority of the examples in this article will be UML-based models.

One caution: Although UML is extremely powerful, the models created using this technique are useful only if they can be easily understood by the intended audience. Further, an outstanding design can be rendered worthless if the construction principles cannot be effectively communicated to those charged with building the system. For example, a class diagram that contains fifty elements crammed into an 8.5 x 11 inch space on a page would be worthless as a presentation mechanism. Unfortunately, too many misguided efforts to create a single "design document" have resulted in such diagrams. Rather than restricting system representations to only one form of display, it is much better to have several presentation mechanisms.

As Edward Tufte notes, "We envision information in order to reason about, communicate, document and preserve ... knowledge."[2] *To this end it is valuable to explore and understand the principles of effective information display.*

Perception and cognition

A primary source of information for people is visual perception. As noted in the previous articles in this series, humans are better able to reason about very complex problems when they are presented in an appropriate visual form. However, finding that appropriate form is often a singular challenge; frequently, it requires having an artist's eye and a teacher's wisdom. Psychologists have developed a theory of perception known as *Gestalt* (from the German word for "whole" or "pattern") that refers to our ability to perceive a pattern (*figure*) against a backdrop (*ground*).[3] However, these patterns must be either familiar to the viewer or arranged in a manner that will evoke the proper response. Similarly, presenting familiar elements in an unfamiliar context or in an unexpected way will likely lead to confusion and misunderstanding. Consider the line elements in Figure 1. Which of these two symmetrical arrangements of lines do you perceive as something familiar[4]?

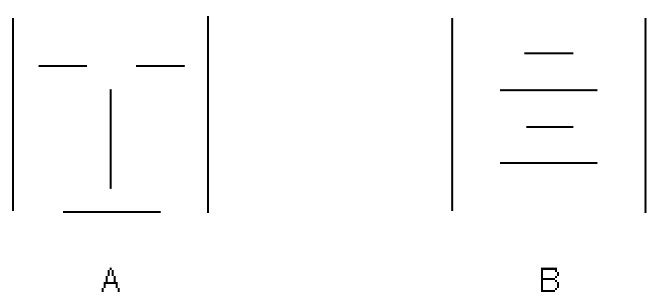


Figure 1: Visual recognition by element position[5]

So the first consideration when modeling with visual elements is whether your audience will *recognize* the symbols and their intended meaning. "Perceptual constancy" -- that is, keeping each element's geometry constant, is key to ensuring understanding.[6] This is important, because in our minds we map new shapes to previously learned ones. This is particularly important if we are using *intuitive comprehension* to map model elements to real-life objects. Intuitive comprehension relies on the model viewers' common experience to correctly infer the meaning of symbols. Road signs, for example, often use intuitive symbols to caution drivers about upcoming conditions or situations (see Figure 2): a mother and child to indicate a pedestrian route (I automatically assume the larger figure is female and not a kilt-wearing Scotsman); a digging man to indicate construction ahead; or a leaping deer to warn that animals may dart across the road. Modern computer operating systems also rely heavily on metaphors for familiar objects to help control complex tasks.

Figure 2: Intuitive traffic signs

The UML use-case model, for example, is close to an intuitive model; simple stick figures represent real life system users. However, since an actor can also represent a non-human system user, these models do not necessarily map directly to people. The modeler must therefore include text-based clues to help viewers interpret the visual elements correctly. Figure 3 shows the difference between an ineffective use-case diagram and an effective one. The ineffective diagram on the left uses nondescriptive labels (e.g., A, B) that probably refer to some other document.

In contrast, the diagram on the right uses clear language to identify system users and options for actions. Those viewing the model are instantly "conditioned" to interpret lavender circles as system activities and stick figures as system users next time they encounter a similar diagram. As we will discuss below, black lines, arrowheads, pastel colors, and rectilinear lines are used in consistent ways in this diagram to help viewers to quickly identify critical diagram elements.

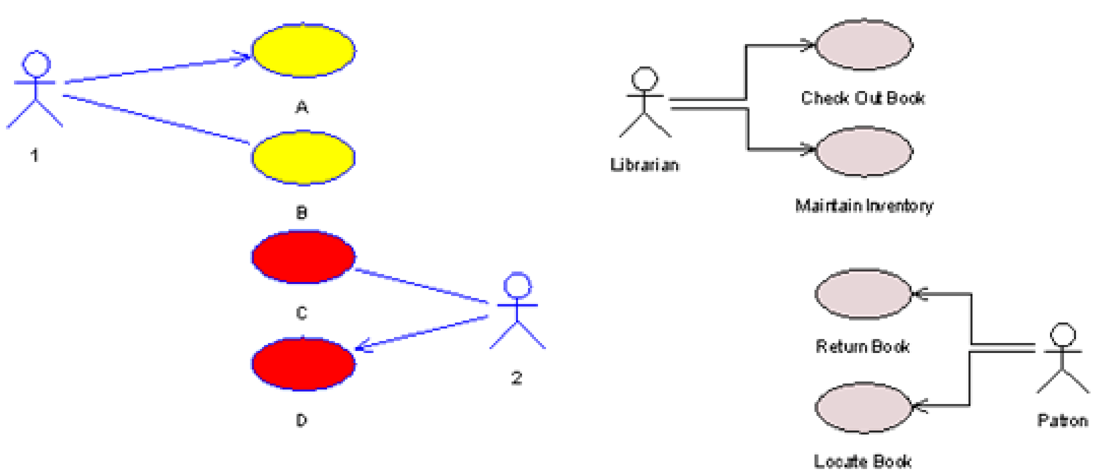


Figure 3: An ineffective use-case diagram (left) versus an effective diagram

Preventing visual overload

In addition to ensuring that their viewers can interpret diagram symbols correctly, modelers must also prevent visual overload. Several powerful techniques are available to control the flow of information to viewers, including *progressive disclosure*, *detail hiding*, and *layering and separation*.[7] As Massironi notes in his treatment of graphic images, "useful drawings are not exhaustive; they are selective."[8]

Progressive disclosure

Progressive disclosure gives viewer successively more detailed information on a specific collection of elements. This incremental revelation of information permits viewer to maintain a continuous connection between increasingly complex model views. The key to this technique is to avoid presenting unrelated information while maintaining a clear connection between one view and the next.

