Architecture
Re-Use
Communications

DICOM Cook Book
for Implementations in Modalities

Chapters 1 and 2

Version 1.1 (Accepted)
14 January 1997

Bas Revet

Internal Use Only

Document Number XPR080-970004.00

Philips
Medical
Systems

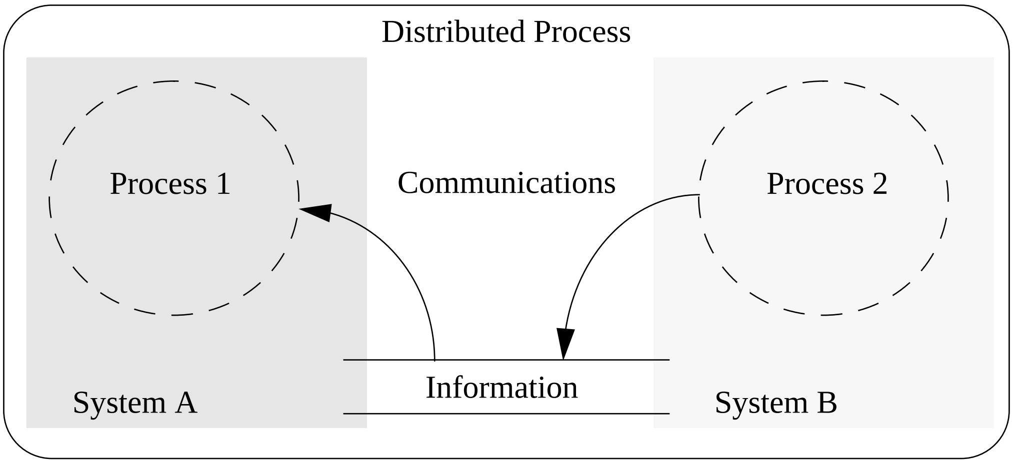
PHILIPS

1 DICOM Distributed Applications

In this chapter a number of concepts defined by the DICOM standard are explained. First a generic model of distributed processing is described as a baseline to introduce the DICOM concepts. The process parts which deal with information processing (Service Classes) and other issues are explained. In the two following sections the network and media exchange of information are described. Finally, features which ensure connectivity and an overview of the parts of the DICOM standard are given.

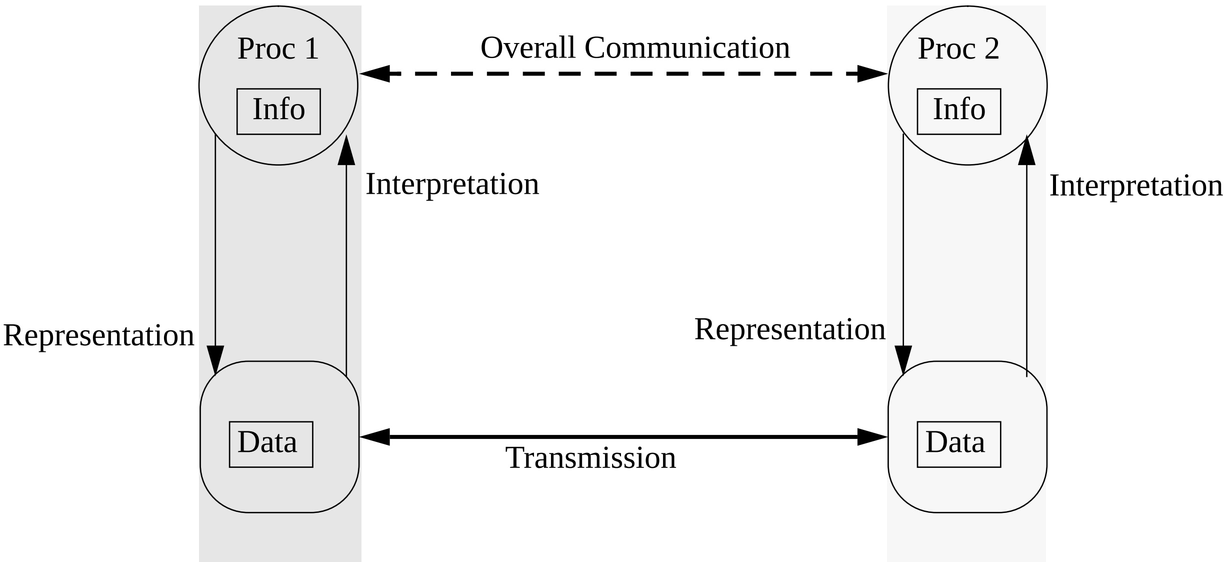
1.1 Distributed Processing

A simple model of distributed processing explains the mechanisms and terminology used in the DICOM standard.



A *distributed process* has at least two processes sharing information, each doing its own processing but relying on the functionality on each side. A number of distributed processes acting together provide *services* for systems in environments such as radiological departments. For instance modalities, archive and workstations provide services such as acquisition, storage and viewing of image data.

In most distributed process scenarios, the application processes are strongly decoupled from the communications processes which coordinate data transmission between systems and compensate for the different ways in which values are internally represented on different systems.



Before processes can act together a number of issues have to be addressed. They have to agree on the role each will play, have an equivalent view on the information and select the operations each side implements. In Figure 1-1 more details are shown.

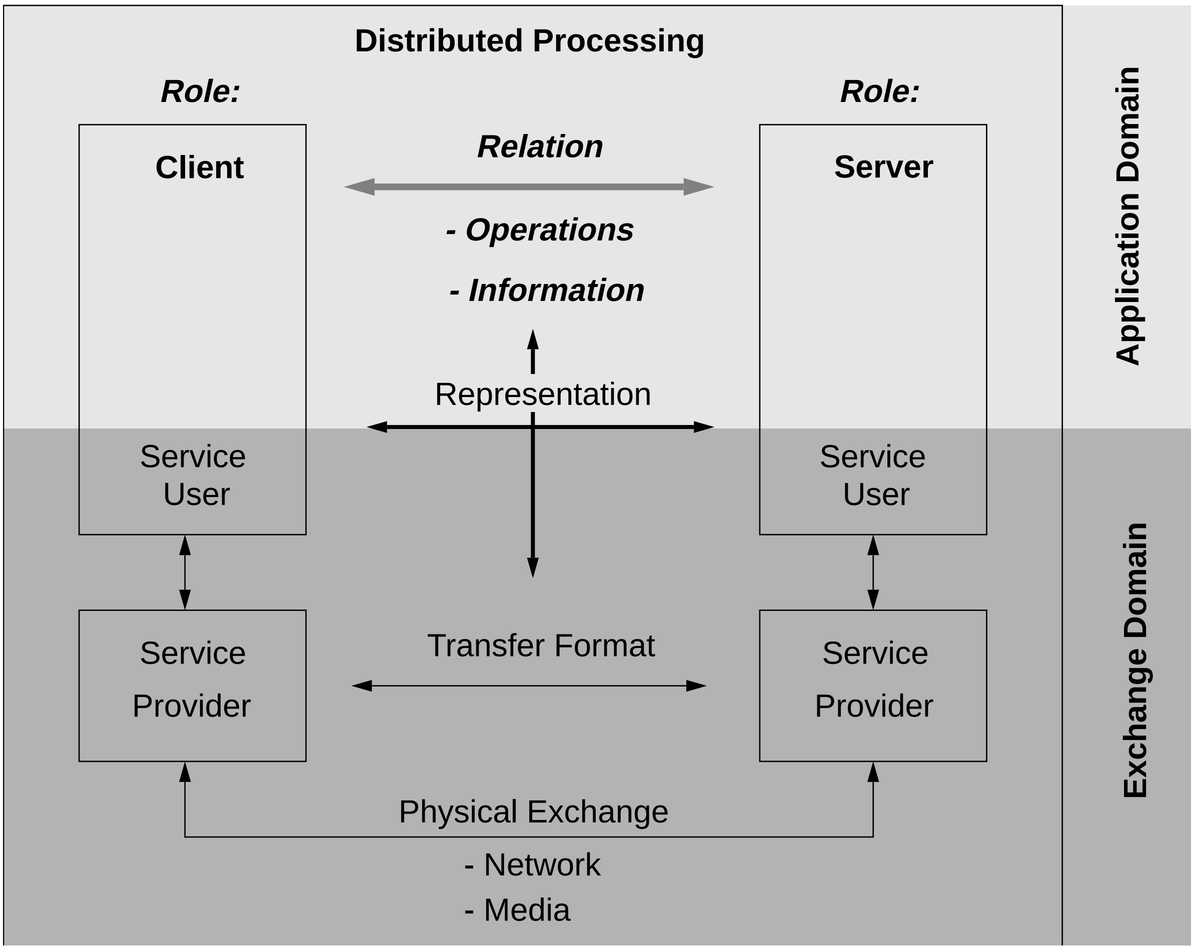


Figure 1-1: Model of *Distributed Processes*

First of all the *role* of each side has to be defined as client or server role. The side that uses the other sides functionality has the role of *client*. The opposite side acting upon an agreed model has the *server* role. The expectations both sides have of each other are defined in the *relationship* they share. The relationship defines which side under which condition takes the initiative in the process. In most cases the client triggers the process, but sometimes the server is the initiating partner.

Besides roles, both sides have to agree on the *information* they exchange. Here, the semantics of the information is considered, not the way it is represented (syntax). The information is defined by the *context* of the service the distributed process are implementing. Each individual process may have a selective view on this information, but this view must be consistent in the whole context.

The *operation* defines how exchanged information must be processed on the other side, such as storing the information, returning a result, etc.

The combination of context, relationship, operations and information is the corner stone of distributed processing and has to be determined before a successful implementation can be realized. All these issues are part of the *application domain* of the distributed processes. They do not deal with the way information is actually exchanged, but rely on lower level services (e.g., TCP/IP) provided in the *exchange domain* to cope with this exchange.

Both client and server parts must be able to issue requests to low level services. Low level services handle the exchange and are hidden for the application domain part of the client or server. The part requesting services is the *service user*. The counterpart is the *service provider*. Both sides may have different implementations, but they share the same knowledge about how to exchange data (protocol) and have the same logical interface (request format) between service provider and service user.

Both sides must determine how information is represented in bit/byte formats. The service provider must determine which *format* the information was transferred and convert it to the *representation* expected by the application domain. The representation is known between the service users and provider on each side and both the service providers. After an exchange, the information presented to the processes using the information is equal on both sides, regardless of how it was exchanged.

The *physical exchange* between service providers can be via *network* or *media* (e.g., DOR, Tape). Each mechanism has its own way of handling the knowledge of the representation.

1.2 General DICOM concepts

DICOM is a standard which partly covers the issues discussed in the previous section. This section discusses the generic concepts with respect to the actual exchange mechanism used. To keep this section in line with the preceding section, some of the concepts in the exchange domain are only relevant with respect to network exchange, and will not be found in media exchange part.

DICOM uses its own terminology to describe the context, relationship, etc. The first step is the same model of the distributed processing with a transformation of Figure 1-1 into Figure 1-2 by applying the equivalent DICOM terms.

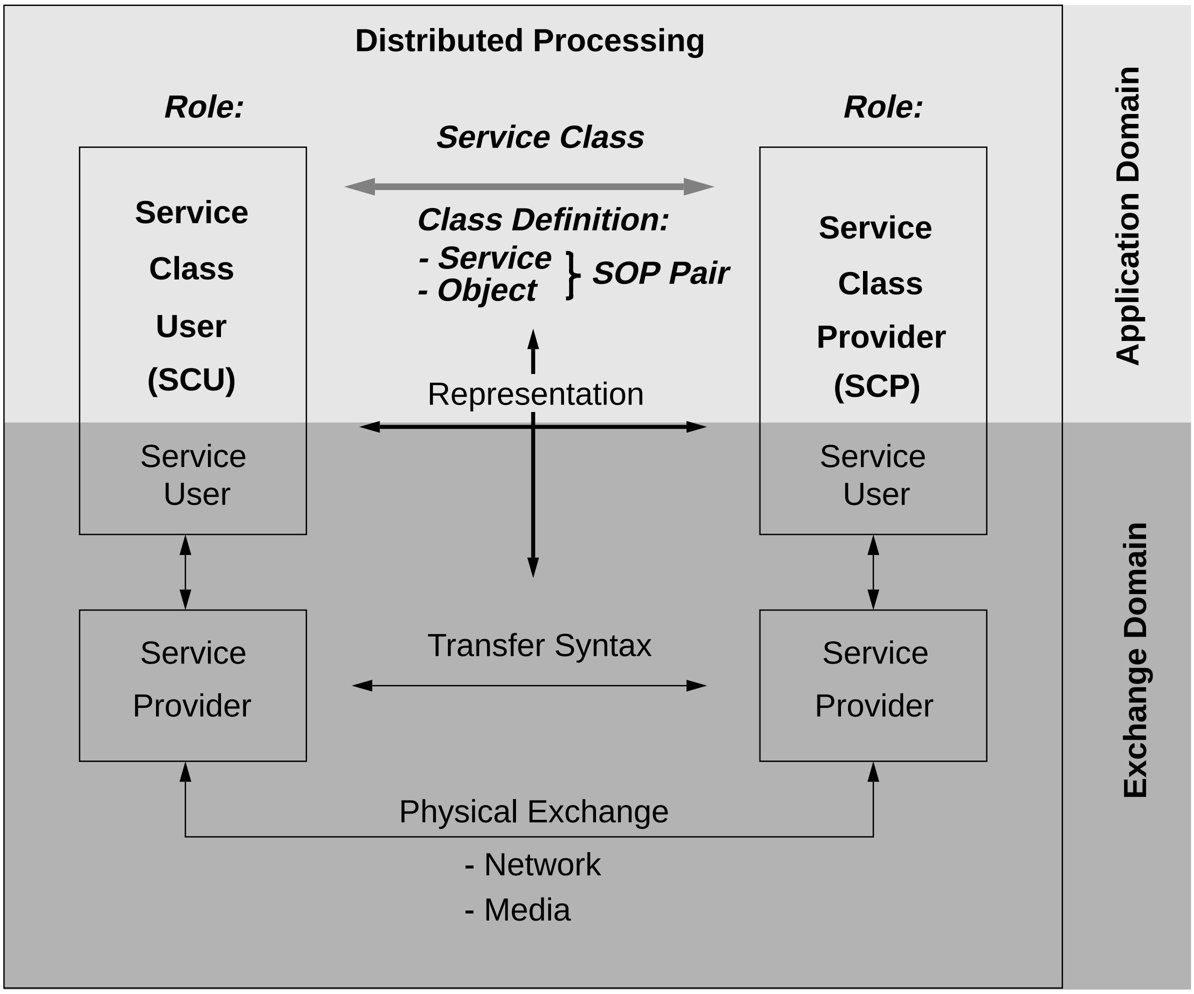


Figure 1-2: DICOM Service Classes

1.2.1 Service Classes and SOP classes

The relationship between both partners is defined by the *Service Class* description. The Service Class explicitly describes the roles both partners play. Depending upon the individual Service Class the context of the services is defined. With DICOM both roles are named: *Service Class*

User or *SCU* (client) and *Service Class Provider* or *SCP* (server). Do not confuse SCU and SCP with the service user and provider role in the exchange domain.

Part of the Service Class is the description of the information and operations. In DICOM these are combined with (object oriented) class definition, called a *Service Object Pair Class* or *SOP Class*. In each SOP Class definition a single *Information Object Definition* or *IOD* is combined with one or more *services*. For each of these services the details of the roles both partners have to play are fixed. More than one SOP Class can exist in a Service Class when more than one IOD is involved. A Service Class denotes the relationship of information defined in different IODs.

SOP classes identify the capabilities of the specific distributed processing for a certain Service Class. When partners agree to use a SOP class, both sides must ensure they will play their role as described, using the context of the enclosing Service Class. Before information exchange can take place the SOP Class identification is an important issue which has to be dealt with first. The mechanism used depends on the type of exchange: network or media.

Using the Service Class and other derived definitions, partners in a distributed processing environment function together via the services provided by the exchange domain.

1.2.2 Information Object Definitions

The information part of a SOP class is defined in IODs. An IOD is a collection of related

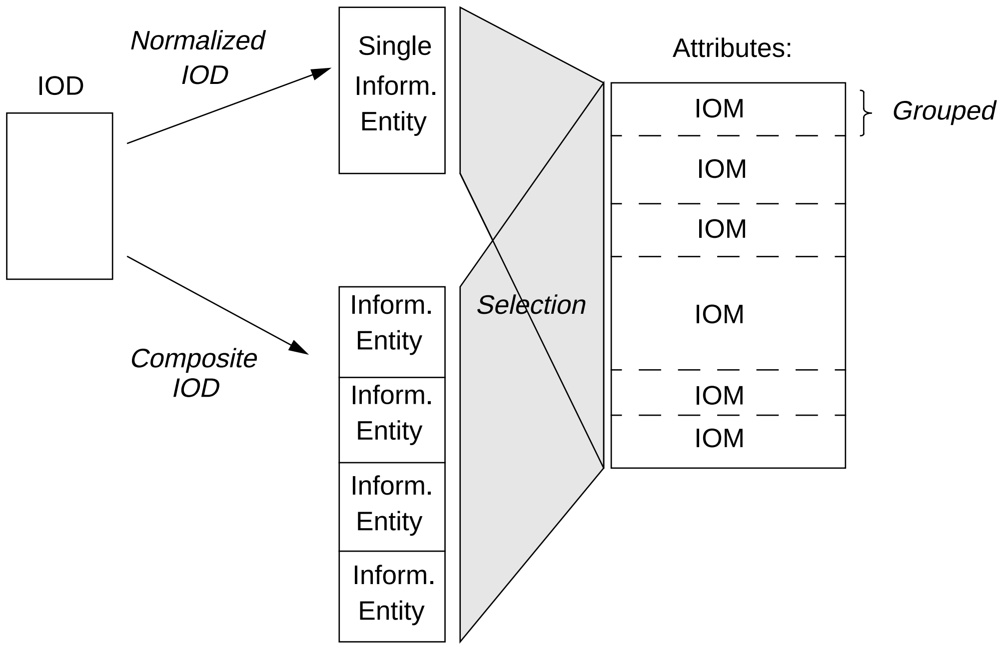


Figure 1-3: Relations IODs and attributes

pieces of information, grouped in *Information Entities*. Each entity contains information about a single (real world) item such as a patient, an image, etc. Depending of the context defined by the Service Class an IOD consists of one single information entity (called *normalized* IOD) or an combination of information entities (called *composite* IOD). Service classes which implement management functions (mostly single items) use normalized IODs, those handling the flow of image data (a complex structure of information) use composite IODs.

