Part Four: A Nontechnical Introduction to DICOM[1]

Steven C. Horii, MD

■ INTRODUCTION

The initial goal in developing a standard for the transmission of digital images was to enable users to retrieve images and associated information from digital imaging equipment in a standard format that would be the same across multiple manufacturers. The first result was the American College of Radiology (ACR)-National Electrical Manufacturers' Association (NEMA) standard, which specified a point-to-point connection. However, the rapid evolution of computer networking and of picture archiving and communication systems meant that this point-to-point standard would be of limited use. Consequently, a major effort was undertaken to redesign the ACR-NEMA standard by taking into account existing standards for networks and current concepts in the handling of information on such networks. The Digital Imaging and Communications in Medicine (DICOM) standard was the result of this effort. Its popularity has made discussion, if not implementation, of the standard common whenever digital imaging systems are specified or purchased.

The DICOM standard is extremely adaptable, a planned feature that has led to the adoption of DICOM by other specialties that generate images (eg, pathology, endoscopy, dentistry). The fact that many of the medical imaging equipment manufacturers are global corporations has sparked considerable international interest in DICOM. The European standards organization, the Comité Européen de Normalisation, uses DICOM as the basis for the fully compatible MEDICOM standard. In Japan, the Japanese Industry Association of Radiation Apparatus and the Medical Information Systems Development Center have adopted the portions of DICOM that pertain to exchange of images on removable media and are considering DICOM for future versions of the Medical Image Processing Standard. The DICOM standard is now being maintained and extended by an international, multispecialty committee.

The DICOM standard has become the predominant standard for the communication of medical images. However, even though the standard is widely available from manufacturers and is rapidly expanding to include nonradiologic imaging, most radiologists' understanding of it is limited. In part, this is because DICOM has a "steep learning curve" and most introductory material has been written either for the engineer and is highly technical, or for the administrator and is rather superficial.

Why all the interest in what would seem to be a simple task? The answer is that it is not as simple as it first appears. Most radiologists are familiar primarily with film images, and film can be viewed anywhere there is a light source. It is the transition from film images to digital images and the need to communicate, display, and store these images that has made DICOM necessary. With film, slight differences in exposure, processing, and viewing will have little effect in these areas. In digital imaging, however, the difference of a few bytes can make it impossible to transfer an image from one system to another.

The DICOM standard consists of multiple documents (1); as of this writing, there are 13 published parts. Each DICOM document is identified by title and standard number, which takes the form "PS 3.X-YYYY," where "X" is commonly called the part number and "YYYY" is the year of publication. For example, DICOM Part 2 has a title of "Conformance" and document number PS 3.2-1996. In informal usage, the year is often dropped.

In this article, a basic, nontechnical introduction to DICOM is presented that will enable the reader to understand the basic concepts and principles used in the standard. A number of key terms and their definitions are listed in the Appendix at the end of this article.

Abbreviations: ACR-NEMA = American College of Radiology-National Electrical Manufacturers' Association, DICOM = Digital Imaging and Communications in Medicine, DIMSE = DICOM message service element, E-R = entity-relationship, ISO-OSI = International Standards Organization Open Systems Interconnection, SNOMED = Systematized Nomenclature for Medicine, SOP = service-object pair, UID = unique identifier, VR = value representation

Index terms: Computers • Digital Imaging and Communications in Medicine (DICOM) • Education

RadioGraphics 1997; 17:1297-1309

[1]From the Department of Radiology, University of Pennsylvania Medical Center, 3400 Spruce St, Philadelphia, PA 19104. Received July 10, 1997; accepted July 16. **Address reprint requests to** the author.

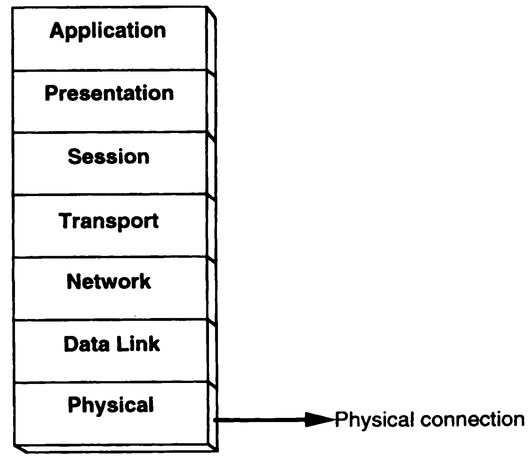

Figure 1. Diagram illustrates a layered communication model, the seven-layer ISO-OSI Reference Model. The top (*Application*) layer is the one that interacts with the user or the user's software. The bottom (*Physical*) layer supports the medium (eg, coaxial cable, fiber-optic cable). In between are layers that handle such tasks as encoding, breaking large data streams into smaller packets, detecting and correcting errors, and controlling the flow of data over the medium. Two such stacks, one implemented in each device, are connected at the physical layer to provide device-to-device communication.

■ COMMUNICATION

The essence of the DICOM standard is that it prescribes a uniform, well-understood set of rules for the communication of digital images. For the purposes of this article, communication is defined as the interchange of information. This sounds simple enough; indeed, it is something we do every day. However, the reason we are able to communicate successfully is that we follow a well-established set of rules that we have, for the most part, mastered from childhood.