Figure 4 shows an example of progressive disclosure through a series of three UML class diagrams detailing an air travel domain model. Figure 4A shows the core travel objects and containment relationships. The successive diagrams add the search and result object and show relationships relative to the core travel object. This allows viewers to "track" the constant core elements as new elements are added.

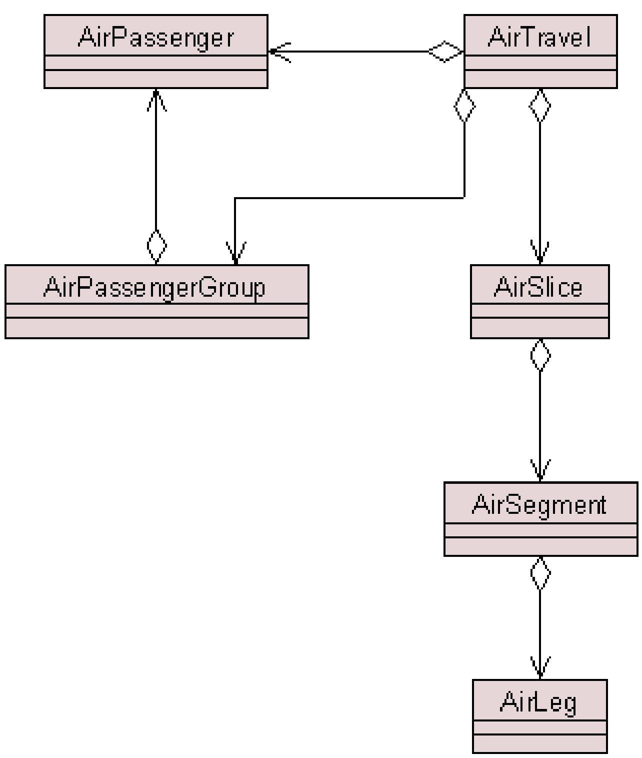


Figure 4A: Air travel domain model: Essential air travel objects

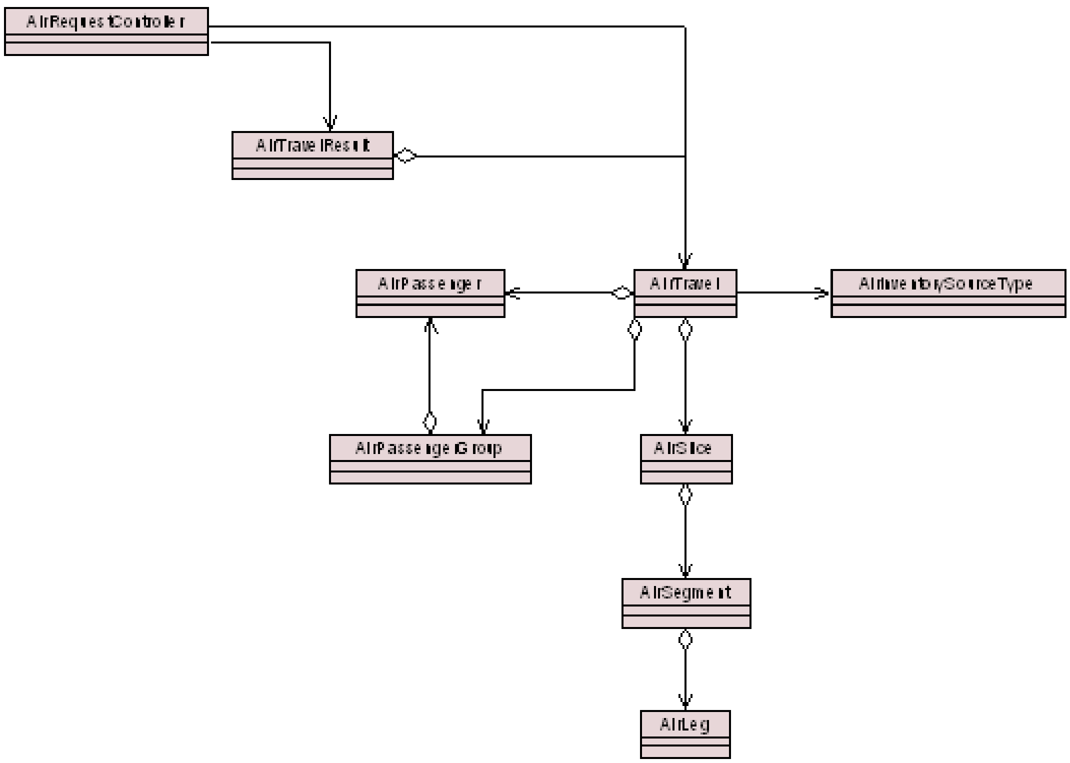


Figure 4B: Air travel domain model:
Essential air travel objects with controller and search request

