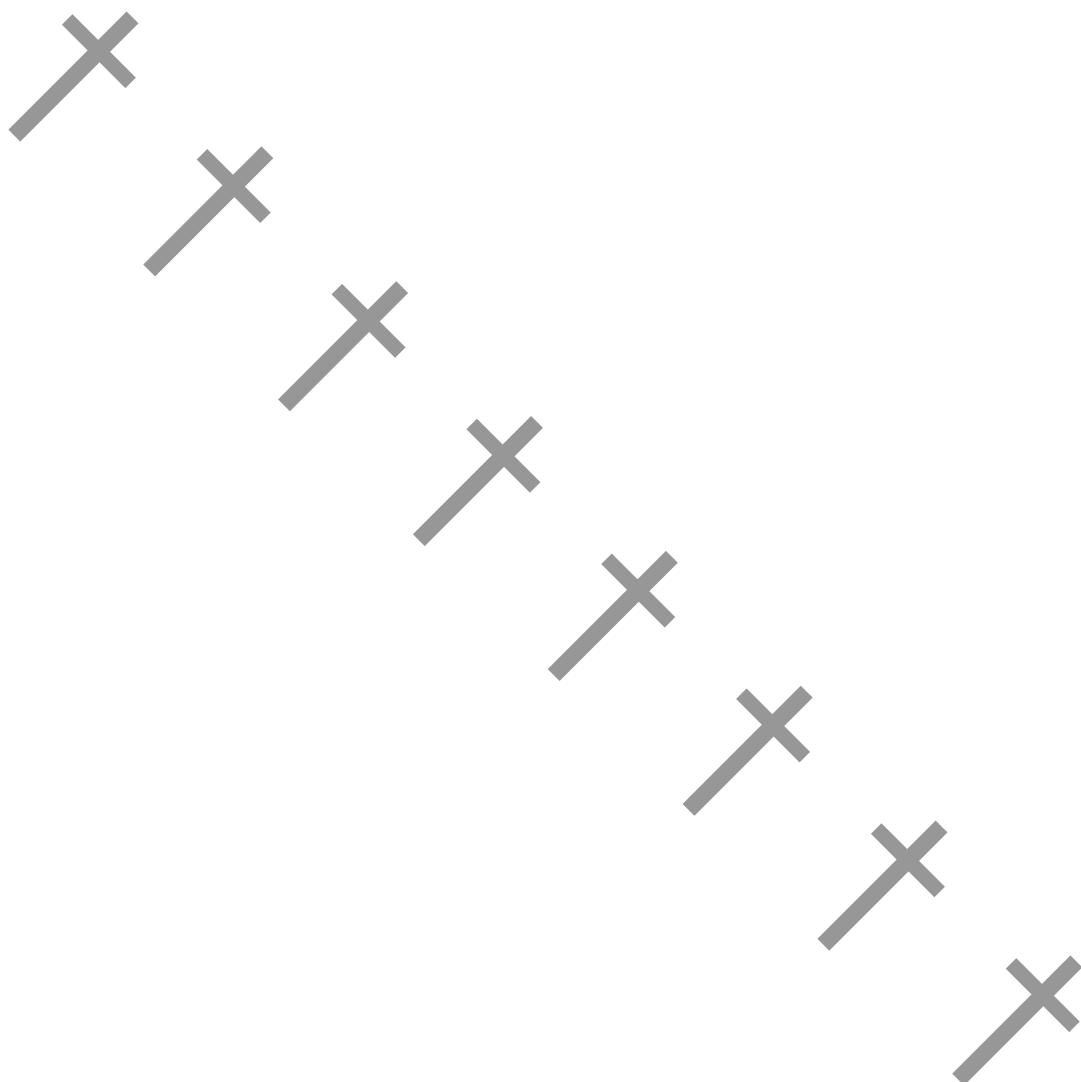


第三部分:

DICOM 通讯



第七章

DICOM SOP：基础

DICOM 网络像胶水一样将各种医学成像系统结合在一起。当我们还在以为 DICOM 仅仅是一种医学影像文件格式时，实际上这个标准就已经比我们想象的宽泛得多了，它直指临床流程以及管理图像文件格式以外的领域。现在，整个数字医疗领域都被在计算机网络间传送和交互的 DICOM 对象所创建和占据。

很有趣的是，DICOM 网络早在计算机网络存在之前就已经涵盖在标准中了。标准中 PS3.9 部分（消息交换的点对点通讯支持）就介绍了使用旧式的针式信号线来连接 DICOM 设备的方法。所有这些，包括 PS3.9 自己，都已经与现代网络硬件及协议一起销声匿迹了，这些消失的标准和协议已经成为了现代 DICOM 数据交换的基础。

坦白地说，当前 DICOM 也使用以 TCP/IP 协议为基础网络，和您发送电子邮件或看在线视频使用的网络协议一模一样（图 31）（TCP 即传输控制协议，IP 即因特网互联协议）。TCP/IP 恰好适应了所有种类的软硬件并且提供我们所需的最基本的网络功能：从一个端口/IP 地址向另一个发送信息（以字节的序列形式）。DICOM 只是在这基础上增加了我们即将去探寻的内容，也就是它自己所使用的网络语言（应用层）。

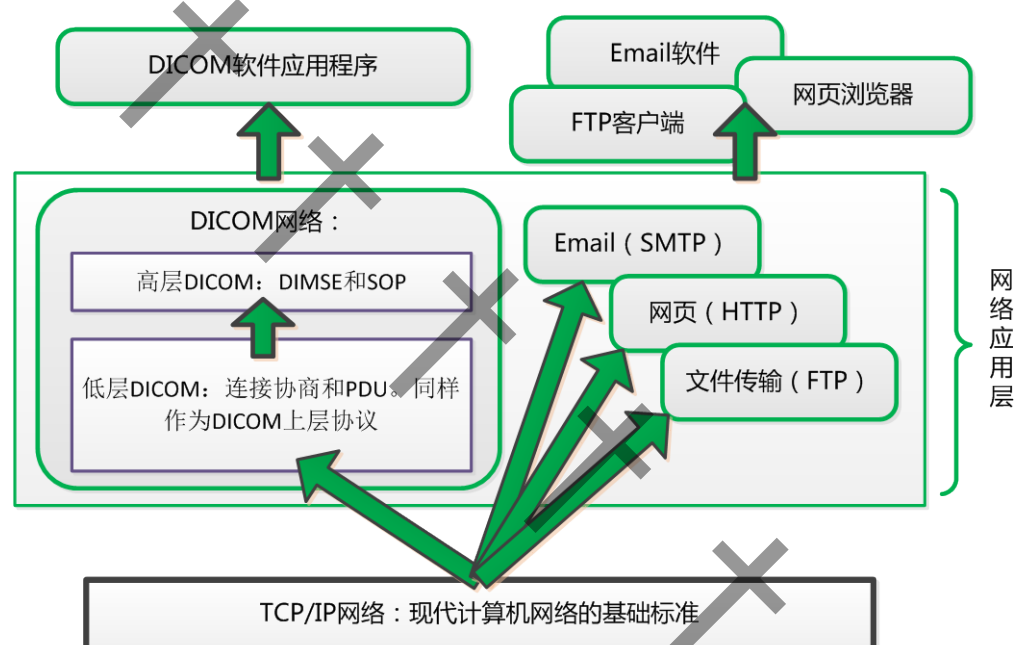


图31 DICOM和流行的网络协议。DICOM网络应用层通过DICOM规定的协议扩展了基本的TCP/IP功能

这门语言由高层服务组成，DICOM 消息服务元素（DICOM Message Service Elements DIMSE），这章的主题，以低层 DICOM 联系原语为基础（DICOM 上层协议、简短的 DICOM-UL、更多技术探讨等内容会在第 9 章介绍。）

虽然实属技术领域，但是所有 DICOM 网络中的概念都很直白，即使你不是一位 IT 专家，你也可以学懂这些内容。此外，了解 DICOM 网络的原则非常有助于提高你对 DICOM 和 PACS 的理解以及你处理相关项目的的能力。基于以上原因，建议您好好阅读本章内容。

7.1

在 DICOM 网络上识别装置

DICOM AE（见 5.3.8）通常相当于连接在网络上的 DICOM 设备所使用的 DICOM 应用程序；比如，一个 DICOM 服务器（归档服务器）、成像工作站、胶片打印机或图像获取设备（影像设备）。这四个实例覆盖了临床环境下约 95% 的 AE 类型，因此我们将使用它们作为我们的标准范本。

每个网络上的设备都应该会有网卡和它自己的 IP 地址，只有这样别的设备才能够找到它。除标准的网络设置之外，DICOM 为每个 AE 规定了它自己的 DICOM 名称，即“应用实体名（Application Entity Title AET）”。在 DICOM 中，AET 使用 AE VR 进行编码的。你可能记得 5.3（表 2）中提到的 VR 表，AE 能够由至多 16 个字符组成。实际工作中的 AET 命名规则通常要么使用应用程序的名称（比如，PACSSERVER（PACS 服务器的意思））要么使用计算机名/位置（DRBOBOFFICE），并且最好别有标点符号和空格；名称只用大写且要尽量避免含混。这会使得 AET 更加方便维护以及识别。不幸的是，某些 PACS 公司在滥用应用实体名方面是出了名的。当 PACS 软件安装在你的场地时，请确认 AET 的设置是清楚的和一致的。

AE 相当于 DICOM 应用程序

注意（这点很重要）顾名思义，AE 名通常用来标明应用程序，而不是计算机。虽然在许多情况下你的每台设备上只有一个 DICOM 程序在运行，因为没有任何理由拒绝你在一台计算机、服务器、工作站、DICOM 打印机服务器等等设备上安装多个 DICOM 程序。事实上，这是一种非常常见的真实情况。在这种情况下，每个 DICOM 应用程序会有不同的 AE 名，其它 DICOM 设备不会与整个计算机交谈，而只是与每个特定的应用程序交流。

当 DICOM 网络用 DICOM 格式交换消息和数据时，这一切都是在 AE 之间完成的。如果你对网络很熟悉，那么你可能知道 TCP/IP 总是发送数据到每个计算机的一个特定端口¹。端口是从 0 到 65535 的数字。比如，你的网页浏览器（基于 TCP/IP 的超文本传送协议 hypertext transfer protocol, HTTP）使用 80 端口，而你的 email（也是基于 TCP/IP 的简单邮件传输协议 simple mail transfer protocol, SMTP）使用 25 端口。DICOM 默认端口是 104，但是当你安装你的 DICOM 软件时，你能将它修改为任何一个其它的可用端口号，只要：

1. 你需要让所有连接在该网络环境下的装置保持一致，只要接收装置与发送装置使用一致的端口
2. 端口号已经被其他的应用程序所使用了。比如，DICOM 使用 80 端口就不算是一个明智的选择，因为 80 端口通常是给网页浏览器使用的，而不能让其它人使用。如果你不喜欢 DICOM 的 104 端口，可以尝试用更大的数字（比如 10000 往上）；因为较低的端口号已被占用的概率较大。

当你的一台计算机上有许多 DICOM 应用程序时，端口号就会显得非常好使了。如果说不同的 AE 名代表了不同的 DICOM 应用，那么不同的端口号就起到在网络上分隔这些应用的作用。虽然它们使用相同的 IP 地址，但是每个应用程序只用其自己的端口进行呼叫。此外，一个 AE 能够设置两个端口：一个用来发送数据，另一个用来接收数据。最新的 DICOM 软件提供 AE 的双端口支持功能。

¹ 如果这对你来说是新知识的话，那么请将计算机名想象成街道名，端口想象成房屋编号，而 AE 想象成房主姓名。你必须知道这一切才能给房主发信，对吗？

不可能搞定的连接

如果你正在考虑购买 DICOM 软件，那么请一定确认一下它能够提供给你配置 AE 参数的绝对自由，包括本地（应该与应用程序一致）和远程（应该与参与通讯的其它软件的 AE 一致）。