Electronic communication is commonly thought of as being divisible into a set of *layers* with each layer performing a defined set of functions (2,3). This model of communication as a set of layers is part of an international standard for communication called the International Standards Organization Open Systems Interconnection (ISO-OSI) Reference Model (Fig 1). The model may be understood from an analogy to the structure of a manufacturing company. At the highest layer of electronic communication, the system interfaces to the user's application (eg, a computer terminal or a personal computer running a particular program that accesses data over a network). This layer corresponds to the planners and decision makers in the company who determine what product will be manufactured and shipped. At the lowest layer of electronic communication is the physical medium (eg, the "wire") over which the information will be sent and received. This layer corresponds to the trucks used by the shipping department. The designation of these layers as "higher" and "lower" does not imply levels of importance; rather, it is used because the layers are viewed as a *stack* with the physical layer on the bottom and the application layer on top.

In between the topmost layer and the physical layer are other layers that deal with matters such as what character set will be used to represent information, how to establish the rules for making connections over the physical medium, and how to handle any errors that might occur in the communication process. These intervening layers can be thought of as different departments in our hypothetical company, each performing specific tasks (eg, parts selection, manufacturing, quality control, planning shipping routes). In both the electronic communication model and the company, each layer (or department) accepts input from the layer above (preceding department), performs a well-defined set of functions, and provides output to the layer below (succeeding department).

The exchanges between layers or departments can go both ways, because most communication (as in the exchange of goods between companies) is bidirectional. In communications terminology, the movement of information between layers uses *services* that are provided by the layer. However, communication means the exchange of information; therefore, there must be a corresponding set of layers in another device to which the physical medium is connected.

Our hypothetical company manufactures and ships its goods to another company that incorporates them into its products. This second company does not ship products back to the first company (unless they are the wrong product or are defective) but sends back payment for the goods it has received. Expediting de-

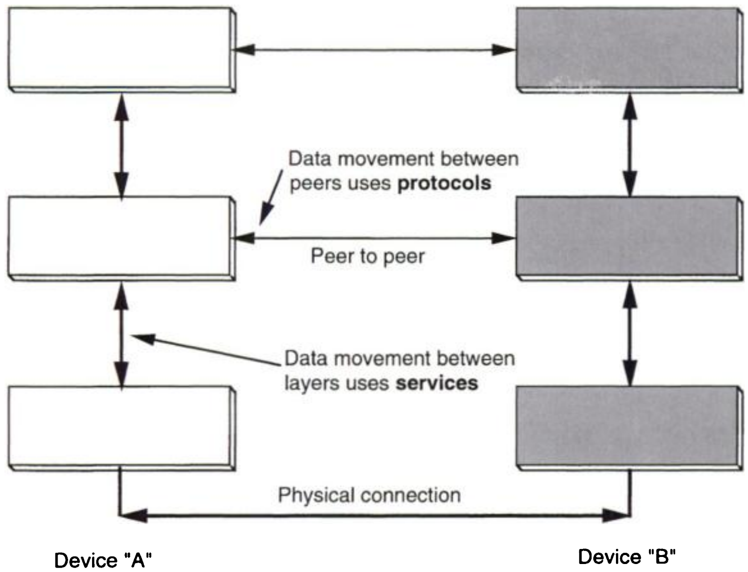

Figure 2. Diagram shows two simplified communication protocol stacks of three layers each. Vertical communication (within a stack) uses services, whereas horizontal communication (stack to stack) uses protocols. Stack-to-stack communication is physical only at the lowest layer. Peer-to-peer communication (ie, communication between corresponding layers in different stacks) occurs when information is moved down one stack, over the physical connection, and up to the corresponding layer in the other stack.

partments in the two companies may handle scheduling of shipments and may establish rules for shipping (eg, not shipping perishable goods over a long holiday weekend). Similarly, communication between two electronic devices uses rules, or *protocols*, that establish how corresponding layers in the two devices interact. In electronic communication, the actual movement of data between two devices occurs only at the physical layer. However, because of the way layered communication works, each layer can be viewed as communicating with the corresponding layer in the other device by means of the established protocols. A layer adds some information to what is going to be communicated and, using the services of the layer, sends it down to the layer below where a similar process may take place. On the receiving side, the information works its way up through the layers with each layer removing the information it needs to perform its functions (Fig 2). In our hypothetical company, the quality control department might add stamps or tags to the product if it meets the quality standards and passes it on to shipping, where it is packaged and placed on a truck. In the second company, the product is unpacked and sent to the quality control (or incoming inspection) department, where the tags or stamps are checked and the product may undergo testing to ensure it has not been damaged in transit. If the product passes inspection, it is sent on for further processing, and final acceptance triggers the accounts payable operation. The protocols used in this process ensure that the shipper has placed appropriate quality control tags on the product so that the receiver can check them against its acceptance documents. If protocol is violated (eg, the shipper does not inspect the product or leaves off the tags), receipt of the product will fail (ie, the receiver will refuse the shipment or may hold it pending resolution of the problem).

One advantage of layered systems in electronic communication is that a layer may be replaced by a newer layer without affecting the other layers. For example, if an alternate physical medium with a transmission speed 10 times greater than that of the existing physical medium becomes available and exactly matches the services and protocols of the existing layer, it may be substituted for the existing layer. If our hypothetical company found a common carrier that delivered freight 50% faster to the same locations, used the same size freight containers, and provided the same insurance coverage as the existing carrier, it could switch carriers without having to modify other operations.

These collections of layers are often called stacks or protocol stacks. The DICOM standard makes use of a layered communications structure and, in fact, is designed to use existing standard communications stacks rather than define its own. One may ask, If DICOM uses existing standards for communication, what then does it do? Why not just use the communications standards as they exist?