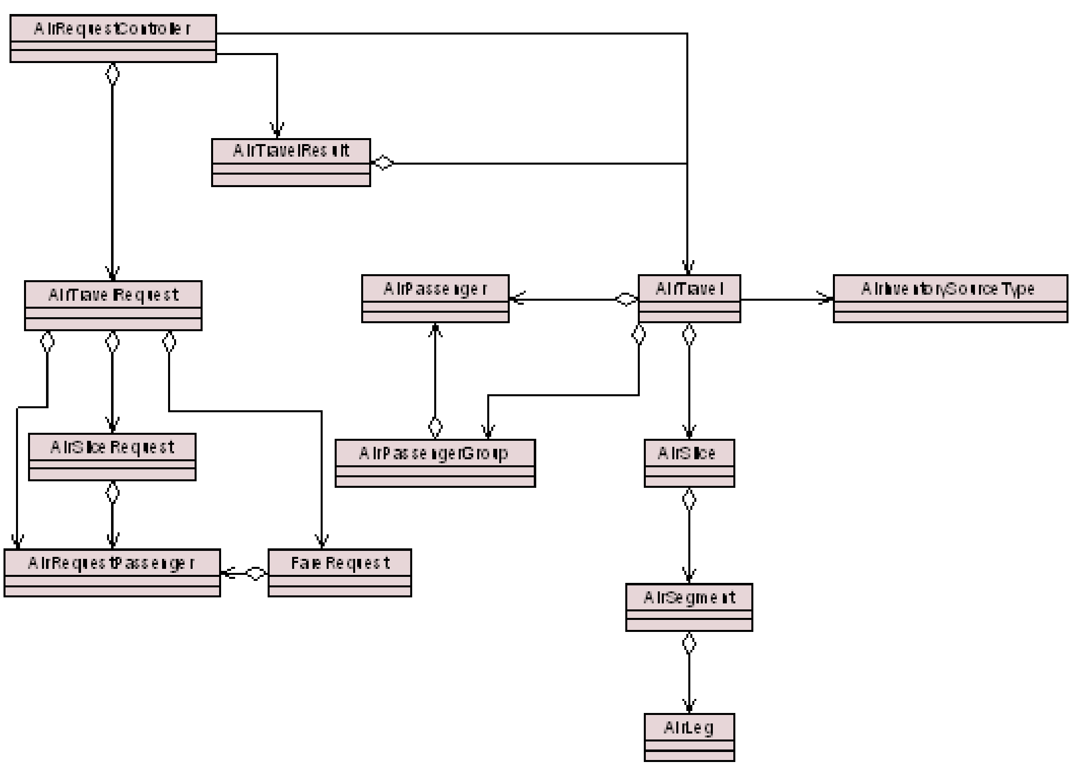


Figure 4C: Air travel domain model:
Essential air travel objects with controller, search request, and search result

Figure 5 shows another form of progressive disclosure; here the core element associations are expanded with multiplicity and constraints.

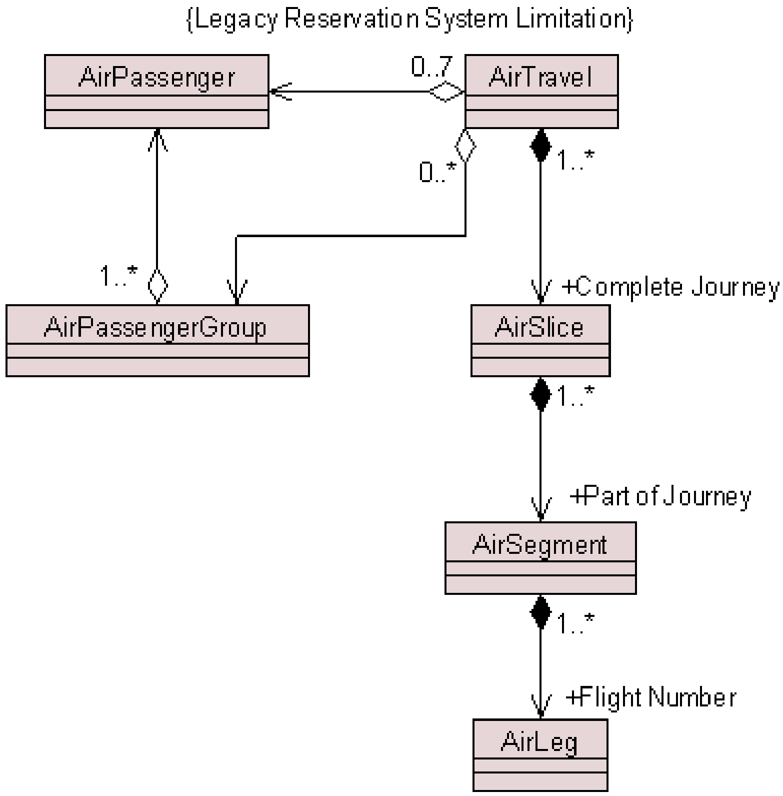


Figure 5: Core air travel with multiplicity, roles, and constraints

Detail hiding

This technique is exactly what its name implies. The UML provides for extensive capture of information about software systems, most particularly for structural elements (e.g., classes and associations). However, a model may contain a great deal more information than is necessary for most people to view. The primary goal of UML diagrams is to allow designers and programmers to intelligently discuss elements of the emerging system. They are not intended to be detailed construction blueprints like those used for civil engineering projects. Complete disclosure of every method, attribute, role, multiplicity, and so forth, for every system class would result in a diagram that is cluttered, incomprehensible, and ultimately useless.

Layering and separation

Layering and separation are similar to progressive disclosure, but these techniques are more effective for displaying disparate but related information. For example, the human body contains multiple interconnected systems. Two of these, the central nervous system and the muscular system, are essential for movement. If you were to display one of these systems in isolation, you would not get a full picture of how muscles contract during movement. However, if you were to display both systems simultaneously, the result would be visually confusing. The best approach is to separate the two views: First provide a picture of the muscles, and then add a separate, progressive overlay of the nervous system. That way, viewers can see one system in isolation first and then come to a gradual understanding of the interconnections between the first and second systems.[9] For software development, you can use a similar approach: First, show a collaboration diagram without method calls to indicate connectivity; then overlay that diagram with details about the method calls (method name, calling order, etc.).

Aesthetics of effective diagramming

Aesthetics in diagramming[10] is more then creating pretty pictures; it is about creating effective views of a model. Unlike some abstract art forms, model diagramming is intended to provide a maximum level of clarity so that the modeler can accurately project his/her understanding to the intended audience. The more attention you pay to placement, size, color, balance, and the overall composition of diagram elements, the more likely your intended audience will retain the model information.

Line and contour

Lines are critical elements of many diagrams. This is particularly true for UML-based diagrams in which lines represent a variety of relationships between elements. For example, typical class diagrams have two primary line forms (solid vs. dotted) and multiple line endings (open arrows for generalization, plain solid lines for associations, diamonds adorning one line-end to indicate containment, etc.). Since each variation has a specific meaning, you should choose your line forms carefully. All relationship lines should be of equal weight and match the weight of the surrounding lines for solid objects. *Relationship lines* of unequal weight will confuse viewers and may incorrectly suggest greater importance for objects connected by heavier lines. In addition, if the *object lines* are too heavy, they will overwhelm the relationship lines and make the overall diagram much more difficult to understand, as Figure 6 shows.

Fortunately, most UML modeling tools have a fixed width for both object and relationship, which minimizes chances for mismatched line widths.