在 DICOM 软件中打开 AE 设置窗口，你不应该看到任何默认或只读的 AE 名或不能修改的端口号。比如，在近期的一个项目中需要从 X 公司的服务器连接到 Y 公司服务器，我们发现：

1. X 想要使用两个不同的端口，一个端口负责接收、一个端口负责发送，发送端口“必须是” 104 而接收端口“必须不是” 104。
2. Y 想要使用一个端口实现发送和接收。

结果，当 X 需要两个不同端口时，Y 坚持只能用一個，因此根本不可能将两个系统进行连接。对于 X 和 Y 公司这种破烂设计，它们根本无法给予合理解释；同样，它们也完全没有理由坚持使用特定端口号。

这几乎是你所要了解的所有 DICOM 网络设置了。因此总结一下，在 DICOM 网络上配置你的设备，你必须将它们设置成一致的：

1. **AET：合适的字母、最多 16 个字符。**注意简单易懂“CTWORKSTATION1”、“ARCHIVE”或“PACS_SERVER”。
2. **AE IP 地址：**确认已经为这个 AE 而提前预留，并且不会再改变。
3. **AE 端口号：**选择任何（如果不是 104）数字并在所有相连的 AE 上保持一致。如果你不会在同一个计算机上运行多个应用程序，那么请在你整个 DICOM 网络中使用单一端口号（图 32）。

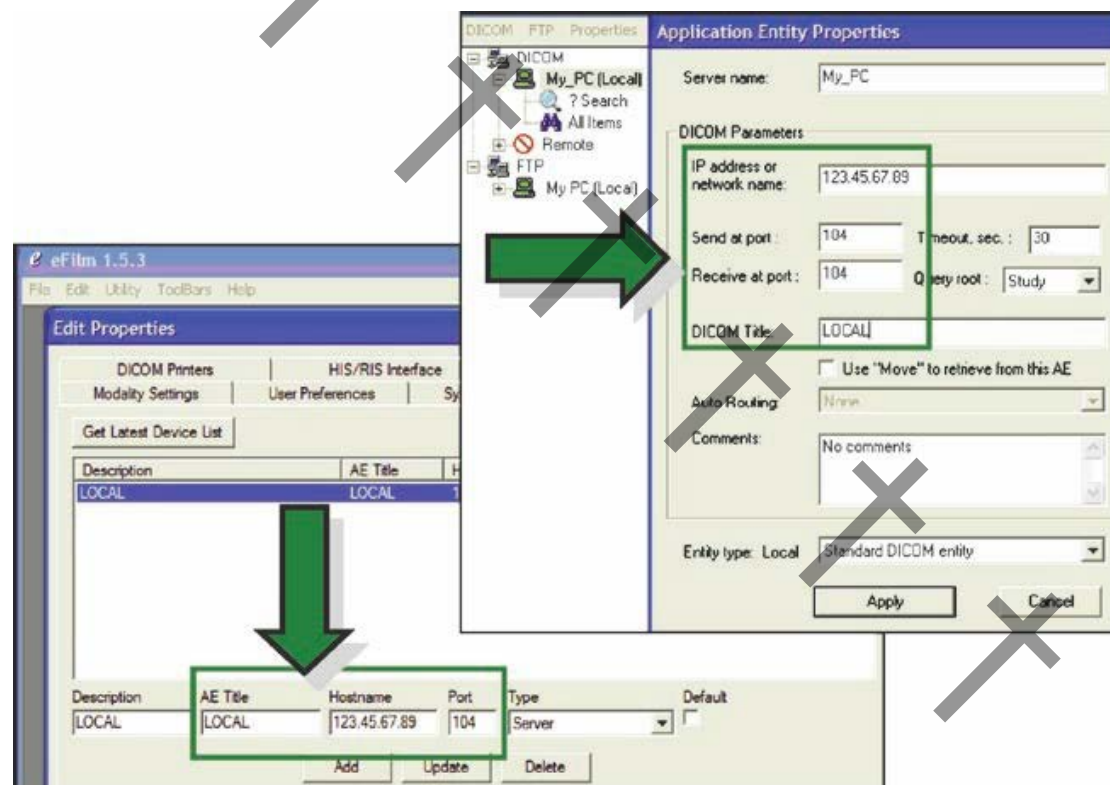


图32 在各类DICOM软件中配置“应用实体”（AE）规格；不同界面，同样的AE参数

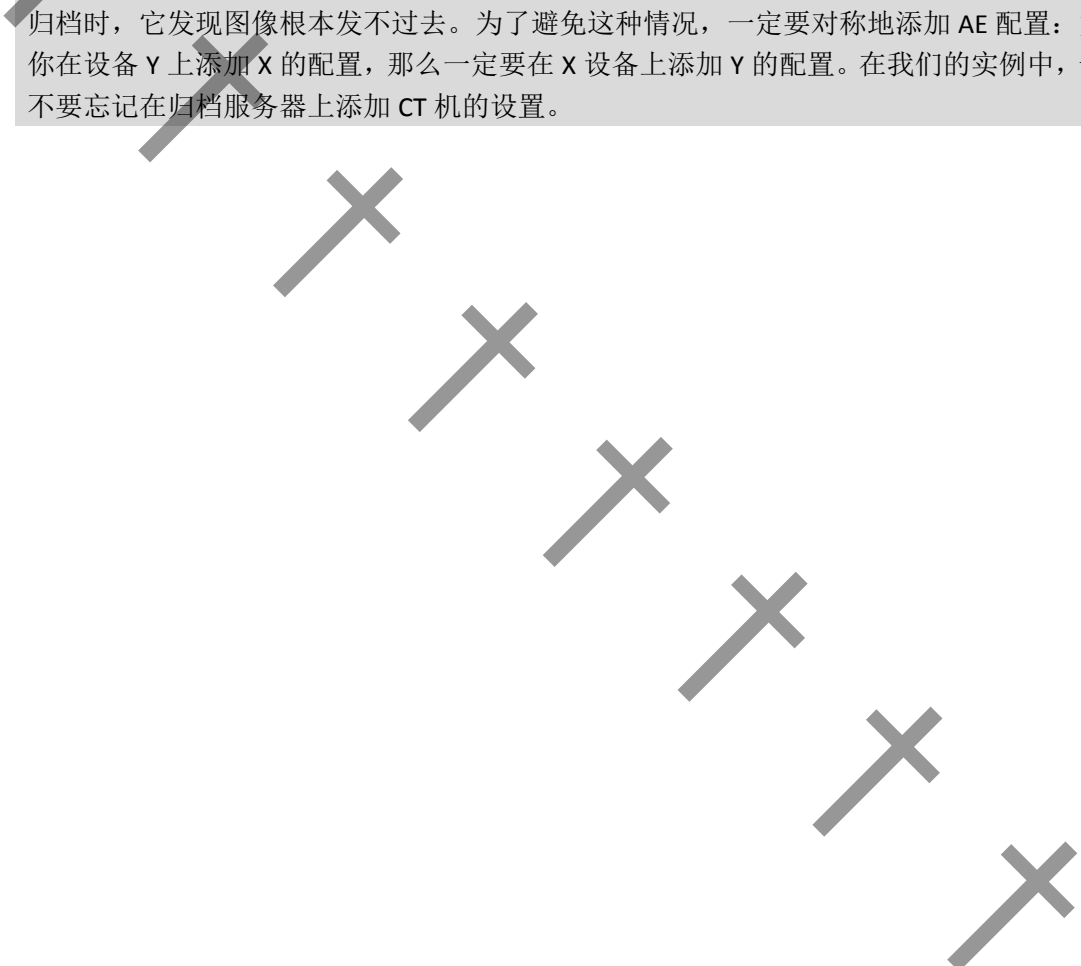
你网络中的任意一个 DICOM 应用都需要设置 AE 属性；所有 DICOM 应用程序都会提供

一些菜单或配置工具来进行设置工作。找找菜单项比如“DICOM 属性”或“增加 DICOM 服务器（节点）”，或者其他相似的地方。AE 设置通常由设备维护人员来完成，也就是 5 分钟的事儿，讽刺的是有时需要折腾几个小时。为什么呢？因为相关支持信息通常不太好找或者不太容易提前预计会用到，可能因为忘记了配置工具的密码；可能因为整个 PACS 不得不重启以及一大堆其它你意想不到的原因。因为所有这些“因为”，设备厂商是很乐于从中收取服务费的，不过就是些孩子都能完成的工作，它们通常要向你收几万块钱。因此，请提前计划 AE 配置，并且确保你了解所有细节——这真的能够既省钱又省时间。

你是谁？

虽然 DICOM 标准没有要求，但是许多 DICOM 厂商会设定额外的 AE 验证逻辑，要求所有建立连接的多个 AE 都要完全了解彼此之间的配置信息。在 DICOM 标准中，如果 AE X 想要连接 AE Y，AE X 需要了解 AE Y 的 AE 配置信息。这很容易理解，因为 X 必须知道在网络上的什么地方（名称、IP、端口）去找 Y。一些设备通常额外需要 Y 了解 X 是谁，即使 Y 从来不会主动与 X 建立任何连接。

这种相互了解的过程也许会在一些看似粗鲁的安全特征（不要与陌生人讲话）中出现，但是事实上它带来了相当大的麻烦：你在 CT 机上设置了一个归档，当 CT 机试图发送图像给归档时，它发现图像根本发不过去。为了避免这种情况，一定要对称地添加 AE 配置：如果你在设备 Y 上添加 X 的配置，那么一定要在 X 设备上添加 Y 的配置。在我们的实例中，千万不要忘记在归档服务器上添加 CT 机的设置。



7.2

服务与数据

DICOM 所采用的数据处理和交换模型是优雅且不失品位的；这种模型是通过 DICOM AE 之间相互提供服务来实现的。一个 DICOM 实体能够请求其他的实体提供服务，被请求的实体就会向前者提供这项服务。在 DICOM 术语中，称请求服务的 AE 为 SCU，称提供服务的 AE 为 SCP。换句话说，为了相互通讯，AE 既可以运行 SCP 应用也可以运行 SCU 应用（图 33）。

SCP 和 SCU 这两个名字表示，所有 DICOM 服务都基于 DICOM 服务类的。DICOM 中的服务类通过数据处理功能来绑定 DICOM 数据。在更严格的情况下，DICOM 服务类通过一个或多个命令来与一个或多个 DICOM IOD 建立联系。以打印胶片为例，DICOM 打印管理服务类（DICOM Print Management Service Class）负责打印（命令）各类图像（IOD，比如 CT 或 MR 图像）。因此，任何 DICOM 打印机都能够提供这个服务（以打印管理（Print Management）SCP 的形式运行）。任何 DICOM 设备发送图像给打印机的活动产生了这个服务请求（以打印管理 SCU 的形式运行）。

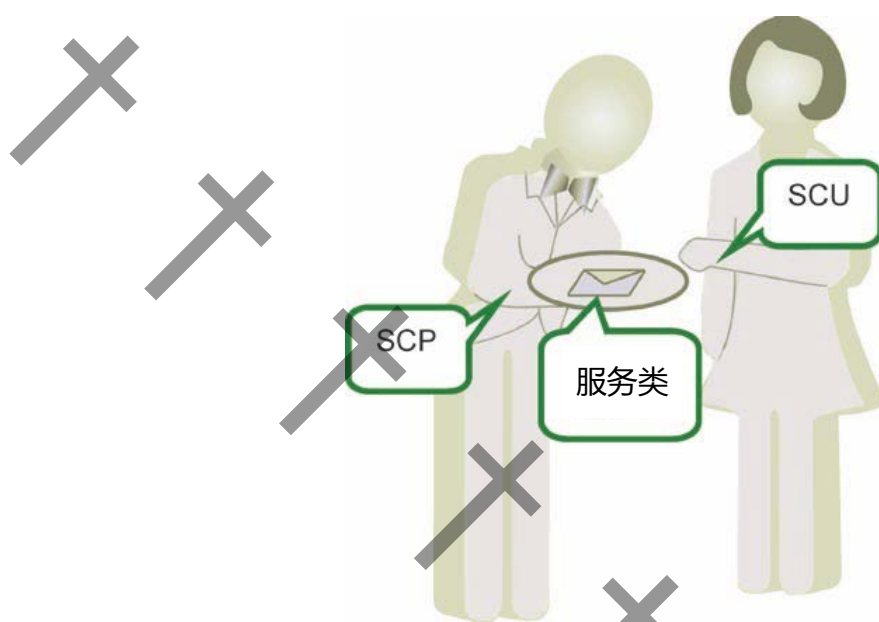


图 33 SCU-SCP 模型

在这一部分，我们将更深入地了解 DICOM 服务的结构和含义。我将会阐明 DICOM 标准在细化服务内容方面正在发挥着巨大的作用，由于这些服务遍布多个 DICOM 卷并且有时候使用的术语可能不一致，因此所有这些服务之间的关系非常容易让人混淆。阅读本部分内容，将有助于您理解以上问题。