The relationship between different information entities (structuring) of composite IODs is described in the *information model* belonging to the service class. With normalized IODs (only one information entity) there is no need for structuring. *Relations* to other pieces of information are done by referring to that information.

Information entities consist of *attributes*, describing a single piece of information, e.g., a patient name. Attributes which have a relation are grouped in *information object modules* or *IOMs*. IOMs are defined in such a way that they can be used in more then one IOD. These IOMs also have the advantage that the semantic descriptions of related attributes can be grouped together. See the overview of relations in Figure 1-3.

As example of a composite IOD an image IOD is shown in Figure 1-4.

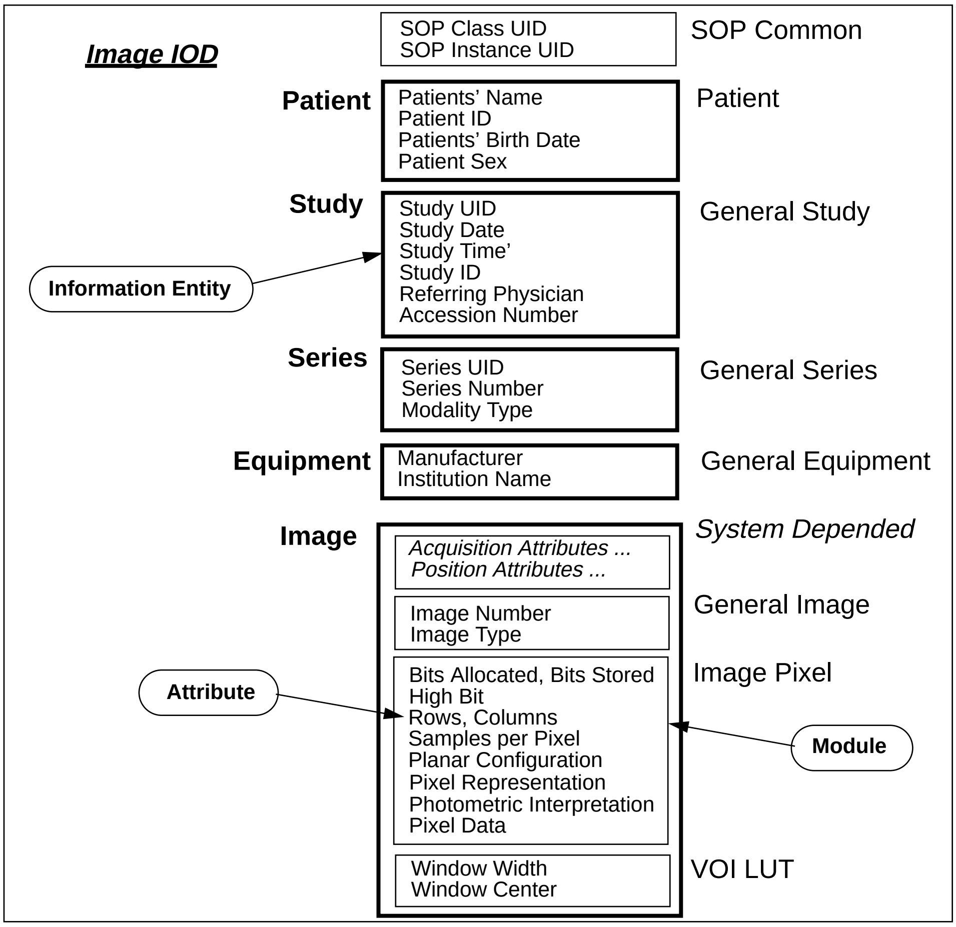


Figure 1-4: Example of composite image IOD

1.2.3 Attributes

Attributes are the basic information entities and have to be described in detail. For an attribute the following features are defined in the DICOM standard:

- unique *Attribute Name* (human readable)
- unique *Attribute Tag* (information systems readable)
- *Attribute Description* (semantics)
- *Value Representation* (syntax)
- *Value Multiplicity*
- Type classification: 1, 1C, 2, 2C or 3 (usage depending context of SOP class, Service Class role, etc.)

Type classification specifies use of the attribute related to the SOP class and SCU or SCP role. Depending upon the situation, each attribute is forced to have a value (*type 1*) or forced with or without a value (*type 2*) or optional (*type 3*).

Inside an IOD, groups or individual attributes may be conditional on the situation the IOD is used in. For example, an examination using a contrast can store the information in a Contrast/ Bolus module. The attributes of this module are therefore available or not available, depending upon the use of contrast. If used, the type classification specified for attributes must be obeyed (defined as *type 1C* and *type 2C*).

1.2.4 Service Elements

Services Elements are the operations allowed on Information Objects for a certain SOP Class. The group of service elements belonging to the SOP Class is called the *Service Group.*

The Service Group of a SOP class is selected from a fixed list of DICOM Services Elements. Some Service Elements are intended for use with a composite IOD, others for use with normalized IODs. A third category, storage media related Service Elements, handles instances of composite and normalized SOP Classes as files.

The context described by the Service Class is limited when using composite IODs (e.g., transfer image). Such Service Elements have a complex meaning, e.g., STORE, FIND, MOVE. There is no relationship assumed between individual Service Elements in a sequence when using composite Services Classes. If a relationship exists, it is outside the scope of Service Classes and should be defined in the process flow using the Service Classes.

In contrast, Service Classes using normalized IODs have a much broader context, such as managing functions. They use primitive Service Elements for operations with simple pieces of information: GET, SET, ACTION, etc. The Service Class defines the relation of a sequence of the primitive requests. With normalized Service Classes both partners keep track of the processing on both sides, using the Service Elements to control them.

Each SOP class uses one or more Service Elements from either the composite group (C-XXXX) or the normalized group (N-XXXX). The next Service Elements are available: C-STORE, C-FIND,

C-MOVE, C-GET, C-CANCEL, C-ECHO, N-GET, N-SET, N-ACTION, N-CREATE, N-DELETE and N-EVENT-REPORT. The semantics of the Service Elements depend on the Service Class and SOP Class in which they are used.

Media-related Service Elements M-WRITE, M-READ, M-DELETE, M-INQUIRE-FILE-SET and M-INQUIRE-FILE define primitive functions for manipulation with file sets.

1.2.5 SOP Instances

The framework of the above definitions takes shape when used in a distributed process. After the agreement which SOP Classes are supported (and implicitly the Service Class), and how the SCU and SCP roles are divided, *instances of the SOP Class* can be exchanged between the two counter parts. Attributes have to be provided with the (semantic) correct values and stored inside the SOP Instance as specified by the attribute definitions.

After collecting the information it will be encoded to the DICOM defined formats, using the Tag and Value Representation to create a DICOM *data set*, in which each attribute in encoded in a *data element*. This data set is handed to the exchange service provider which ensures the counter part receives an equal data set. Differences in system specific representation are taken into account during the exchange, ensuring the semantic values remain intact.

The receiver of the data set will decode the data set to extract the information it needs and acts as agreed by the SOP Class semantics.

1.2.6 Identification

As part of the creation process of a SOP Instance an identification is generated as attribute of the SOP Instance. The identification is intended for use by information systems rather than humans and has two aspects: the class identification and the instance identification.

This identification has to be used in a multi vendor environment on different places in the world. To ensure global uniqueness of each identification a mechanism is used to generating a string of characters (called *Unique Identifier* or *UID*) as follows:

<root>.<suffix>

The root part is supplied by an authority which guarantees nobody else will use this root. This number will be allocated by standards organisation to companies such as Philips or hospitals, that must ensure it remains unique across their own systems. By using a unique system identification each system will have a world wide unique root. The suffix has to be created dynamically by the system at creation of the instance.

Once an instance is identified by a UID it must be used consistently. If copies are made or the instance is recreated without any modification, it must have the same UID, otherwise two pieces of identical information will exist with different identifications, which can lead to confusion.

1.2.7 Relations

Besides the SOP Class and SOP Instance identification, UIDs are also used to identify a *relation* between instances. In a composite instance which contains a single image belonging to a series of images, the Information Entity which contains the information of the series will be common for all those instances. The relationship is identified by using the same UID for the Series Instance UID attribute for all the composite instances belonging to that series. In this case only an instance UID is required, the attribute itself identifies which type of information entity is identified.

In the case of normalized instances only references to instances outside itself are possible, here the combination of a class and instance identification is required. This is also the case when images are referring to each other when they have a certain relation.

With the method of uniquely identifying information using UIDs it is only possible to compare if instances are equal. The value of the UID has no meaning, it can not be used for sorting, etc. By using other, more meaningful attributes such as date and time and sequence numbers, the relationships between information can be established.

1.2.8 Value Representation

For each attribute a *Value Representation (VR)* is defined. A Value Representation describes how an attribute is encoded in a data element. The knowledge of the Value Representation is shared by the partners in the information exchange, the encoding and decoding process has to take care of selecting the correct VR for an attribute (identified by its tag).

Two ways of sharing this information is possible: sharing a data dictionary which contain all possible exchanged attributes or by including the value representation as part of the data element. The last approach increases the overhead of information exchange, but is much more flexible compared to the use of a shared data dictionary. Especially in a multi vendor environment synchronizing the data dictionary is difficult.

When the Value Representation is included, the message is encoded with *explicit VR*. In the other case the encoding take place with *implicit VR*.

1.2.9 Transfer Syntax

Before the data set of a SOP instance can be exchanged, the way the data set is encoded to a byte stream has to be fixed, either by agreement when a network exchange is used, or stored together with the data on a medium. The way of encoding is specified by a Transfer Syntax.

Three aspects have to be defined by the transfer syntax:

- How a *Value Representation* is specified (see Section Section 1.2.8 “Value Representation”).
- The *byte ordering* of multiple byte number (words, longwords): little endian or big endian.
- In case of encapsulation (compression): the *compression format*.

The handling of the transfer syntax is part of the service provider. However both processes have to initiate the setting of a correct transfer syntax, acceptable for both partners.

Analogously to SOP Class identification, a transfer syntax is identified by an UID.

1.2.10 Overview

See Figure 1-5 for an overview of the encoding and decoding flow. Services provided inside the Exchange Domain has to ensure that the SOP instances on both side contain the **same information**, regardless the representation and method of transfer.

The encoding and decoding process has two stages: First transferring the internal representation in the DICOM defined format (Data Set) where each attribute is stored according the value representation defined for that attribute. The second stage transfers the data set into a stream of bytes which can be handled by the lower layers. For the second stage the byte ordering has to be used as agreed with the Transfer Syntax.

The application which is using the information has to know the meaning (semantics) of the information inside the data object.

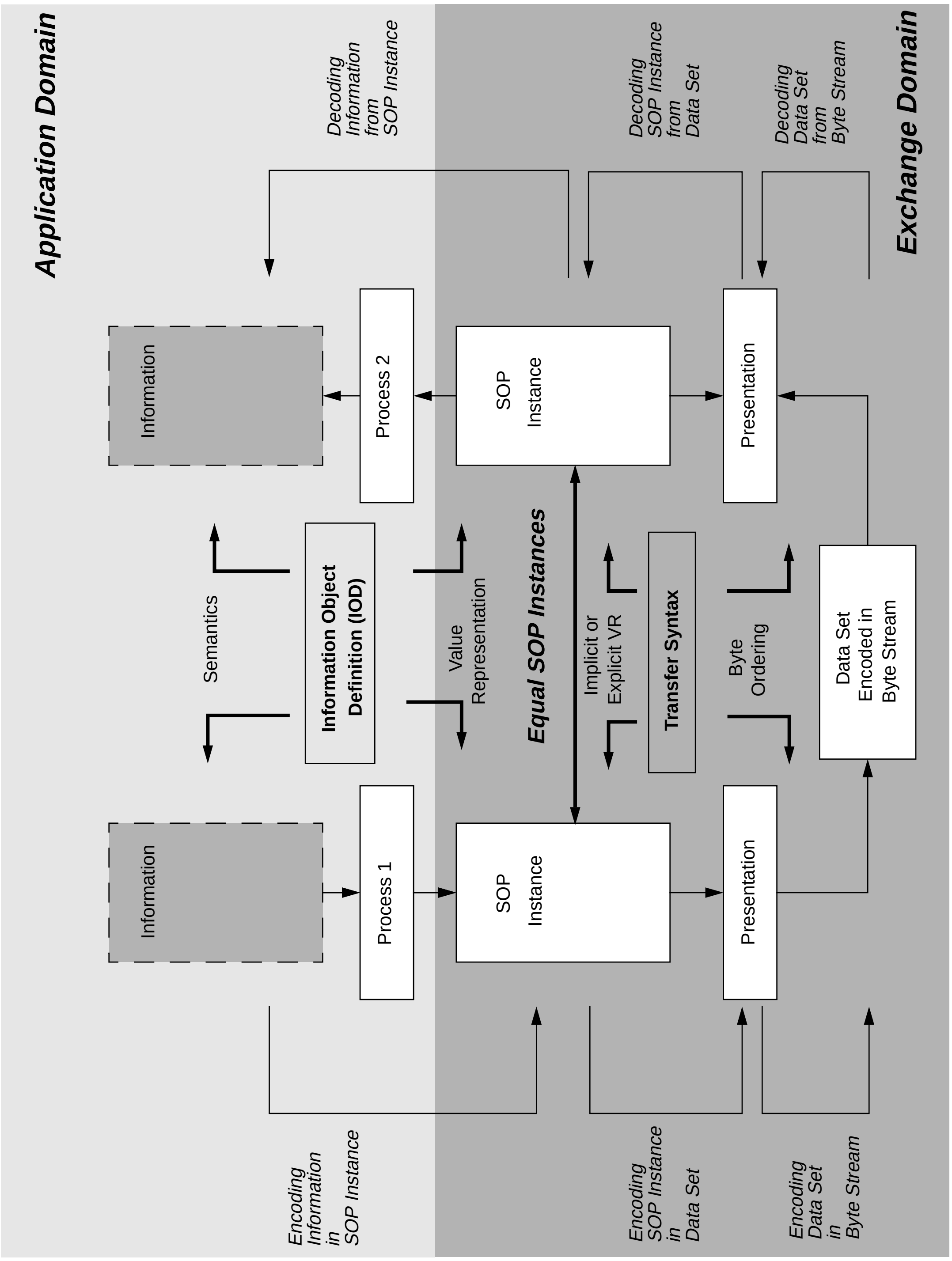


Figure 1-5: Overview Encoding and Decoding SOP Instances

1.3 DICOM Network Concepts

In the preceding section the DICOM concepts of the application domain are discussed. When a network is used for information exchange, the exchange domain will contain functions required for communication: the communication domain; see Figure 1-6.

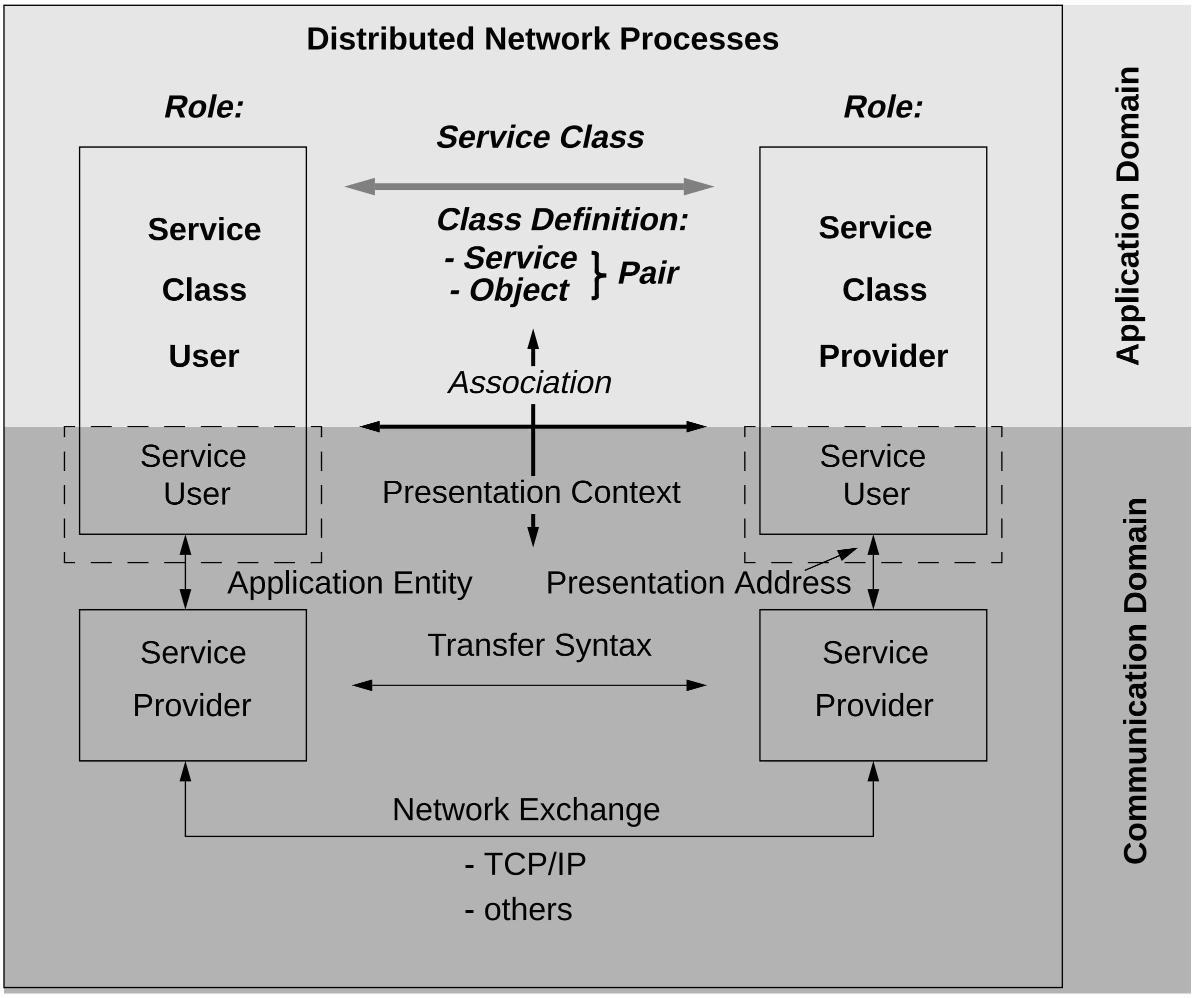


Figure 1-6: DICOM with Network Exchange

1.3.1 Application Entity

A major issue in networked distributed applications is how applications can contact each other. Arrangements have to be made to address the counterpart and agree about various topics before SOP instances can be exchanged. In DICOM Network partners recognise each other via Application Entities. An *Application Entity* is that part of a process that deals with the communication. It contains the Service User of the process, containing functions to setup connections and transfer the information. An Application Entity has a name *(Application Title)* that has to be used when setting up the communication.