■ A DICOM INTERCHANGE

What DICOM does and the importance of its functions are best illustrated by a hypothetical transaction between two manufacturing companies (1,4).

Suppose that you, as the director of product development, decide that you need a dozen widgets to create a prototype product. You tell Betty, your engineering purchasing manager, to buy the widgets from the Acme Widget Company. Betty knows that the quickest way to get them would be to call Acme, but she also knows that the widget you want is popular and may be out of stock. She picks up the telephone, calls Acme, and hears the usual greeting: "Hello, this is Acme Widget; how may I direct your call?" Betty replies, "Please connect me with Bob Roberts in sales." The receptionist puts through Betty's call.

What Betty has done so far is much like what the device using DICOM does when it asks to communicate with another device. When she picks up the telephone, she is requesting that the line be opened to her. The dial tone she hears tells her that the line is available (no dial tone would indicate a problem with the line or that it was in use). Similarly, DICOM uses services to request communication with another device over the network. The network protocol will indicate that the network is either busy or available. If it is available, DICOM initiates a series of actions that request what is called an *association* with the other device.

By simply dialing (ie, using either a push-button or dial telephone) the desired number and listening for an answer, you have accomplished a surprising amount of communication protocol. First, you follow certain rules in dialing the number. If you are calling from within an office, it is likely that you must first dial a number to request a line outside your local switchboard. Then, you dial a "1" plus an area code if the number you are calling is not in your area code, or you leave those digits out if the number is local. When you hear the telephone on the other end ring, you know the line is not busy. If it is busy, your "protocol" will likely be to hang up and try again later.

When the person on the other end answers, the first few words tell you quite a bit. The two things of most importance to you are, first, that the person is speaking English, and second, that you have reached the correct number. If you hear, "Moshi moshi, Fujiyama Denki desu," you might either hang up or ask if you are talking to Acme Widget. In this case, some negotiation is taking place; you want to know if you have the correct number and if the person speaks English instead of Japanese. You are presuming that the person will recognize your language as English. The person's reply will determine what you ask next.

DICOM does some initial negotiating to establish the association mostly by determining what it is the requesting device wants to do and what the receiving device is capable of doing. With DICOM, it is not the device itself, but the software running on the device that does much of this negotiation. DICOM refers to the devices as application entities because it is the application layer (the uppermost layer in the communications stack) that initiates the communication process.

Just as your initial "negotiations" determine basic capabilities (eg, the language being used, the identity of the other person), so the association establishment in DICOM involves negotiating capability. The application entity requesting the association sends what amounts to a list of things it wants to do. The receiving application entity then replies with which items on that list it can do. Subsequent exchanges are based on the capabilities that the two entities have in common. An example is the way in which digital pixel values larger than 8 bits (1 byte) are represented. Some computers represent 2-byte (16-bit) numbers with the least significant byte (ie, the one representing the least significant digits of the number) being stored or sent first ("little endian"). Other computers do just the opposite ("big endian"). If two devices using the opposite representations exchange numeric data (eg, pixel data), the values will be incorrectly represented unless the devices know this and can effect conversion. DICOM allows for both methods of representation, so that one matter for negotiation is which method the devices will use during the exchange of information.

Another difference among imaging devices is the manner in which they represent values. As will be discussed later, DICOM breaks the information it needs to send into a series of *data elements.* Applications may have different ways of representing the value contained in any given data element. DICOM has very specific definitions of the different types of representations allowed; these are called *value representations* (VR). Examples include text strings of differing maximum lengths (a text string is a collection of successive characters with a starting and ending character, much like a sentence), binary numeric data, time and date data, and person name data. In the earlier ACR-NEMA standard, these VRs were defined in a section of the standard called a data dictionary. If you were writing software that received ACR-NEMA information and you wanted to understand the meaning of the data elements you were receiving, you would use the data dictionary to look up the ACR-NEMA element *tag* (which identifies the element) to find out what data were in the element and how the element was represented. With the DICOM standard, new VR types were added, and any future additions will increase the size and complexity of the data dictionary. This is one reason why DICOM defines both an ACR-NEMA method and a new method for handling the VR of a data element. The new method defined by DICOM includes the VR in the element, thus avoiding any possible ambiguity. For software designers, this method reduces the need to refer to the data dictionary. The ACR-NEMA method of defining the VR in the data dictionary is called *implicit VR.* The new method used in DICOM, in which the VR is included in the data element, is called *explicit VR.* DICOM allows both methods, and this is important to know as abilities are being negotiated.

Both the VR method and the order of the bytes are part of a set of information that is crucial to successful information exchange. This information set is called a *transfer syntax* and is defined in DICOM. Which transfer syntax is to be used for exchange of information is negotiated very early in the association process. Thus, when people familiar with DICOM speak of "implicit VR, little endian," they are referring to one of the DICOM transfer syntaxes.

After Betty is connected with Bob in the sales department at Acme, they exchange some pleasantries and then Bob asks her what she needs. Betty replies, "I need 15 widgets." Bob thinks for a second, tells her he is checking inventory, then says, "Yes, I have them in stock." Then he adds, "You know, Betty, we have just released the Mark II Widget. It is 10% faster than the Mark I Widget for the same price." Betty asks, "What about the interface?" Bob replies, "The Mark II is a drop-in replacement for the Mark I. We will be phasing the Mark I out as we use up inventory." Betty says, "Sounds good; I'll take the 15 widgets as Mark IIs, then."

Although these transactions seem straightforward, it is because the process is one that is very familiar. If Betty did not know Bob in sales, for example, she might simply have asked for the sales department.