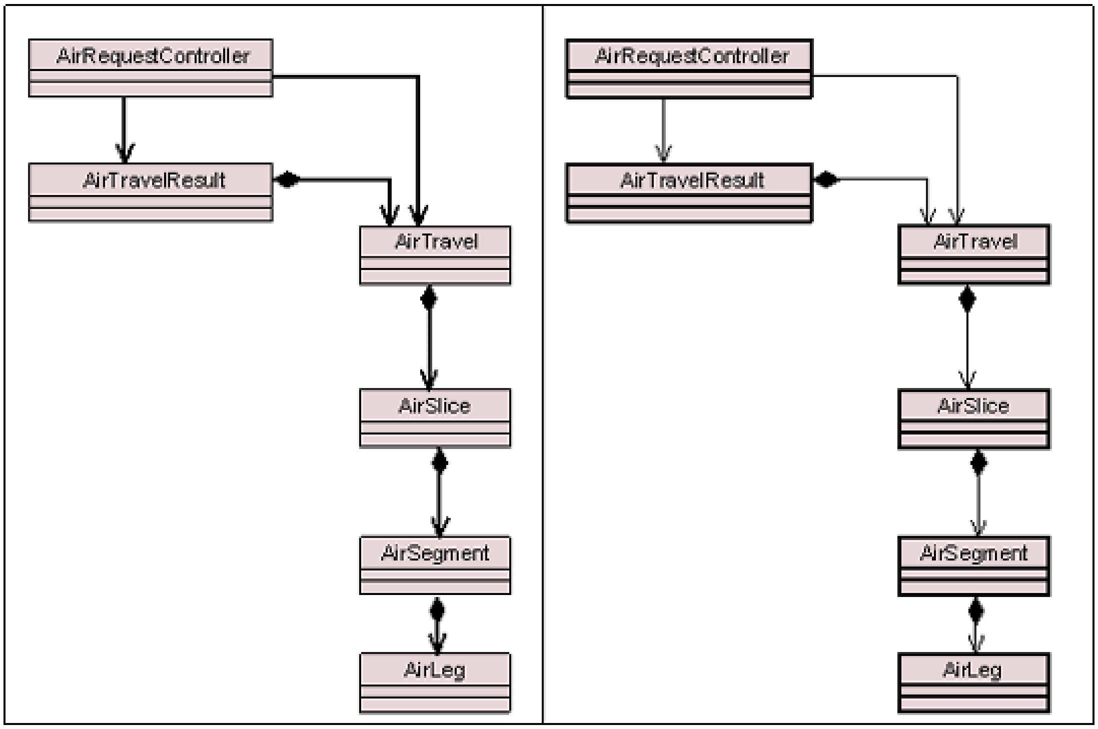


Figure 6: Diagram with inconsistent (and therefore confusing) line weights

Another decision is whether to use rectilinear or oblique lines. Rectilinear lines are straight and form 90° angles, whereas oblique lines can create any angle. In my experience, oblique lines make diagrams visually confusing because of their inherent irregularity; it is far easier for us to interpret straight edges and regular shapes. In my opinion, the ease of recognition and tracking you get from using straight lines between objects overrides any benefit you might derive from using oblique lines for the same purpose.

Consider the diagrams in Figure 7: Which one is easier to understand? Note that I may have influenced your answer by arranging the subsidiary elements (core air travel objects) on the left side of the diagram in panel A. Western languages typically read from left to right/top to bottom, which conditions us to interpret other visual forms of information in the same way. Typically, we expect information in the upper left corner of a display to be the starting point and that in the lower right corner to be the end point. By doing the reverse of these expectations, I subtly introduced additional tension into the diagram. Panel B, which shows subsidiary information below and to the right, is more "comfortable" for most viewers.

As an aside, some tools allow modelers to add "drop shadows" around elements that create a three-dimensional effect and increase the contrast between elements and relationship lines. In my experience, this design fillip adds little to the content or understandability of elements and relationships in the diagram; drop shadows, like other forms of "eye-candy," should be avoided.

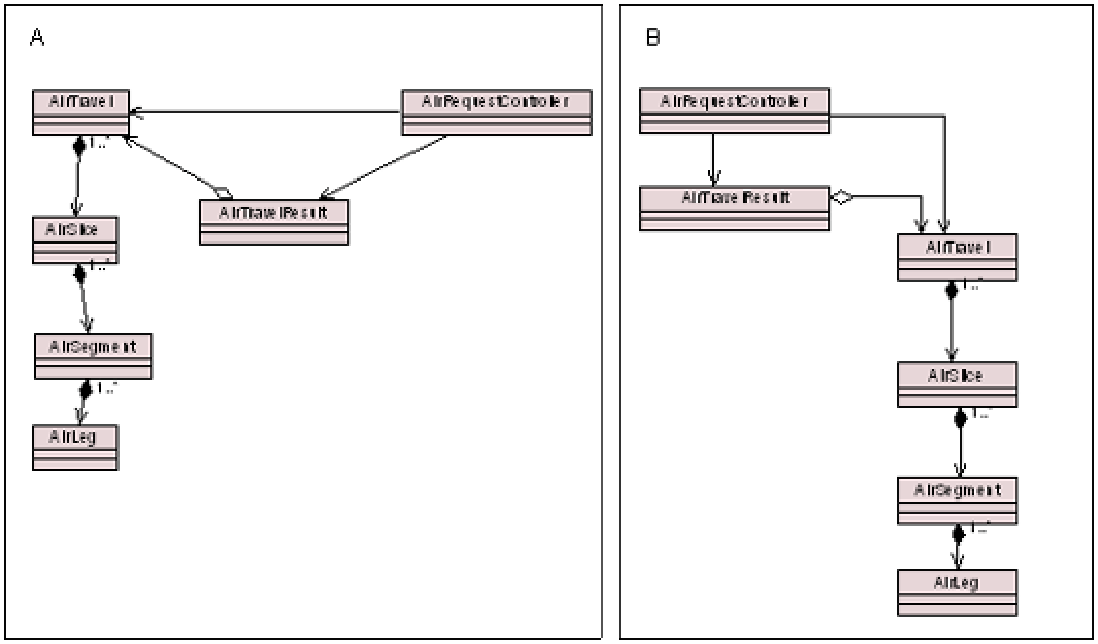


Figure 7: In depicting objects and relationships, rectilinear lines are easier to interpret (and look cleaner when you create them with a modeling tool) than oblique lines.

It is also important to avoid crossed lines and junctions in your diagrams. Multiple crossed lines create a "spider-web" effect: The eye can no longer track a particular line as it intersects with other lines or objects. This is especially true for oblique lines (just try tracing all the routes depicted in a major airline's in-flight magazine, for example). This effect is also obvious in class diagrams and collaboration diagrams, which have large numbers of connecting lines. Fortunately, it is almost always possible to avoid crossed lines by moving elements around and rerouting the lines (see Figure 8). If crossed lines are unavoidable, try to minimize the negative impact by making them cross at a 90° angle at the midpoint of both lines.