1.3.2 Presentation Address

Application Titles are symbolic names for the processes involved in the communication. In a real network a network address has to be provided. This is called the *Presentation Address* and points to the Application Entity addressed. It is called presentation address because the service user is the (OSI) Application layer, the service provider the (OSI) Presentation layer (and lower layers). The boundary between both layers is the *network access* point where the data is transferred from the application layer to the networking layers. Each access point in a network has an unique address.

The mapping of the Application Title to the Presentation Address does not have to be unique, because the Presentation Address is used for connection initiation, etc. However at application level the Application Title is often used to identify an application as source or destination of information in a directory or catalogue. If this cannot be registered unambiguously the co-operation of systems can become a problem.

The format of the Presentation Address depends on the network protocol used. DICOM networks are in most cases realized using the TCP/IP protocol stack. In this case the presentation address is mapped to a *TCP/IP socket;* see Section 1.3.6 “TCP/IP Protocol Stack”. In case of a OSI protocol stack a valid OSI Presentation Service Access Point (PSAP) must be used.

1.3.3 Association Negotiation

The connection for information exchange between two Application Entities is called an *Association*. For an Association a number of communication issues are fixed as the context in which information can take change. This context (called *Application Context*) is defined in the DICOM standard and both sides must agree upon acting according to this context definition.

An Application Context is identified with a UID and during the initiation of an association this UID is transferred to the partner. By comparing this UID of the Application Context the partner can decide if it is capable of handling this request for an association. It will either accept the establishment for the association or reject it.

The Application Context covers the global functionality for the information exchange. Which type of information exchange will take place across the association is defined by the SOP classes and the Service Classes of those SOP Classes. The initiating partner of the association proposes the SOP classes it will use, the SCU / SCP role for each SOP class and the way of representation of the information. Depending upon the capabilities of the other side it may accept or reject each individual SOP class.

After this negotiation process both partners know of each others capabilities and limitations. The real information exchange can take place according to the Service Class and SOP Class rules defined for these classes. When an Association is no longer required the Association is terminated.

1.3.4 Presentation Context

For each SOP Class negotiated during Association initialization an agreement has to be reached between the processes involved about the transfer syntax used between the processes. The initiating partner proposes all transfer syntaxes it can handle for a certain SOP Class. The other side selects one of these transfer syntaxes, fixing the *Presentation Context* for this SOP Class. After the negotiation a Presentation Context for each accepted SOP Class is settled.

A Presentation Context is identified by an agreed number between both partners. In the context of an association a number of Presentation Contexts can exist. The Presentation Context number identifies the SOP Class for which the information exchange takes place.

1.3.5 Network Protocols

The actual Network Protocol has to comply with the standard services as defined for the OSI protocol stack. Besides the hardly ever used OSI protocol stack other protocol stacks are possible which must be adapted to the OSI services, see Figure 1-7.

The left part of the picture shows the Application Entity for a process with communication in general, the right part the DICOM functionality of the Application Entity.

For the application layer two groups of services have to be available for a DICOM implementation: the *Association Control* protocol *(ACSE)* and *DICOM Message* protocol *(DIMSE)*. ACSE is a standard OSI protocol, DIMSE implements the DICOM services discussed in the Section Section 1.2.4 “Service Elements”. [The extension “SE” means Service Element, a part of the services provided in the application entity].

The interface between ACSE, DIMSE and the application is the *DICOM Interface* as described in the DICOM standard. This specification describes which parameters are required for each function of the ACSE and DIMSE requests.The ACSE, DIMSE, and DICOM interface are part of the DICOM Application Context.

The interface towards the application (API) is not specified by the DICOM standard, but depends on the implementation. In general this API provides functions to connect to other applications, construct or process SOP Instances and transfer these to a remote application.

1.3.6 TCP/IP Protocol Stack

The combination of a TCP/IP stack and an extension for OSI application services is widely used to implement DICOM across networks. Because no upper layers are defined by TCP/IP, the application, presentation and session layer functionality as required for DICOM is described in the DICOM standard. This functionality is combined in one layer: the *DICOM Upper Layer* or *DUL*.

The DUL uses the same DICOM Interface for TCP/IP protocol stack as for the OSI protocol stack. At the lower level the DUL has an interface to the TCP layer. The DICOM Association between the Application entities is mapped to a TCP connection. The Presentation Address is mapped to a *TCP port number*, combined with the *IP Number* or *Host name*. This combination

Internal Use Only

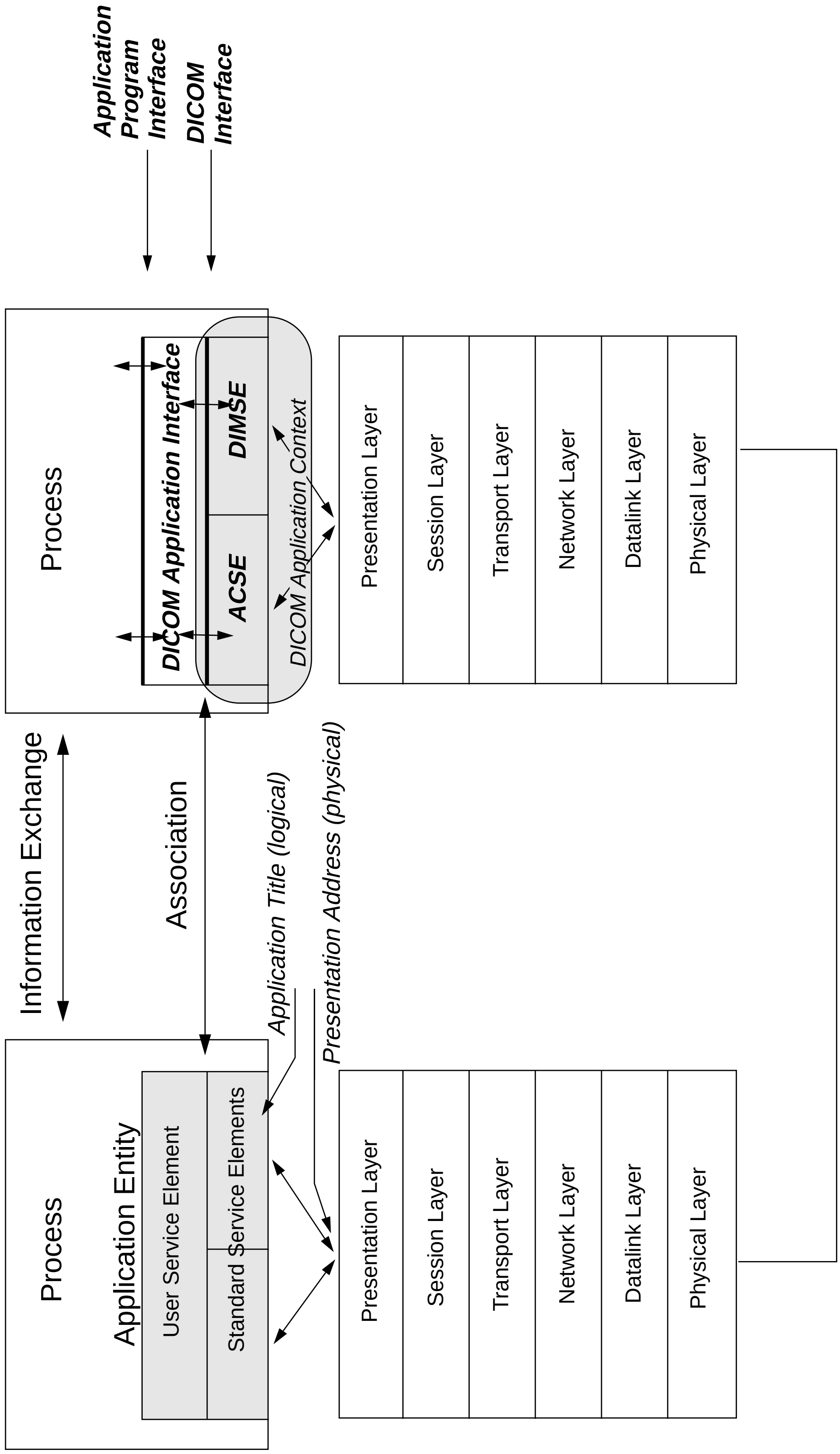


Figure 1-7: OSI Layers

of TCP port number and IP number is called the Socket Address, see Figure 1-8. In a network this combination is always unique.

A TCP connection is defined by the **combination** of the local Socket Address and the remote Socket Address. By keeping the IP numbers network wide unique and the TCP port number unique inside a system, each TCP connection is uniquely identified by the combination. The management of connections is done by a facility called the *Socket Interface* which provides functions to setup connections, transfer byte streams, etc.

The TCP port of the partner called during a connection initialization must be known. This can be either an agreed port number between two applications, or a port number (called a *well known port number*) reserved for DICOM implementations (port number 104).

Internal Use Only

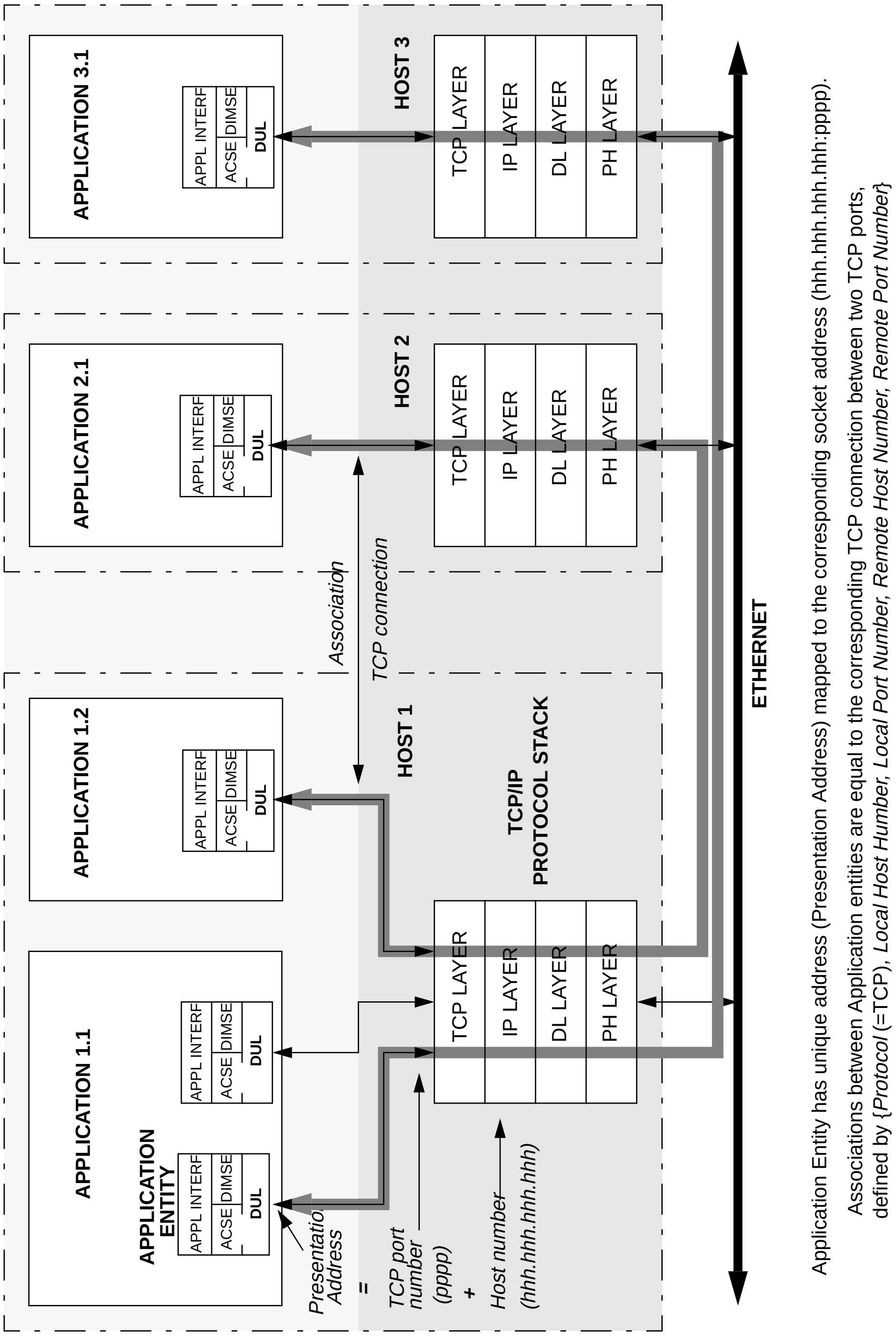


Figure 1-8: TCP Connection

1.4 DICOM Storage Media Concepts

When storage media are used to implement DICOM based distributed processing, the way processes work together is different compared to network distributed processes. In the first place no direct link is available: the data is stored and used at an other moment in time by the other process. There is no agreed Presentation Context, the information representation has to be handled on a different way. Secondly, there is no control over the capabilities of both sides as defined by the Service Classes for networking and agreed by the SOP Class during association.

The way DICOM has defined media exchange is illustrated in Figure 1-9.

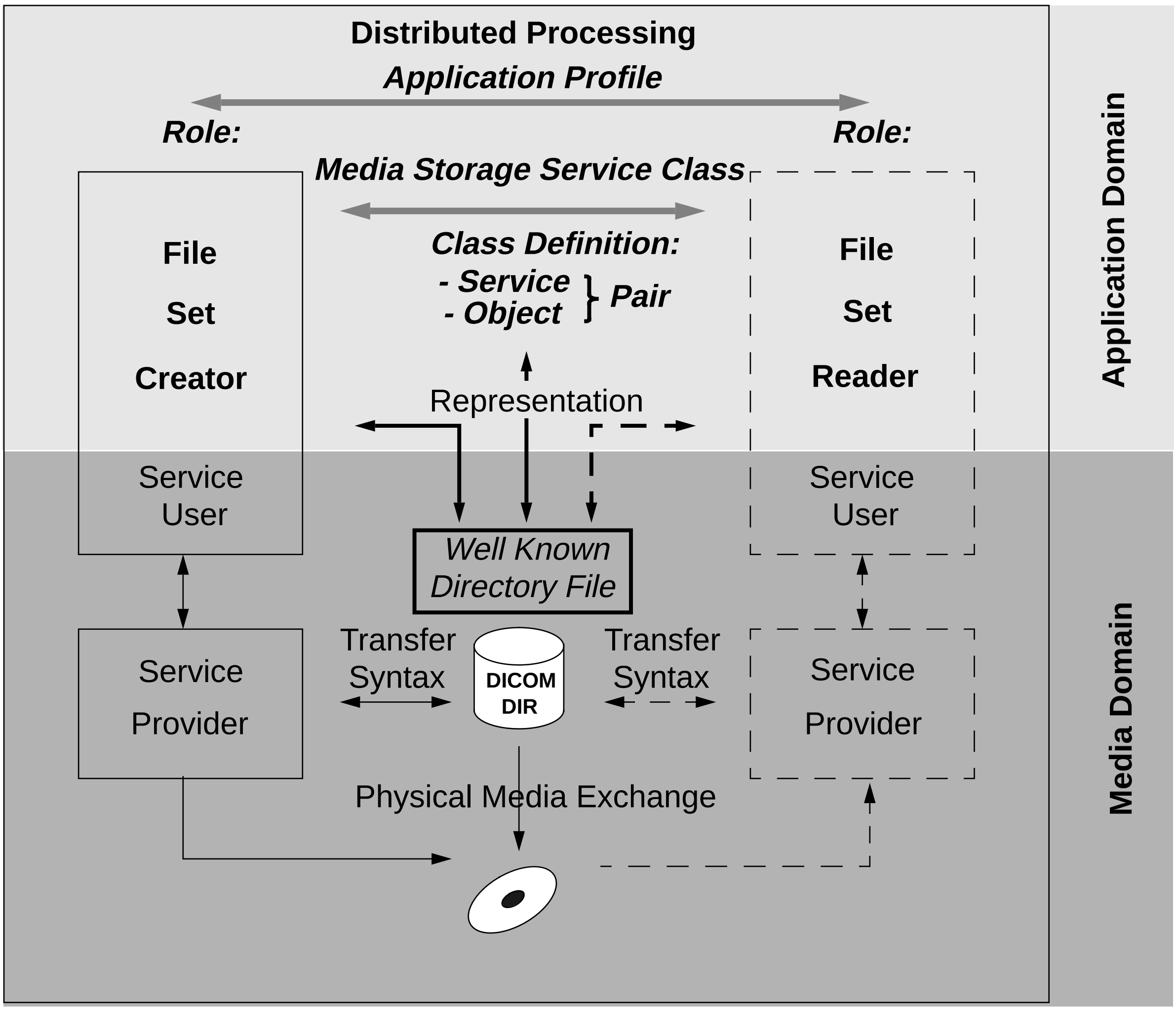


Figure 1-9: DICOM with Storage Media Exchange

1.4.1 Media Storage Service Class

The *Media Storage Service Class* defines a set of services which allows the exchange of data using Storage Media. Media will be used for the next two reasons:

- Images are stored on media for *exchange* between two processes without further specification about the processing, just the transfer of the information.
- Images are stored for *printing* organized as Film Sessions. The receiving process must handle the Print Management information on the media, and maintain on that media status information about the progress of the print job.