What is happening in this scenario is something that happens so many times during a typical day that we tend to ignore it unless something goes wrong. In this example, the definitions of terms and the model of the way things work are held in common by you, Betty, and Bob at Acme. This is true in virtually all of our communication: We work from common definitions and models, and if the terms or model are unfamiliar, we either ask for clarification or we run the risk of encountering problems. In this case, you (product development manager) asked for a dozen widgets, but Betty (engineering purchasing manager) ordered 15 of them. Betty is working from the historical information she has that, in your development projects, you invariably forget about the three widgets that will be used up in safety compliance testing and need to order more. Betty knows enough about your prototype project design to ask whether the Mark II Widget has the same interface as the Mark I. She also knows that the design would benefit from an increase in widget speed, so she sees no reason not to switch to the Mark II. There is an even higher level model invoked in this scenario. Some engineers would be wary of using a just-released product for fear that it is likely to cause more problems than a more time-tested product. However, in this scenario, both you and Betty know that Acme has such good quality assurance that the likelihood of encountering any problem with the Mark II Widget compared with the Mark I is negligible.

The most important functions that DICOM performs are to define as unambiguously as possible the terms it uses and to define models of image communication that are agreed on by those who adopt the standard. At a very high level of communication (that between users), it is important to agree on terminology. Both within radiology and between radiology and other specialties, use of unambiguous terms is vital if actions taken on the basis of the communication are to be correct. A difference of opinion about what constitutes "medial" and "lateral," for example, could have disastrous effects if surgery is undertaken on the basis of the radiology report. The College of American Pathologists has for some years been working

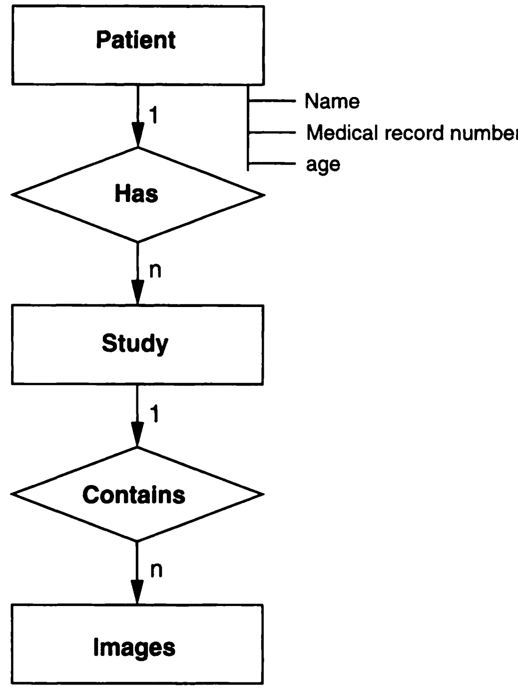

Figure 3. Simple E-R model shows the relationship between a patient and the images obtained of that patient. The entities are represented by rectangular boxes and the relationships by diamond-shaped boxes. The small characters next to the arrows indicate that one patient may have *n* studies and that one study may contain *n* images. The arrows do not indicate data movement; rather, they are included to avoid ambiguity or nonsensical relationships (eg, it would make no sense to say that an image contains a study). Only three patient attributes—name, medical record number, and age—are shown. If the number of attributes is large, they may not be included in the E-R diagram.

on the Systematized Nomenclature for Medicine (SNOMED). This nomenclature proposes standard terms, also called a controlled vocabulary, for anatomic structures and pathologic conditions. The College of American Pathologists is now a DICOM Standards Committee member and is working with the committee to develop the sections of SNOMED that will address imaging (5). These sections are referred to as the SNOMED-DICOM Microglossary.

DICOM has also adopted conventions from other standards where appropriate, such as the person name format proposed by the Health Level 7 standards body. This format clarifies how names are represented and is of value when there are prefixes (eg, Dr, Fr) and suffixes (eg, Jr, II; academic degrees) along with the name. The format divides a name into components such as "family name," "given name," and "prefixes," and each component is single or multiple to account for unusual constructions like multiple middle names or suffixes.

DICOM extends its definitions beyond terms, measurement values, conditions, and the like. It also defines the model of how these things, or entities, relate to each other. For example, how is a patient related to a study done on that patient? A simple relationship in this case is expressed by the word "has": A patient *has* a study. In turn, the study itself *contains* images. This process can be extended so that, through careful examination of a clinical operation, a model can be built that includes all the entities in the operation (eg, patients, studies, images, reports) and the relationships between them. This process is called entity-relationship (E-R) modeling. It is important to note that the resulting model, usually a diagram, does not depict the direction in which data move (Fig 3).

Each of the entities in an E-R model has other descriptors, or *attributes*. For example, a patient might be described in terms of name, age, sex, and medical record number, and perhaps height and weight. All of these are attributes of the patient; that is, they each carry some identifying or descriptive information about that particular person. DICOM defines not only E-R models for imaging but also the attributes that describe each entity. In fact, the data elements of the ACR-NEMA standard are the attributes used in DICOM. Figure 4 shows the structure of a DICOM attribute.

