

Developing and Using an Integrated Collection of Rich Real-Life Analogies and Metaphors in the Teaching of Complex IT Concepts

Thang N. Nguyen
tnnguyen@csulb.edu
California State University Long Beach
Long Beach, California 90840, USA

Abstract

Common IT (information technology) concepts/notions are generally complex and rather difficult for new students to grasp. These include, for example, (a) OOP (object-oriented programming) notions of *abstraction*, *encapsulation*, *inheritance* and *polymorphism* or (b) those associated with ISO/OSI (International Standards Organization/Open System Interconnect) reference model such as the notions of *protocols*, *services*, and *standards*. The basic concepts of OOP in example (a), if well presented, will facilitate the understanding of other relevant concepts, e.g. class versus instance methods/data, method overloading and overriding, use of foundation classes and interfaces as well as OO program design. Similarly, the basic concepts of ISO/OSI in example (b) should be well understood before the introduction of other related concepts such as internetworking with circuit-switching or packet-switching, forward and reverse address resolution, hop, time-to-live, communications gateways, message-oriented middleware, security public/private key or socket programming. It is a real challenge for instructors to adequately convey the complex IT concepts to students, especially to those majored in Information Systems and/or other non-Computer Science disciplines. Quite often, the instructors use fragmented, piecemeal and ad hoc analogies, and refer to isolated metaphors drawn from numerous and disparate sources, in the teaching of such complex concepts. This paper presents a simple and practical method to developing and using an integrated collection of rich real-life analogies and organized metaphors for better conveying and teaching complex IT concepts. For illustration of the method, the paper uses two examples: the first with emphasis on the development of an *analogous model*, and the second focusing on its *metaphorical usage*. The model in the first example is developed for the teaching of internetworking concepts and facilities and is labeled as a human communication model, analogous to the OSI model. The second shows the use of an extended version of the analogous model for the explanation of basic OOP programming concepts and capabilities using their real-life metaphor counterparts. Discussions and concluding remarks follow.

Keywords: Analogies, analogous modeling, metaphors in IT education, OOP concepts

1. INTRODUCTION

We use analogies and metaphors in casual conversations and in other forms of communication everyday. One of the many reasons for using analogies and/or metaphors is to enhance the explanation and understanding of the concepts or notions to

be communicated. We use (1) an *analogy* i.e. part of A resembles B, or (2) a *metaphor* i.e. some A is B, so that A is better understood since the meaning of B is more obvious or already established (Lofting 2003). An analogous example may be "... life is like a box of chocolates, ..." (Forrest Gump movie 1994) and a metaphorical example

may be the recycle bin on the Windows desktop for storing files removed from the computer, awaiting to be emptied, restored or recycled much like the recycle trash can.

The teaching of Information Systems or Information Technology in particular and/or Computer Science in general involves the introduction of complex concepts, principles, models, facilities, algorithms, processes, products, and applications. These are commonly very hard for students, especially business students or other non-Computer Science students, to grasp, retain, describe, understand, let alone explain them adequately. The instructors explore the use of analogies and metaphors in their teaching of the said discipline or of the teaching task itself (Bowman, 2003). The analogies and metaphors used can be spontaneously created (in classroom) or part of a planned lesson. Commonly, the instructors use the analogies and metaphors drawn from numerous, independent sources of information to explain complex concepts. The analogies and metaphors frequently cited in many textbooks or online tutorials are however piecemeal, fragmented and are of limited scope (Feig, 1994).

This paper, via examples, illustrates a simple development method and use of an *integrated collection of human-related and rich real-life analogies and metaphors* for the teaching of complex Information Technology concepts and implementation. The first example emphasizes the development of an *analogous model* called *the human communication model* for the teaching of internetworking concepts and facilities. The second focuses on the *metaphorical usage* on the model to illustrate the basic OOP concepts and capabilities by using their real-life metaphor counterparts.