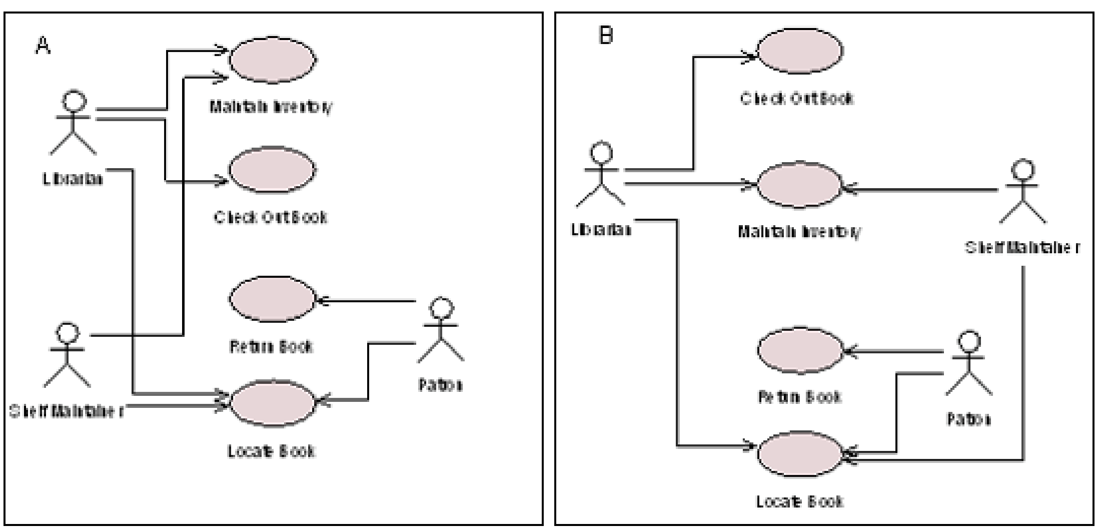


Figure 8: Avoiding crossed lines through rearrangement

As a final note on line arrangement, the way you choose to place combinations of elements horizontally or vertically affects your viewers' ability to identify relationships. For example, refer again to Figure 6, which arranges the air travel objects vertically to shown containment, and places unrelated elements -- for control and holding of search results -- off to the left. Such thoughtful arrangements will help viewers rapidly determine which elements are related and/or highly interdependent.

Scale and proportion

A UML diagram often contains multiple objects in the same view. In a class or collaboration diagram these objects are boxes; in an activity/state diagram the boxes have rounded corners; in a sequence diagram the boxes have lines emerging from the bottom, and so forth. Most UML tools allow you to resize each of these multiple objects, based on the text or other information displayed within the object box (e.g., class methods/attributes). *Scale* refers to the overall diagram size, and *proportion* refers to the relative size of each diagram element. Both the overall scale and the proportion of each element must be balanced in order to achieve a visually pleasing display.

The first rule of proportion is the *golden mean*, which is a ratio of approximately 1:1.62.[12] The ancient Greeks discovered that many natural objects, such as pine cones, flowers, and animal shells, are constructed according to this proportion; it is what we have in mind when we say that something is "well balanced." For the nautilus shell shown in Figure 9, for example, the ratio between successive segments approximates the golden mean.[13]

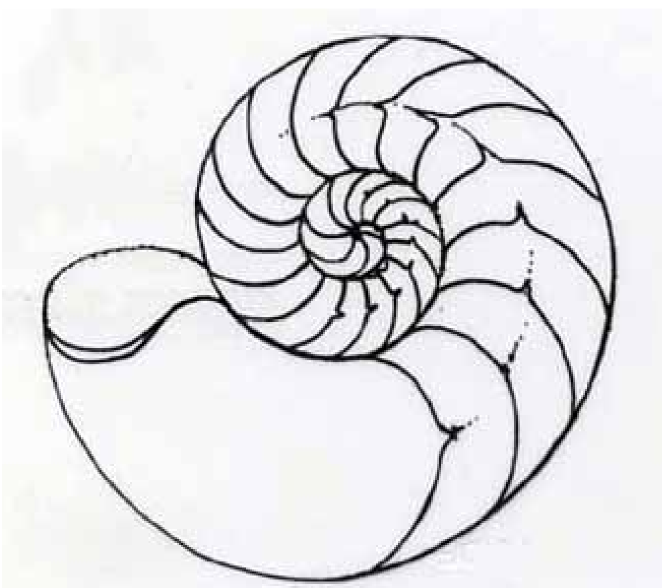


Figure 9: Nautilus shell: The ratio between segments approximates the golden mean

For diagramming, this value (1:62) is most useful when creating quadrilateral (e.g. rectangular or oblong) objects, like those in all UML diagrams. In Figure 10, the diagram on the left has elements of all different sizes, even though the tool maintains the same relative shape (a UML activity icon). In the diagram on the right, each element has been resized to a proportion roughly equivalent to the golden mean. Again, I encourage you to decide for yourself which approach is more aesthetically pleasing.

As we noted above, the human eye is very sensitive to changes in shape and contour. As a result, differences in the size of elements tend to distract viewers from the content of a diagram. Uniform sizing of elements or groups of elements reduces this visual tension and allows viewers to concentrate on the model's content rather than on presentation details.