7.2.1

DIMSE（DICOM 消息服务元素）服务

由于我们对于 DICOM IOD 的结构已经非常熟悉了，因此让我们仔细看看服务类定义的其他部分：IOD 处理命令（IOD-processing commands）。DICOM AE 之间是如何请求服务的呢？

就像我们人类一样。DICOM AE 之间通过发送服务消息来请求和提供服务信息。这就是为什么所有服务命令在 DICOM 中被称作 DIMSE。DIMSE 协议为 DICOM 服务交换设置了规则，这就是 DICOM 网络的骨干了。因此，每个 DIMSE 服务通常具有请求和响应两个消息组件。请求部分是由 SCU AE（比如，CT 机希望在归档系统上保存一个新的 CT 图像）发送的；而响应则是由 SCP AE 来提供的（比如 CT 归档系统）。

我们可以使用人类逻辑来继续分析一下。为了使消息能够通过网络进行交换，我们需要

记录下他们，并把它们放在 DIMSE 网络“信封”中。我们已经了解了“写信”的过程，即 DICOM 以相同的 VR 规则记录下他自己的网络对象，这部分我们在第 5 章已经了解过。为了区分“服务”属性和“数据”属性，DICOM 数据字典保留了一个 0000 组，专门用作所有的服务标签（这类标签比较少）并且称这些服务对象为“DICOM 命令对象（DICOM command object）”。如此命名是为了与我们之前介绍的“DICOM 数据对象（DICOM data object）”区分开。

比如，我们需要发送一个图像给一个归档；即如果我们需要通过服务传送数据那又会怎样呢？请看图 34。在每个服务对象的结尾都有一个数据集类型（Data Set Type）(0000,0800) 属性。如果这个属性被设置成 0101，那么就意味着服务不需要传送 DICOM 数据；即使这些数据都保存在服务属性中。然而，如果数据集类型标志被设置成了非 0101 的任何值，那么数据对象（Data Object）会跟在命令对象（Command Object）服务的后面，在命令对象之后紧跟着就发送出来。这样，DICOM 服务就像穿梭运输车一样，装载着数据在应用实体（Application Entity）之间传送；数据对象决不可能自己运送自己。

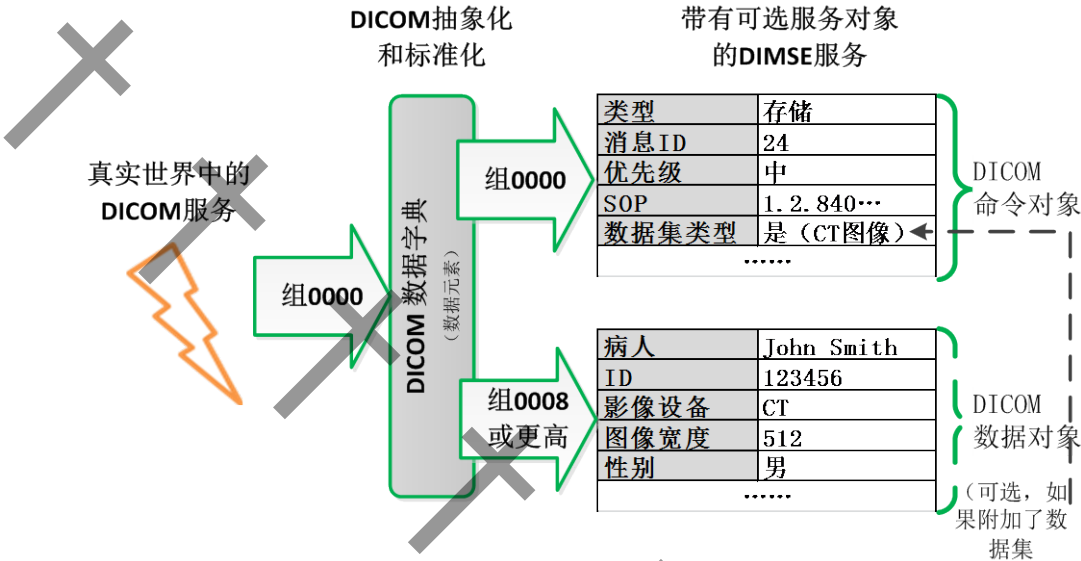


图 34 建立 DICOM 消息服务元素 (DIMSE)

处理复合 (composite) 数据的 DIMSE 服务称作“DIMSE-C”服务；而处理常规数据的 DIMSE 服务称为“DIMSE-N”服务。“C”和“N”这种标记经常会作为服务名称的前缀，放在服务名称之前，比如，C-Store 服务存储 DICOM 图像（这是复合对象）。类似的，带有服务请求的 DIMSE 消息是用“Rq”后缀标记的（比如，C-Store-Rq 就是一个 CT 机发给数字归档，请求图像存储的服务请求）。带有服务响应的 DIMSE 消息使用“Rsp”的后缀标注的（比如，C-Store-Rsp 用作归档向 CT 机发送的答复）。

受够了抽象介绍了吧？实例自然比定义更有效，那么让我们来看看最基础的 DIMSE 服务吧，这可以帮助我们更好地理解它是如何工作的。

7.2.2 简单的 DIMSE 例子：C-Echo

C-Echo 是最简单的 DIMSE 服务。它常用来校验一个 DICOM AE 是否与另一个 AE 建立了连接。因此，针对这个简单的事情建立了 C-Echo 服务，它是最基础和最常用的服务，因为：

1. 只是知晓两个 DICOM 设备通过网线建立了物理连接还不够。
2. 只是实现从一个设备 ping 另一个设备还不够。即使 ping 证明了设备已经连接了 TCP/IP

网络，但是不能证明这就是“DICOM 网络”（或者确实能够运行 DICOM 软件了）。

我们不会无聊到又开始讨论 AE 名、端口和 IP 地址。验证两个 AE 是否配置正确的唯一方法是从其中一个向另一个发送 C-Echo 请求。

表 20 C-Echo-Rq: 请求 DICOM 连通性校验

消息字段	标签	VR	值/描述
组长度 ^a	(0000,0000)	UL	从(0000,0000)值字段的结尾到 C-Echo-Rq 消息的结尾，偶数字节
受影响的服务类 UID	(0000,0002)	UI	1.2.840.10008.1.1
命令字段	(0000,0100)	US	0030
消息 ID	(0000,0110)	US	消息的数字唯一 ID
数据集类型	(0000,0800)	US	0101

^a如我们在 5.5.2 中学到的，这个项目对所有 0000 组之后的项目总长度进行编码。

C-Echo DIMSE 的运行模式是非常简单的。请求的 AE 发送一个 C-Echo-Rq（一个 C-Echo 请求）。如果它对面的 AE 返回一个有效的 C-Echo-Rsp（响应消息），那么两个 AE 就一定建立了正确的 DICOM 连接。如图 35 所示。

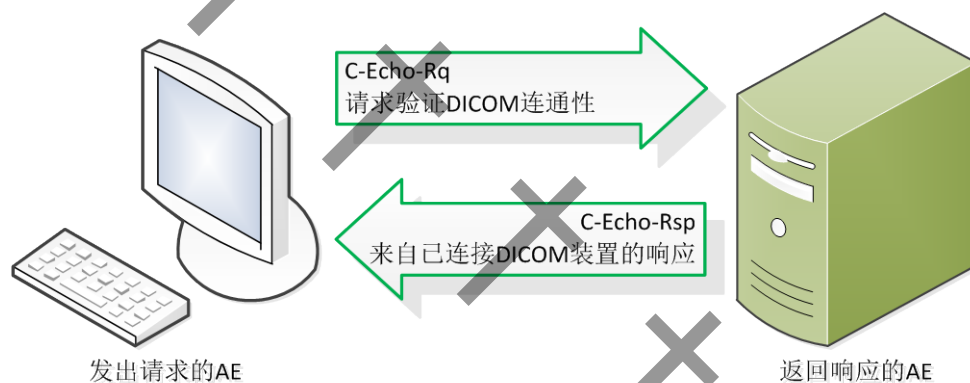


图 35 C-Echo 协议

C-Echo-Rq 和 C-Echo-Rsp 消息的内容是由 DIMSE 协议设置的，如表 20、21 所示，并且没有为实施留下什么出错的余地。

表 21 C-Echo-Rsp: 对 C-Echo-Rq 请求的响应

消息字段	标签	VR	值/描述
组长度	(0000, 0000)	UL	从值字段的结尾到下一个组开始处, 偶数个字节
受影响的服务类 UID	(0000, 0002)	UI	1. 2. 840. 10008. 1. 1
命令字段	(0000, 0100)	US	8030
响应用消息 ID 的用途	(0000, 0120)	US	应设置成消息 ID (0000, 0110) 字段的值, 用来与 C-Echo-Rq 消息关联
数据集类型	(0000, 0800)	US	0101
状态	(0000, 0900)	US	0

让我们看看表格中的一些项目吧。

1. **(组, 元素)标签**, 如我们所知, 在 **DICOM 数据字典**中对不同的数据字段进行编码。因为处理服务 (即 **DICOM 命令对象**) 时我们只使用来自组编号 **0000** 的命令元素。前面讨论过, 它们的 **VR (UL、UI、US)** 定义了它们的数据格式。
2. **受影响的服务类 UID (Affected Service Class UID)** 里面放的是与 **C-Echo 服务** 一致的 **UID**; 即 **1.2.840.10008.1.1**。所有 **DIMSE 服务** 都有它的 **UID 字符串** (需要懂得, 这就是在 **DIMSE 消息** 中识别这些服务的方法)。在我们之后讨论 **SOP** 时会更加突出标识符的作用。
3. **命令字段 (Command Field)**, 同样包含一个这个消息类型预设的 **ID**: **C-Echo-Rq 通常为 0030** 而 **C-Echo-Rsp 通常是 8030** (这些数字都有二进制 **US VR**, 因此他们是十六进制的)。当 **AE** 接受一个 **DIMSE 命令** 时, 它会首先看一下命令字段标签: 如果是 **0030**, 那么命令就是 **C-Echo-Rq**。在 **Rsp 命令字段** 中的数字 **8** 通常用在另一个 **DIMSE 服务** 中, 用来区分 **Rsp** 和 **Rq** 消息。
4. **数据集类型 (Data Set Type)** 参数被设置成 **0101**, 这在 **DICOM** 中意味着 “**空数据 (NULL data)**” (没有所属数据)。C-Echo 纯粹是一个命令对象, 它绝不会去传送数据 (比如图像、病人信息等等, 他们编码所使用的组编号绝不会是 **0000**)。
5. 在 **C-Echo-Rq** 中的消息 **ID** 包含这个消息产生的一个唯一编号。所有 **DIMSE 请求消息** 都有 **UID**。与 **DICOM UID** 不同, 消息 **ID** 是短型 **2 字节数字 (US VR)** 这已经足够在它们短暂的生命周期中标识每个消息了。当一个 **Rsp** 消息被创建, 来自 **Rq** 消息的消息 **ID** 被复制到 **Rsp** 消息的 (0000,0120) 字段, **DICOM** 就是这样将请求消息和响应消息配对的。假设在一个繁忙的 **DICOM 网络** 中, 相同的 **AE** 可能在每秒钟接受一堆 **DIMSE 消息**。这种情况下的消息 **ID** 成为了唯一能够区分它们的信息, 只有这样才能追踪发出请求的 **AE**, 并且无误地返回给它响应。
6. **状态字段 (The Status field)**, 在 **C-Echo-Rsp** 消息中通常设置成 **0000**, 表示成功。这意味着配对的 **AE** 已经收到了 **C-Echo-Rq**, 理解了请求, 并且用一个成功的 **C-Echo-Rsp** 进行了回复。我真的不知道其他 **C-Echo** 状态的选项了; 只需简单地接收一个 **C-Echo-Rsp** 答复就知道 **C-Echo** 已经成功了。只有在预设的超时时间 (几秒钟) 内没有任何响应时, 才考虑 **C-Echo** 消息传送失败了。假使某处出现问题, **DIMSE 服务** 为了返回各类警告和错误才会在消息状态字段中使用非零值。比如, 如果一个 **DIMSE 服务** 试图打印一个图像然而图像却不可用, 状态字段会为该服务提供一个错误值。