2. EXAMPLE 1: DEVELOP AN INTEGRATED SET OF REAL-LIFE ANALOGIES AND METAPHORS FOR THE TEACHING OF OSI NETWORKING CONCEPTS AND FACILITIES

The right-hand side of Figure 1 shows a typical OSI layered organization that is commonly found in many textbooks or online tutorials on an introductory internetworking topic (e.g. Comer 1995,

Murphy et al. 1995; Tang et al. 1992). As an example for illustration, we use the following definitions found at www.whatis.com, searchword=OSI:

“Layer 1: *The physical layer*...This layer conveys the bit stream through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier.

Layer 2: *The data-link layer*...This layer provides synchronization for the physical level and does bit-stuffing for strings of 1's in excess of 5. It furnishes transmission protocol knowledge and management.

.....

Layer 4: *The transport layer*...This layer manages the end-to-end control (for example, determining whether all packets have arrived) and error-checking. It ensures complete data transfer.

.....

Layer 7: *The application layer*...This is the layer at which communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. (This layer is not the application itself, although some applications may perform application layer functions.)”

The above definitions for the most part, appear dry and either too abstract (e.g. transmission protocol knowledge and management in the definition of layer 2) or too detailed (e.g. bit-stuffing for strings of 1's in excess of 5). They refer to other concepts and/or terminologies and therefore require additional explanation and clarification, such as bit stream and carrier (in the definition of Layer 1), bit-stuffing, transmission protocol knowledge and management (Layer 2), end-to-end control, packets and error-checking (Layer 4), and quality of service, authentication and constraints on data (Layer 7), to say the least. Furthermore, it is not obvious from the above definitions, that the seven architected layers underline the three basic notions of protocols, services and standards which are fundamental and crucial in internetworking.

To explain the basic notions, the instructors will have to offer more details and/or a set of examples to help visualize the layered organization and associated concepts. The instructors would explain that the physical

layer consists of the cable and the connector inserted to the NIC card (network interface card) at the back of the computer. The link layer is the logic encoded on the NIC card that interprets the binary stream

sent/received, and so forth. To cover all the layers, this explanation effort may take hours of instruction to illustrate relevant concepts.

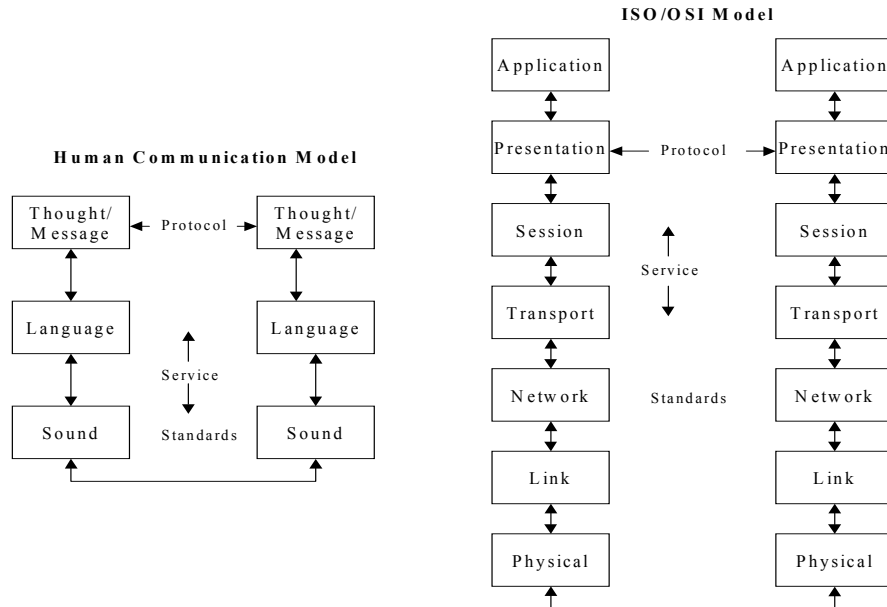


Figure 1: Real-life analogies and metaphors to explain OSI layers

In this section, we show a simple 3-step method of developing a model as an integrated collection of real-life analogies and metaphors for the teaching of complex IT concepts. The model called *human communication model*, considered as an analogous model of OSI, is rich enough to facilitate the understanding of basic OSI concepts, namely protocols, services and standards, and relevant issues surrounding them as described below.