The role which a process plays in this Service Class is not related to the role of the partner as with the network situation, but related with the operations on the media. Three roles are defined: *File-set Creator* or *FSC, File-set Reader* or *FSR* and *File-set Updater* or *FSU*, the name refers to the operation(s) allowed.

The Service Elements used in the SOP Classes of this Service Class specify operations on instances of the SOP classes as a file set or the management of a complete file set. IODs used with these services define the information to be stored in a file. This information can be a mix of composite and normalized objects.

This Service Class deals only with the storage of information in a file, regardless of the contents. The exception is a special SOP Class (Media Storage Directory Storage) which handles information about the file set and a directory (DICOMDIR).

The other SOP Classes of the Media Storage Service Class are identical to the SOP Classes used with the network Storage Service Class for image data, Detached Patient Management, Detached Study Management, Detached Result Management and Print Management Service Classes. SOP Instances stored in files can be used directly by the Service Class of the corresponding SOP Classes after access using the services of the Media Storage Service Class; see Figure 1-10.

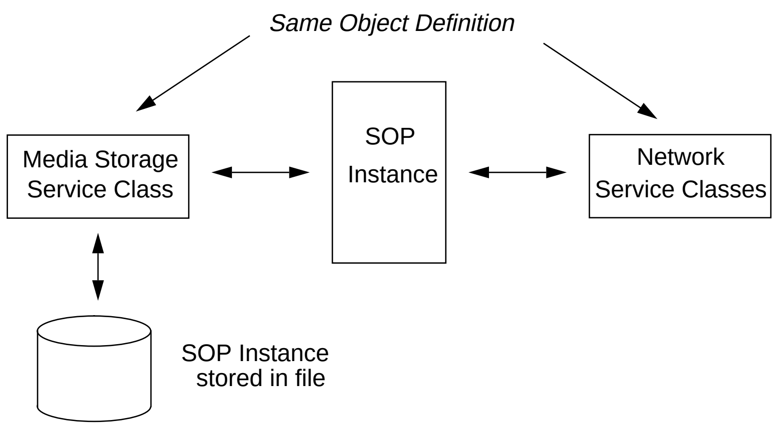


Figure 1-10: Shared Object Definition with Media Service Class

Processes on both sides must agree what information is exchanged by the media by specifying a list of SOP classes and other issues. As no mechanism such as association negotiation exist, there must be an arrangement conforming to an *Application Profile,* see Section 1.6.2 “Application Profiles”.

1.4.2 DICOM File Format

The DICOM File Format describes how a data set representing a SOP Instance has to be stored in a byte stream in a file on a physical medium. At the front of the data set a header is created containing the SOP Class, SOP Instance and Transfer Syntax identification in the format of UIDs. The transfer syntax of the header is fixed to always enable the reading process to obtain this information before it can proceed with reading the data set itself.

The files containing the header and data set are stored in a media dependent way. The file is identified by a directory path and file name as a sequence components separated by back-slashes (such as the path names in MS-DOS).

1.4.3 DICOM Directory Format

A special data set contains the SOP Instance of the Media Storage Directory Storage SOP Class. The file identifier is fixed to “DICOMDIR”, this allows access to this directory without further knowledge.

For each file written by a Media Storage Service Class service there are one or more *records* kept inside this directory. One record contains information about an Information Entity part of the SOP Instance. A record contains always a Record Type specifying type of information (Patient information, Study information, Image information, etc.) and a number of specific keys containing attributes extracted from the stored SOP instance. The records are hierarchically ordered (an example is the study, series, image relation). They are linked to each other at the same level and linked to a lower level by pointing to a next record. The keys in a record can be used to list the directory or search for a certain SOP Instance, without reading the data sets itself.

Entries referring to a SOP Instance captured in a file list the SOP Class, SOP Instance, Transfer Syntax (all in UID format), and file identification (path name).

1.4.4 Physical Medium

The definition of the DICOM file and directory format is independent of the implementation of the file system and physical medium which stores the data, as long as byte stream files are supported. DICOM files are contained in the directory structure maintained by the medium defined file system, regular file access mechanisms being used.

For each supported physical exchange media DICOM specifies the file identification mapping, location of the directory file (DICOMDIR), physical format, logical format and physical medium.

1.5 Supported Service Classes

DICOM has defined a number of Service Classes. They can be grouped into a number categories. This list of Service Classes will grow as new functionality will be standardized in the DICOM standard.

1.5.1 Image Storage Service Classes

The first category contains the Service Classes dealing with the image data. Image data is always encapsulated in a composite IOD and using the composite Services. The Service Classes of this group are:

- *Storage Service Class*, consisting of SOP Classes for each modality type of image data: Computed Radiography (CR), Computed Tomography (CT), Magnetic Resonance (MR), etc. This Service Class specifies the exchange of the data across a network. It does not specify what has to be done with the image data, that has to be managed by other Service Classes.

- *Query/Retrieve Service Class*. Includes the FIND, MOVE and GET SOP Classes for a number of query models. The FIND can be used to query for a collection of images. The MOVE and GET can be used to initiate the transfer. The actual transfer is done by using the Storage Service Class.

- *Study Contents Notification*, is used to notify an image management facility about the images created during a study and may be used to initiate the transfer of the image data or to check if all image data is completely transferred.

1.5.2 Management Service Classes

The management oriented Service Classes are using a mix of normalized IODs with normalized Services and a query Service Class which is handling information in the composite way. This second category consists of:

- *Detached Patient Management* handles the information required to schedule visits to one or more modalities. The patient demography information and study information are sent from the administrative systems (RIS) towards the modality.

- *Detached Study Management* receives information of the created series of images from the modalities and order all the acquired data into a complete study of related images and all other type of information.

- *Detached Result Management* is used to keep track of the reports and impression of a study.

- *Basic Worklist Management*, the only not normalized Service Class. It complements the Detached Patient, Study and Result Management Service Classes with a query facility to be used for obtaining information about a single or list of information entities. This allows a more flexible way of acquiring information compared with the other Service Classes.

- *Print Management* Service Class to manage the process of formatting and printing a collection of image data on film.

1.5.3 Media Storage Service Class

This category consists of the Media Storage Service Class, see Section 1.4.1 “Media Storage Service Class”.

1.5.4 Verification Service Class

Finally, there is a Service Class that does not fit into one of the two categories: the *Verification* Service Class. This Service Class is used to test whether an association can be set up between two processes and exchanges a command without any data (C-ECHO), in which no IOD is involved. It is intended for testing purposes on connectivity level.

1.6 Connectivity

Before two DICOM compliant implementations can be connected to each other, some investigation is necessary whether that connection is possible. This is reaching from the low level physical connection to the implementation of the same Service Class at application level.

The approach for a network connection is different compared to an exchange by media. During the Association negotiation in a networked environment a number of details can still be settled. In the case of using media this is not possible and should be addressed in a different way.

DICOM solves this issue by using system profiles for implementations and application Profiles in an environment with media exchange.

1.6.1 Conformance Statement

A *System Profile* contains lists of supported functions and limitations or extensions to that functions. Together they form a profile which must fit in the profile of the partner system(s) it has to co-operate. These system profiles are described in a document that must be supplied with each DICOM implementation: the *Conformance Statement* see Figure 1-11.

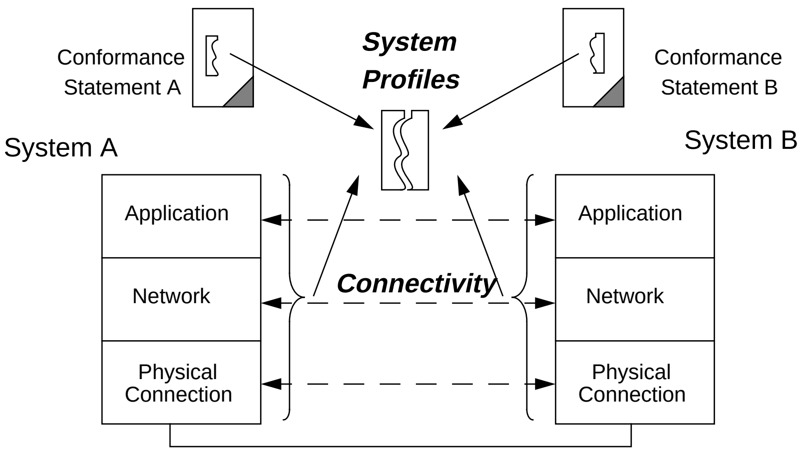


Figure 1-11: Conformance Statement with System Profile

At application level a functional description of the Application Entity, the supported SOP Classes, Transfer Syntaxes and the role both systems will play are described. When applicable more details about the implementations are given, specified by the SOP Class conformance requirements in the standard.

For the implementation of the network protocols can be referred to appropriate standard documentation, with stating the exceptions which restrict the usage in a network environment. The possibilities of the physical connection is also an issue to be addressed.

The configurable items of an implementation, such as Application Title, Presentation Address of both own implementation and partner implementation, are mentioned together with information how this can be configured. Other configurable items such as protocol data unit (PDU) size must be listed.

Finally the support for character sets other then the standard ASCII set (such as extensions for European languages, Japanese, etc.) is described.

By comparing Conformance Statements it can be verified if connectivity at all levels is possible. If the interpretation of the information by all partners involved is equal can not certain by verified with a Conformance Statement. Depending how strictly the semantics of all individual attributes can be interpreted, the level of interoperability is more predictable. Currently there is no method to ensure the interoperability. Thus solving ambiguous issues will take some time and needs fine tuning of distributed application.

Conformance Statements can add more information describing in more detail the information they handle. When it is stated what relations are available and which selections are made by the implementation compared to the standard, this will help to increase connectivity and interoperability.

1.6.2 Application Profiles

For media a detailed system profile makes little sense because matching will not take place before connecting the systems, but at the moment the medium is carried to another system. In this case both systems must be guarantee that they conform to a generic format which enable the application they both implement. This generic format is called a *Application Profile.* For instance, a system which produces image data on a medium must do it according a certain Application Profile. A system using this image data can rely on this Application Profile to be successful.

Two aspects are important: the format of the medium and the extent of information captured on the medium. An Application Profile fixes this two aspects and provides a kind of label which can be attached to the systems involved and the medium containing the data; see Figure 1-12.

The physical medium aspect refers to the defined format in the DICOM standard.

The information part described by the SOP Classes is the second aspect included in the Application profile. It always contains the Media Storage Directory Storage and one or more Media Storage SOP Classes. If a special application is intended with the Application profile, constraints or extensions have to be added to the Application profile. They can applied to the information object, but also to the Record Keys in the directory.

In normal cases no extensions are needed for the SOP Instances used with network Service Classes when stored on media, except that for the Record Keys some attributes has to be applied with a value, opposite to the SOP Instances exchanged with networks.

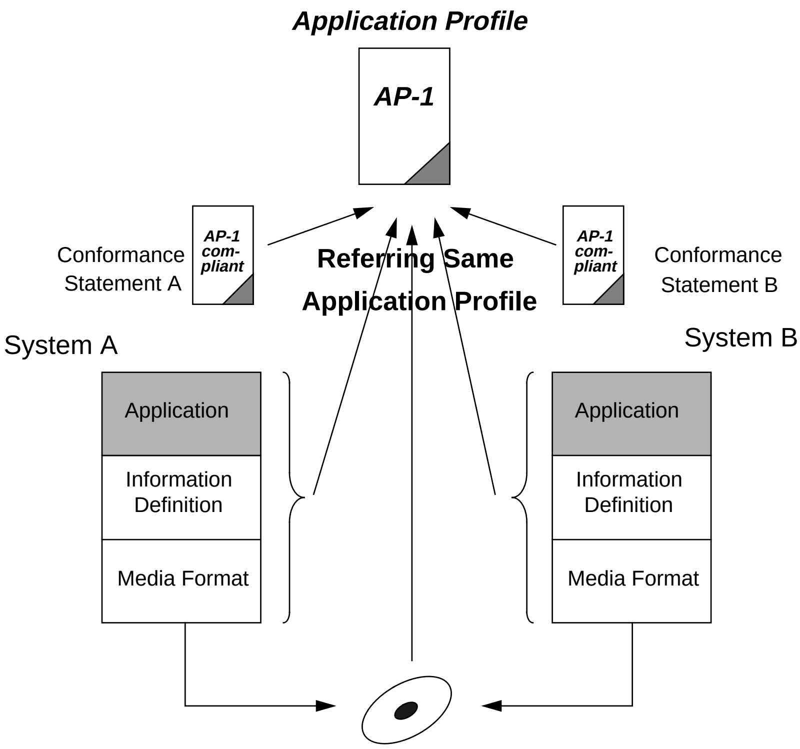


Figure 1-12: Conformance Statement with Application Profile

1.7 DICOM Standard

The DICOM standard is split into several parts, each part describing a major topic such as Service Classes, IODs, Network and Media related issues, etc. In Figure 1-13 an overview is given of the relation between the different parts.

In this section the different parts are discussed in the same order as the topics discussed in the preceding section of this chapter. It can be used as a guideline how to start with reading the various parts of the DICOM standard.

Part 1 gives an introduction and overview of the DICOM standard and its relation to other information systems found in a clinical environment.

The *Service Classes* and the *SOP Classes* included in the Service Classes are defined in Part 4. For Each Service Class the functionality is outlined followed by a description of the individual SOP Classes. For each role a process can play the requirements are given together with details of the usage of attributes if applicable. Depending the type of Service Class (composite or normalized) the description is giving little context or a detailed context. Also the topics which have to be described for each role of the SOP Class in the Conformance Statement are listed. Part 4 uses the IODs and Services defined in Parts 3, 7 and 10.

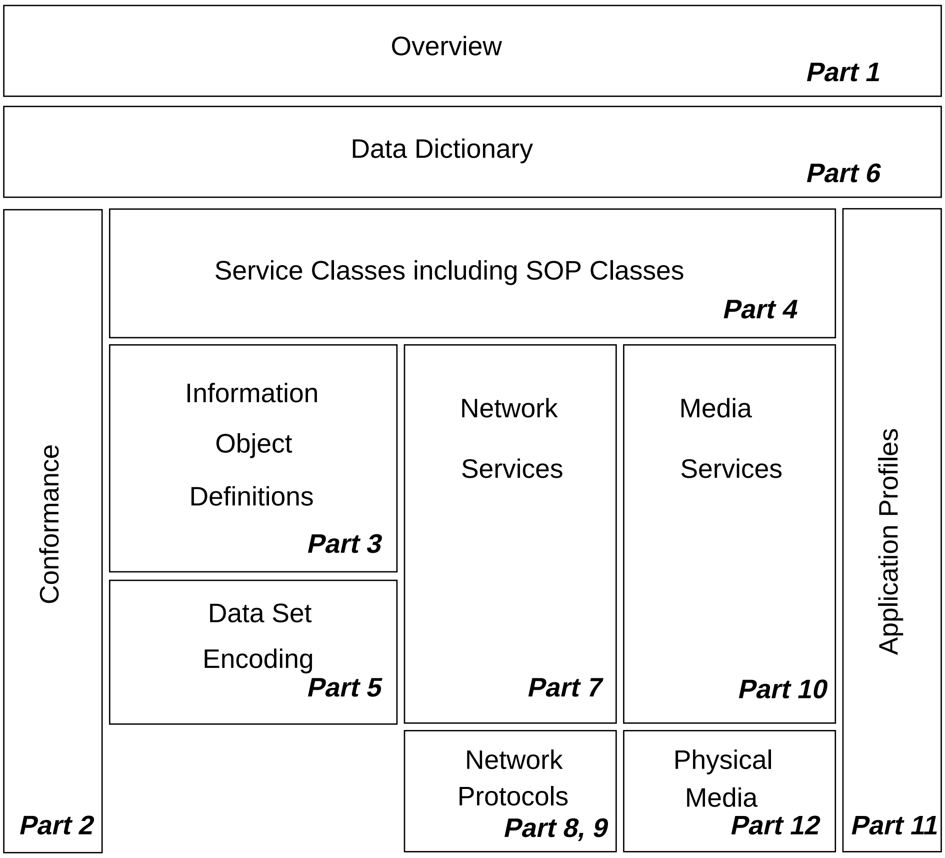


Figure 1-13: Relation between Parts DICOM Standard

The *Information Object Definitions* used by the SOP Classes are described in Part 3. It starts with the description of the full IOD, split into the groups of composite and normalized IODs. From each IOD a list of included Information Object Modules is given. The last part defines the individual attributes grouped in the IOMs in detail. For composite IOD all details are listed in this part, for normalized IODs the actual use of attributes is depending the applied service and described in part 4.

In Part 5 the *encoding* of the SOP Instances to data sets is described. The rules for the various Value Representations and Transfer Syntaxes are defined.

The *Services Elements* used by the SOP Classes are divided in two parts, Part 7 for the Network Services and Part 10 for the Media Services. In these parts the encoding of the network message header and media file header are defined. The result is a message or file which can be handled by the corresponding exchange mechanism.

The two lower level groups deal with physical exchange of data. In Part 8 the *Network Protocol* issues are described, Part 9 defines the point-to-point connection (rarely used) and Part 12 defines the format issues of *Physical Media*.

All defined attributes and UIDs by the various parts of the DICOM standard are listed in the *Data Dictionary* (Part 6).

Conformance issues are described in Part 2 including the way a Conformance Statement has to be setup. The *Application Profiles* used for the Media exchange are discussed in Part 12 together with a Application Profile layout.

2 DICOM Image SOP Instances

In the preceding chapter, the DICOM concepts are explained without describing in detail how images are captured inside a SOP Instance. This chapter will go into more detail about how the information is modelled. The difference between types of images is explained, together with the way in which the creation process produces the image data. Finally, the correct use of the information by a consuming system is discussed.