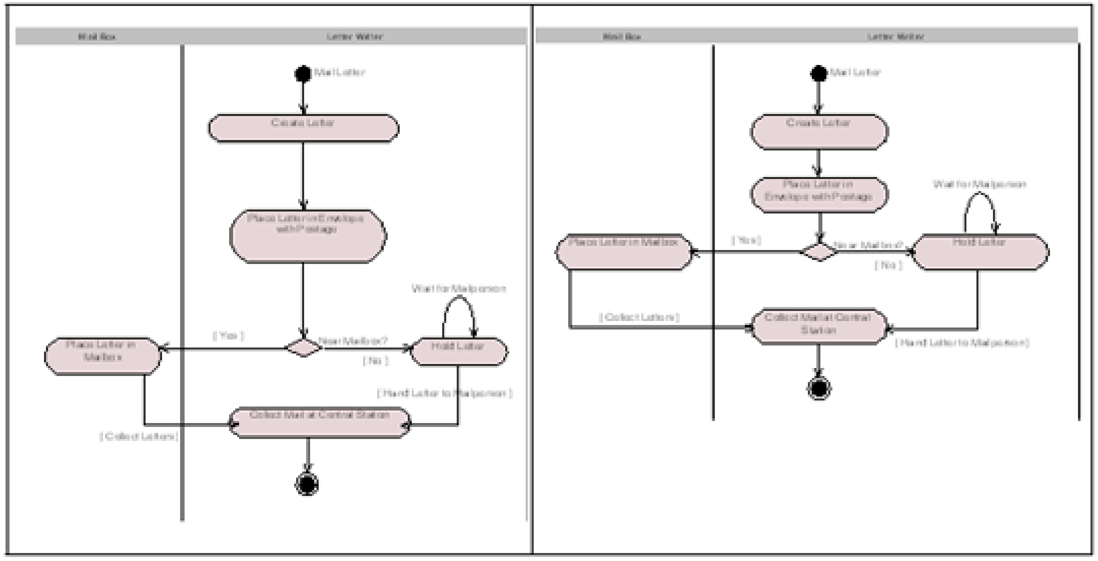


Figure 10: Activity diagrams with elements of different sizes (left) and uniformly sized elements based on the golden mean (right)

Balance

With respect to diagrams, the term *balance* refers to the layout of the elements. In most cases it is desirable for the diagram to be balanced symmetrically, both vertically and horizontally (see Figure 11). This means that it should have as many elements on the right side as on the left, and as many on the top half as the bottom. Most humans find that symmetrically balanced images convey stability and calmness – a desirable effect if you are constructing a cathedral, for example. In addition, symmetrical images mirror many living things, which tend to be either radial or bilaterally symmetric (e.g., starfish and people, respectively).

By contrast, asymmetrical diagrams can be useful for implying activity or motion, or to emphasize specific groups of elements (see **Emphasis in composition** below). Diagrams with a central focus on a few elements will draw the eye *toward* the center. Diagrams that cluster elements in the corners will draw the eye *from* the center. When used with care, such diagrams are particularly good at indicating the relative importance of diagram elements. On a UML collaboration diagram, asymmetry can be used to emphasize critical control elements. Keep in mind, however, that asymmetric diagrams can be disjointed and confusing if you do not construct them carefully.

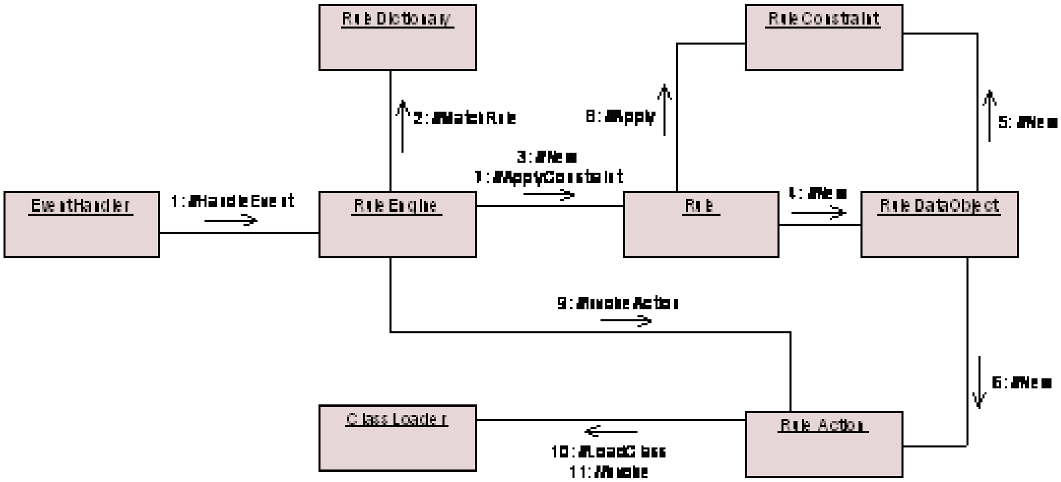


Figure 11: A symmetrically balanced collaboration diagram

Color

The theory and treatment of color is a huge topic that has spawned a massive collection of publications with conflicting notions, often based on empirical trial and error. Although it is clear that color can be used effectively to differentiate model elements, there is no single accepted theory about what color combinations are most effective. Although color can make a dramatic difference in the aesthetic qualities of a diagram and in our understanding of the content, it can also cause visual "noise," confusion, and even discomfort for the viewer.

The most common representation of color relationships is the color wheel invented by Sir Isaac Newton in 1666 (published in 1704 in his seminal work *Optick*). A traditional color wheel (see Figure 12) shows the visible spectrum of colors, starting with a set of *primaries* -- red, blue, and yellow. These colors can be added together to form all other visible colors. The human eye perceives all colors through a complex of specialized structures in the retina at the back of the eye that respond to the various wavelengths of light with nerve impulses to the brain.[14]

More recent advances in our knowledge of human anatomy and the physiology of our visual system have led to updated theories that treat red, blue, and green as "additive primary colors" because these are the colors that the three types of cones in our eyes can perceive.

Figure 12: Standard twelve-part color wheel

Our updated knowledge about human physiology can also help us derive some rules about color use in diagrams. These rules take into account the distribution of rods and cones in the retina as well as changes that must occur in the lens to focus properly on an image. When creating color diagrams, it is important to consider the following[15]:

- Avoid the simultaneous display of highly saturated, spectrally extreme colors. This causes the lenses in our eyes to rapidly change shape and so leads to fatigue. Instead, reduce the saturation of colors (e.g., use pastels), or else use colors that are close together in the spectrum.
- Avoid using pure blue for text, thin lines, and small shapes. Since there are no blue cones in the center of the retina, these elements would be difficult to see. However, light blue makes an excellent background color.
- Avoid using adjacent blues that differ only in intensity. Since blue does not contribute to brightness, the edges will appear fuzzy.
- For elderly viewers, use higher brightness levels to distinguish colors; older people have fewer rods and cones in their retinas.
- Be aware of the ambient light level; it affects how we perceive a color's hue. Avoid subtle variations in hue to distinguish elements.
- Use black lines to distinguish edges. It is difficult to perceive edges created by color alone.
- Since the rods/cones that detect red and green are located in the center of the retina, avoid red and green in the periphery of large displays, and keep viewers' focus where it belongs.
- Opposing colors go well together; so do neighboring colors.
- For color-deficient observers, avoid relying on color distinctions between red and green or blue and green; this is important for the most common form of color blindness (red-green) as well as the less common form (blue-green).
- Printed colors rely on a subtractive method rather than an additive method to produce hue. Colors on a monitor will not exactly match the more saturated colors on a printed page.