Step 1: Sketching a simple, analogous model

When a person (sender) has a thought or a message to communicate to another (receiver) - left-side Figure 1, the thought or message from the sender can be translated into a statement or a series of statements using a selected language such as English. The English statements or sentences are then converted into sound spoken by the sender. The converted sound travels through a physical media such as air to reach the intended receiver, another person. At the

receiving end, the sound received, assuming that it experiences no distortion, is converted back to the same language used by the sender, and subsequently, results in the thought or message being heard, translated and understood (Human Communication Model, Figure 1).

Three observations can be drawn from this human communication model example. First, at the language layer of the human communication model, it is expected that the *same language* is used between the sender and the receiver, thus the notion of *protocol*. Sender and receiver must have the same protocol. If the language (e.g. Greek) at the sending side of the human communication model is different from that (e.g. Chinese) at the receiving side, the communication fails because the sending message will not be understood by the receiver.

Second, the lower layer, e.g. sound, is to translate the English statements into audible *sound*, hence the notion of *service*. The

lower layer (sound) in fact services the immediately upper layer (language). If the language is translated into a text format recorded in some media other than air (e.g. paper) by the sender while the receiver expects sound rather than written text, the communications also fails.

And of course, finally, the language used has to follow certain grammatical rules in a particular language's constructs to be meaningful to the receiver, thus the notion of *standards*. The same standards must be applied to both sender and receiver, else the communication fails.

If we use the human communication model of Figure 1 as analogy and the real-life metaphors (human thoughts, human sound, air, etc.) to explain ISO/OSI model (right-side, Figure 1), we will find that the three notions, protocols, services and standards defining the basic key concepts of OSI internetworking are more easily understood as further explained below.

In fact, the three concepts defined in the human communication model are applicable to explain a simple client/server model (Figure 1, right side) under OSI. The physical layer (i.e. the telephone plug/communication cord and the modem or the NIC - network interface card - in the back of a computer, e.g. the Ethernet or Token ring card) is to *service* the link layer immediately above it, *much like the sound layer servicing the language layer* in the human communication model. The link layer (in which an encoded firmware e.g. the Ethernet - IEEE 802.2 - protocol logic to translate the hardware address to the internet address, among other things) in turn services the network layer. The network layer then services the transport layer, and so forth (session layer and presentation layer) all the way up to the application layer.

The same layer level at each side (client or server) must have the same protocols (e.g. http - hypertext transfer protocol - at the client requiring http at the server) for possible communication between them *just like the same language must be used* in the human communication model to allow one human at one end to understand the other human at the other end. If different protocols are used, e.g. FTP - file transfer

protocol - is used at the client while Telnet is expected at the server (equivalently, different languages in the human communication model), the message will not be understood.

The simple human communication model can also be used to easily explain other concepts such as *synchronous* or *asynchronous* messaging. In fact, if sound and the air media are used then the receiving party must be present to receive the sound via the air, hence synchronous. Analogously, a server using OSI reference model must be active (i.e. present) for the synchronous client/server communication to be possible.

On the other hand, if the sound is recorded for playback (e.g. message left on voice-mail) or if a text message is used (such as e-mail) instead, then the receiving party does not need to be there at the time the message (as a recorded message or a written text) is sent. The message can be saved some place, allowing the receiver to retrieve the message any time later. The communication is therefore asynchronous. This analogy applies well for OSI synchronous and asynchronous messaging.

Although the analogy appears to work well using the human communication model to explain concepts of OSI in step 1 (this step), the simple human communication model as shown is not adequate to convey additional concepts such as routers, time to live (TTL) or other important concepts in internetworking. Some concepts such as hardware address or internet address have no equivalent counterparts based on the previous human communication model. Thus, the human communication model must be extended (step 2) and detailed out (step 3) in a fashion that is deemed to be adequate to explain other relevant concepts of the OSI model.