DICOM SOP Classes contain an object definition (IOD) and services to be applied to the object. In most of the sections below only the object definition is discussed. Image data handling, as described in DICOM, is limited to transfer (Storage SOP Classes) and media storage only. In this chapter, instead of using the DICOM term Storage SOP Class/Instance, the non-DICOM term *Image SOP Class/Instance* is used to refer to the processing of the image data. DICOM has no way of describing this type of image data handling; the term Storage SOP Class makes little sense and is confusing when used in other contexts.

2.1 Image Information Model

The electronic handling of information requires a model to represent the way the information will be structured. This structuring is needed to have uniform instances and to make it possible to describe the relations between instances on an umambigous way.

An image information model is derived from the way images are handled in a radiology department. Images are collected, from one or more modalities, into a *patient folder.* Images are ordered in the patient folder based on the type of examination (series of images which have a certain relationship).

The users of each type of modality have their own terminology for this ordering, such as examination, run, scan, slice, etc. When the image data from different sources has to be collected in a single environment, it must be possible to order the image data from the different sources. This is only possible when all the image data is structured according the same information model.

2.1.1 Mapping Real World Examinations

The *DICOM Image Information Model* is based on assumptions about the way in which information from different modalities is related; see Figure 2-1. The four levels of this information model are Patient, Study, Series and Image.

2.1.2 Patient Level

The Patient level contains the identification and demographic information about the patient to which a study belongs. Because more than one study of a patient can exist, the patient level is the highest level (when all the information collected for a single patient is taken in account). However, the normal practice is to use the study level for the collection of information handled by the various systems for a single examination request.

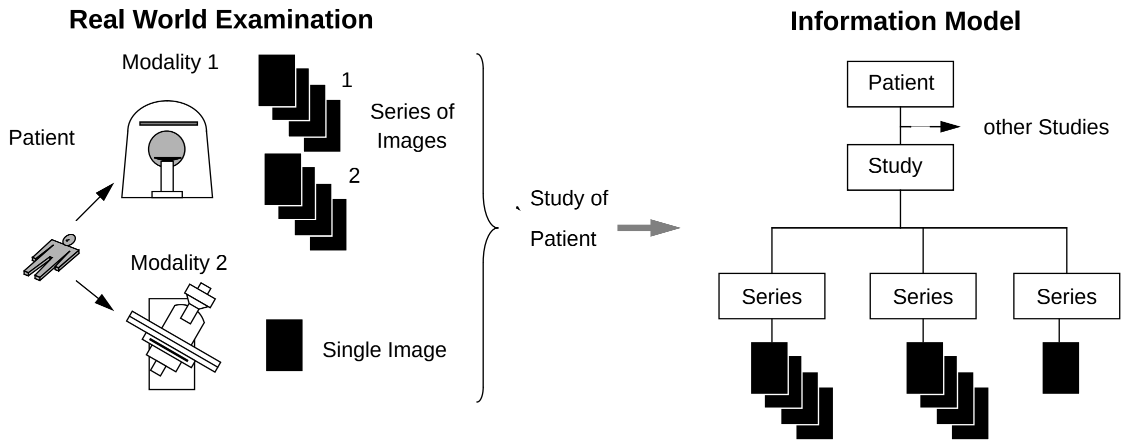


Figure 2-1: Mapping Real World Examination to Information Model.

2.1.3 Study Level

The Study level is the most important level in this information model. A study is the result of a request for a certain type of examination. All the activities in a radiology department center around the correct handling of the study. At study level identification information is kept and it can also contain references to information related to the same study in an administration system.

In general, a request may involves examination procedures on different modalities. This results in series of one or more images, depending on the protocol defined for the examination. All the image data is collected together with the same study as the root. A single patient can have multiple studies as the result of other (previous) requests for an examination procedure.

2.1.4 Series Level

Below the study level all the series of images are collected. The Series level identifies the modality type creating the images; the date/time when the series was made, and details about the examination type and equipment used.

Mapping the terminology used in different types of modalities has to be carefully considered. Terms such as run and acquisitions are apparently the same, but are used differently in various contexts.

Series are always a collection of related images coming from a s*ingle modality.* The way images are grouped into series is *depending on their clinical usage*. How the images are acquired by the modality is less important for this grouping. However, various attributes will identify the acquisition and can be shown when displayed.

In a number of cases the image relationship is defined by the way the acquisition takes place. When acquisitions in a sequence has a spatial or temporal relationship, the images resulting of this acquisitions can be grouped into a series. A new series must start when the existing relationship between images is no longer valid.

An other criteria to group images can be collecting images of a single body part made during the complete examination at the modality. For example, when a modality produces a number of images of a patient's stomach from different position and different moments in the examination, the images can be collected into a single series.

Some systems produce more then one image from an acquisition. For example, when scans are made on a CT system, the images reconstructed from each scan are collected in a series and have a spatial relationship. The next scan will be a new series, because in most cases the scan is made from a different position. In a series, a reference image can be included as an overview of the position of the individual slices, see Figure 2-2. Different reconstructions of the same

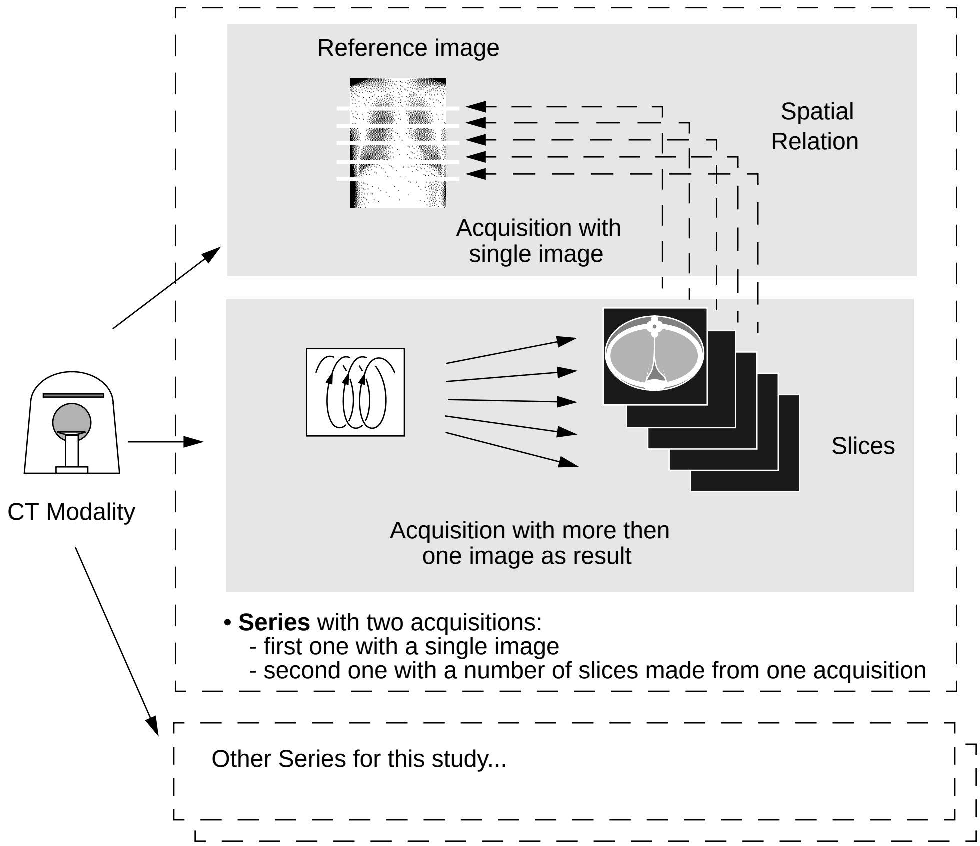


Figure 2-2: Example of Mapping CT Series

acquisition can also be stored in separate series.

For each type of modality the rules defining the contents of a single series must described. DICOM gives no modality dependent definition of what has to be collected in a series. The rules used by a given system are part of a system profile in the DICOM Conformance Statement.

2.1.5 Image Level

The lowest level of the information model is the *Image* level. Each image contains acquisition and positioning information as well as the image data itself. Depending on the type of modality, the image level contains data for one image (single), two images (*biplane* system) or a collection of images taken in a single gathering of image data in a relative short period of time (*multiframe images*).

The use of multiframe images saves duplication of information at the higher levels, but is only possible when the relationship between the frames can be described in a simple manner. For example, the increments in the movements of the system and time are equal for all the single frames.

Creating multiframe images is more complex and consumes more resources than creating a single image. The relationship between frames, the capability of the modality, and the amount of image data produced should be used to determine whether a series of single images or a multiframe image is most applicable.

2.2 Image SOP Instances

The information model shown in Figure 2-1 is a simplification of the complete DICOM Image Information Model in Figure 2-3. Each block in this diagram represent an *Information Entity* (see Section 1.2.2 “Information Object Definitions”) of the composite IOD. The relationships indicate the cardinalities for each relation of the Information Entity used in a SOP Instance.

Each Image SOP Instance has to contain the information structured according to the DICOM Information Model. Each Image SOP Instance (single or multiframe) is a composite SOP Instance containing the *whole information tree* from the information model. All images in a series contain the same patient, study and series information entities; all series, the same patient and study information entities, etc. In each composite, all information related to the image data is available.

This self-containing composite format makes exchange and handling (especially the storage) of information easier but increases the amount of data when a whole study is transferred. In this case the patient and study information entities have multiple instances in the collection of SOP Instances. In contrast, normalized SOP Instances (with single information entities) use references to other information entities, allowing a more efficient protocol, but requiring a more complex handling.

2.3 Relations and Identification

When collecting a group of image SOP instances which have a relationship with eachother, but are created on different modalities, it is important to be able to match the information entities at the different levels. Two aspects are important:

- All modalities must have a *consistent mapping* of their image data to a SOP Instance.
- The individual information entities must contain *sufficient identification* to make a correct match of equivalent information entities in other SOP Instances

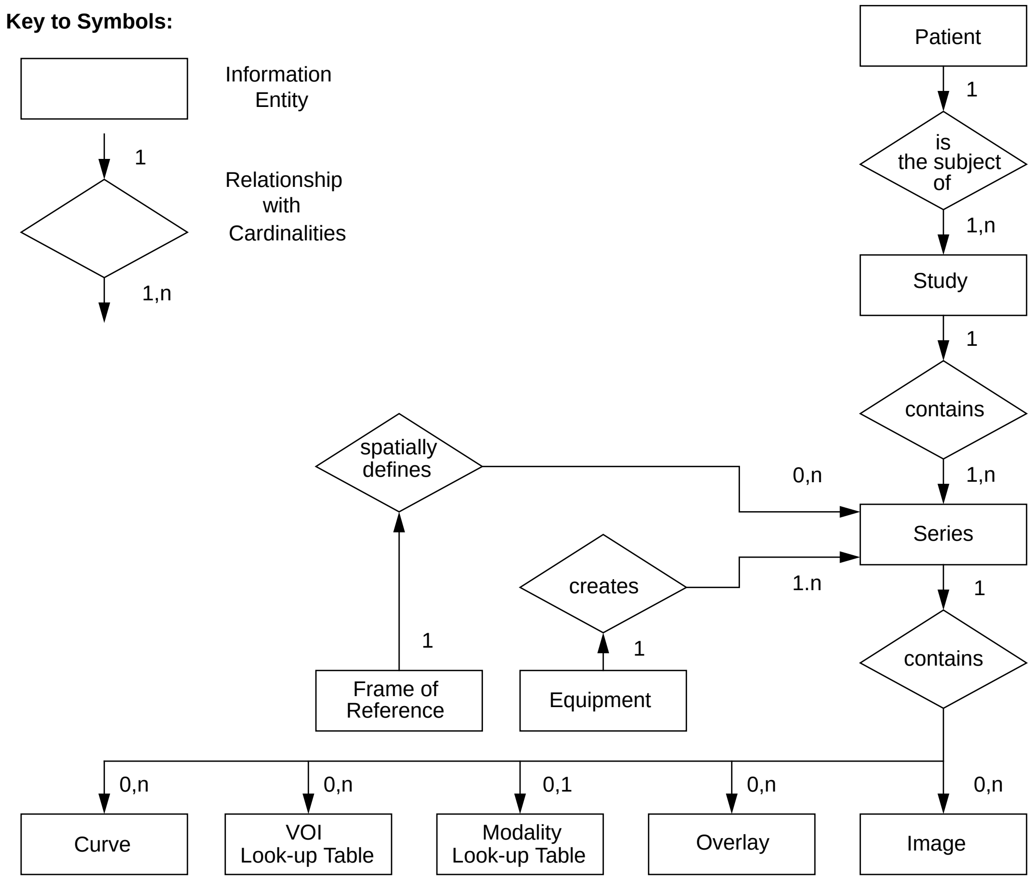


Figure 2-3: DICOM Composite Image IOD Information model

2.3.1 Mapping of Image Data

The first aspect requires that the image data produced by the modalities is ordered into a series which have a relationship as described in Section 2.1.4 “Series Level”. At the series and image level, the image sequence inside a series must identified on a modality dependent way.

The information entities above the series level should contain information belonging to the study and patient that must be comparable with information from other modalities. Most of this information comes from an external source such as a scheduling system. It can be supplied to the modality by either its user interface (from paper) or a connection to an information system.

2.3.2 Identification

If image data have to be stored in a storage systems which order the data by examining the information entity content, there must be a consistent and agreed way of identifying an information entity by all systems (modalities, storage systems, workstations, etc.) which handle the information.

The scope of identification is broader then just ordering images. The identification is also used to access *data from other information systems.* Information systems typically use keys which do not need to be interpreted by human beings, but have to be unique to the environment in which they are used.

The mechanism DICOM has defined for this identification are the UIDs as discussed in Section Section 1.2.6 “Identification”. Each of the Information Entities in the Information Model has its own UID, except for the patient Information Entity.

The way the patient information has to be identified is defined by other information systems (outside the scope of DICOM) dealing with the patient administration. In this case a hospital wide or even nation wide Patient Identifier (Patient ID) is used.

2.3.3 Study Identification

Because the request for an examination by a referring physician is the center of all activities in a radiological department, it must be reflected in all pieces of information which are related to the examination request. For DICOM this information is identified at the study level. In most cases the *Study Instance UID* attribute identifies the Study Information Entity belonging to this real world examination request.

When this UID is used in a consistent way by all systems involved, it is not difficult to relate all the pieces of information with the image data in the DICOM SOP instance. However this requires a *link between all the systems involved* in transferring this system key. UID transfer by human beings is not an acceptable practice due to length and meaningless content of the UID string, it would be error-prone.

Besides this link, UIDs have to be *supported* by all other systems, not only the systems involved in image data handling. A system which generates the study UIDs plays a major role by distributing the UID to other systems involved. Typically, this should be done by a Radiology Information System (RIS) or Hospital Information System (HIS), which currently may not always support the UID concept.

When the support for the Study Instance UID is not available, it is not possible to use this UID as a link to all other information. It has to be replaced in these cases by other keys. RISs are already using one or more keys to access their internally stored information, e.g., examination registration number. These keys are mostly printed on the papers belonging to that study. This information has to be included in the Study information entity and used as a *replacement of the Study UID*.

Using the Study UID as a link-pin to related pieces of information is an important aspect in providing a consistent DICOM Information Model, which can be expanded into other parts of information management within a radiological department. This consistency is very difficult to maintain when the Study UID is replaced by some RIS or environment specific identification method.

2.3.4 Other Identifications

Besides the system keys, users also need access to the information and want to use meaningful identifiers such as patient's name, birth date, date and type of examination, etc. Modalities have to provide as much consistent information as possible to enable human identification. The identification information can be supplied from a single source when a link between information systems is available. For instance the RIS supplies the patient's name, birth date, etc. as part of the information to perform an examination. This method prevent typing error and allows a more efficient way of working.

2.4 Classification of Image Data

The Information Model defines the hierarchical model of Information Entities to make clear how information inside different SOP Instances can be grouped at the different levels. In this section the SOP Instance information is classified according to the function it has, but regardless of its place inside the Information Model. Of course, there is a strong relationship between model and creation process. The following sections describes the view of the producer of image data.

In Figure 2-4 an overview of the classification and the relation with the system architecture of a (example) modality is shown. The different classes are created at different moments in time when performing an examination. Each sub-system adds information attributes to the final result: the Image SOP Instance. The following sections discuss the information classes in more detail.

2.4.1 Patient Information

This class contains information about the patient undergoing an examination. In a radiological department the patient information is already known from other sources, such as information systems or the request for the examination on paper. It only has to be registered in a formal way by a number of attributes such as Patient's Name, Patient ID, Patient's Birth Date, etc. The information in this class is stable, except for the correction of typing errors and changes to the name in the case of marriage, etc. The maintenance of this information is done by systems which act as source for modalities.

One or more attributes (most likely Patient ID) are a key to information in other information systems. Other attributes identify the patient as a person or give more details about his condition.

A number of these attributes are very important for the whole process in a radiological department for identification and connections to other information. To allow identification of the patient and study during the review of a study, the modality has to include this attributes in the SOP Instances which are created.

Procedures in the hospital also have to cope with the handling of information in exceptional cases. For example, when an unknown emergency patient is examined, steps have to be taken to allow the information to be correctly re-identified when the patient's name is known.