The process of developing E-R models and identifying the attributes that describe the entities is part of an information analysis and modeling method called *object-oriented analysis.* This technique has caught on rapidly in com-

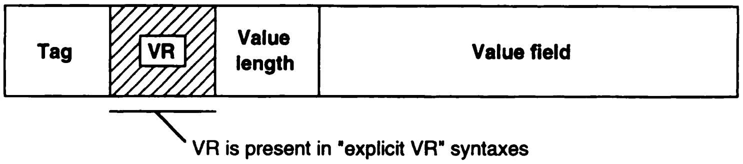

Figure 4. Diagram illustrates a DICOM attribute (data element). The attribute includes a tag that identifies the element and consists of two 16-bit integers. The first integer is derived from the old ACR-NEMA group number and, for DICOM standard attributes, is always an even number. The second integer is also based on ACR-NEMA practice. The two numbers together form the attribute tag and are usually shown in parentheses—for example, (0010, 0020), which happens to be a patient identification number. The second field (VR) is present only if the negotiated syntax includes DICOM explicit VR. It describes how the value field to follow is represented. The length field is a 16- or 32-bit field (16 bits if VR is explicit because the VR field takes up 16 bits itself) that gives the length of the value field that follows. In some DICOM-defined instances, the length may not be known; in such cases, the length field is filled with a representation that tells the system that the length is not known by the application that generated the attribute. The advantage of this attribute structure is that it allows for variable-length values.

puter science disciplines until now it is nearly impossible to study in any branch of information science without encountering object-oriented methods. Part of the goal of creating E-R models is to achieve an object-oriented result; the entities are objects and the attributes describe them.

DICOM adopted the object-oriented approach as part of the design philosophy. Things such as images, reports, and patients are all objects in DICOM and are called *information objects* because their function is to carry information. The definition of what constitutes an information object in DICOM is called an information object definition, which is nothing more than a list of which attributes must be present (mandatory attributes), which are optional, and which are conditional (ie, must be present only in some situations). There is a subtle distinction that is important to make when describing information objects. The information object definition can be thought of as a "form" with a number of "blanks" to be filled in with information. Each piece of information is an attribute, such as patient name or medical record number. Even if the form is not filled in, the various blanks on the form give the information some structure. When the blanks are filled in, values are given to the attributes so the form is no longer generic; it applies to a specific patient, image, or other type of object. This filling-in process, or assigning of values to attributes, creates what is called an *information object instance.*

The ACR-NEMA Version 1 and 2 standards did not use object-oriented analysis or design. Instead, attributes (or elements, as they were called) were grouped according to use. For example, there were groups of elements that carried identifying information about the patient and others consisting of elements that described the methods of image acquisition. Because they were developed without an E-R model, these groups do not conform to conventional object-oriented definitions. For example, a collection of elements used in the ACR-NEMA Version 2 standard to identify and describe a computed tomographic (CT) image would also contain the patient name. In an E-R model, however, the patient name is an attribute of the patient object, not of the image object. In other words, the patient name is not needed to *describe* the CT image, even though it would be needed to *identify* the image. One might also view these complex objects as consisting of parts of more than one entity in an E-R model.

The problem for the DICOM developers (at that time, still the ACR and the NEMA) was that they wanted to maintain some compatibility with the earlier versions of the standard. In addition, there were some computer scientists who believed that objects that "broke the rules" by containing attributes not strictly inherent in that object had some value. For one thing, retrieving such complex objects from storage would be more efficient: If the objects were broken up into smaller ones, assembling the smaller objects for use would mean searching the storage device for all of them. Such a task would be more time-consuming than recovering the complex object in a single retrieval. Consequently, these complex data structures were retained in DICOM as *composite objects.* New objects that were defined by the E-R models of DICOM followed object-oriented design rules and did not contain attributes that were not inherent in the object. Such objects in DICOM are called ***normalized objects.*** The image objects (eg, CT, ultrasound [US], magnetic resonance [MR] imaging) are all composite; the objects that are used for image and results management are normalized.

At the hypothetical company of which you are product development manager, an important aspect of quality assurance is keeping track of what parts are incorporated into your products. That way, if a problem with your product is discovered either in your testing laboratories or by your customers, you can find out what parts went into that product and determine if the problem is a design flaw or a faulty part. To facilitate this process, parts would be assigned part numbers in your products. This would enable you to identify a part that failed, but it would not help you figure out which manufacturing lot it came from. In other words, the part number identifier you assign may be unique in a product line, but it would not be unique across multiple versions of that product. For example, the widgets that Betty ordered might be assigned part number 2011030-001 on the basis of your design drawings. However, all the products you make containing that widget will reference that part number. To identify a specific part, you need a ***unique identifier*** (UID). If you know that Acme assigns unique serial numbers, for example, you might use them to uniquely identify a particular widget in a particular product. Only one product should contain the widget with part number 2011030-001 and serial number 97-2003, so any subsequent follow-up with Acme necessitated by product failure will be made simpler.

Just as certain elements in industry are assigned numbers for unique identification, so the images, reports, or other information that are transmitted from one device to another in medical image communication must be identified in a unique fashion. DICOM uses UIDs to identify information objects in this way. The form of the UID conforms to an international standard, and, if properly applied, provides an identifier that is unique not only within an institution but worldwide. UIDs are designed mainly for computer software interpretation; as a result, their form is a bit cumbersome. The UID is used by DICOM whenever one thing is referenced by another. For example, the transfer syntaxes described earlier have UIDs so that the different machines using them can refer to a particular transfer syntax by its UID. As part of the international standardization process, the committee responsible for the DICOM standard applied for, and was granted, a numeric field to use as part of any UID that DICOM defines. This numeric field is called the organizational root; for DICOM, it is 1.2.840.10008. To this organizational root are appended additional numeric fields. For example, the UID for the DICOM explicit VR little endian transfer syntax is 1.2.840.10008.1.2.1. The organizational root will differ depending on whether the UID was assigned by a manufacturer or a user. The periods separating the numbers make these numeric fields look as though they should have some particular meaning, but they do not. A UID exists to give a unique identity to something, not to carry information about the thing it identifies.

