

## ► **The art of modeling**

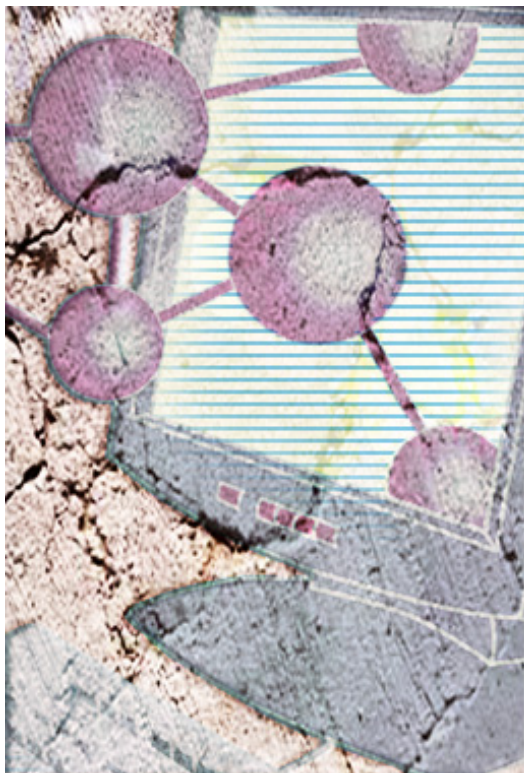
### **Part I: Constructing an analytical framework**

by [Ben Lieberman, Ph.D.](#)

Principal Architect  
BioLogic Software Consulting,  
LLC

*Models are now, and always have been, an integral part of the human experience. We create models of the world about us based on our five known sensory inputs: visual, auditory, textile, olfactory, and gustatory. These models inform us of changes in our immediate perception and also permit understanding of these changes in comparison to past events. As infants we develop sophisticated internal mental models of motion, shape, distance, time, and cause/effect in an effort to relate to the new and confusing world around us. Starting from a nearly clean slate, we must build an internal understanding for every new experience.*

*Our internal mental models of the world are also key to the communication of complex concepts to other people<sup>1</sup>. Since no two people share the exact same set of experiences, no two mental models will be exactly the same. For example, it would be impossible for people living in a primitive hunter-gatherer culture, without contact with the modern world, to understand the principles of refrigeration simply by hearing a description of the mechanics; the concepts are completely foreign to their experience.*

[► subscribe](#)[► contact us](#)[► submit an article](#)[► rational.com](#)[► issue contents](#)[► archives](#)[► mission statement](#)[► editorial staff](#)

*To succeed, first you would need to have a shared communication medium (language, pictures, gestures, etc.), and translate your internal mental model into ones that can be related to their understanding of the world: examples of food storage, effect of cold temperatures on spoilage, and so forth. In other words, you would need to align your explanation of a phenomenon to a shared set of knowledge with your intended audience.<sup>2</sup> Without such a common information base, it is very difficult for one person to explain his or her understanding of a phenomenon or system to another. Moreover, the probability of misunderstandings increases dramatically.*

*Similarly, when developing software systems, every person on the development team must have the same understanding of the needs and purpose for the system. This information is often gathered by a few specialists (e.g., requirements engineers) who then present this information to the remainder of the team, using a series of visual and textual models. Because these models are built from the viewpoint of the specialist, it is important for the "modeler" to understand his own biases and assumptions, since they will not likely be shared by all members of the development team. This need for clear communication of complex concepts is the purpose of creating models.*

*This series of articles will focus on the principles and practices of model conception, creation, and presentation. This first article will be primarily concerned with the principles of system analysis, observation, experimentation, and the techniques that can be employed to study a selected system. The second article will focus more on the practical applications of modeling, with an emphasis on software development models. The final article will focus entirely on techniques and considerations for presentation of information, using UML for the visual examples.*

## **Perception and representation**

A model can be defined as the simplification of a complex system for the purpose of communicating specific details. In this way a model operates much like a lens; it focuses attention on items of direct concern while obscuring or omitting everything else (Figure 1).