Step 2: Extending the analogous model (the breadth of the model)

We extend the human communication model in this step to include the fact that humans commonly live in different and distant houses. Human communication needs be carried out with the communication means (such as mail, telephone, etc.) other than

just face-to-face conversation (via sound) as shown in the previous model. The houses in which humans reside are analogous to computer hardware systems in which applications run under a particular operating system (e.g. Microsoft XP). The address of the physical lot as shown in the house blueprint, on which the house is built is analogous to the Ethernet (physical) 48-bit address associated with the computer, which is universally and uniquely defined. Then, the universally unique street address of the house is like the universally unique 32-bit IP address. For mails to be received, the street address has to be placed on the envelope containing the message from a human sender to a human receiver, much like the 20-byte message header found in the transport layer (TCP/UDP) datagrams. The post-offices can be thought of as analogous to the routers. The type of mail delivery by USPS is labeled connectionless and unreliable just like UDP, as opposed to the connection-oriented and reliable transport by telephone that is similar to TCP.

The crucial concept of routing or gateway providing all interconnections among physical networks (not individual machines) is similar to the concept of post offices providing the interconnection to all physical groups of housing and establishments identified by zip codes (not individual humans). Similarly the concept of multiplexing using ports associated with application processes is analogous to the distribution of incoming mails to different humans residing at the same street address.

At this point, we wonder whether the human communication model can be used to deal with OSI implementation's technical details. We may ask for example how the information processing can be explained at the physical layer with the network interface card (NIC) using frames (sent/received). In step 3, we detail the model to cover the technical depth as shown below.

Step 3: Detailing the analogous model (the depth of the model)

The physical layer interface can be explained using the analogy of human hearing anatomy that was taught to students during their high-school days. For example, the sound that arrives at the outer ear of a

human causes vibrations of the tympanic membranes which produce movements of the middle-ear ossicles which in turn press against a membrane. Pressure waves generated from the membrane cause the bending of the sensory hair cell. This stimulates AP (action potentials) in the sensory fibers. The AP is transmitted to the brain. The receiver recognizes it as sound. This process involves many structures from the ear to the brain (Alberts et al., 1998). It also involves many physical processes that translate signals from one form to another (Fox, 1996).

This is in essence parallel to the modulation and demodulation process that converts analog sound signals from a telephone line to digital signals (sending) and vice versa (receiving). The latter is interpreted by the logic of the firmware (similar to its counterpart portion of the brain) on the NIC card. The process involves many components in the NIC card for the appropriate translation of information (Comer, 1995).

When the words of the intended message arrive at the brain, they are accepted by a process that produces as output an understanding of the words. This process involves a look-up of the meaning of the words in a particular natural language used. If the information carried by the thought or message reaches the receiver and is understood, that is because it is expressed in terms of nerve impulses in the receiver's brain (Fox, 1996). In our human communication model, between the word that is formulated by the language and the sound that is spoken, there are also many different forms of signals that are used to carry the information, one of which is the mechanism of cell signaling. At this interface, a particular type of molecule is produced by the sending cells and detected by the receiving cells by mean of receptor proteins, a process that is analogous to a IPC (inter-process communication such as named pipes, RPC-remote procedure call-and the like) in the operating system supporting the corresponding layers of OSI.

The whole process from sound-to-thought transformation is analogous to the transformation of digital frames at the physical layer and link layer to packets at

the network layer, to datagrams and segments at the transport layer, and to messages at higher OSI layers.

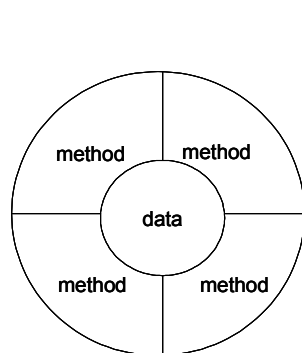
In summary, the scheme of sound-to-language, language-to-message, message-to-thought transformation, can be explained using the simple model as in step 1, expanded to cover a variety of related concepts as in step 2 and can include technical details as rich as those described in step 3. Using the analogy between human communication model and the OSI reference model, the basic OSI concepts, facilities and implementation can be explained such that they become easier to understand. In addition, the human communication model is quite robust. It can be further expanded to parallel the communication between multiple humans (n-tier client/server), from one person (one client) to many persons (multiple clients, multiple servers), one person addresses a group leader (hub-and-spoke, bus or star architecture), and the like. With the human communication model, we can equally introduce the notion of sensation (message initiated by the senses) as analogous to event monitoring. This robustness allows students to think beyond the analogy, possibly to conceptualize new ideas to IT.