Some basic formulae for aesthetically pleasing color combinations are based on the way our brains interpret colors (i.e., on cognitive recognition). For example, pleasing *analogous color combinations* consist of any three contiguous colors on a twelve-part color wheel, with one color that dominates the other two. Attractive *complementary color combinations* use any two colors that are opposite from one another on the wheel. Figure 13A shows the effective use of complementary colors. Combinations of natural colors (such as forest shades of brown, grey, and green) will seem familiar and attractive to viewers, although they do not fall into standard relationships on the color wheel.

Personally, I have found that pastel colors (i.e., colors with lower saturations), combined with slighter more saturated colors to emphasize selective points, work well in UML diagrams As noted in the color rules above, heavily saturated or primary colors should be used sparingly because they command attention and draw the eye away from other elements. Overuse of strong colors can also lead to viewer fatigue and obscure text within or near the strongly-colored objects. Compare Figure 13A and 13B, for example. In Figure 13A the colors help us group and distinguish the elements; the text is easy to read, all elements seem of equal importance. In Figure 13B, however, note how the highly saturated blue obscures the text of the classes; the bright red creates a misleading focal point, and the bright green creates an almost fluorescent effect that draws our eye to the diagram's periphery -- for no apparent reason.

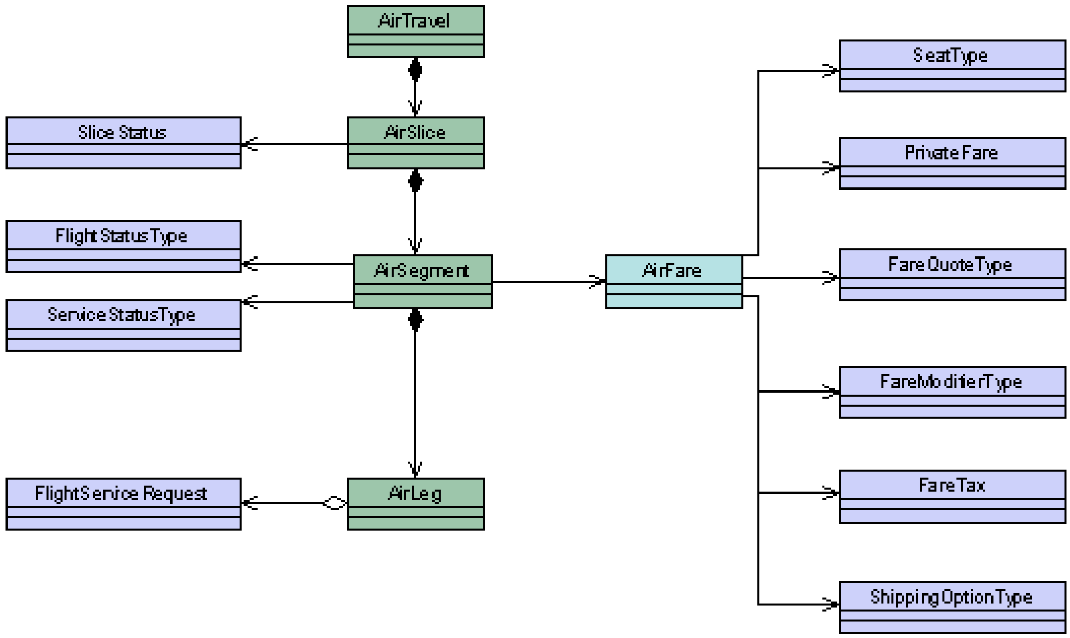


Figure 13A: Effective use of complementary colors

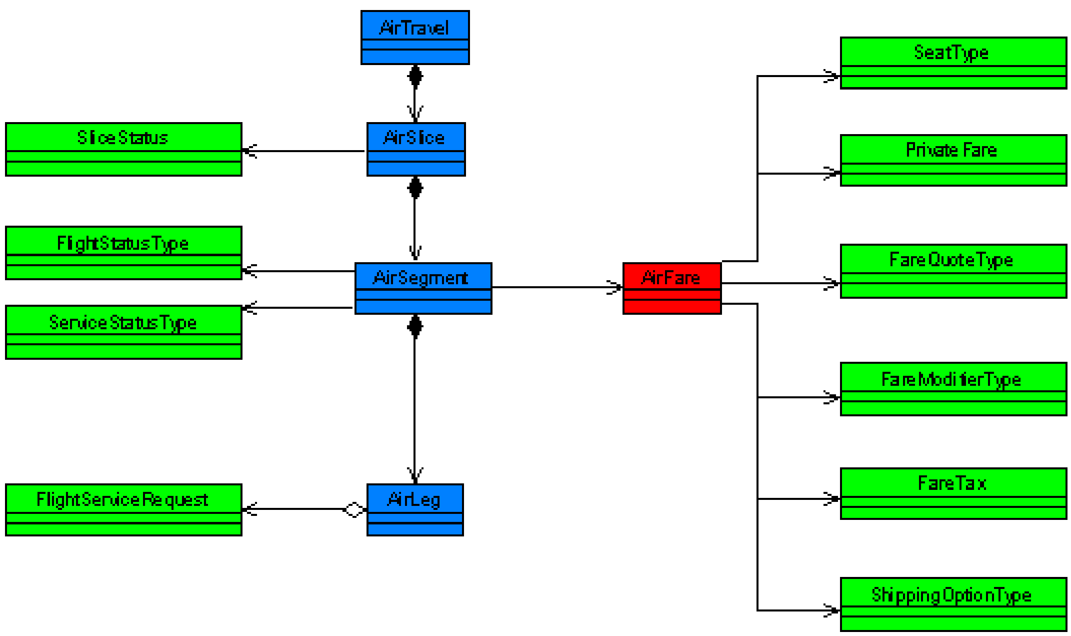


Figure 13B: Overuse of highly saturated/primary colors

Emphasis

Visual distinctions that provide emphasis in a diagram should be subtle but still clear and effective.[16] In UML diagrams, the most subtle change is often to alter the relative position of elements. This follows the Gestalt principle of element grouping, or associating nearby elements with one another. You can also emphasize an element by isolating it from other elements; the eye will be drawn to the isolated element and its unique aspects.