A model expands one area



expense of others

### **Figure 1: The effect of modeling on focus of attention**

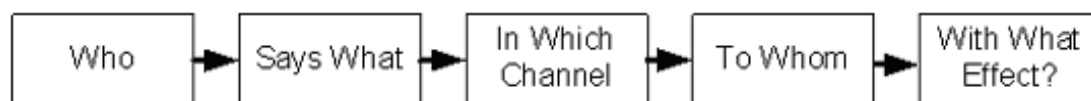
For example, recall how book reports are constructed (for those who can remember back to elementary school). The intent is to capture key elements of the book; highlighting primary plot lines, characters, situations, and so forth, rather than copying the book in its entirety. In this way someone reading the report can gain a reasonable understanding of the important aspects of the work without having to read the full text. This is also a good example of the concept of *abstraction* and how it is used to conceptualize a model (see the **Conceptualization and capture** section below).

In my opinion, there are no "bad" models; some are simply less suited for their intended purpose. For example, the least useful model has a scale of 1:1, which is to say that a 100 percent accurate description of the system is identical (or, in "system analysis speak," *isomorphic*) to the system itself, and thus is not considered a model. So, for any system of significant size, such as one that is larger than can be held completely in mind, it is necessary to sacrifice some accuracy in favor of general understanding. Further, since the human mind has a finite capacity to capture and retain information,<sup>3</sup> in order to reason about a complex topic, it is often necessary to break it down into smaller sections that can be solved and re-synthesized back into the whole.

There are many ways to create models (as will be explored in Part II of this series). Selecting the correct modeling form is very like choosing the form for telling a story. Should the subject be presented as a lyrical poem or a multi-volume novel? Each form has its advantages and drawbacks, and selecting the correct model depends on the experience and education of both the modeler and the audience. Each modeler must carefully consider the intended audience's level of sophistication when choosing a model form and presentation mechanism.

Since there are many ways to represent a system, and since a model is used for the purpose of communication, the model should not be built until the modeler understands the intended audience. Building a model without this knowledge would likely lead to an ineffective model and

miscommunication. To communicate ideas via a model, it is useful to consider how information flows between people. Figure 2 shows two models, taken from the works by Lasswell and Satir listed in the **References** below. Several theories have been developed to explain the process by which information flows from one person to another; each posits the need for a "communication channel" through which the information flows. Often this occurs through spoken language, as is the case in a lecture hall or a meeting room. Visual communication is also used to convey information, as is the case with paintings and advertisements. In all cases, there are two key aspects to understanding: the generation of a message, and the interpretation of that message.



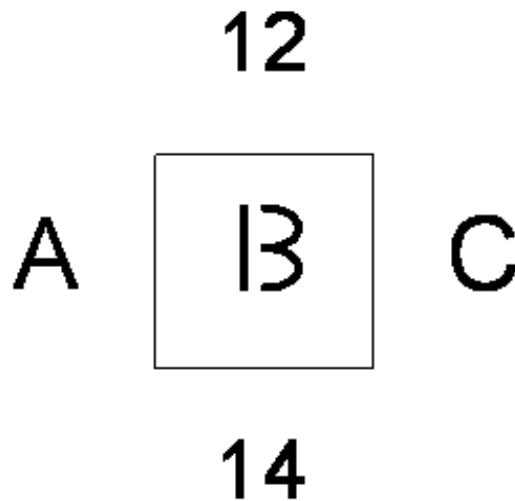
Lasswell, H.B., 1948



Satir, V., 1988

**Figure 2: Two models for communication flows between people (Lasswell and Satir)**

As was mentioned above, individuals perceive the world through the use of mental models. However, no two people share the exact same experiences, so no two mental models are alike. This can lead to errors of interpretation based on differences in perceptions. For example, consider Figure 3, from the work by Meyhew listed under **References** below. What is in the box? Depending on which context you choose (vertical or horizontal), the answer may be the letter B or the number 13. In point of fact, the contents of the box are simply a straight line next to a line with two curves. It is the mental model of letters and numbers that we have constructed that influences our interpretation of the symbology. To someone who cannot read or write (or uses different symbology for those operations), these figures would be meaningless, and this example would represent a very poor model indeed.



**Figure 3: How perception is affected by context (Meyhew)**

So the goal of modeling is to match the description of a complex phenomenon or system to the expectations of the audience that will receive that description. Moreover, we wish to create a representation that faithfully exhibits specific aspects of the system in way that furthers the audience's understanding.