In summary, the development and use of an integrated set of real-life analogies and metaphors help the students better understand the complex IT concepts as they relate to prior learning and experiences with real-life facts and processes. The analogy also intrigue the students with more insights into IT schemes.

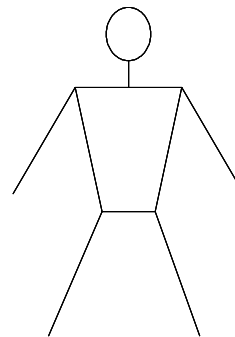
3. EXAMPLE 2: USING AN INTEGRATED SET OF REAL-LIFE ANALOGIES/METAPHORS IN THE TEACHING OF OOP CONCEPTS

The model developed in example 1 (section 2) can also be used for explaining OOP concepts as argued in this section.

There are four basic object-oriented concepts (Morin, 2002) that every OOP textbook addresses: *abstraction, encapsulation, inheritance* and *polymorphism*. These concepts are very difficult to understand, even for students already OO programming courses such as C++ or Java and require a great deal of effort by students who are previously experienced with traditional, conventional programming such as COBOL, PL/I, Pascal or C.



OO class and object



Thang is an object that belongs to the class of Person.
 A person has name and other attributes whose values are data.
 A person can do many actions such as run, speak, reveal... The actions are called methods.
 No one will know the person's name unless s/he reveals it. So data are hidden, and methods and data are encapsulated in the class

Figure 2: Human analogy and use of metaphors for explaining OO encapsulation concept

Take for example, the concept of OO *encapsulation* (Chalk 2003, Sun 2003), a concept that is rooted in the *information hiding* concept (Parnas 1972). As commonly shown in textbooks, to illustrate encapsulation/information hiding, some authors draw a circle, place data inside an inner circle and methods between the two circles and say something along the line that data are hidden and inaccessible. The picture (left side of Figure 2) suggests that the data can only be accessed by using methods.

But the drawing is not really clear. Most students would accept the argument implied by the picture but there are a lot of why's and a lot of what-if's in their mind. Why the data are hidden? Why not the methods? What happens if the data are not? What difference does it make?

The analogous model can be used to explain the information hiding concept and the encapsulation concept. To start, one might ask the students to think of real-life "things" as objects. For example, a particular person is an object. The author of this article is an object. He belongs to a collection of persons – class of persons (hence, notion of class and object). A person has a name, and other attributes such as height, weight, intention, thoughts, the amount of money in his pocket and many other attributes. The values of the attributes are data. For example, the author's first name is Thang. His height is 5'5". As a person, he also can do a lot of different things. He can write. He can speak. He can run. He can communicate with another person in English. The actions he can do are called methods, his methods.

Normally, no one will know his name if he does not use one of his methods, such as *speak* or *write* method to reveal his name. No one will know how much money he has in his pocket if he does not use his own hands to reach the pocket, gets the money out, counts them and then exposes the amount (equivalent to a *get* method). Therefore, the data are commonly hidden or kept private. Only the methods that are public can place (*set*) and/or retrieve (*get*) or manipulate the data, thus, the notion of information hiding. Both data and methods are encapsulated within the notion of object class. The circle containing data and methods in Figure 2

now makes sense. The human analogy and metaphors serve well to explain the concept of information hiding/encapsulation, and other issues such as the implementation of the encapsulation concept in Windows.