■ DICOM SERVICES

The information objects of DICOM are used to communicate the various images and related data between hardware components. However, such information alone is not sufficient to ensure proper operation. When Betty called Acme, she not only communicated information about what was needed but also initiated a series of actions. Betty said she wanted to order 15 widgets. Placing that order (which would likely involve a written purchase order after the telephone call) would start the process of picking the widgets from Acme's stock, moving them to the shipping department, and packing and shipping them. Along the way, there would surely be paper or electronic forms filled out to track progress and inventory changes. In effect, the communication involved a set of services provided by various departments at Acme. If the widgets were to reach your receiving department, it would set off a set of services provided by departments in your company. Thus, in addition to exchanging information (eg, "I want to order 15 widgets," "We have replaced the Mark I Widget with the Mark II Widget"), the communication involved a series of services needed to satisfy your company's request.

Communication of medical imaging information is similar. In addition to communicating with a device, you need that device to do something (eg, workstations display information, printers print it, archives store it). The negotiation process described earlier is a method whereby devices declare what it is they do. DICOM provides standardized services that are used with the information objects. These services are built on a set of elemental services. Because DICOM has both composite and normalized information objects, there are both composite and normalized services. Services are performed using a service element. In our example, a service such as filling a customer's order actually consists of a number of simpler actions or services. Your receptionist answers calls and would direct a call to the sales department if necessary. The salesperson writes up the order and sends a copy to your billing department. All of these are component services of the larger task of filling an order. Similarly, DICOM builds its more complex services out of a set of service elements that are called DICOM message service elements, or DIMSEs (pronounced *dim*-see). There are five DIMSEs (called DIMSE-C) that are used for composite information objects and six (called DIMSE-N) that are used for normalized information objects. These DIMSEs fall into the categories of operations (such as "store," which would cause data to be stored) and notifications (such as "event report," which would notify a device that something had taken place). These simple DIMSEs are used to build the services that most of us would expect in a picture archiving and communication system. These include services such as "storage," which causes information objects to be stored, and "query-retrieve," which causes a storage device to be queried and information retrieved from it. Some services, such as "storage," have a direct DIMSE counterpart, in which case the DICOM storage service would invoke the DIMSE "store" service. Other DICOM services such as query-retrieve require more than one DIMSE for implementation. The DICOM query-retrieve service makes use of the "find," "get," and "move" DIMSEs.

Because of the object-oriented nature of DICOM, services are actually referred to as *service classes.* In part, this is because a given service may be applied to a variety (or class) of information objects (eg, storage service class). It is also important to know whether a device *provides* a service (such as the disk file system on a workstation, which would provide the service of storing images) or *uses* a service (such as a US machine that would use the storage service on a workstation to store and eventually display images). DICOM refers to descriptions of this behavior as the service role. Devices may be service class providers, service class users, or both. Clearly, these roles need to be understood by the devices in a system if that system is to communicate and operate properly. This is also a part of the negotiation process discussed earlier: The list of capabilities includes not only which service classes are supported but in which roles they function.

■ THE ELEMENTAL UNIT OF DICOM

The information object and the service class are the two fundamental components of DICOM. An understanding of these components makes it possible to comprehend, at least at a functional level, what DICOM does and why it is so useful. Information objects define the core contents of medical imaging, and service classes define what to do with those contents.

The service classes and information objects are combined to form the functional units of DICOM. This combination is called a service-object pair, or SOP. Since DICOM is an object-oriented standard, the combination is actually called a *service-object pair class*, or SOP class. The SOP class is the elemental unit of DICOM;

everything that DICOM does when implemented is based on the use of SOP classes. Figure 5 shows an analogy between constructing a sentence and building a DICOM SOP class.

Combining a service and an information object is straightforward. For example, DICOM defines a series of storage SOP classes (eg, CT image storage SOP class, MR image storage SOP class). The CT information object definition and the storage service class are combined to form the CT image storage SOP class; other storage SOP classes are formed in a similar fashion. Because the SOP classes are referred to as a way to describe DICOM functionality, they carry UIDs. Furthermore, once the attributes in the information object and the variables of the service class are "filled in" by values representing a real patient, a particular piece of imaging equipment, and a resulting image, the SOP class becomes a SOP instance and is assigned its own UID.

The process of DICOM communication involves the exchange of SOP instances with use of DICOM messages. The DICOM message is the communication version of the SOP class; it contains the commands that use or provide the specified service and the *data set* made up of the properly encoded information object instance.

Verb: Store → **Service (DIMSE)**

Noun: CT Image → **Information Object Definition**

Generic Sentence: Store a CT image → **SOP Class**

Specific Sentence: Store *this* CT image → **SOP Instance**

Figure 5. Diagram shows an analogy between constructing a sentence and building a DICOM SOP class. Note the distinction between a SOP class and a SOP instance: In the latter, a specific image has been requested.