## **Systematic approach: The case of software development**

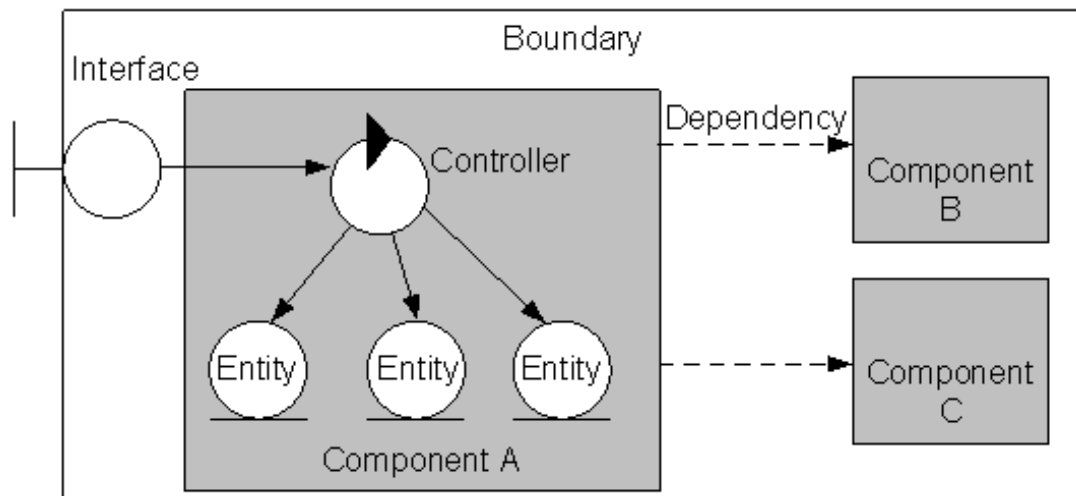
Most models are created to describe complex systems. This is particularly true in the software development field. By its very nature, a computer system is composed of a myriad of interacting parts that all combine to produce visible behavior.

I have found three effective approaches to the analysis of complex systems, including both software systems and other types of systems I've engaged through scientific study: top-down, bottom-up, and middle-out. Top-down analysis starts with a high-level description of the entire domain before moving down to the details. Bottom-up analysis moves in the opposite direction by starting with the concrete details of the system and moving to a more abstract or functional description. In contrast to each of these approaches, middle-out is often useful for dealing with existing systems that are poorly documented. Here, the trick is to pick some point to start (e.g., a section of code), and drill down to the details while simultaneously discovering the functional system description.

Imagine that you are presented with a poorly documented legacy software system. You have access to the user interface, so you could begin investigation of the user functionality by inspection of the interface behavior, perhaps with a subject matter expert as a guide. This would represent a top-down approach. Perhaps you wish to better understand the data model before moving into the business logic that manipulates that data. This would be a bottom-up approach. Or, you could start with one section of the code base and determine the screens that it supports as well as the data that is manipulated. This would be a middle-out

investigation.

Another avenue to investigation of a complex system is to use classical systematic descriptions (systems analysis), based on boundary, interface, component, and dependency, as shown in Figure 4, from the work by Athey listed under **References** below. Boundaries are represented by the differentiation between what is inside and what is outside the system. The Interface represents the communication across the boundary between the system inside and the outside. Components are the elements of the system that interact to perform behavior. Dependencies are the relationships between components in complex systems.



**Figure 4: Classical system analysis approach (Athey)**

The technique of describing a system via systems analysis works well with the three other directed approaches, since the modeler can begin by looking at the system boundary or by working outward from the system components. As illustrated in Figure 4, icons can be used to represent the various system elements such as the interface (which is exposed externally to the system), the controller (which provides for the system logic), and the entities (which are the manipulated system elements) as the system details are captured and elaborated. These systems elements are grouped by components to show cohesive behavior and connected by dependency relationships. (For additional information, see the works of Athey and Jacobson in the references list below.) Systems analysis is often used to investigate and model existing real-world systems into software programs, but it is useful when considering any system (e.g., experimental investigations).

A simple example of this approach could be the analysis of a door lock. The *boundary* is the wall, the *interface* is the door, the *controller* is represented by the keyhole and tumblers, and the *entity* is the deadbolt. The *component* is the full locking mechanism, which has a structural *dependency* on the door and door frame.

## Observation vs. experimentation

There are two principal ways to discover information about a system: by

direct observation, or by experimentation -- both of which have their place in the art of software development. Observation is used when you want to gather information without disruption to the system under study. For example, cultural anthropologists often attempt to have minimal impact on the culture they are studying, since direct involvement between observers and their subjects tends to distort the data they collect.