In fact, users of Window (e.g. 95, 98, NT, 2000, XP) see the Window desktop every time they power-on their computer, but not many realize that the OO encapsulation concept is realized by this icon. Little attention is paid to the meaning of the icon *My Computer* on the desktop. It represents and maps a physical object (the local computer itself) to the desktop. It means that anything a user wants to know about or to do with this local computer is underneath this icon. Although the users right-click the icon all the times, not many know that by right-clicking the icon, they will obtain a popup showing a list of the methods that the object represented by the icon is able to perform and that the properties of the object represented by the icon are also included (data and methods encapsulated in the object). Very few suspect that this is an implementation of the OO encapsulation concept.

Other concepts such as public, private, or protected can also be explained using the same real-life metaphors. For example, information about the author such as his face is public, anyone can see it. Information about his SSN is protected since he only gives it to IRS, his employer or other selected people he can trust. Information on his opinion about someone may be classified as private. He will not give the information to anyone if not asked, and there is no way anyone would know it, if he does not answer.

The concept of abstraction (generalization, specialization and aggregation), inheritance and polymorphism are also easily explained using the extended human communication model of real-life metaphors/analogies (section 2). Using the same analogy of class (person) and object (Thang), one may say that a person may have a child (i.e. notion of subclass). The person is now the parent of the child (i.e. notion of superclass). The child may have some traits (phenotypes) that are inherited from the father (i.e. notion of inheritance), from the mother or from other relatives (i.e. notion of multiple inheritance).

The child may have his/her own attributes and capabilities different from those of the parents (i.e. notion of specialization). Parent, children and the house they live in, collectively give rise to the notion of family (i.e. generalization or aggregation). Both the father and his son can eat food, but they do so differently: the father uses the chopsticks, the son uses the fork and knife, thus the notion of polymorphism (poly = many, morphism = form). The father can lift a heavy object (method) with both arms/hands, the son can do the same with only one arm/hand (i.e. overloading method). Both father and son can present the same topic but the reasoning and arguments used by the father may be different from those by the son (i.e. overriding method).

In this section, we have shown a formulation of metaphors based on the extended analogous model for use on explaining complex OOP concepts previously. We may expand further the analogous human model to address concepts beyond those OSI and OOP basic concepts cited in the previous sections. In fact, the natural/biological life and OO programming have a lot of similarities as frequently argued by numerous authors. For example, OO foundation classes (such as Microsoft MFC) where all programs are derived are analogous to genome and genes are frequently equated to OO classes (Nguyen, 2003). Cells are considered analogous to executable programs (Ray, 1997). The set of integrated real-life analogy and metaphors can be used to explain many other complex IT concepts and can be used to get insights into IT research.

4. DISCUSSION AND CONCLUDING REMARKS

The two examples have shown the formulation of an integrated model of real-life analogy that can be developed via the three steps as shown in section 2. The model called human communication model, helps explain many IT concepts and body of knowledge (e.g. OSI and OOP concepts) as presented in section 3.

The collection of real-life analogies and metaphors as illustrated can be used during the first few sessions of an MIS course.

During the last semester (Spring 2003), I have used the said model in one of my classes as pilot. The subject of discussion no longer appeared "dry". The classroom has been fun. The student's participation was amazing. The feedbacks were also amazing. The students reported that they understood better and were capable to explain the concepts to fellow students. When they read the textbooks, the analogies and metaphors helped them to understand even more. The average student grades went up. The student evaluations scores on my course went from an average of 4.25/5 to 4.52/5 for the pilot course taught using the analogous and metaphorical model.

The model of real-life analogies and metaphors as exemplified in the previous sections can be used to address other topics such as OO program design, the concept of message queuing, message exchange and message handling in e-business multi-tier client/server applications, the notion of communication brokers, integration brokers, gateways, transactional systems, data base systems, and the like. The analogies and metaphors can range from those used to explain simple DOS command such as ping (asking if anyone is alive) to networking concepts and infrastructure, and to IT enterprise architectures (analogies for explaining Zachman's model). Researchers have also used many analogies and metaphors for insights into their research. For example, human brain and memory are metaphors used to formulate Von Neumann's computer model (von Neumann, 1958). Genetic algorithms (Holland 1975) as well as genetic programming (Koza 1992) are patterned after genetic metaphors (mutation, cross-over) and natural selection. Artificial Life has been studied by Langton (1989) at Santa Fe Institute and other researchers at MIT Alife lab with insights gained from observations on natural life, and vice versa. Many other biologically-inspired research and studies have surfaced (Hu, 2002) in many different disciplines, which were due largely to biological metaphors.