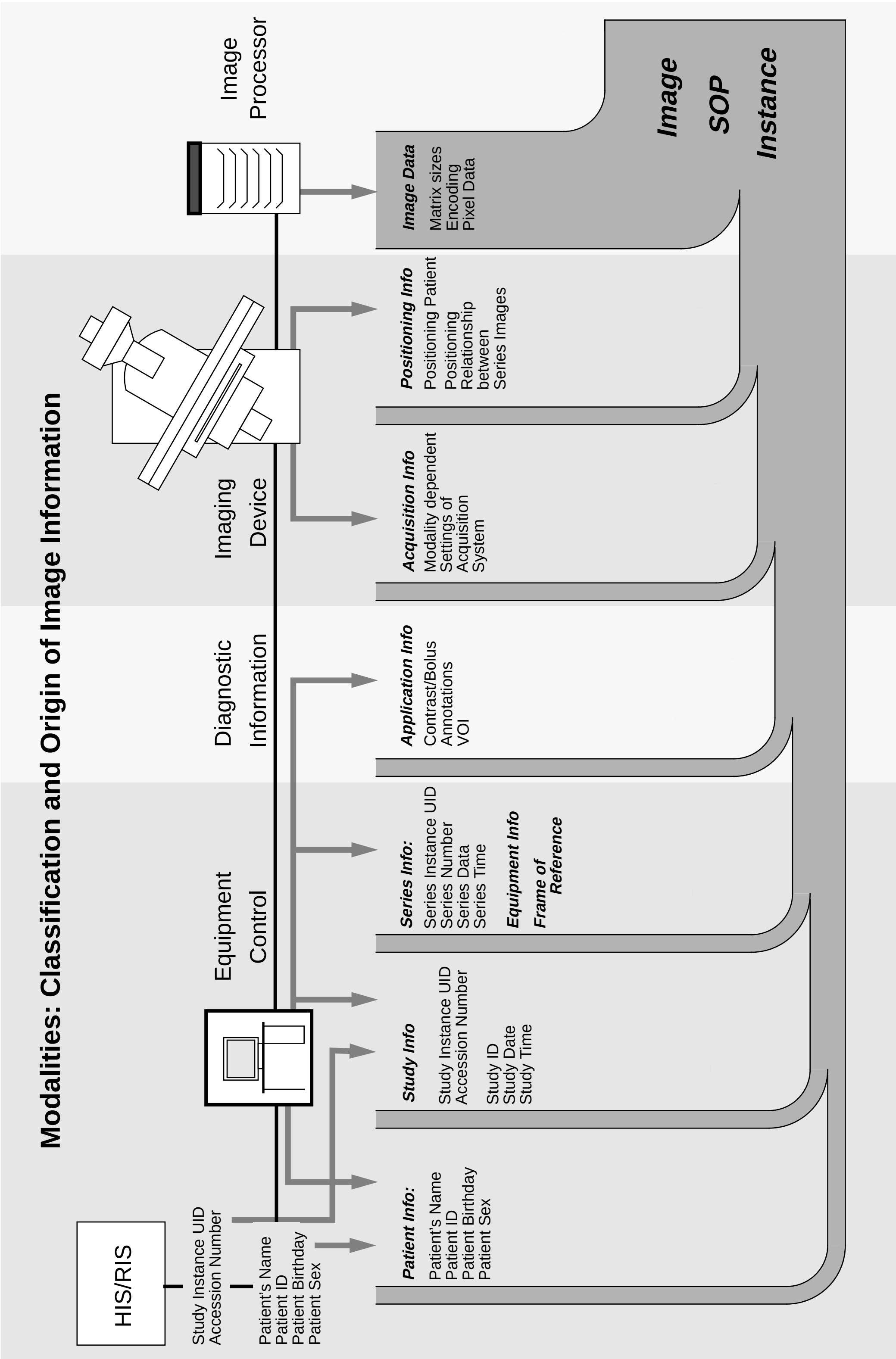


Figure 2-4: Classification of Image Information

2.4.2 Study Information

Study Information is a class with a mixed source of information. On the one hand, information will be supplied from a system such as a RIS which identifies the study across more the one system. On the other hand, the modality will add study information about the patient at the time the examination takes place.

Information from other systems includes an identification of the study. A Study Instance UID is the most efficient way of identification, but has drawbacks (see Section Section 2.3.3 “Study Identification”). An alternative attribute, called Accession Number, can be used in a RIS dependent way. In the case no Study Instance UID is available from outside the modality, the modality has to generate the UID in such a manner as to guarantee that it is unique in the whole system.

When images of a study are copied from the local storage to a remote destination, it is very important that the *same Study Instance UID* is used. This prevents the existence of images with different study identitifications from a single original examination. Such images can never collected together without operator intervention.

Other information supplied to the modality are the names of Physicians requesting or reading the images and patient information which is dynamic such as age, weight, occupation, etc.

Information, included locally by the modality, identifies the study by providing a value for the Study ID attribute and the actual study date and time. The Study ID is only relevant for the modality used to perform the examination.

2.4.3 Series Information

The Series Information class is the first one which is completely generated by the modality. In this class the type of system, the location and identification of the system is given. The identification of the series itself consists of the Series Instance UID, which uniquely identifies the series in the image data environment, and a locally used Series ID that can be used to sequence a series in a study. The Series ID has only a meaning for the modality itself, no rule for its usage is given.

With the Series Information more details are supplied about the way the series is performed, people involved, positioning relevant for the whole series, part of body examined, etc.

The *Equipment Information* part contains general information about the system used for this series. It includes information about the location, type and serial identification, calibration issues, etc. This information can be shared by series belonging to the same study and made on the same modality.

A *Frame of Reference* is used to group images which have a spatial or temporal relationship. This can be used to divided a series into parts or across more then one series if the same relationship is applicable. Such a relationship is identified by a Frame of Reference UID shared between the images involved.

2.4.4 Application Information

The attributes in this class give information about the image contained in the SOP Instance required for the diagnosis and other applications. Examples vary, from simple text added as comment, to details about the contrast, therapy and devices used during the examination. Another group describes the body part examined using coded values.

The *Value of Interest (VOI)* settings, in most cases called the *window width* and *window center* (or window level), are very important members of this class. The VOI is the selection out of the full range of pixel values which are clinically significant when displaying or printing the image. Only the specified range has to be converted to the available gray levels, all values outside this range are display as back or with values.

Information which draws lines or add text to the displayed image can be in the form of overlay matrices which have to be added to the display by a viewing station, or already applied to the pixel matrix (burnt in). By supplying the overlay as a separate piece of information from the image data, the image can be displayed with or without overlay, allowing the image data to be used as input for further processing.

2.4.5 Acquisition Information

In this class of information the settings of the acquisition equipment are stored. The extent of information depends on the type of modality and can range from a few attributes for simple systems to a complex structure for systems such as MR or Angio systems. It contains details of the acquisition system settings (such as X-Ray kV values, Collimator shape, Image Intensifier diameter, etc.). Also included, for MR systems, are the identification and details of the scanning sequence used.

Images as result of the same acquisition can be identified with an *acquisition number.* This grouping is system dependent and can be part of a single series, but a single acquisition can also result in multiple series of images, each with different characteristics. The acquisition has no relation with the DICOM Information Model and it has no equivalent UID identification.

2.4.6 Positioning Information

An important class is the information given about the positioning of the image inside a patient. Depending on the type of modality, the way in which the image matrix is positioned is described using simple terms such as anterior, posterior, right, head, etc. Care must be taken to ensure that there is sufficient information provided with the image to allow an unambiguous display (particularly left/right issues).

In a series that has a spatial relationship, such as a series of CT or MR images, much more detail about the position of the images in the three-dimensional space of the patient's body has to be provided. This information allows systems such as radiotherapy treatment planners to use the three dimensional positioning for their processing of the image data.

Other usage of the positioning information are for vascular systems to describe the dynamics of the movements when tracing a Contrast/Bolus flooding through the veins. This information is used for post-processing a group of images to create a single picture, with elements from the collection of images, showing the progress of the Bolus.

2.4.7 Image Data Information

Finally, the image data is acquired by the acquisition system and processed to produce a viewable image in digital format. This class describes details about the way pixel information has to be interpreted - such as the size of the pixel matrix, the representation of a pixel value, and how it is encoded. When modalities are able to produce colour pictures, information about how the image data is ordered in different colour planes has to be provided.

Besides the formatting information this class contains the pixel data itself in a single frame, two combined frames for biplane systems or a multiframe. When a multiframe image is generated by a biplane system it is possible to stored the frames of both planes together. In this case the frames of both planes are stored alternated (A-B-A-B-...). For multiframe images the relationship (in time) between the individual frames is described in a number of attributes.

The image is uniquely identified by the *Image UID*. Because a single SOP Instance of an Image SOP Class always includes an image portion, the Image UID is also used as *SOP Instance UID*. This UID is used to identify the instance when transferred or retrieved from an image store or, to identify the image entity itself when using it in a hierarchical tree of information.

2.5 Extension of Information

For the storage of all the information in the classes described above attributes are defined, grouped into Information Object Definitions which give a generic description of the SOP Instance for a certain type of modality. The actual attributes used must be described in the Conformance Statement (system profile).

It is not always possible to store all information generated by a modality in a standard IOD. In a number of cases equipment has gone through new developments which require additional information to be stored in an extended IOD. Care must be taken that parties using this information will understand this new information. Details have to be published in the Conformance Statement. If the usage is accepted, the new information becomes part of the standard definition. The extension may not influence the semantics of the information stored standard attributes. It has to be a proper subset, compatible with the IOD from which it is derived.

In other cases equipment from a single vendor can add information to be used only in that combination of systems or only by the same system that generated the data (re-use). In this situation, no details about the information have to be published in the Conformance Statement. There is no intention for other parties to use the additional information.

To enable the extension of information DICOM has defined two type of attributes: Standard Attributes and Private Attributes. Standard Attributes are used to encode the attributes described in the standard IODs (see Section Section 1.2.3 “Attributes”). If no extension or changes are made to the IOD the SOP Class is a *Standard SOP Class*.

By adding privately defined attributes or using standard attributes not belonging to the IOD of a specific SOP Class, it may no longer be called a Standard SOP Class and depending on the effect it changes to one of the following types:

- *Extended SOP Class* - when the additional attributes do not change the usage of the SOP Class. In this case it is a super-set, when used by systems which are not aware of the additions, they can be ignored, and the image can be handled as defined for the Standard SOP Class. An Extended SOP Class uses the same UID as the Standard SOP class from which it is an extension. The differences between both classes is shown in the Conformance Statement.

- *Specialized SOP Class* - when the additional attributes follow the Information Model, but the class is not longer a super-set. As a consequence, the UID of a Standard SOP Class may not be used; a privately defined UID must be used for this SOP Class. Partners handling SOP Instances know the private UID and can handle the information. Others cannot accept the SOP Class during association negotiation or when opening a DICOM file when reading it from a DICOM formatted medium.

- *Private SOP Classes* do not follow the DICOM Information Model and are used in a completely private context. They make use of the mechanisms provided by DICOM to transfer the information. Private SOP Classes use privately defined UIDs to prevent incorrect use of the information.

If one of the three above mentioned SOP Classes is defined with the intention to become part of the DICOM Standard the details are published in Conformance Statement. Otherwise, they are only used in a closed environment.

2.6 Image Types

DICOM defines a number of Image SOP Classes types, depending on the modality which creates the image data. Each type has its own IOD to add modality specific information to an image SOP Instance.

All Image SOP instances share a minimum set of information which allow a display application to handle the images regardless of it type.

A dedicated Image SOP class is available to encapsulate images which are not available in digital format, but captured from video or film format.

2.6.1 Generic Image Type

The Image SOP Class instances have a common basic set of attributes; see Figure 2-5. The minimum set of attributes required for an Image SOP Instance consists of the following group of attributes:

- Identifying attributes: SOP Class UID, Study Instance UID, Series Instance UID and Image Instance UID (= SOP Instance UID)

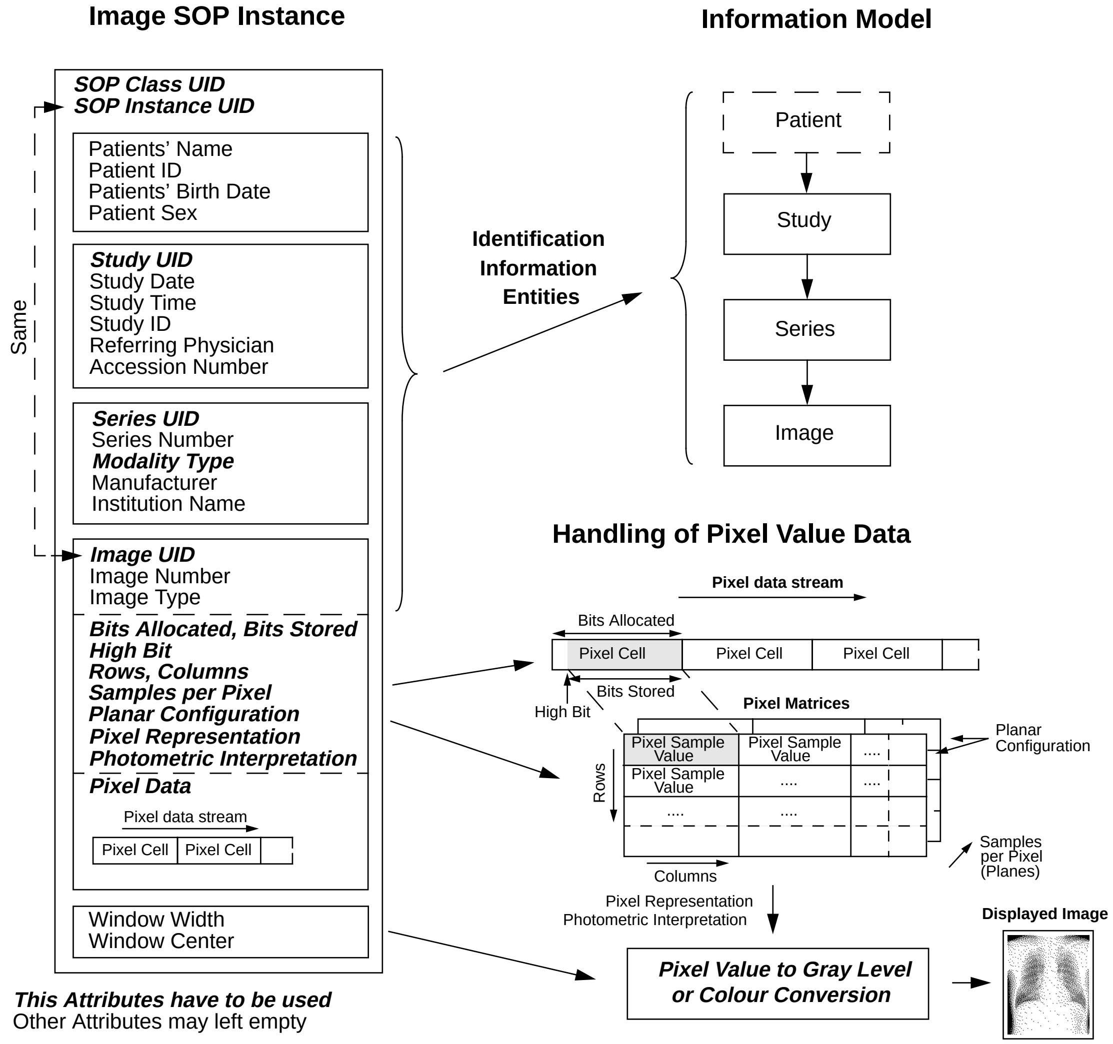


Figure 2-5: Basic Set of Attributes Image SOP Instances

- Modality type
- Pixel Matrix description: Sample per Pixel, Rows, Columns
- Pixel Value Interpretation: Photometric Interpretation
- Pixel Encoding: Bits Allocated, Bits Stored, High Bit, Pixel Representation, Planar Configuration
- Pixel Matrix: Pixel Data

This minimum set allows the display of the pixel data, and provides identification at system level, in order for the SOP Instance to adhere to the Information Model. By adding more information at least for the first three levels of the Information Model, it makes the SOP Instance more understandable. Attributes which identify the SOP Instance for human beings and allow the image to be displayed with the correct window settings are:

- Patient Level: Patient's Name, Patient ID, Patient's Birth Date, Patient Sex

- Study Level: Study Date, Study Time, Referring Physician’s Name, Study ID, Accession Number
- Series Level: Series Number, Manufacturer, Institution Name
- Image Level: Image Number, Image Type
- Presentation Settings: Window Width, Window Center

The attributes listed above are in most cases Type 2 attributes (must be supplied, but may left empty), or Type 3 (optional).

2.6.2 Specialized Images Types

The generic format described above is used in every image SOP Class definition, but depending on the type of modality, it is extended with dedicated information about the acquisition, etc. The number of specialized image types is growing by the addition of new types of modalities. Currently, the following modalities have an image Storage SOP Class definition in the DICOM standard:

- *Computed Radiography IOD*, used for the traditional Radiographic systems working with phosphor plates to be read by systems such as a PCR. Also the Thoravision system belongs to this group of modalities. Images created by this type of modality contain no extensive information about the acquisition and positioning. In most case there is no relation (spatial or temporal) with other images in a series.

- *Computed Tomography IOD* for CT scanners. For this type of modality the positioning information is important, for processing the image stacks, to create views in the three dimensional space represented by these images.

- *Magnetic Resonance IOD* for MR systems. Besides the same positioning information as for CT scanners also extensive information about the acquisition protocol is given.

- *Nuclear Medicine IOD* for cameras tracing radioactive isotopes. Contains special image format and acquisition information for this dedicated type of modality. Images are in the multiframe format.

- *Ultrasound IOD* for Ultrasound imaging equipment. This type of modality contains details about the positioning and the acquisition of the image. Images can be in colour format, and can use the multiframe format.

- *X-Ray Angiographic IOD* for the digital cardio and vascular systems. This format can capture a run in multiframe format or single images. Inside a run a description of which images are a mask for image subtraction can be added. Extensive information about the equipment positioning and acquisition is given to allow the image data to be processed.

- *X-Ray Radiofluoroscopic IOD* for systems such as a Angiographic system, but making use of a table with a column instead of a C-arm. The major difference is the way the positioning of the pieces of equipment is described.

For each of the IODs, a list of modules is described in the DICOM standard. The use of a certain module is sometimes dependent on a condition or capabilities of a certain system. The modules are selected from either a group of common modules, used for all Storage SOP IODs, or specific modules for a single type of IOD only.

The specific modules contain the special attributes for that type of IOD. These modules, and sometimes the individual attributes, redefine and extend ones defined for the generic IOD.

In PS3.3 of the DICOM standard the IOD, IOM and Attributes are listed. Annex A contains the list of IOMs for each IOD. In Annex C.7 the common modules of Image IODs are described, in Annex C.8 the modality specific modules. Annex C.9 through C.12 define a number of modules which can be added to a image object, such as an overlay module, Value Of Interest (VOI) information, etc.