Biological studies, on the other hand, often require the use of direct manipulation of the organism under study. For example, to determine the genetics of yeast, the geneticist will interbreed different genetic strains and study the outward appearance and behavior of resulting cells. In this way the geneticist can infer mutation points, genetic mechanisms, and other details of the mechanics of yeast biology.<sup>4</sup>

In software development, observation is frequently used as a mechanism for determining requirements, whereas experimentation is the basis for many design decisions.

## **Conceptualization and capture**

Once a model form and investigation mechanism has been selected, the next step is to capture the information from that investigation into a usable form. Here are four general techniques for creating a conceptual model that can be formally captured into a physical model for software development.<sup>5</sup>

### **Technique 1: Abstract/specialize**

The first and most useful technique in the modeler's toolkit is the ability to find commonalities and differences among the different data elements that comprise the system under development. The term *abstraction* is used to indicate the modeler's attempt to denote common characteristics between two or more system elements, with the goal of reducing redundancy and providing for a common description. The direct opposite of this is to *specialize*, or find the distinctions between two system elements to highlight the unique characteristics of each.

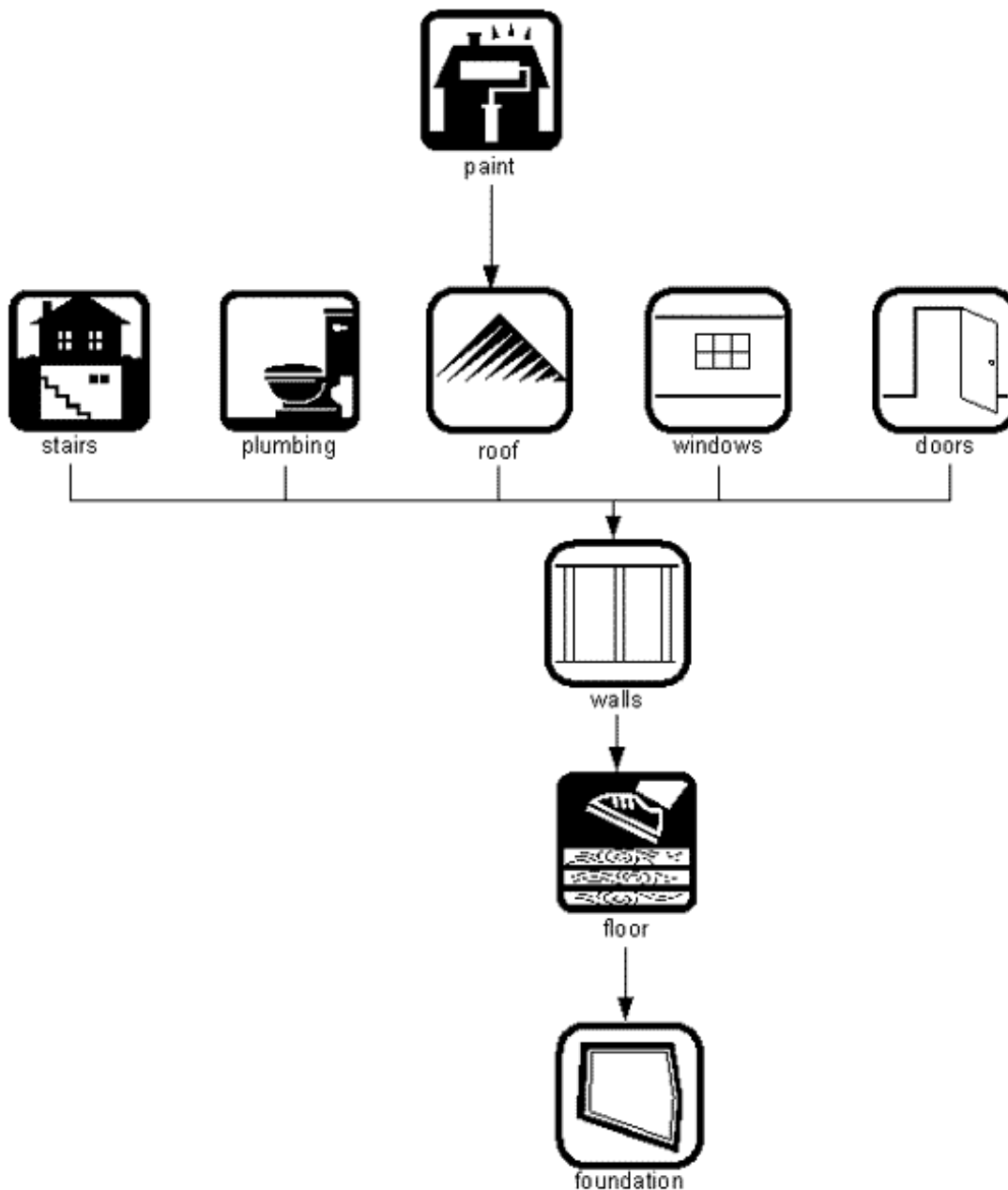
Figure 5 illustrates the idea of grouping for a defined abstraction. Here a group of shapes can be sorted into a variety of categories based on their characteristics: color, shape, and size. For example, if this set of shapes is intended for use in a visual recognition study of colors, then the first model is more appropriate than the other two. Alternatively, this set of shapes may be intended to illustrate the principles of congruent geometry, in which case the second grouping is more useful. Finally, the intent may be to illustrate effects of volume changes, in which case the third grouping may be most appropriate.