基于所有知识, 来看一看表 22 中 **C-Echo-Rq** 进行最终 **DICOM** 编码后的实例。就像任何

DICOM 命令对象一样，它也遵循隐式 VR 数据编码（默认的 DICOM 编码方式，见表 5）。

在二进制行中这个 68 字节的序列就是在 DICOM 网络上即将从一个 AE 向另一个 AE 发送的 C-Echo-Rq。需要注意的是，这个序列使用的是小端字节序（见 5.2）；当有多字节数字时，最低有效字节在前面。在我们的例子中，字节编号 15 和 16 形成了元素编号 $e = 0002 = 02$ （最低有效字节）和 00（最高有效字节）。

C-Echo-Rsp 采用相似的方法建立，返回情况如表 23 所示。我们标记了在 C-Echo-Rsp 中修改过（与 C-Echo-Rq 比较）的字段。首先是命令字段(0000,0100)，它反映了命令类型的变化。其次是消息 ID，它在 C-Echo-Rsp 中存储在(0000,0120)中，并且相当于 C-Echo-Rq 中(0000,0110)的值。

简单地说，C-Echo DIMSE 服务真没什么可让人费解的内容；基本上所有服务参数都是可以确定的。如果你正在开发 DICOM 软件，唯一需要你在 C-Echo 协议中创建的参数就是消息 ID 编号。每个 C-Echo 请求的消息 ID 必须是唯一的，并且要与 C-Echo-Rsp (0000,0120)字段匹配。只要这些都搞定了，就算万事大吉了。

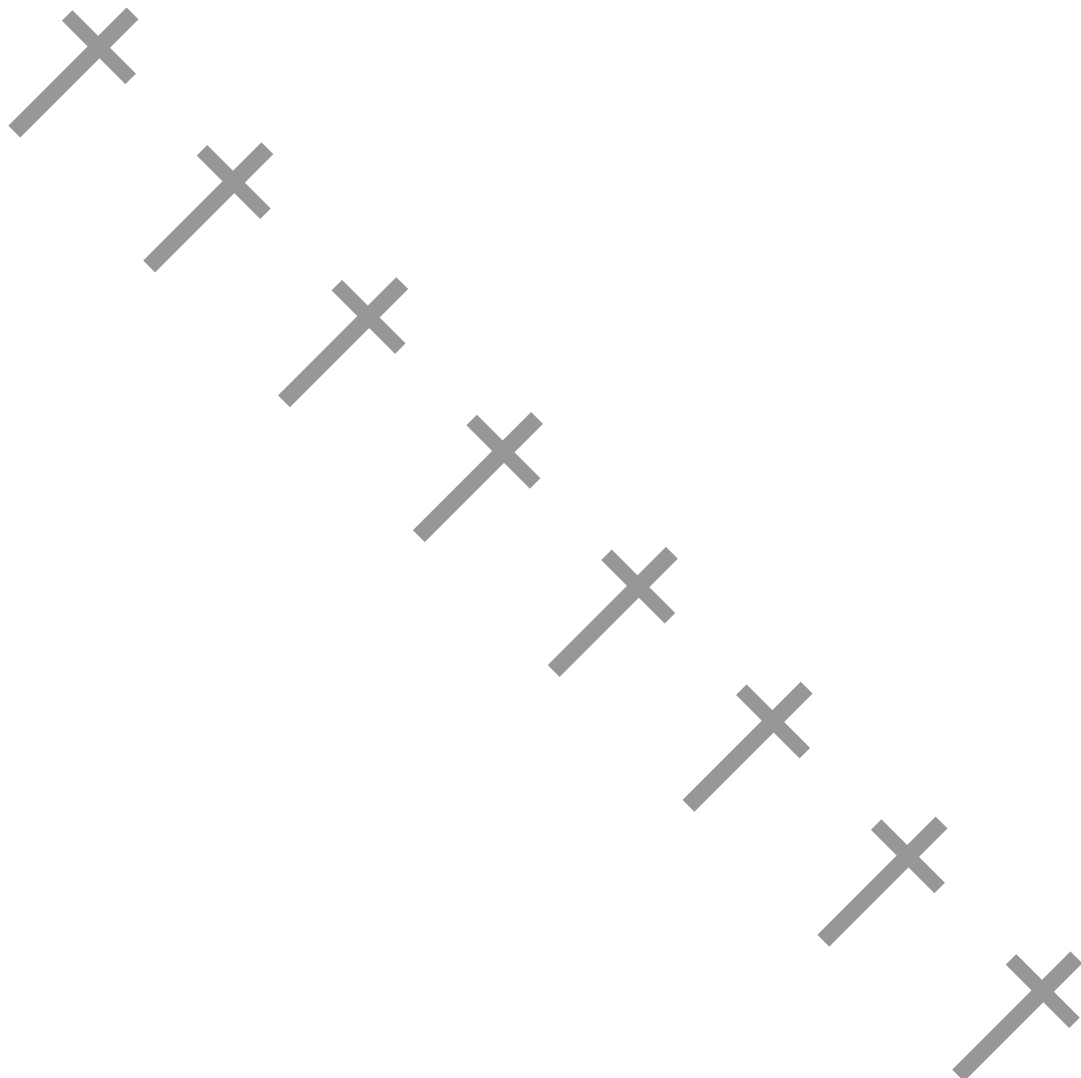


表 22 C-Echo-Rq 在最终 DICOM 字节编码

字节 #	1	2	3	4	5	6	7	8	9	10	11	12	13	14
十进制	0	0	0	0	4	0	0	0	56	0	0	0	0	0
二进制	00	00	00	00	04	00	00	00	38	00	00	00	00	00
	g=0000		e=0000		VR 长度 = 4				VR 值= 56				g = 0000	
字节 #	15	16	17	18	19	20	21	22	23	24	25	26	27	28
十进制	2	0	18	0	0	0	'1'	'.'	'2'	'.'	'8'	'4'	'0'	'.'
二进制	02	00	12	00	00	00	31	2e	32	2e	38	34	30	2e
	e=0002		VR 长度= 18				VR 值 = 1.2.840.10008.1.1 (17 字符结尾空白, 0 字节)							
字节 #	29	30	31	32	33	34	35	36	37	38	39	40	41	42
十进制	'1'	'0'	'0'	'0'	'8'	'.'	'1'	'.'	'1'	0	0	0	0	1
二进制	31	30	30	30	38	2e	31	2e	31	00	00	00	00	01
	VR 值 = 1.2.840.10008.1.1 (17 字符结尾空白, 0 字节)										g = 0000		e = 0100	

字节 #	43	44	45	46	47	48	49	50	51	52	53	54	55	56
十进制	2	0	0	0	48' 0'	0	0	0	16	1	2	0	0	0
二进制	02	00	00	00	30' 0'	00	00	00	10	01	02	00	00	00
	VR 长度= 2				Val = 0x0030		g = 0000		e = 0110		VR 长度 = 2			

字节 #	57	58	59	60	61	62	63	64	65	66	67	68
十进制	2	0	0	0	0	8	2	0	0	0	1	1
二进制	02	00	00	00	00	08	02	00	00	00	01	01
	Val = 0x0020		g = 0000		e = 0800		VR 长度= 0002				Val = 0x0101	

全部 DICOM 对象长度: 68 字节 = 13 字节 + 56 字节

此处:

(0000,0000) “组长度” 元素长度: 12 字节

(0000,0000)元素之后的长度: 56 字节 (等于(0000,0000)的元素值)

表 23 C-Echo-Rsp 在最终 DICOM 字节编码

字节 #	1	2	3	4	5	6	7	8	9	10	11	12	13	14
十进制	0	0	0	0	4	0	0	0	56	0	0	0	0	0
二进制	00	00	00	00	04	00	00	00	38	00	00	00	00	00
	g=0000		e=0000		VR 长度 = 4				VR 值 = 56				g = 0000	
字节 #	15	16	17	18	19	20	21	22	23	24	25	26	27	28
十进制	2	0	18	0	0	0	'1'	'.'	'2'	'.'	'8'	'4'	'0'	'.'
二进制	02	00	12	00	00	00	31	2e	32	2e	38	34	30	2e
	e=0002		VR 长度 = 18				VR 值 = 1.2.840.10008.1.1 (17 字符结尾空白, 0 字节)							
字节 #	29	30	31	32	33	34	35	36	37	38	39	40	41	42
十进制	'1'	'0'	'0'	'0'	'8'	'.'	'1'	'.'	'1'	0	0	0	0	1
二进制	31	30	30	30	38	2e	31	2e	31	00	00	00	00	01
	VR 值 = 1.2.840.10008.1.1 (17 字符结尾空白, 0 字节)										g = 0000		e = 0100	

字节 #	43	44	45	46	47	48	49	50	51	52	53	54	55	56
十进制	2	0	0	0	48' 0'	128	0	0	32	1	2	0	0	0
二进制	02	00	00	00	30' 0'	80	00	00	20	01	02	00	00	00
	VR 长度 = 2				Val = 8030		g = 0000		e = 0120		VR 长度 = 2			

字节 #	57	58	59	60	61	62	63	64	65	66	67	68
十进制	2	0	0	0	0	8	2	0	0	0	1	1
二进制	02	00	00	00	00	08	02	00	00	00	01	01
	Val = 0x0020		g = 0000		e = 0800		VR 长度 = 0002				Val = 0101	

全部 DICOM 对象长度: 68 字节 = 13 字节 + 56 字节
此处:
(0000,0000) “组长度” 元素长度: 12 字节
(0000,0000)元素之后的长度: 56 字节 (等于(0000,0000)的元素值)

7.2.3

服务对象对 (Service-Object Pairs)

让我们看看我们的 DICOM 工具箱里都有些什么了。首先,我们有 IOD,它可以定义 DICOM 数据。其次我们有 DIMSE 服务,它可以定义 DICOM 命令。应当足够了,命令需要应用于数据(打印 CT 图像、存储 MR 序列等等)上,因此我们将兼容的 DIMSE 服务和 IOD 对象配在一起,称作 SOP。也就是说,我们用实现数据处理(服务)的指令来捆绑 DICOM 数据对象(IOD)。

一直以来,我们都在讨论服务和数据,但实际上我们讨论的是 SOP。图 36 呈现了 DICOM SOP 类(DICOM SOP Class)的总体结构。如你所见,它确实将功能性(服务)和数据(IOD)结合在一起。重点是由于服务组合的目的就是处理特定的 IOD,所以 SOP 将 IOD 和 DIMSE 服务组结合在了一起。