The next most subtle approach is to introduce variations in shape or proportion. For example, if you have one rectangular element but all the others are similar squares, the rectangle will garner more attention than the squares.

Finally, you can add color for emphasis; a judicious splash of color in an otherwise plain diagram can direct viewers' attention to the most important elements. However, do this carefully. As we noted above, using color incorrectly can emphasize the wrong elements and lead to visual confusion.

Repetition and rhythm

Multiple diagrams of the same type should position and size elements consistently from diagram to diagram. For example, if you use the *progressive disclosure* technique, keep the foundation diagram consistent to serve as an "anchor" for the new information being presented. If you are creating multiple model views, you can give your display a consistent rhythm by maintaining the overall diagram size and balance (as described in **Scale and proportion** above), even though you are displaying information about different parts of the system.

Unity and harmony

Returning again to Gestalt theory, humans tend to seek consistencies among visual elements. That is why it is effective to group elements that are co-located and/or share properties of shape, color, and size. We learn these cognitive skills very early in childhood as a way to understand and interpret sensory information. We seek the familiar in the unfamiliar in order to understand new environments and avoid disorientation.

When creating diagrams to represent concepts, it is valuable to include familiar elements that will "anchor" viewers as they discover new information. As was shown in Figures 4A-C, you can do this by placing new information around a stable element, or by layering new information onto a stable *ground* (e.g., to display human anatomy or multiplicity/roles on classes).

We create models to tell specific stories, with each diagram depicting one aspect of that story. UML models describe the story of developing a software system, starting with use cases, and then moving through analysis and design, and finally implementation and deployment. Each part of the story should trace to each other part so that the viewer can follow from one model to the next without becoming lost. These model links are almost as important as the content of the model itself, since you can get the full story only through this progression. Think of the plot in a novel; each new incident typically relates in some way to an earlier incident, which moves the story along and gives it consistency and cohesiveness.

Presentation

We have all seen different ways to present a model. A sculptor might create a small three-dimensional model out of clay for a larger piece she plans to chisel from marble. An architect might create a scale model of a new building with balsa wood, paper, and other materials. For UML models, I have found four useful presentation options: screen projection, hard copy printouts, a "system architecture wall," and a portable portfolio.

Screen projection. Since computer monitors have limited screen space and viewing angles, if you are working with a group, you can project the model onto a screen or wall. This technique is particularly valuable when moving through a sequence of diagrams: you can use a pointer to highlight various elements. This technique also provides a rapid and accurate mechanism to update a model with comments and corrections from the audience and to facilitate discussion.

Hard copy printouts. If you create a model with a plotter, you can produce a hardcopy printout. The advantage of a printout is that you can edit it directly and write comments during reviews and discussions. A large printout also allows you to see a substantial portion of the model at a time. Most models do not translate well to an 8 ½ x 11 format, so you may want to purchase a wide-carriage printer, many of which sell for under $5,000. I find 24 x 36 printouts good for small team discussions but have created prints as large as 36 x 60 to place on the wall during a review.

System architecture wall. Printouts also allow you to publicly display the development team's work. I have found that displaying model views on a "system architecture wall" helps "advertise" the progress of the development team and also elicits hallway discussions on specific points of the architecture.

Portable portfolio. If you have to take your diagrams off site to present to a small client audience, you can safely organize and carry your printouts in an art portfolio with protective plastic sleeves. These portfolios come in a variety of sizes, but I find the 14 x 17 size easiest to manage. You can print the model views on paper that will fit inside the sleeves, and then organize them for a progressive disclosure presentation (see above).

Conclusion

In this article series, I have presented key concepts involved in the selection, creation, and presentation of models, with a focus on software development models. My intention was to provide an overview of important considerations as you conceptualize, create and present models. Given the critical importance of modeling with respect to software development, it is in the best interest of all members of a development team to understand and practice sound modeling techniques. A model is a window on reality; a *good* model should present a clear and concise view of that reality.

References

G., J.Booch et al., *The Unified Modeling Language: User Guide*. Addison-Wesley, 1999.

M. Massironi, *The Psychology of Graphic Images*. Lawrence Elbaum Associates, 2002.

J. Rumbaugh et al., *The Unified Modeling Language: Reference Manual.* Addison-Wesley, 1999.

D. T. Sharpe, *The Psychology of Color and Design*. Nelson-Hall Company, 1974.

E. R. Tufte, *Envisioning Information*. Graphics Press, 1990.

E. R. Tufte, *Visual Explanations*. Graphic Press, 1997.

Notes

1 See Grady Booch et al., *The Unified Modeling Language: User Guide*. Addison-Wesley, 1999, and Jim Rumbaugh et al., *The Unified Modeling Language: Reference Manual*. Addison-Wesley, 1999.

2Edward R.Tufte, *Envisioning Information*. Graphics Press, 1990, p.33.

3 See M.Massironi, *The psychology of graphic images*, Lawrence Elbaum Associates, 2002, and D. T. Sharpe, The psychology of color and design, Nelson-Hall Company, 1974.

4 For those with a background in electrical engineering, the figure on the right may recall the diagram symbol for a battery!

5 M. Massironi, *Op. Cit*.

6*Ibid*.

7Edward Tufte 1990, *Op.Cit*.

8 M. Massironi, *Op.Cit*., p. 93.

9 For a particularly fascinating use of this kind of display, see work from the Visible Human project:
home: http://www.nlm.nih.gov/research/visible/visible_human.html

video: http://collab.nlm.nih.gov/webcastsandvideos/visiblehumanvideos/visiblehumanvideos.html

10 From work by G. Scott Owen, Ph.D., Georgia State University: http://www.etsimo.uniovi.es/hypgraph/design/composition/composition_main_page.htm

11 Edward Tufte, 1990, *Op.Cit*.

12 Mathematically, the golden mean is described by the Fibonacci sequence (0,1,1,2,5,8,13,21...) which results in a ratio of approximately 1:1.62.

13 For a full discussion see: http://www.sacredarch.com/sacred_geo_exer.htm

14 For an excellent review of contemporary color theory, see Josef Albers, *The Interaction of Color*. Yale University Press, 1963.

15 Scott Owen, Ph.D., Georgia State University, http://www.siggraph.org/education/materials/HyperGraph/color/coloreff.htm#Guidelines

16Edward Tufte, "Visual Explanations." Graphic Press, 1997.

For more information on the products or services discussed in this article, please click here and follow the instructions provided. Thank you!