2.6.3 Secondary Capture Image

The Secondary Capture SOP Class is a special image SOP Class. It is intended to be used for the storage of non DICOM formatted images inside a DICOM environment by converting them to the DICOM format. In this way image nformation from not DICOM compliant systems can be merged with image information of DICOM compliant systems belonging to the same study.

This SOP Class includes images captured from frame grabbers, film digitizing equipment, etc. The Secondary Capture IOD contains no details about the modality and acquisition of the image data. It only gives details about how the image data was captured.

The IOD allows the image to be handled like any other modality. In a number of cases it contains only the *gray level values* of a screen capture which can only be displayed “as is”. For instance, a image made with frame grabber only contain the gray levels in it pixel matrix (a “photograph”).

But, in other cases, it contains a real pixel matrix which needs a pixel value to gray level conversion, allowing manipulation of the representation. This makes it possible to use the Secondary Capture SOP Class to store image data from modalities for which *no standard IOD is available*. This requires the removal of all acquisition, positioning and other modality related information. Only patient, study, series, overlay, and other additional information is available.

The format allows the possibility to save a screen dump of a modality. This method has the drawback that the image matrix, in most cases, is not square and contains burnt in patient and other information which is duplicated by the standard attributes. When displayed without special measurements, it will display the image with a reduced format (in most other cases the image display area is square) and show patient information twice, etc.

2.7 Image Processing Pipeline

An image processing pipeline describes the processing steps which translate the acquired information (by X-Ray, MR, Ultra Sound, etc. equipment) to an image presented on a video display or a film sheet. Some of the processing steps depend on the acquisition system, others improve the presentation by enhancement, or use a series of acquired information to create derived images (subtraction, 3-D images, etc.). The following processing steps can be distinguished:

- *Acquisition processing steps* which include conversion to digital data, corrections, reconstructions, etc. These steps are in most cases performed by the acquisition system.
- *Intermediate processing steps* to enhance the presentation or create derived images.
- *Presentation processing steps* resulting in an image being displayed or printed.

A number of the processing steps are performed by the acquisition system. Other processing steps can be executed on a different system in a distributed environment. In this case a transfer of the information is necessary. This requires a definition of the information and a protocol between both systems. Two types of information exchange can be defined:

- *Processed Image Data* which only needs proper conversion to gray levels for display.
- Image Data suitable for further processing steps together with processing parameters. This group can be split into:
  - using processed image data or
  - *Raw Image Data* which is not suitable for display without the intermediate processing steps.

The different types of processing steps and transferred image data are shown in Figure 2-6.

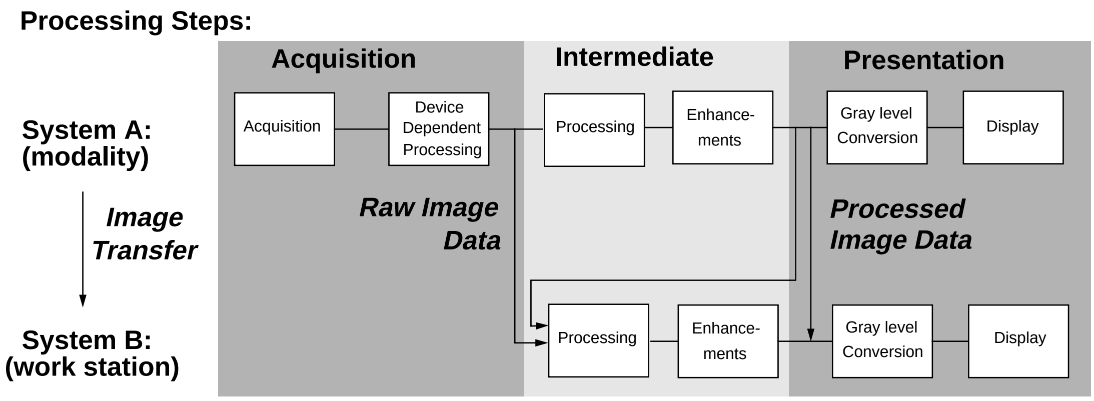


Figure 2-6: Processing Step and Image Data Types

Note: The acquisition processing steps and intermediate processing steps are not discussed in this document. Only the presentation processing with processed image data is described. The current version of DICOM image definitions allows only transfer of image data in that stage of processing.

2.7.1 Raw Image Data

The exchange of image information in the image processing pipeline containing raw image data needs extra care. Additional processing is required for a correctly presented image.

Processing for this type of images includes enhancements such as digital subtraction, filtering certain frequency domains or combining parts of images to a larger single image.

For this type of processing the image data has to be accompanied with information of proceeding processing steps which must be reversed, processing parameters and hints for the steps to be performed, additional acquisition and positioning information, etc. The SOP Instances used for this type of image are not intended for general use, so a Specialized or Private SOP Class is necessary. Information is only disclosed to parties involved.

2.7.2 Processed Image Data

Processed Image Data is discussed in more detail in this document because the current interoperability between producers (such as modalities) and consumers (such as review stations) is based on the information transferred inside this type of images.

Processed Image Data require no further processing steps except the presentation processing steps which results a *correct conversion to gray levels* to ensure:

- correct representation of the acquired information independent of the display device used and
- (clinical) application dependent selection of the information to provide the best possible diagnosis.

2.7.3 Perceptual Correct Image

The correct representation is an important factor in preventing the wrong interpretation of an image when displayed on different systems, with different methods (video screen versus film) and a variety of environments. The gray scale conversion must take care of all the non-linear behaviour of the display device, environment and human perception of the human eye; see Figure 2-7.

The result of the presentation processing steps is a *perceptually correct image* as interpreted by the user of the display system.

The input of this presentation process requires that the preceding processing steps creates pixel values on such a way that all the *pixel values have a meaningful value with respect to each other*. These values depend on the type of system or/and the application of the image data.

In the case of X-Ray systems the absolute values of the pixels is not significant, only the relative value is used for conversion to gray levels. For CT and MR images the pixel values are an important factor for clinical use and must be presented together with the displayed image. The processing steps must take care to supply correct scaled pixel values.

2.7.4 Conversion and Selection of Pixel Values

In a number of situations the pixel values supplied by the preceding steps (acquisition and intermediate processing) have to be used for further processing. This processing sometimes requires an other relation between the pixel values as expected by the presentation processing

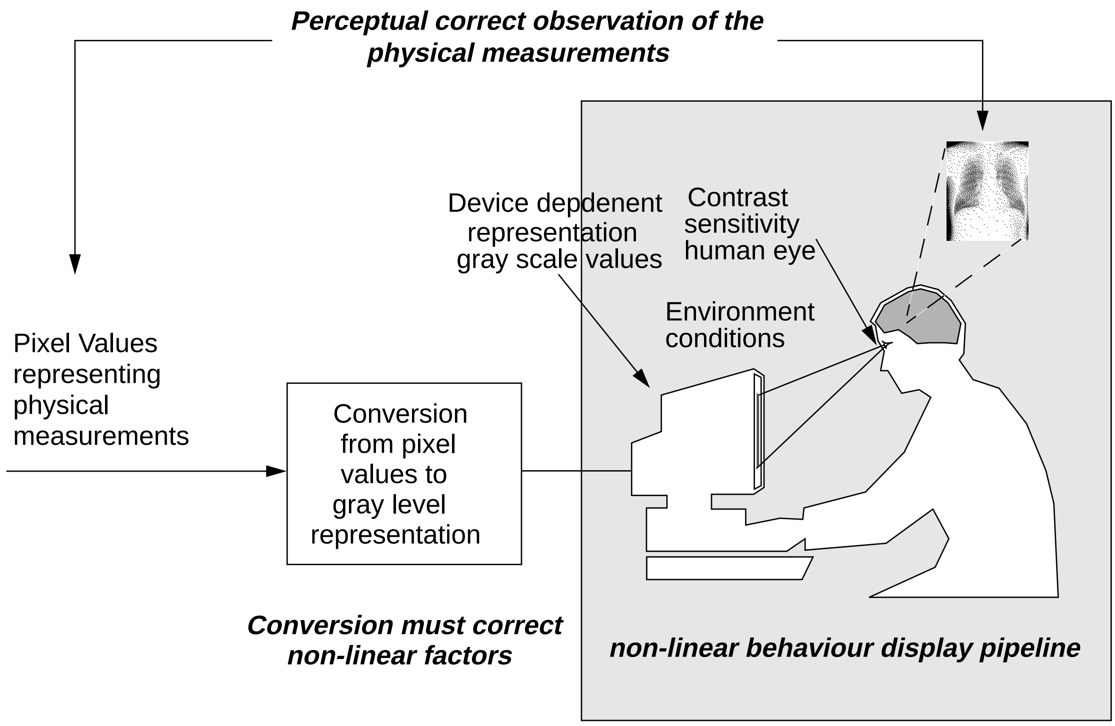


Figure 2-7: Display Pipeline

steps. For instance the relation between the pixel values is on a logarithmic scale and not proportional to the physical measurements. Before the conversion to gray levels can take place a correction step must take place, based on values supplied together with the pixel values; see Figure 2-8.

For some clinical applications the display of the acquired information has to be adjusted to a subset of the full range of pixel values, dropping meaningless pixel values. The result is the display of the relevant pixel values using the full gray level scale. This subset is called a *window,* and can be specified by its center value and the size of the window.

The above conversions and selections depend on the type of systems and the application of the image data. Some systems already apply these adjustments to the image data before transfer. Other systems transfer the original image data with a description of the functions to be applied by the viewing system. In the first case no re-adjustment is possible, this is suitable for system producing always images for a single type of application, e.g. CR images (Thoravision, PCR).

Systems with more then one clinical applications the display only, should use the original pixel values to prevent the loss of information due to already applied conversions. It also allows the user of the viewing system to change these settings to values for a better understanding of the image presented. Examples of these changes are the *contrast/brightness* settings for X-Ray images and the *window width/ window center* for CT and MR images which all effect the window selected from the pixel value range, see Section 2.7.10 “Gray Level Conversion Step”.

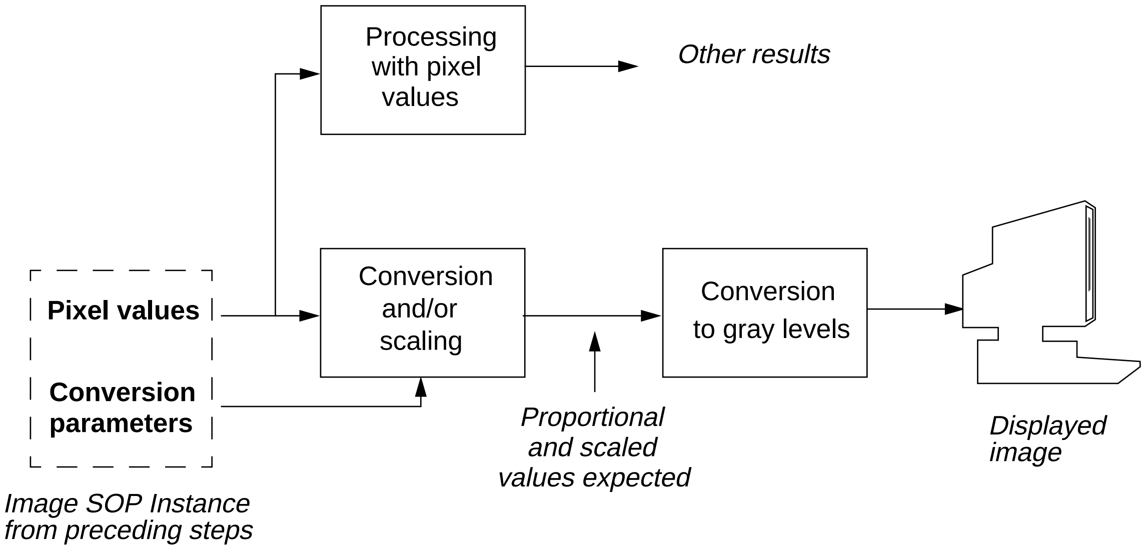


Figure 2-8: Pixel data with Conversion of Pixel Values

2.7.5 Presentation Steps

The presentation steps convert the pixel values to a displayed image on video screen or film. These steps have to take the following points into account:

- The pixel values can have no semantically correct relationship or value (non-linear, not scaled, etc.).
- A sub-range of pixel values must be presented.
- The representation of the pixel value on the video screen or film must be perceptual correct.

For an overview of the presentation processing see Figure 2-9.

The first two functions depend on the contents of the image information and have to be stored with the SOP Instance. The last function is device dependent. The result of the first two functions must result in a range of gray level values that ensure that the result of the device dependent correction is equal on different systems. These two or three functions have to be applied to the pixel data in **one processing step** to prevent the loss of image quality due to accumulation of rounding errors in each function.

2.7.6 Processed Image Requirements

The conclusion from the previous section leads to the requirement that the Processed Image Data exchanged between systems must contain sufficient information to result in equivalent images when used on different systems each with their own device dependent correction. This information must be structured in such a way that it allows an implementation to combine all the necessary functions in one step.

Internal Use Only

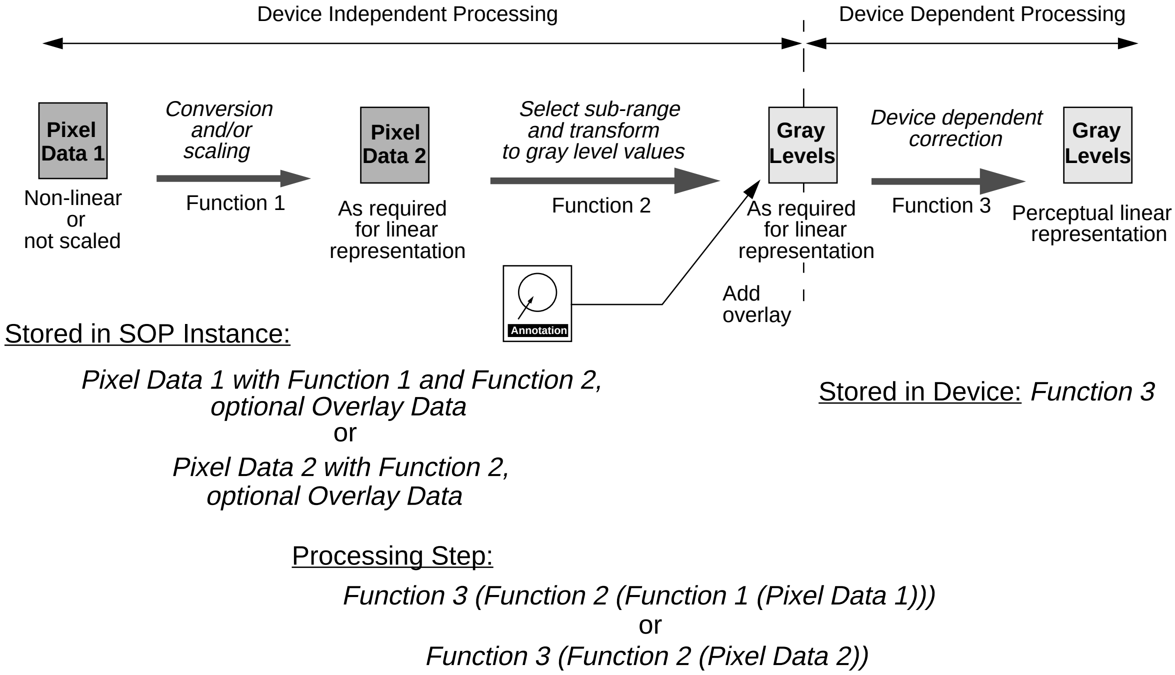


Figure 2-9: Image Presentation Processing Steps

In DICOM there are currently two ways of describing the functions:

- for a linear function the necessary factors are given ($y = \boldsymbol{a}x + \boldsymbol{b}$),
- for non-linear conversion the mechanism of a *look-up table (LUT)* is available: for each range of input values an output value is stored. An example for a non-linear conversion are the smooth rounded curves for the top and bottom of a window; see Figure 2-12.

The latter way of describing a non-linear conversion has a major drawback. It introduces abrupt changes in the output value when the input values passes a range boundary. If a sequence of these look-up tables is being used, it downgrades the quality of the presented image. It also does not allow the composition of the functions to be performed in one step (to prevent loss of the quality of the presented image). The lack of possibilities to specify a non-linear function in the form of a mathematical formula is a major drawback of the current DICOM definitions.

2.7.7 DICOM Processed Images

For images transferred with DICOM a number of modules containing information for the Presentation Step as described above are defined:

- *Image Pixel Module* which contains the Pixel Sample Values, stored in a stream of Pixel Data, with attributes which describe the encoding and format of the Pixel Matrix.
- *Modality LUT (MOD LUT)* with a function description for rescaling or conversion (function 1 from Figure 2-9).
- *Value of Interest LUT (VOI LUT)* with a function description to select a window in the range

of pixel values (function 2 from Figure 2-9).
- *Overlay modules* which add graphical information to be display overlaying the displayed image.

Depending on the required processing the MOD LUT module, the VOI LUT module or both may be present beside the Image Pixel Module. A VOI LUT is very likely to be present to enable a correct display of the image for a certain clinical application.

Overlay information can contain lines and circles to show the bounders of area of interest, or a bitmap with character strings to annotate the information in the image displayed. This information is supplied as a separate entity. When this information is added to the pixel data (‘burnt in’) very severe limitations with the use of the image data are applied. In this case the value of some of the pixels is changed to the overlay value. This will disallow processing with the pixel value and also prevent a display system to exclude the information in the displayed image area when image data without overlay is requested by the user of the system.