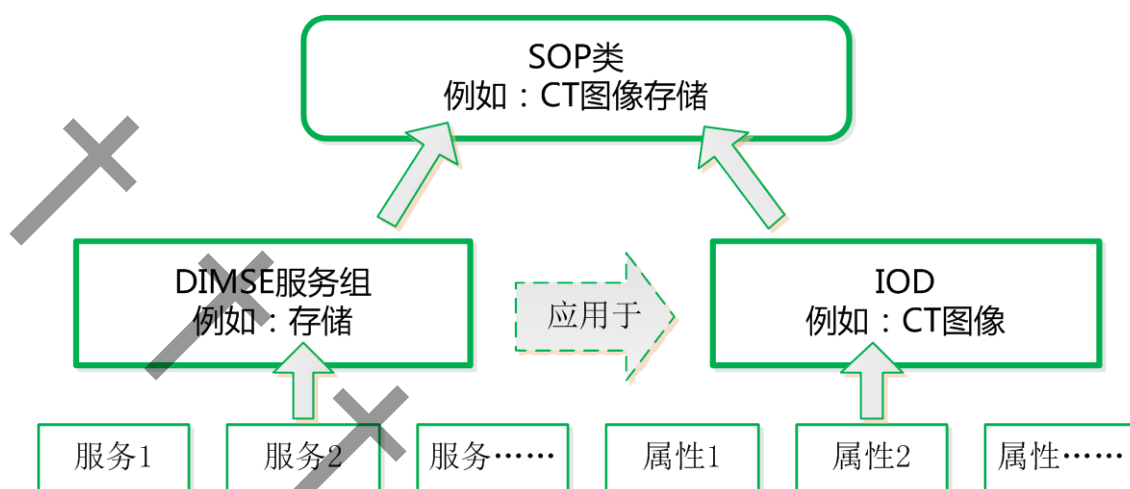


图 36 SOP 类结构：DIMSE 服务应用于 IOD 实例

DICOM PS3.6 提供一个所有标准 DICOM SOP 的列表,并且不是很长(见我们的附录 A.2)。在列表中,每个 SOP 都有一个描述名和一个相关联的 DICOM UID。当 DICOM 需要决定处理什么或者如何处理的时候,它都会按照 SOP 来处理的。这使得 SOP 类(SOP Class)变得特别重要;它们实实在在地告诉你 DICOM 应用到底能起什么作用。所有 DICOM 一致性声明(DICOM Conformance Statements)都是用 SOP 语法描述的。这方面的内容在早前的 DICOM 对象和 IOD 讨论中已经有所触及,但是那并不是介绍网络功能性的最好时机,现在时机成熟了。

在之后的段落中,我们将回顾所有最常用的 DICOM SOP 类,从最基础和最常见的开始。

7.2.4

校验 SOP

校验 SOP 验证两个 AE 之间的 DICOM 连接性。我们不是在表 24 中已经见过 1.2.840.10008.1.1 了吗?没错,在 C-Echo 中这个数字串会作为 SOP 的参数使用。校验 SOP 它自己具有 DIMSE 部分的 C-Echo,并且由于连接性校验并不需要数据处理,所以它没有 IOD(图 37)。

因此,校验 SOP 和 C-Echo 做的事情是一样的:它验证两个 DICOM 应用之间的 DICOM 连接性。校验 SCU 发送一个验证请求(C-Echo-Rq)给其他应用程序。如果其他设备正确连接并且支持一个校验 SCP,那么就会返回一个 C-Echo-Rsp,作为连接成功的确认消息。实际上就是校验 SCU 问它通讯的对方:“你能和我进行 DICOM 对话吗?”并且期望听到“是的”。

如果校验 SOP 失败会发生什么,如果在连接超时之前没有收到任何响应又会怎样呢?

大多数情况下，这意味着以下两种情况中的一种：

1. 你有一个常见的网络连接性错误，比如没插网线、IP 地址错误、通讯被阻断（防火墙）、网络故障、或其他问题。为了解决问题，请确认你能够从一个 AE ping 通另一个。
2. 网络是好的，但是 DICOM 配置可能是错的。这通常意味着两个 DICOM 通讯设备中的一个并不知道另一个设备正确的 IP 地址、端口、或 DICOM 名。如果任何一个参数是错的，那么发送 C-Echo-Rq 的 AE 就肯定敲错了门。为了解决问题，请确认每个 AE 了解对方正确的配置（IP、端口、名）。

表 24 校验 SOP

SOP 类名	SOP 类 UID
校验(Verification)	1.2.840.10008.1.1

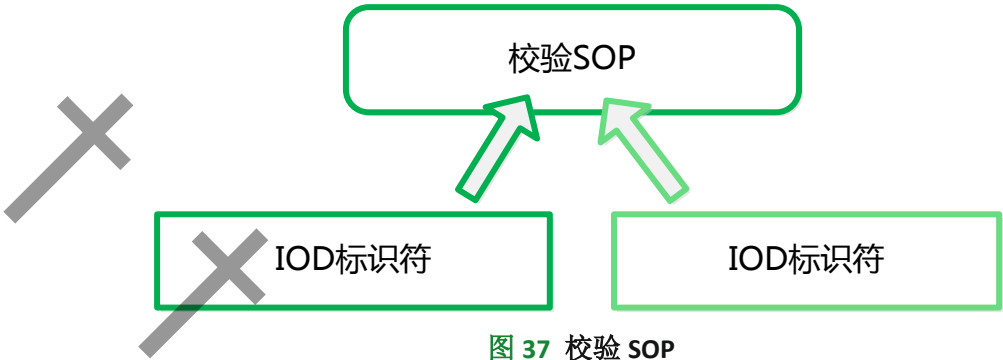


图 37 校验 SOP

校验 SOP 类在控制 DICOM 网络的正常工作方面起着最基础的作用。因此，在许多 DICOM 界面上会有类似“连接验证”的按钮，用户可以点击它们来确认是否已经与远端设备建立了 DICOM 通讯，这类应用使用的恰恰就是校验 SOP 类。假如在你的 DICOM 连接中发现了任何问题，从校验 SOP 开始来解决问题吧。

因此，一定要避免购买没有校验 SOP、SCP 功能的 DICOM 设备和软件。你能猜到，这些设备将不能够对 DICOM 的验证请求给予回应；坦率的说，验证这些设备的 DICOM 连接性将是一个非常艰巨的工作。

真实案例：校验 SCP 与安全性？

曾有一次，我们试图将一个 CR 装置与一个知名成像厂商的软件进行连接，而我们发现这个 CR 装置根本不支持校验 SOP SCP。

“难怪”，正在努力搞定 DICOM 验证问题的厂商工程师这么解释，“我们不希望你们从 CR 上导出图像数据，这可是一个安全性要求”。

许多人都会用相同的说辞来粉饰他们的无知：我们都知道，校验 SOP 并不传输图像数据或者任何其他保密信息。因此，它不能用来盗取任何私有或公开的信息。

别把安全性说的那么神秘，也别用安全性作为掩饰功能设计缺陷的借口。

7.3 存储

如果说 C-Echo 在 DICOM 连接性查验方面的作用（校验 SOP）很重要，那么 C-Store（存储 SOP（Storage SOP））就是 DICOM 数据的搬运工，它负责在 AE 之间移动 DICOM 影像（以及其他数据类型）。尽管名字可能有些误导，但是 C-Store 仍然能够在 DICOM 网络上从一个 AE 向另一个发送 DICOM 数据对象。这是因为不同的对象有不同的处理方法，DICOM 为每个设备类型或数据类型设置了各自的 SOP 类（比如 CT 图像存储 SOP、MR 图像存储 SOP 类等）。因此，DICOM 存储是通过一系列数据特有的存储 SOP 体现的，每一个 SOP 都有它自己的 UID。你能够在附录 A.2 中找到所有的存储 SOP。我们在表 25 中简略地列出最常用的 SOP。

表 25 存储 SOP

SOP 类名称	SOP 类 UID
CR 图像存储	1.2.840.10008.5.1.4.1.1.1
数字 X 线图像存储 - 展示用	1.2.840.10008.5.1.4.1.1.1.1
数字 X 线图像存储 - 处理用	1.2.840.10008.5.1.4.1.1.1.1.1
数字乳腺 X 线图像存储 - 展示用	1.2.840.10008.5.1.4.1.1.1.2
X 线血管造影图像存储	1.2.840.10008.5.1.4.1.1.12.1
正电子成像（PET）图像存储	1.2.840.10008.5.1.4.1.1.128
CT 图像存储	1.2.840.10008.5.1.4.1.1.2
增强 CT 图像存储	1.2.840.10008.5.1.4.1.1.2.1
核医学图像存储	1.2.840.10008.5.1.4.1.1.20
超声多帧图像存储（弃用）	1.2.840.10008.5.1.4.1.1.3
超声多帧图像存储	1.2.840.10008.5.1.4.1.1.3.1
MR 图像存储	1.2.840.10008.5.1.4.1.1.4
增强 MR 图像存储	1.2.840.10008.5.1.4.1.1.4.1
MR 波谱存储	1.2.840.10008.5.1.4.1.1.4.2
基本文本 SR（结构化报告）	1.2.840.10008.5.1.4.1.1.88.11

增强的 SR	1.2.840.10008.5.1.4.1.1.88.22
综合的 SR	1.2.840.10008.5.1.4.1.1.88.33
放疗图像存储	1.2.840.10008.5.1.4.1.1.481.1
核医学图像存储（弃用）	1.2.840.10008.5.1.4.1.1.5
超声图像存储（弃用）	1.2.840.10008.5.1.4.1.1.6
超声图像存储	1.2.840.10008.5.1.4.1.1.6.1
二次获取图像存储	1.2.840.10008.5.1.4.1.1.7
12 导联 ECG 波形存储	1.2.840.10008.5.1.4.1.1.9.1.1
通用 ECG 波形存储	1.2.840.10008.5.1.4.1.1.9.1.2
基本语言音频波形存储	1.2.840.10008.5.1.4.1.1.9.4.1
挂片协议存储 ^a	1.2.840.10008.5.1.4.38.1

^a 挂片协议在诊断显示器上定义图像排布格式。

除了影像设备特定的存储 SOP 类型（UID）之外，在 DICOM 存储中的其他东西都是为所有存储影像设备设计的。首先，所有存储 SOP 类的结构都是一样的（图 38）。这里 DIMSE C-Store 服务一如既往地包含两部分内容：命令部分以及依附它的且包含所有实际存储对象的 IOD 部分。C-Store IOD 是影像设备特定的那一类，而 C-Store DIMSE 则是影像设备无关的那一类。



图 38 存储 SOP

从 C-Store SOP 表格中可以看出，DICOM 存储可用于不同的数据类型：图像、波形、报告等等。图像是最普通和最常用的，因此我们在本章之后的部分也常提及它们。

7.3.1

C-Store IOD

C-Store IOD 组件没有什么神秘的，它其实就是承载着要存储的数据。数据（MR、CT、CR、超声等）的类型定义了特定的 C-Store SOP 类 UID，如表 25 简述的。

如果我们需要存储多幅图像（通常会遇到），每个图像都会通过它自己独立的 C-Store 请求来传输数据。即不能在一个 C-Store 中批量处理多个 IOD 实例。唯一的例外是多帧 IOD

（比如超声多帧图像存储，1.2.840.10008.5.1.4.1.1.3.1），这个 IOD 实例（图像）会包含多个图像帧。超声电影回放在 DICOM 中是以一副图像的多个帧来显示的，也因此可以只用一个 C-Store 请求来发送。

相较之下，CT或MR序列中的每幅图像都将会以一个新的C-Store来处理。一方面，它将特定的处理和通讯架空了¹，但是另一方面，每个图像传输的结果都是清晰的：等待、失败或发送成功。

C-Store 瓶颈

如果你在编写 DICOM 软件时用 C-Store 来发送图像序列，那么永远不要将一个图像存储的失败解释为整个图像序列的失败。一个序列中的一幅图像可能由于许多非常轻微的问题（比如，图像在目的地已经存在）导致 C-Store 失败，这并不会导致剩余图像传输的停止。