**Figure 5: Symbolic groupings**

There is no "right" or "wrong" way to group this set of shapes; the correct "model" is dictated by the intended use. A real world example can be found in human anatomy books, which show each of the human body systems (e.g., circulatory, skeletal, nervous, lymphatic, endocrine) in isolation from the others. Thus, the skeletal grouping is based on bones and the concept of structural support, and the circulatory grouping is based on artery/veins/capillaries and the concept of blood flow.

Figure 6 illustrates a dependency map for groups of related elements. In this example, some of the components of a house are dependent on the construction of previous components. The walls and floors are dependent on a foundation, and all the rest of the house components are dependent on the creation of walls. Finally, painting can occur only after the creation of a roof to protect rooms from the weather. This example also shows the use of icon-based symbology to represent the various house components and the concept of *obscuring* detail, as there is no indication of what comprises each of these components (e.g., is the roof slate? Tile? Wood?).



**Figure 6: Components of a (very) simple house**

### **Technique 2: Fragment/synthesize**

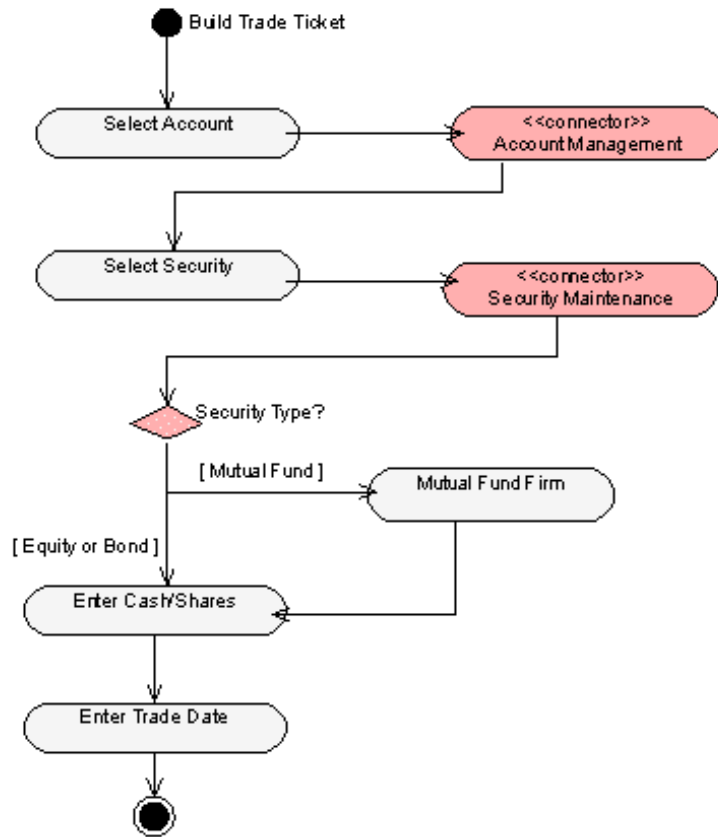
The second technique uses the well-known strategy of "divide and conquer." By *fragmenting* the problem under study into component parts, each piece of the problem can be investigated independently prior to *synthesis* back into the full model. Someone creating a software program would not attempt to solve the entire problem at once, but would instead create smaller program elements (e.g., objects) that would perform relatively simple but well-defined tasks. These elements can then be joined together to create robust functionality. As shown in Figure 6, the elements of our simplified house have been broken up into a series of components that can then be considered independently of the others.