With the explosion of the Internet, its availability and its increasing use, the growing number of portals, e-commerce and e-business applications, and the uncountable new IT products supporting them, most instructors have a hard time to keep track

the new information, new concepts, new facilities, new techniques, and new acronyms in IT, in their IT teaching. The method presented in this paper offers a simple and practical method for the development and use of an integrated collection of real-life analogies and metaphors for the ease of teaching complex IT concepts and facilities.

5. REFERENCES

- Alberts, Bruce, Dennis Bray, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts and Peter Walter (1998), Essential Cell Biology, Garland Publishing.
- Bowman, Mary Ann. (2003), Metaphors We Teach By: Understand ourselves as teachers and learners, Essays in Teaching Excellence, Center for Teaching Excellence, <http://www.cte.umd.edu/podresourcepackets/definingteaching/metaphorts.html>
- Chalk, Peter. (2003). Encapsulation, online. <http://homepages.unl.ac.uk/~chalk/proj/oootutor/encapsulation.html>
- Comer, Douglas E. (1995). Internetworking with TCP/IP, 3rd edition, Volume I, Principles, Protocols and Architectures. Prentice-Hall
- Comer, Douglass E. and David L. Stevens. (1994). Internetworking with TCP/IP, Volume II, Design, Implementation and internals, Prentice-Hall
- Comer, Douglass E. and David L. Stevens. (1994a). Internetworking with TCP/IP. Volume III. Client/Server Programming and Application, Prentice-Hall
- Feig, Rami. (1994). The OSI Reference Model, (<http://www2.rad.com/networks/1994/osi/intro.htm>).
- Forrest Gump. (1994). Robert Zemeckis, director, starring Tom Hanks and Sally Field.
- Fox, Stuart Ira (1996). Human Physiology. Wm.C. Brown Publishers.
- Holland, John H. (1975). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence, University of Michigan Press
- Hu, Jianjun. (2002). WWW Resources on Biologically-Inspired Research, online, www.egr.msu.edu/~hujaaju/Biologically_inspired.htm
- Koza, John. (1992). Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press
- Langton, C. G. (1989). Artificial Life, in Artificial Life (Santa Fe Institute Studies in the Sciences of Complexity, Langton, ed. Addison-Wesley
- Lofting, Chris. (no date). Analogy versus Metaphor, online <http://members.ozemail.com.au/~ddiamond/analog.html>
- Morin, Randy C. (2002). OOP Concepts by Examples, online, <http://www.kbcafe.com/articles/OOP.Concepts.pdf>
- Murphy, Eamon, Steve Hayes and Mathias Enders. (1995). TCP/IP Tutorial and Technical Overview, 5th edition. Prentice-Hall
- Nguyen, Thang N. (2003). An Embryonic Approach to Object-Oriented Program Development, Proceedings of 34th Conference of Decision Sciences Institute, Washington D.C., Nov 2003
- Parnas, David L. (1972). On the criteria to be used in decomposing systems into modules, CACM, December, 1972
- Ray, Thomas (1997). An Evolutionary Approach to Synthetic Biology, Zen and The Art of Creating Life. Artificial Life 1(1), Langton C. (ed.), MIT Press
- Sun Microsystems (2003). The Java Tutorial: OOP concepts, <http://java.sun.com/docs/books/tutorial/java/concepts/class.html>
- Tang, Adrian and Sophia Scoggins. (1992). Open Networking with OSI. Prentice-Hall
- von Bertalanffy. (1968). General System Theory, Braziller
- von Neumann. (1958). The Computer and The Brain, Yale University Press