2.7.8 Decoding Step

The decoding process of Pixel Matrix from the Pixel Cell streams uses two groups of attributes from the Image Pixel Module; see Figure 2-10:

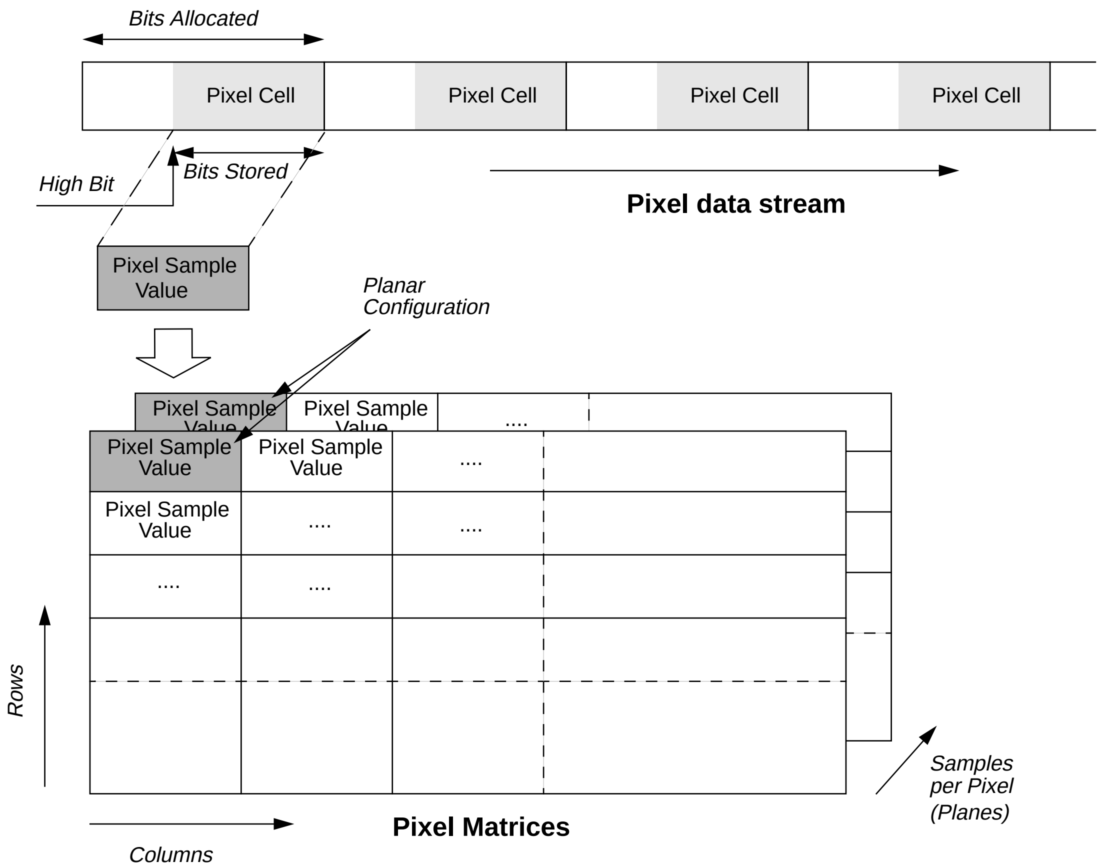


Figure 2-10: Decoding Pixel Data

- Bits Allocated, Bits Stored, and High Bit to decode the Pixel Sample Values from the Pixel Data Stream, and
- Rows, Columns, Samples per Pixel and Planar Configuration to order the Pixel Samples in the Image Matrix.

A single Pixel Value Sample is contained in a Pixel Cell. Besides the Sample Value, other Pixel related information such as overlay indications can be stored in the space not occupied by the Pixel Cell. The sequence of Pixel Sample Values are stored in the pixel matrix with the dimension in the Column, Row, Samples per Pixel attributes. When more than one plane is used, the Planar Configuration describes how the Sample Values are ordered in the Pixel Data Stream.

The attribute *Pixel Representation* contains the data format of the Pixel Sample Values: signed or unsigned integers.

2.7.9 Normalization Step

After decoding the conversion to meaningful pixel values takes place if required for a certain type of Image SOP Class. The result is a normalized range of pixel values suitable for conversion to gray levels and according to what is expected for that type of modality and clinical application. For example, for X-Ray systems the received X-ray intensity is proportional to the pixel values, for CT systems the pixel sample values are converted to the Hounsfield scale, etc.

In case only a rescale can be used, two attributes are necessary to describe the function: *Rescale Slope* and *Rescale Intercept;* see Figure 2-11.

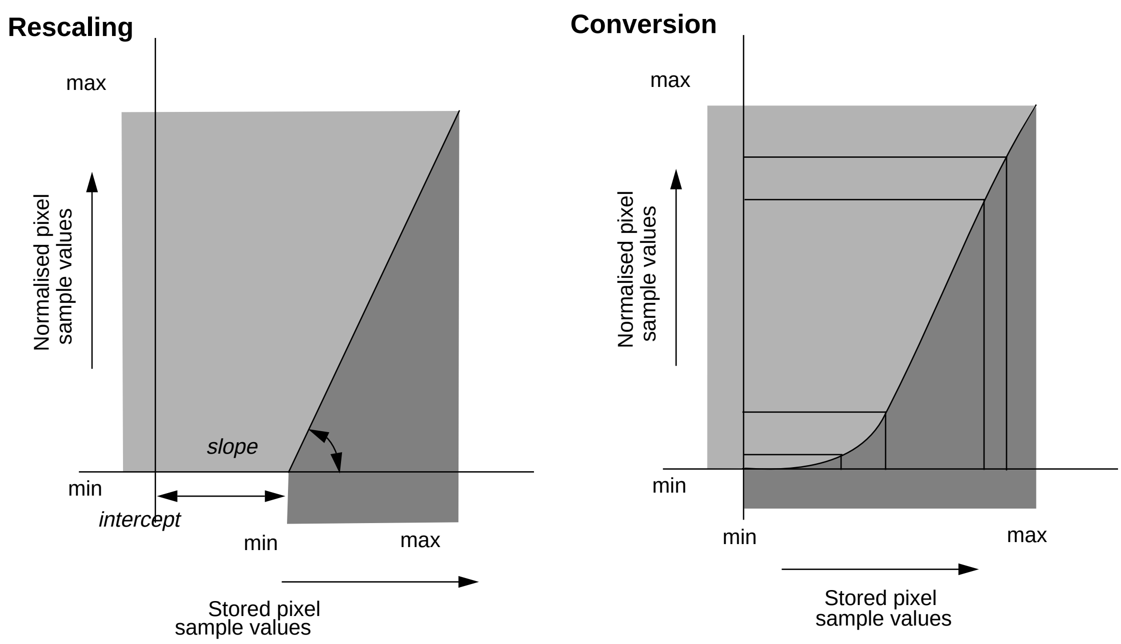


Figure 2-11: Modality Dependent Rescale and Conversion

When a non-linear conversion has to be applied the look-up table mechanism is used.

2.7.10 Gray Level Conversion Step

In most cases the full range of normalized Pixel Sample Values has to reduced to a sub-range which contains the valuable information for the application of the image. On its turn, this has to be converted to gray level representation. For some applications more then one sub-range is selected. In that case, the separate windows have to be mapped to different sub-ranges of gray levels.

The window is described by two attributes or another look-up table. The two attributes allow only a linear conversion of the selected range(s), non-linear conversion can be achieved by means of a look-up table. When no look-up table is used the window(s) are described by the center pixel value (Window Center) and the size of the window (Window Width); see Figure 2-12.

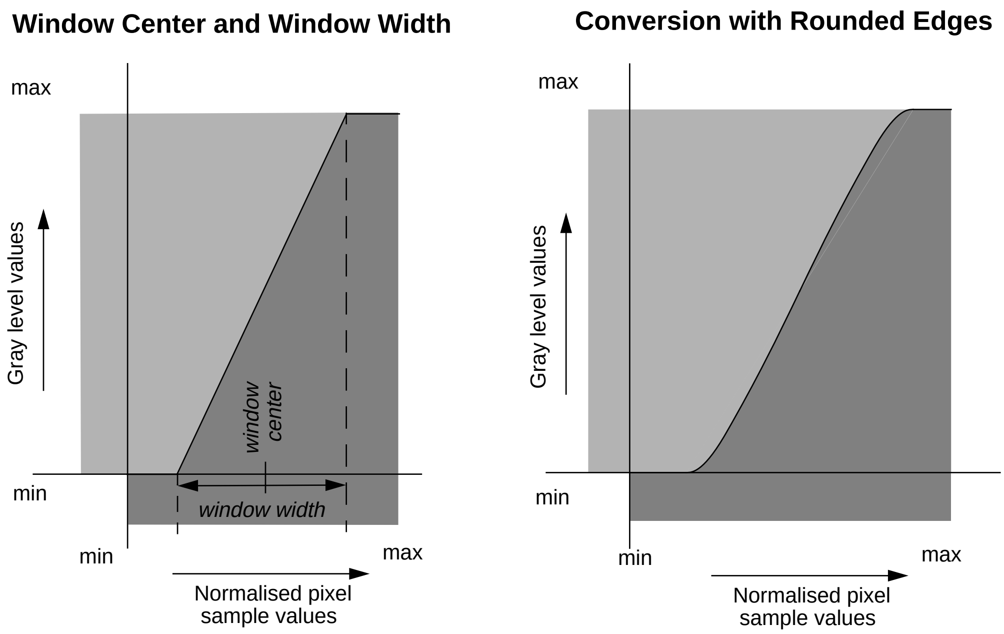


Figure 2-12: Gray Level Conversion

Depending on the value of the *Photometric Interpretation* attribute the minimum and maximum gray level are mapped to white and black or conversely. The value MONOCHROME1 means that the minimum value is mapped to white and the maximum to black, whereas MONOCHROME2 means the minimum value mapped to black and the maximum value to white.

The Window Width and Window Center attributes are used on systems such as CT and MR. Traditional X-Ray systems use a *contrast/brightness* mechanism to adjust the window. DICOM has no attributes for this mechanism, but there is a translation possible between both representations; see Figure 2-13.

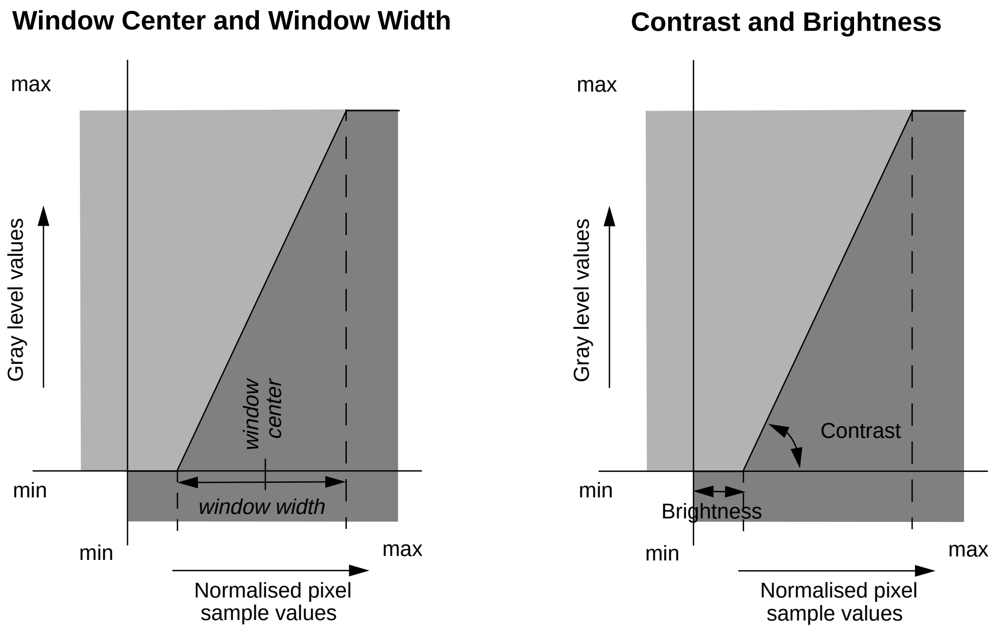


Figure 2-13: Window Attributes vs. Contrast Brightness

2.7.11 Overlay Step

On or more separate modules specify where the overlay bitmaps must be positioned on the image matrix. Optionally, the colour of the contours can be specified. The bitmaps belonging to these images can be included in the Pixel Data Stream (unused bits in Pixel Cell) or in separate Pixel Matrices in the overlay modules.

The corresponding positions in the pixel matrix with have to be set to values for the overlay contour before sending the gray level image to the next step.

2.7.12 Device Correction Step

The converted Pixel Sample Values have to be corrected to reach a perceptually correct understanding of the image. The corrections are device and environment dependent and have to be determined by calibrating the device and entering the calibration result as a correction function to be applied to the gray level values.

To prevent the loss of quality due to executing the two or three functions (see Figure 2-9) separately, all functions have to be combined into one look-up table which converts the stored pixel values directly into a perceptually correct representation. This, however, will only work when all the functions are mathematically described and not stored in look-up tables. There is still some effort needed to implement mechanisms, in the DICOM standard, which allow the use of non-linear conversion without using look-up-tables.

2.8 Application of Image Data

Image SOP Classes are in general generated on modalities or processing workstations. The result are displayed on review stations or printed on film. Storage systems buffer the images in between, or archive these images for reference at a later stage. The requirements for the information comtained in the Image SOP Instance differ for each type of system which is involved in the lifecycle of an image.

By exchanging data between systems, each system may have a different view of the information, but care must be taken that **all** the information in the Image SOP Instance is transferred between each system involved. Even when a system in a chain is not using the information, another system which is using the information in a later stage is relying on the complete passing of the information in the whole chain, see Figure 2-14.

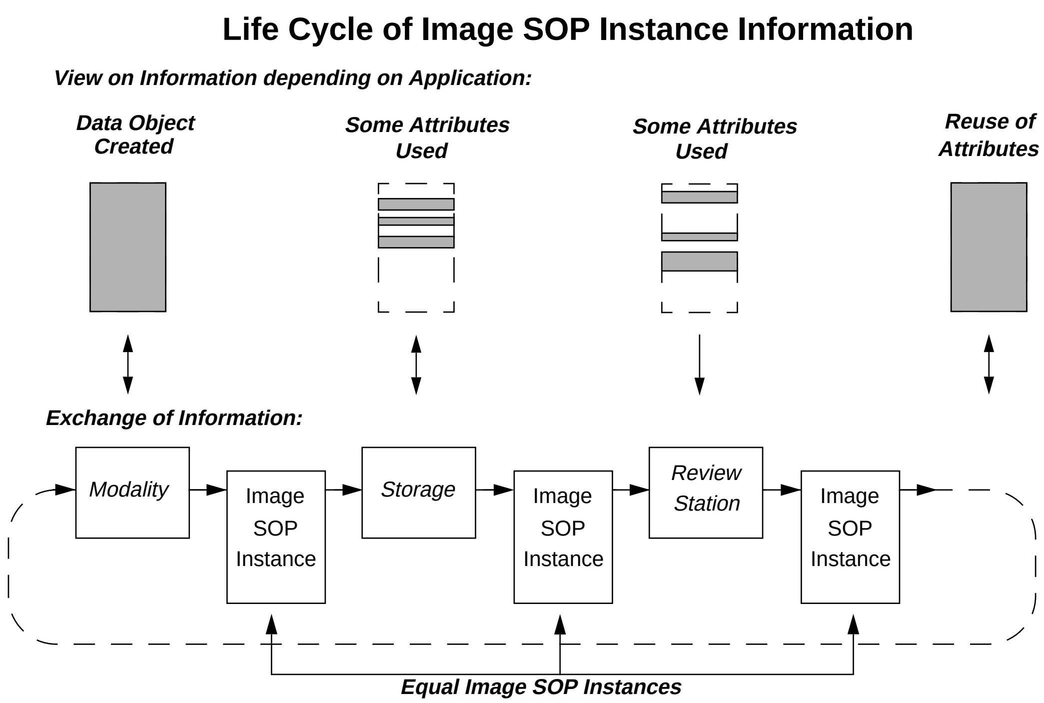


Figure 2-14: Life Cycle of Image SOP Instance Information

2.8.1 Image Storage Systems

Image storage systems use a number of the identifying attributes to store the Image SOP Instances. In the first place, these attributes are used to collect all image data belonging to the same study. The Study Instance UID is the key attribute. But when this is not consistently used, other attributes have to be used (such as Patient ID, Accession Number, etc.).

Secondly, a number of attributes can be used by systems which want to query Image SOP Instances from the storage system. Primary keys in this case are the Study Instance UID and Series Instance UID. Queries based on Patient’s Name, Study Date, etc., are also possible.

For the use of storage systems a meaningful value for these type of attributes must be supplied by the creators of Image SOP Instances. The attributes which contain the acquisition parameters and image data are stored but have no meaning to a storage system.

2.8.2 Review Stations

A review station is mostly used for displaying the images made on one or more modalities. It receives or queries Image SOP Instances, from the storage system, belonging to a certain study. It will display the image together with information about the patient, acquisition settings, diagnostic information, etc.

The settings for the presentation processing, as selected on the modality, are very important. When the presentation steps are processed in the correct way, the resulting display should be equal to the original displayed on the modality.

For accessing additional information from other systems, identifying attributes such as the Study Instance UID are used.

2.8.3 Image Processing Stations

Workstations capable of processing the image data have additional requirements. Acquisition and positioning parameters are required to perform the additional processing steps. Depending on the type of processing the input is a stack of images in processed or raw format. In this case the relation between the images is important to order the image data in the correct way for the processing.

Results of this processing is new pixel data which is stored into a new Image SOP Instance which has its own life cycle, with in most cases relationships pointing to the original data used for this image.

2.8.4 Reuse by Modality

A final category of applications is the system which created the Image SOP Instance. In this case old image data can be used when a new visit of the same patient, with the same examination type, takes place. The acquisition and positioning data can be reused, or the image can be displayed as a reference for the new examination.

2.8.5 Application Categories

As shown above the requirements of the individual systems in the lifecycle of an Image SOP Instance are different. When a system produces image data it must be aware that all the systems which are intended to be a part of the lifecycle must receive sufficient information. The conformance statement must describe, for which type of system the information is adequate and which processing can not be applied.

To help the selection of these types of systems, the requirements can be divided into *Application Categories*. A higher numbered category includes the lower numbered category. The following categories are defined:

1. *Storage category* - only identifying attributes required.
2. *Display category* - viewing and printing images, only the attributes for correct presentation required.
3. *Simple processing* - distance and volume measurement, requires some more attributes

describing what information is contained in the image.

4. *Complex processing* - MPR and Image subtraction, requires specific information about positioning and relationships.

The DICOM Standardisation Committee is considering a classification as listed above to be included in the Conformance Statement.