■ CONFORMANCE

A major issue for any standard is determining conformance. In many situations involving public health and safety, conformance to standards is required by law. For many other types of standards, including DICOM, conformance is voluntary. The DICOM Standards Committee does not have any enforcement authority. However, the DICOM standard includes a whole section devoted to conformance. Anyone claiming that their equipment or software conforms to the DICOM standard must be able to provide a conformance statement that describes exactly how that device or software conforms to the standard. A question that is frequently asked about the DICOM standard is, If it is a standard, why is a conformance statement required? Isn't it sufficient to simply state that equipment conforms? As has already been explained, DICOM can support many different types of images, transfer syntaxes, service roles, and the like. This flexibility is needed if DICOM is to be useful across many different medical imaging applications, but it also means that possibilities or options used in a particular implementation of the standard need to be made known. The conformance statement is a prescribed way of doing this.

A conformance statement follows a standard structure; use of a standard format makes comparison between such statements simpler. A user or manufacturer trying to determine if two DICOM devices will communicate to suit a particular application can compare the conformance statements side by side. This process does not guarantee that the two devices will communicate properly, but obvious problems, such as one device not supporting the service needed by the other, can be caught. The engineer familiar with DICOM should be able to ascertain the compatibility of the two applications.

The contents of the conformance statement include (*a*) the implementation model of the application; (*b*) the presentation contexts to be used; (*c*) the manner in which associations are to be handled; (*d*) the SOP classes to be supported; (*e*) the communication profiles to be used; and (*f*) any extension, specialization, or privatization to be used.

The implementation model of the application is a simple diagram that shows how the application is associated with both local (in the proposed equipment) and remote (across the DICOM interface) activities. For example, the local activity might be the creation of a DICOM image information object and the remote activity, the display of that object.

Presentation contexts consist of an abstract syntax (another term for the SOP class) and the transfer syntaxes used for that abstract syntax. The term abstract syntax is used in part because it is defined in one of the international standards that DICOM references. A DICOM conformance statement lists both presentation contexts that an application will propose during negotiation and those that it will accept.

The conformance statement must describe how an activity handles associations (ie, whether the activity initiates associations and accepts multiple associations) for each activity in the model. Some devices, such as the archives in a picture archiving and communication system, must support multiple associations if performance is to be acceptable. Otherwise, only a single activity (eg, DICOM storage) could be handled at any given time.

The list of SOP classes supported is one of the key aspects of the conformance statement. This list describes which service classes and information objects will be offered and accepted by the application. By understanding the SOP class, the reader of the conformance statement will be able to judge whether two conformance statements are describing applications that "match" (ie, if the presentation contexts [and therefore the SOP classes] offered by one application match those accepted by the other). If not, it is very unlikely that the two applications will operate together successfully, even if other aspects of conformance do match.

The communication profile used simply states which of the DICOM-supported communications stacks is used. Those available are the point-to-point, the ISO (an implementation of the ISO-OSI standards), and the transmission control protocol/Internet protocol (TCP/IP) stacks. Sections particular to the communications stack chosen are included.

The final part of a conformance statement details any extension to or specialization or privatization of SOP classes. Extension of an SOP class means the addition of standard attributes that are not mandatory but that the creator of the SOP class may use in a particular application. In effect, an extended SOP class is a superset of a standard SOP class, and most applications will not have difficulties handling it. A specialized SOP class has mandatory or conditional standard attributes added to it. A specialized SOP class may also contain private attributes that the creator of the SOP class considers mandatory. A private SOP class conforms to the structure of a standard SOP class but may contain completely private attributes. Neither specialized nor private SOP classes use the DICOM-defined UIDs, whereas the extended SOP class uses the UID of the standard SOP class on which it is based. Specialized and private SOP classes may cause problems for applications that try to interpret them unless the applications are designed to do so. The reason for such SOP classes is to allow manufacturers and other implementers to use the DICOM structure for their own purposes where there is no intent or need to communicate with these SOP classes outside their own equipment. The use of specialized and private SOP classes was never intended as a way to circumvent the DICOM standard. The pharmaceutical package insert familiar to physicians is analogous to the conformance statement. Just as the insert gives in standard format information such as the chemical structure, origin and nature, indicated uses, and adverse effects of the contents of the package, the DICOM conformance statement provides the engineering information about an implementation.

Because the conformance statement enumerates all that DICOM applications have to do or define, anyone wishing to gain a deeper understanding of DICOM would do well to read the part of the standard on conformance (PS 3.2) after reading the introduction (PS 3.1).

■ A DICOM WALK-THROUGH

As a way of summarizing, let us assume we want to send a CT study from a CT scanner to a workstation and follow how this would be accomplished in DICOM. You or the technologist would interact with software on the CT scanner to set up the particular study to transfer. The scanner might ask you to specify the workstation in addition to the study. This interaction is at the level of the medical imaging application, much of which takes place without DICOM. However, the scanner may store the study in its console in DICOM form, which means that interactions with the scanning application created the DICOM CT information object instances by assigning values entered (eg, patient name, medical record number) and those generated by the scanner itself (eg, date, time, institution, and scanner identification) to the images as they were generated. Thus, asking for a particular study to be sent to a workstation begins by requesting the name or network address of the workstation.

The communication protocol (most commonly TCP/IP) will handle the physical connection to the workstation from the CT scanner. By selecting the function that will send the study, the application software at the CT console begins assembling a series of SOP instances—in this case, CT image storage SOP instances. In our example, one instance will be created for each image to be sent. When the location of the workstation is entered, the DICOM software begins the communication process by requesting an association with the workstation. The communication network will handle the steps preceding this by setting up the communication channel. During the association, the CT scanner will provide its presentation context, telling the workstation that it supports the CT image storage SOP class as a service class user and that this particular application accepts only the verification service class (designed for testing) as a service class provider. The CT software also declares that its transfer syntax is implicit VR little endian. The workstation replies that it supports the CT image storage SOP class as a service class provider and also as a service class user. The workstation replies that it can use implicit VR little endian transfer syntax. Notifications of the acceptance of the association and capabilities are sent back to the respective devices.