### **Technique 3: Obscure/emphasize**

With this technique, the goal is to create a model that will selectively draw attention either toward or away from selected system elements. By using the principle of *obscuring* or hiding elements, the modeler insulates the viewer from non-essential elements. By *emphasizing* other elements, the modeler can draw the viewer's attention to exactly the system elements that the modeler is interested in discussing. This technique can often be seen in artistic presentations or heard in musical compositions. In music, for example, one instrument may have emphasis because it carries the melody, while the other instruments carry the rhythm or underlying musical structure (e.g., the Baroque idea of *continuo*). In software development models, this principle is at work when a class diagram omits the attributes and operation of every object in the system and shows only the attributes and operations relevant to the current activity. As in Figure 6, the details of each component are obscured so that the dependencies between components can be emphasized.

#### **Technique 4: Chaining**

In the creation of models it is often necessary to segment the model into a collection of specialized views. This is a direct result of the application of the other three techniques; the system under study is, for example, fragmented, abstracted, and emphasized. In order to connect the various views or to lead the model reviewer from one section of the model to other logically connected sections, it is necessary to *chain* elements of the model. This kind of logical connection is frequently seen in software development models, since software is a highly complex system with many interacting parts. One form of this kind of linkage is shown in Figure 7, which depicts the creation of a trade ticket (shown in activity diagram form) that requires the selection of an account and a security. For both of these activities, the model shows an activity that is stereotyped to «**connector**»; this represents the linkage to another diagram that describes the steps for the selection of an account and security, respectively. In fact, the diagram suggests that there are at least three types of security that can be selected: an equity stock, bond, or mutual fund.



**Figure 7: Activity diagram for creation of a trade ticket**

By creating chaining elements in the model, it is possible to lead the viewer to exactly the area of the model that you desire -- a form of *emphasis*. Chaining also allows for the indication of parts of a system that are dependent on other parts, a key feature of any system analysis.

## Conclusion

Arguably, anyone can create a model; but creating a truly effective model is hard work. As mentioned previously, we continuously create models of the world we experience. Our difficult goal is to communicate those models to others in a way that enhances their understanding.

The techniques and frameworks described in this article are one path toward achieving this goal. In the next article we will delve deeper into the techniques of model capture, including examples of the different forms models take, and the types of reasoning that can be employed when creating a physical model from a mental one. Finally, in the third and final article for this series, we will look at visual model presentation and discuss concepts of color, balance, composition, and form.

## References

C.Alexander, *A Timeless Way of Building*. Christopher Alexander, 1979.

T. H.Athey, *Systematic Systems Approach*. Prentice-Hall, 1982.

H.Lasswell, "The structure and function of communications in society," in *The Communication of Ideas*. Harper and Brothers, 1948.

T. Mandel, *The Elements of User Interface Design*. Wiley Computer Publishing, 1997.

D.Meyhew, *Principals and Guidelines to Software Interface Design*. Prentice-Hall, 1992.

G. Miller, "The magical number seven plus or minus two: some limits on our capacity for processing information." *Psychological Review* 63 (1956): 81-97.

J.Morgan and P. Welton, *See What I Mean?* Edward Arnold, 1992.

V.Satir, *The New People Making*. Science and Behavior Books, 1988.

W.Schramm, *The Process and Effects of Mass Communications*. University of Illinois Press, 1954.

---

## Notes

<sup>1</sup> J.Morgan and P. Welton, *See What I Mean?* Edward Arnold, 1992, and T. Mandel, *The Elements of User Interface Design*. Wiley Computer Publishing, 1997.

<sup>2</sup>W.Schramm, *The Process and Effects of Mass Communications*. University of Illinois Press, 1954.

<sup>3</sup> G. Miller, "The magical number seven plus or minus two: some limits on our capacity for processing information." *Psychological Review* 63 (1956): 81-97.

<sup>4</sup> It is interesting to note that this is a form of *indirect reasoning*, because the conclusion is reached by inferring the mechanism from the experimental result. We will investigate various reasoning methods in the second article of this series.

<sup>5</sup> Please do not confuse conceptual and physical models used in this context with the Rational Unified Process definition, based on the 4+1 architectural views proposed by P. Krutchen. Here the term is meant to express differences between a mental model and the constructed physical model.



---

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