更重要的是，绝不要将一些图像的 C-Store 失败解释为停止任何后续通讯的借口。我们就遇到过一个 CR 设备发送图像到数字归档失败的经历。因为一个非法的病人 ID（病人 ID 不允许为空）致使一个 CR 检查传输阻塞。目的地归档发现了 ID 问题并且拒绝接受这个检查。令人咋舌的是 CR 装置居然将单个检查失败解释为一个重大灾难，即使后续检查的病人 ID 是完全正确的，在这一幅图像失败后它仍然停止发送所有后续检查！过了一会，所有未发送的检查堆积在 CR 中，将它本地硬盘填满了，造成设备拒绝执行新的检查任务。

这个例子完美地体现了一个小问题如何放大了软件设计的缺陷，造成整个 PACS 流程的瓶颈。

7.3.2

C-Store DIMSE

C-Store 负责向图像接收 AE 发出指令，告诉他该做什么。就像 C-Echo 一样，C-Store 有一个请求部分（从 C-Store SCU 发送）以及一个响应部分（从 C-Store SCP 返回）。如果我们用表 26 和前面学到的 C-Echo-Rq（见 7.2.2）进行比较，我们会发现一些新的参数，如下所示：

表 26 C-Store-Rq

消息字段	标签	VR	值/描述
组长度	(0000,0000)	UL	从(0000,0000)值字段的结尾到 C-Store-Rq 消息的结尾为偶数字节
受影响的 SOP 类 UID	(0000,0002)	UI	包含用于这个图像类型的 SOP UID。比如，1.2.840.10008.5.1.4.1.1.2 是 CT 图像存储用的 SOP UID
命令 ID	(0000,0100)	US	0001
消息 ID	(0000,0110)	US	用于消息的唯一数字 ID
优先级	(0000,0700)	US	任选其一： 0002（表示低优先级） 0000（表示中优先级） 0001（表示高优先级）
数据集类型	(0000,0800)	US	任何非 0101 的值
受影响的 SOP 实例 UID	(0000,1000)	UI	包含需要存储的 SOP 实例的 UID（即，需要存储的图像的 UID）

¹ 私有的、非 DICOM 标准的、医疗设备厂商在设备内部使用的图像格式，也许会使用将所有的图像序列批量处理，放在同一个数据缓存中并且一次性将其发送出去。

移动发起者的应用实体名	(0000,1030)	AE	包含 DICOM AE 的 DICOM AET，在 C-STORE 的子操作执行过程中，该 AE 调用了 C-MOVE 操作。
移动发起者消息 ID	(0000,1031)	US	包含 C-MOVE-Rq 消息的消息 ID (0000,0110)，消息来自 C-Store 子操作的执行过程中。

1. 优先级：如表格所示，优先级可以是低、中（默认）、或高。许多其他 DIMSE 消息在其规定中都有优先级字段，但是是否支持优先级是可选的并且默认使用中级（0000）。事实上，DICOM 厂商几乎不会在产品中使用消息优先级，当然这会在厂商的一致性声明中写清楚。这是有实际意义的，因为维持 PACS 网络的正常运行比实现消息优先级更为重要，而任何优先级的设置机制只是为系统增加了不必要的复杂性以及不可预知性。在数据获取、传输和提取中的演示，在现实的临床环境中根本不可避免，思考如何定义消息语义比起思考预先定义消息优先级显得更有意义。因此，SOP 消息优先级几乎不会用到。
2. 受影响的 SOP 实例 UID：这是个自解释字段。一个需要存储的图像会跟在 C-Store-Rq 消息的后面，并且图像会有他自己的 UID，这个 UID 就存放在这个字段中（见 5.6）。
3. 移动发起者名称和消息 ID 字段，如我们即将在 C-Move SOP 部分所见，C-Store 会被其他 DIMSE 命令调用，比如 C-Get 或 C-Move。既然这样，调用 C-Store 的那些 AE 名称和原始的消息 ID 就需要保存在这个字段中。
4. 数据集类型：你可能记得在 C-Echo 中这个字段被设置为 0101，在 DICOM 中认为它表示 NULL（没有附加数据）。然而，一个 C-Store-Rq 肯定要携带数据（需要存储的图像）这个字段必须要设置成非 0101 的值。具体是什么值就无关紧要了，由 DICOM 厂商自行决定。有些会用 0102 有些则会用 0000；具有讽刺意味的是，在 DICOM 中居然用 0101 表示“空”而用 0000 表示“非空（not-NULL）”（有附加数据）。

由于数据集类型字段是非空的，因此一个 C-Store-Rq 消息后面紧接着就是一个需要存储的图像（一幅，只有一个图像对象）。换句话说，一个 C-Store-Rq 总是发送一个图像到它的配对 AE 去，要求其进行存储（图 39）。

C-Store-Rsp 是数据接收 AE 发回的消息。见表 27。几乎在 C-Store-Rsp 中的所有内容都是来源于原始的 C-Store-Rq 消息，除了总是很重要的“状态（Status）”字段。状态告诉你图像是否存储（发送）成功（Status = 0000）；是否正在传送（Status = FF00）；或者产生某种警告和错误（Status = 除 0000 和 FF00 外的其他值）。

表 27 C-Store-Rsp

消息字段	标签	VR	值/描述
组长度	(0000,0000)	UL	从(0000,0000)值字段的结尾到 C-Store-Rq 消息的结尾为偶数字节
受影响的 SOP 类 UID	(0000,0002)	UI	包含用于这个图像类型的 SOP UID。比如，1.2.840.10008.5.1.4.1.1.2 是 CT 图像存储用的 SOP UID
被响应的消息 ID	(0000,0120)	US	应该被设置为在相关 C-Store-Rq 消息使用的消息 ID 的值 (0000,0110)
数据集类型	(0000,0800)	US	0101（意味着没有数据附加在 C-Store-Rsp 上）

受影响的 SOP 实例 UID	(0000,1000)	UI	包含需要存储的 SOP 实例的 UID（即，需要存储的图像的 UID）
状态	(0000,0900)	US	0000（如果成功），FF00（如果等待），或者其他厂商支持的警告和报错值（见厂商的一致性声明）

数据所依靠的 C-Store SOP 列表在各个 DICOM 修订版中一直在不断变化着；旧数据/图像类型最终会退休，而新类型会增加进来。如果你正在考虑买一个新的 DICOM 软件或装置，一定要确认你所使用的数据类型在装置的 DICOM 一致性声明中可以看到。

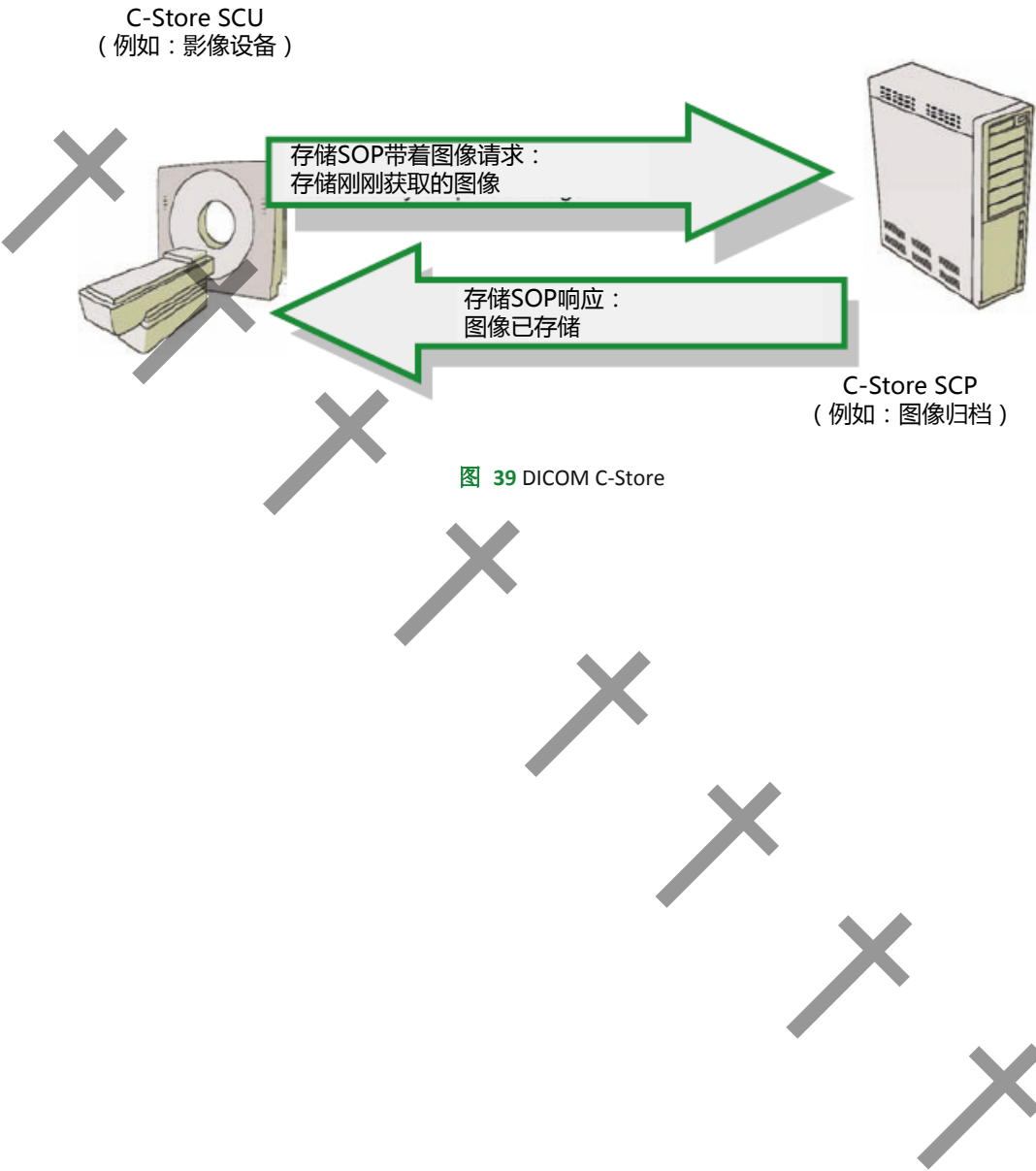


图 39 DICOM C-Store

7.4
查询：查找

现在我们能够校验 DICOM 连接性了。我们甚至能够从一个 AE 向另一个发送图像了。但是我们怎么找出需要发送的数据呢？三个 C-Find SOP 类提供 DICOM 查询的实现方法，如表 28 所示。

查找成像数据不是影像设备特定的，因此 C-Find 不包括许多像 C-Store 那样、基于影像设备的 SOP。因此可以认为，C-Find 与 C-Echo 非常相似（图 40）。然而，C-Find 中还有一样是需要全新介绍的：DICOM 将所有 C-Find 数据查询分为了三个数据级别：病人、检查、病人-检查。这些级别称作根，并且 C-Find 在这三个根上使用不同的 SOP 来实现数据查询。

表 28 查询（C-Find）SOP

SOP 类名	SOP 类 UID
病人根 Q/R 查找	1.2.840.10008.5.1.4.1.2.1.1
检查根 Q/R 查找	1.2.840.10008.5.1.4.1.2.2.1
病人-检查根 Q/R 查找（弃用）	1.2.840.10008.5.1.4.1.2.3.1