The CT software sends the request for storage service along with the images to the software that assembles the DICOM message by putting together the necessary command and data set elements. Next, it sends the message down through the communication stack. The DIMSEs needed for the storage service class and the data set containing the image and other data are handled by the communication stack, where the message will usually be split up into smaller *packets* to be reassembled into the DICOM message in the communication stack of the workstation. The communication stack detects errors that might occur during transmission and is responsible for getting the packets to the proper location.

The reassembled DICOM message is received by the application layer in the workstation, which uses its software to store the CT images locally. The workstation may also send information (eg, patient name, study type, date of examination, number of images) to other software that will display this information in a worklist.

This lengthy-sounding process is usually accomplished quite rapidly. Typical end-to-end transmission time for CT images might range from less than 1 second to several seconds for each image, depending on the software and communication hardware used. Once the association is established, it can be used for multiple SOP instances, unless the SOP instance was not included in the presentation context list at negotiation. In our preceding illustration, however, the SOP instances containing the CT image object instances can be sent by using the same association once it is established.

■ CONCLUSIONS

DICOM is, of necessity, a complex standard. To make matters more difficult for those encountering it for the first time, it is written in the dry language required of standards and with a minimum of explanatory (called "informative" in the DICOM documents) information. Nonetheless, the standard has proved to be practical. The main purpose of this introductory article has been to demystify DICOM to some extent. The author begs forgiveness from those who are DICOM experts for what must surely seem to be oversimplifications and a glossing over of detail. However, the author certainly does not intend for any reader to go out and implement DICOM armed only with the knowledge contained in this article; such an undertaking requires an understanding of DICOM achieved only through a thorough study of the source documents.

For the reader whose interest or curiosity may have been piqued by this article, the author recommends some of the excellent resources available on the World Wide Web. The following are good starting points: *http://www.nema.org/* (NEMA's home page), *http://www.xray.hmc.psu.edu/* (Penn State University at Hershey), *http://dumccss.mc.duke.edu/standards/* (Duke University), and *http://www.merge.com/DICOM/* (a commercial vendor but with many links to DICOM resources).

■ APPENDIX

Association.—A communication connection established between two DICOM applications by which DICOM information is exchanged. A device may support one or more associations simultaneously.

Attributes.—Items that describe something. In object-oriented analysis, part of the process is determining what attributes of an entity are needed to describe or identify it. In DICOM, attributes are used to describe information objects.

Composite objects.—Those objects defined in DICOM that correspond to multiple entities or parts of multiple entities in the E-R model.

Data elements.—Instances (real-world value assigned) of the descriptive attributes that provide the characteristics of entities in the E-R model. The term *data element* is used to describe the contents of data sets.

Data set.—In DICOM, an information object in which real-world values have been provided for all attributes, which thus become data elements.

Entity-relationship (E-R) model.—A formal description of entities (eg, patients, equipment, images) and how they are related from an information organization perspective.

Explicit VR.—A method of including the VR of an attribute in the attribute itself.

Implicit VR.—The ACR-NEMA method of defining the VR of an attribute in the data dictionary.

Information object instance.—An information object to whose attributes real-world values have been assigned.

Information objects.—In DICOM, objects such as images, reports, and patients whose function is to carry information; entities in an E-R model whose descriptive attributes have been listed and defined.

Layer.—A set of software or hardware that performs specific functions needed for the communications process.

Normalized objects.—Those objects defined in DICOM that correspond to a single entity in the E-R model.

Object-oriented analysis.—The process of determining the object (or E-R) model that describes a particular activity.

Packet.—A small (usually) portion of a larger message that is being communicated. In addition to the message fragment, the packet has header information that allows it to be sent to the correct location and to be put in correct order should the multiple packets of a message arrive out of sequence. The packet also usually contains information that allows a communication system to determine if it got corrupted on the way to its destination.

Protocol.—The set of rules that allows two devices to communicate. In layered communications designs, peer-to-peer communication (wherein a layer in one device communicates with the corresponding layer in another) is described by protocols.

Service.—A set of functions performed to communicate between layers within a device.

Stack.—A set of layers designed to provide communications services to applications.

Tag.—In DICOM, the numeric name of an attribute or data element.

Transfer syntax.—In DICOM, a formal description of the manner in which the VR of data elements is presented and their encoding (byte order and compression type, if used) is performed.

Unique identifier.—A specific numeric construct used when an entity is referenced. It can be thought of as a unique name that will allow the desired entity to be found, retrieved, and distinguished from other entities.

Value representation (VR).—In DICOM, the description of how the attribute value is represented (eg, text, binary data, patient name).

■ REFERENCES

1. National Electrical Manufacturers' Association. Digital Imaging and Communications in Medicine (DICOM). Rosslyn, Va: NEMA, 1996; PS 3.1-1996-3.13-1996.
2. Tanenbaum AS. Computer networks. 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1988.
3. Helsall F. Data communications, computer networks, and open systems. 3rd ed. Wokingham, England: Addison-Wesley, 1992.
4. Bennett B, McIntyre J. Understanding DICOM 3.0. Dallas, Tex: Kodak Health Imaging Systems, 1993.
5. Bidgood WD Jr, Horii SC. Modular extension of the ACR-NEMA DICOM standard to support new diagnostic imaging modalities and services. J Digit Imaging 1996; 9:67-77.