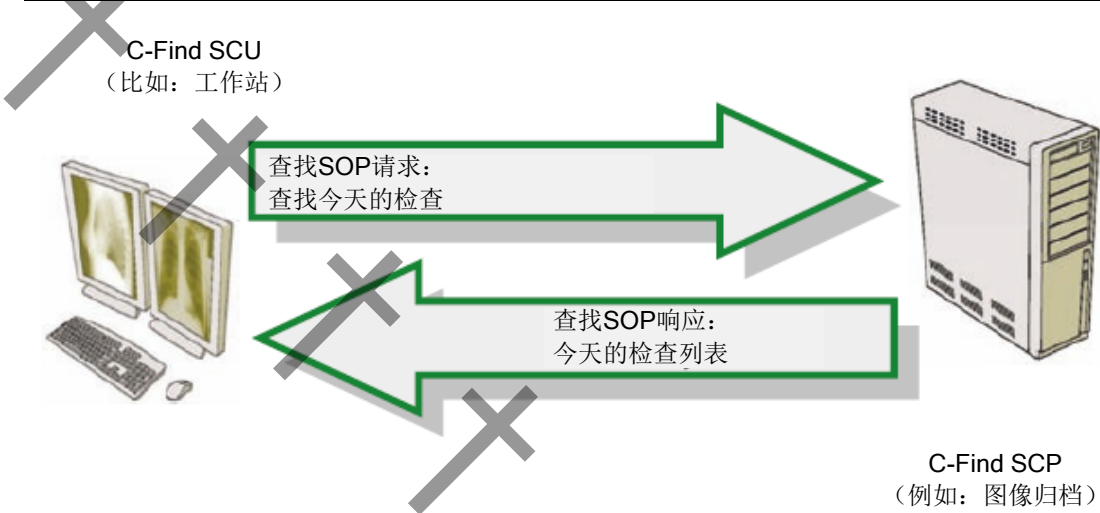


图 40 DICOM C-Find 例子：从图像归档获取所有今天检查的列表

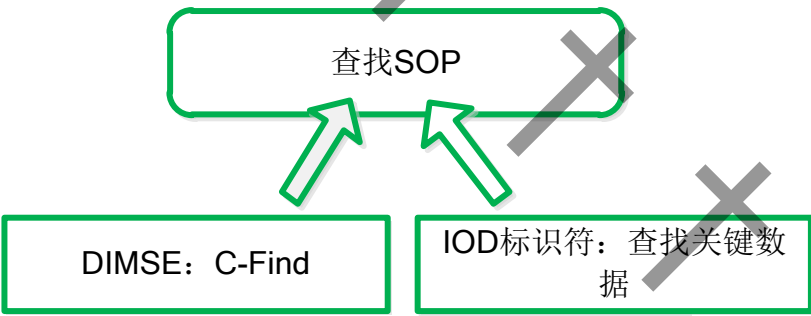


图 41 查询 SOP

DICOM 查询根符合 5.6 中介绍的 DICOM 病人-检查-序列-图像的数据等级。DICOM 用四个等级来组织数据，并且查询数据与等级限制配合的非常好。比如，如果支持病人查询/提取作为根，那么查询和提取只会将限制在病人-检查-序列-图像这个级别；如果使用检查查询/提取作为根，那么查询和提取将会限制在检查-序列-图像这个级别。在后面的例子中，所有病人属性（比如病人姓名或者 ID）将会包含在检查属性中用来识别不同检查。

由于目前的放射工作流是以检查为中心的，因此以检查为根在 DICOM 应用程序中会非

常广泛地使用并且通常作为默认设置。病人和病人-检查 SOP 作为查询层级则很少使用。事实上，以病人-检查为查询根最近已经在 DICOM 标准中弃用了，这意味着，在未来的 DICOM 版本中将不会支持这种做法，并且会浅浅地在 DICOM 实现过程中消失。

勿须赘言，C-Find SOP 也符合 DIMSE + IOD 结构，如图 41 所示。DIMSE 对象（编码来作为 DICOM 命令对象，组 0000）传送 C-Find 消息参数，并且附加 IOD 对象（编码来作为 DICOM 数据对象）用来承载查询条件去 C-Find SCP 上找匹配的信息。

7.4.1

DICOM 中数据匹配方面的一些词汇

当 C-Find SCU 发送一个查询请求时，请求会包括希望获得匹配的属性集合。这些属性会作为查询字段与目标 AE 中的数据进行比较。总的来说，DICOM 会提供一些设置查询值的方法：

1. 通配符匹配。我们之前已经有所了解。我们曾在 5.3.2 中第一次提到它，当时我们正在讨论 DICOM 通配符（*?）。不仅仅在 DICOM 中，很多地方都会常常用通配符来匹配字符串。比如，如果我们知道一个病人名字是以“Smit”开头，那么我们可以在 C-Find 请求中将病人姓名（Patient Name）(0010,0010)属性里面输入“Smit”并且在其后面使用（*）代表任何字符序列。我们会找到“Smith”、“Smithson”、甚至“Smit III the Great”，这些都是符合此次通配符查询的。其他 DICOM 通配符，比如问号（?），表示单一位置字符，但是很少会用到它，因为我们在不了解字符是什么的情况下，也很有可能不知道字符的数量，因此干脆用星号就好了。
2. 罗列匹配。此技术使用反斜线（\）通配符，这一位置“或”。比如，病人姓名中的“Smith\Graham”将用来查找所有叫 Smith 和 Graham 的病人。当然可以将罗列和通配符结合使用比如“Smit*\Grah*”，特别是当你处理由录入错误风险的数据时。（手工录入的病人姓名很容易造成错误录入）
3. 整体匹配。这是一种最常见的匹配规则。如果被匹配的属性是零长度的，那么就意味着匹配所有。比如，如果我们在病人姓名栏中留空，并以此条件执行 C-Find 查询，那么在目标归档中的所有病人姓名都将符合该查询条件。事实上，这种情况下整体匹配意味着任何可能的值相当于使用星号（*）。然而，在 C-Find 中提供一个零长度匹配属性与什么都不提供是完全不同的两个概念。包含零长度属性意味着我们希望（或必须）使用这个零长度属性进行匹配并且我们希望在 C-Find 响应中获得匹配结果。不使用属性进行查询意味着我们不希望查询内容获得匹配（可能根本不支持这种做法）并且我们也不想提取结果属性。比如，如果我们不想提取病人姓名，无论病人姓名中是否有值，我们需要在 C-Find 请求中去掉病人姓名属性。
4. 范围匹配。这种匹配类型用作有范围的属性，比如日期或时间。在范围匹配中，你能够规定开始和结束的属性值，通过连字号（-）分隔两个值。任何在该范围内的内容都将匹配此次查询。比如，日期范围匹配 20000201-20100202 意味着在 2000 年 2 月 1 日和 2000 年 2 月 2 日之间的所有内容。如果开始或结束时间不能确定，可以留空；比如 20000201-表示“2000 年 2 月 1 日及之后”。
5. 序列查询。这也许是最复杂的 DICOM 属性匹配形式。这种形式只有在属性的整个序列（由 SQ VR 组成，见 5.3.10）用作匹配数据时才会采用。比如，比对影像设备已预约病人检查时，DICOM 会使用病人 ID、检查日期和检查号组成属性序列，序列中的每个属性既可以是确定值，也可以使用前面所提到的某一个匹配类型。在一个 SQ 序列中的属性是按照逻辑“与”来组合的（意味着所有属性必须匹配）。与表示“或”的反斜线（\）通配符正好相反。

6. 单一值匹配。这就是简单地使用准巨额属性值作为匹配参数：使用“Smith”用做病人姓名查询；“20000201”用作日期查询等等。很自然地，单一值匹配的属性必须不包含通配符和范围查询。单一值匹配通产雇佣在各种 ID 和 UID 的查询上，并且特别用在分级的关键属性（即唯一 ID，如病人 ID、检查实例 UID 等）上（见 5.6.3）。

在 DICOM 数据匹配中的所有属性既可以是必须的也可以是可选的。必须的属性，人如其名必须在匹配请求中出现。如果我们不知道应该将这类查询值设置成什么，那么我们可以简单地将它们设置为空值（整体匹配）。这就表明我们保证会收到匹配值。如果我们有兴趣使用，那么可以使用可选的属性。DICOM 标准通常会对每个查询类型（比如在病人、检查、序列、图像层面上的 C-Find）规定其必须的和可选的属性，但是你总是需要对你所查询的设备进行 DICOM 一致性声明的核实，因为他们通常不会顾及那些看起来更一般的 DICOM 规定。这在某种程度上是可以理解的，因为用在超声扫描仪上的属性肯定与用在 CT 教学归档上的属性不同。然而，这却总是带来一个实施上的问题，你不得不针对不同设备来设置不同的查询属性。

最后要说的是，在 DICOM 中几乎所有的匹配都是大小写敏感的。只允许在某些特定的属性中使用大小写不敏感，比如姓名（PN VR），在这些属性中也允许空格和强调符不敏感。以我个人的看法，我认为大小写敏感根本没能给 DICOM 带来任何好处：首先大小写很可能在录入时进行转换，其次这几乎没有人关心。更糟糕的是，一些主流的 PACS 界面会要求用户只能使用大小写中的一种来如入某些数据：比如，只能用大写来录入病人姓名。很明显这么做有个重大弊端，除非你录入 SMITH 否则你根本别想找到病人 Smith。实际上，这种要求实际上逼着用户在使用 PACS 时按下“大写锁定”，但是这样不可避免地会产生一系列其他的录入问题。因此，只要可以实现，DICOM 和 DICOM 应用程序就应该避免大小写敏感问题。

7.4.2

C-Find IOD

DICOM 属性匹配大陆上航行过后，我们就可以在更高的层次上对 C-Find 查询进行深入了解。C-Find IOD 包含用来在 C-Find 服务提供者（如数字归档）那里进行匹配的查询参数。表 29 提供了一个真实的一致性声明的实例。¹

如 5.6.3 所述，在 DICOM 中的所有数据符合病人-检查-序列-图像等级制度，而 C-Find 查询也不例外；他们必须属于四个级别中的一个。C-Find 查询层级会反映在（0008,0052）属性中（必须的），内容会是病人（PATIENT）、检查（STUDY）、序列（SERIES）或图像（IMAGE）。这同样说明在 SOP 根之外，这种规则也一直在 C-Find 的各个查询层级中遵从着：

1. 所有较高层次的关键属性值²必须是已知的。比如，在开始进行检查层次的查询时，我们必须知道病人 ID。
2. 所有当前层次的关键属性将通过 C-Find SCP 返回。比如，在一个检查层次的查询中，“检查实例 UID”（0020,000E）将总会出现在 C-Find 请求的响应中。这符合规则 1，即允许我们处理更低的层级（序列、图像）。
3. 在任何的 C-Find 请求中，我们只能从当前 C-Find 层次来查询和匹配属性。比如，一个检查层次的查询只能查询检查属性而不能单独查询序列或图像。

请看我们表 29 中的样例 IOD；它是什么层次的呢？我们会立即发现其中有一个（0008,0052）值：这是检查层次。但是即使你不知道这个值是什么，你仍可以去推断：IOD 里看起来有病人 ID（病人层次），但是还没有检查实例 UID（检查层次）。根据规则 1 和规则 2，

¹ 由于 C-Find IOD 样例来自于一个特定的 DICOM 应用程序，因此不同程序可能有不同的可选项目列表。

² 关键层次的属性值，如 5.6.3 所述，分别为病人 ID(0010,0020)、检查实例 UID(0020,000D)、序列实例 UID(0020,000E)、图像或 SOP 实例 UID (0008,0018)。

这显然是个检查层次的查询。这个逻辑在定位 DICOM 问题时也很有用。比如，如果在关键属性中我们决定采用的层次与(0008,0052)声明的不同，那么 C-Find 查询一定会失败。

(组,元素)	名称	必须的/可选的	实例	匹配（见 5.3.2）
(0008,0052)	查询层次	R	STUDY	可以是“PATIENT”、“STUDY”、“SERIES”、“IMAGE”。这个元素定义了层次和等级查询。
(0010,0010)	病人姓名	R	Smit*\Gr ah*	通配符匹配：*可代替任何子字符串（包括空）；? 可代替任何单一字符；\代表“或”
(0010,0020)	病人 ID	R	12345	具有代表性的，需要一个准确的值。一些系统支持通配符 ID 匹配。
(0008,0020)	检查日期	R	2006123 1-20070 201	典型的，两个日期格式为 YYYYMMDD 的日期范围匹配
(0008,0030)	检查时间	R	015500- 235559	典型的，两个时间格式为 HHMMSS 的时间范围匹配。如果限定了检查日期范围，那么开始时间需要位于开始日期而结束时间需要位于结束日期。在我们的例子中我们反复查找在 2006 年 12 月 31 日 01:55:00 和 2007 年 2 月 1 日 23:55:59 之间的所有完成的检查。
(0008,0050)	检查号	R	Abc789	典型的，精确匹配。经常从放射信息系统（Radiology Information System）中导入检查号。
(0020,0010)	检查 ID	R	1.234.56 7	几乎总是精确匹配
(0008,1030)	检查描述	O	*knee\el bow*	几乎能够使通配符匹配，此处关键词用来定位具有特定描述的检查。
(0008,0090)	相关医师姓名	O	*Sinitsyn *	对于任何名称，通配符是比较好的选择
(0008,1060)	读片医师姓名	O	*Ustiu zh anin*	对于任何名称，通配符是比较好的选择
(0008,0061)	检查所用影像设备	O	MR\CT	精确匹配或列表匹配
(0010,0030)	病人出生日期	O	1956010 1-19860 101	范围匹配。用来确定病人年龄

当然，你没有必要一定建立你自己的层次属性表格。每个 DICOM 应用程序的 DICOM 一致性声明都会针对每个支持的层次（并且用必须的或可选的来标识它们）列出所有支持的属性。事实上，一个特定的设备真不一定能够支持所有四个等级层次，它们的一致性声明将仅包含所支持的层次。

差的不能再差的影像设备

最常见错放 层级的属性是成像“影像设备”(0008,0060)。影像设备是非常重要的；然而，多数放射从业人员总是将影像设备和检查联系起来，他们会对于检查层次的查询不能返回影像设备信息非常失望。

但是这确实是显而易见的结果，因为几乎所有 DICOM 一致性声明都将影像设备列为序列层次的属性。这意味着两件事：

1. 一个 DICOM 检查可以是由多个影像设备完成的。只有很少的人有这样的意识，但这里有一个简单的例子：在一个带有一些三维重建图像的 CT 检查中，三维重建的屏幕截图在 PACS 中可能会将影像设备的值设置为 SC（二次获取）而不是 CT。
2. 为了查找检查的影像设备，你应该即在序列层次查询(0008,0060)影像设备属性，又在检查层次查询(0008,0061)“检查中的影像设备”。进行序列级别的查询是比较费劲的，这会要求从检查层级开始向下查询一级，并且查询设备可能并不支持这种方式。换句话说，(0008,0061)“检查中的影像设备”属性，也不是都能够支持的。

以上这些情况常常造成提取影像设备值出错。但是如果我们不能解决我们自己的问题，我们至少应该明白是怎么回事，对吗？

在一些先进的应用程序中，我们也可以使用关联的 DICOM 查询，在使用这种查询时，我们不需要遵循任何死板的等级结构。如果关联的查询在所有查询目标设备（C-Find SCP）上都能够支持，那么通过 C-Find IOD 向目标设备抛出任何信息；但是当然必须是目标设备 DICOM 一致性声明属性列表中列出的内容。这就是与层级查询底线相反的扩展 C-Find 行为。这会使 C-Find 执行起来更加简单，但是我们需要记住两件事：

1. 如前所述，99%的 DICOM 和医疗成像工作流都是以检查为核心的。放射医生、临床医生以及归档都愿意使用定义在检查级别的功能。如果你提供给他们混合和匹配任何信息的能力，他们也许根本不买你的好。
2. 很少有 DICOM 应用程序支持这种相关的查询。

在这些前提下，关联的查询仍然在实现一些复杂的和非常规的查询时有用武之地，但是临床工作中不会对它有类似“马上搞定”这种较高的要求。

7.4.3

C-Find DIMSE

就像 C-Store 那样，C-Find DIMSE 就像个网络运输者一样在各类应用实体之间移动 C-Find IODs。因此，它的结构和 C-Store IOD 不太一样，如表 30 所示。

表 30 C-Find-Rq

消息字段	标签	VR	值/描述
组长度	(0000,0000)	UL	从(0000,0000)值字段的结尾到 C-Find-Rq 消息的结尾的偶数个字节。
受影响的 SOP 类 UID	(0000,0002)	UI	包含针对这个 C-Find 查询根的 SOP UID；将会使用下面三者之一： 1.2.840.10008.5.1.4.1.2.1.1（病人） 1.2.840.10008.5.1.4.1.2.2.1（检查） 1.2.840.10008.5.1.4.1.2.3.1（病人-检查）
命令字段	(0000,0100)	US	0020
消息 ID	(0000,0110)	US	消息的唯一数字 ID

优先级	(0000,0700)	US	三种选择之一： 0002（低优先级） 0000（中优先级） 0001（高优先级）
数据集类型	(0000,0800)	US	任何非 0101 的内容

就像 C-Store-Rq 那样，数据集类型字段(0000,0800)是非空的(在 DICOM 中 0101 代表空)这意味着一个 C-Find-Rq 消息后面紧跟着 C-Find IOD，用来传送查询属性。DIMSE + IOD 的组合在网络上从 C-Find SCU 应用实体传送到 C-Find SCP 应用实体。后者首先接收到 C-Find DIMSE 并识别它，然后用 C-Find IOD 来查询它自己本地的数据库，找出是否有匹配的结果。当匹配的结果找到后，C-Find SCP 将以 C-Find-Rsp 消息作答，每个回应消息都会附加一个作为结果的 C-Find IOD（图 42 和表 31）。

需要注意的是，总的来说，我们期望一次查询获得多个匹配结果（比如，给定病人的所有检查）。对每个匹配结果（除了最后一个），C-Find SCP 通过一个 DIMSE + IOD 对来响应，这时 DIMSE 自己的状态(0000,0900)设置为 FF00（等待，处理中）并且通过 IOD 传送当前匹配结果。最后的匹配结果将状态设置为 0000 并返回给 SCU，即表示这个操作已经圆满完成。如果什么都没有找到（没有匹配结果），那么 C-Find 通过单个 C-Find-Rsp 消息来响应，这个消息中数据集类型(0000,0800)将被设为 0101（空），并且状态(0000,0900)将被射程 0000（成功）。

如果你不是很幸运遇到了错误，那么 C-Find 会以一个状态字段中的错误值而终止。既然这样，C-Find SCP（你查询的那台设备）的 DICOM 一致性声明应该提供一些它自己所采用的状态错误代码。最容易出现错误的地方包括：

1. **DICOM 配置不正确。**比如，C-Find SCU 的 AE 没有录入到 C-Find SCP 配置中，因此作为非法数据（最常见的 DICOM 通讯错误）被拒绝。
2. **C-Find 请求 IOD 格式不正确。**没有满足等级查询规则；比如，如果你不知道检查关键属性，却试图查询一个影像设备。
3. **查询 SOP 的错误匹配。**如果一个应用程序只支持以病人为根的查询，但是另一个应用程序却向其发送一个检查根的 C-Find 查询，那么这个 C-Find 显然会失败的。再强调一次，对于同一个 SOP 类，两个 AE 只有作为 SCU 和 SCP 时才能正常通讯。在 C-Find 情况下，这个通讯还必须在相同的查询层次上。
4. **C-Find SCP 不支持：**不能指望任何 DICOM 设备或应用程序都能支持 C-Find SCP（将对查询给予响应）。比如，许多影像设备可能只是设计向远端归档推送图像（C-Store）。这将是它们的主要功能，它们可能不支持任何其他 DICOM 服务。所以还是需要在你试图对一个应用程序进行查询之前，先查看一下 DICOM 一致性声明中应用程序对 C-Find SCP 的支持情况。

表 31 C-Find-Rsp

消息字段	标签	VR	值/描述
组长度	(0000,0000)	UL	从(0000,0000)值字段的结尾到 C-Find-Rsp 消息的结尾的偶数个字节。
受影响的 SOP 类 UID	(0000,0002)	UI	包含针对这个 C-Find 查询根的 SOP UID；将会使用下面三者之一： 1.2.840.10008.5.1.4.1.2.1.1（病人） 1.2.840.10008.5.1.4.1.2.2.1（检查） 1.2.840.10008.5.1.4.1.2.3.1（病人-检查）

命令字段	(0000,0100)	US	8020
需要响应的消息 ID	(0000,0120)	US	需要设置为消息 ID(0000,0110)的值，这个字段在相关联的 C-Find-Rq 消息中使用。
数据集类型	(0000,0800)	US	如果要匹配返回的 IOD，任何非 0101 的值。
受影响的 SOP 实例 UID	(0000,1000)	UI	含有找到的 SOP 实例的 UID
状态	(0000,0900)	US	0000（如果成功），FF00（如果等待），或其他代表警告或错误的值

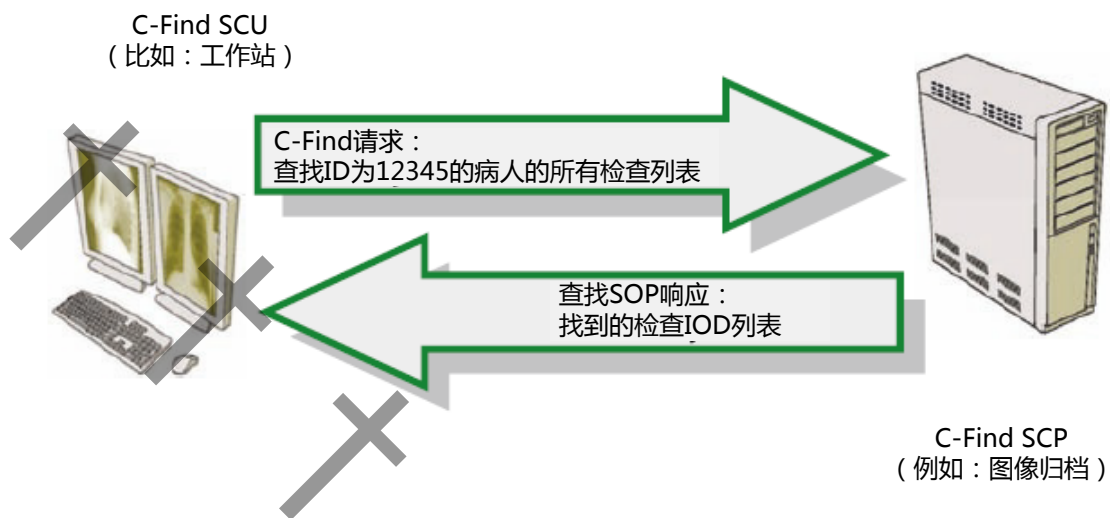


图 42 DICOM C-Find

C-Find，有点复杂，比 C-Echo 还复杂些，因此我们在附录 A.3 提供了 C-Find DICOM 编码实例。如果你真的想要深入了解所有 DICOM 字节，那么去看看吧。如你所见，即使是在 C-Find 层面，DICOM 消息也已经渐渐变得复杂了。

7.4.4

C-Cancel

假如你想要查询一个非常大的 PACS 归档，并且你没有设置特定的匹配属性值来限制你的查询。会发生什么？一个足够聪明的 DICOM 软件会认为你的查询太宽泛了，并且既会提醒你查询范围过大又会人为地依据阈值（许多系统使用 500 或类似最大匹配数量等）来限制匹配返回的数量。比较傻的系统（呜呼哀哉，这种比较常见）将会一直让你等着，取回一大堆你永远不需要的数据。

你可能会说：“我想取消这个！”；DICOM 标准委员会的人一定是听到了这样的呼声。

一个 C-Cancel(C-Cancel-Find)消息允许你来取消一个正在运行中的 C-Find 查询。C-Cancel 不需要响应，不传输数据（IOD 标识符），并且结构简单，详见表 32。唯一重要的元素是 (0000,0120)；它含有我们想取消的那个 C-Find 消息的 ID。所有 C-Cancel 要做的事情就是告知 C-Find SCP（正在处理已提交的 C-Find）需要停止(0000,0120)消息 ID 对应的 C-Find。换句话说，C-Cancel 不具有 C-Find 特定属性：它取消消息是基于消息 ID 的。因此，C-Cancel 和许多其他的 SOP 有关，不只是 C-Find（虽然当 C-Find 被取消时，DICOM 使用的是 C-Cancel-Find 名称）。

事实上，真的没法断言取消到底要花多少时间。C-Find SCP 应用可能要花相当长的时间

来终端它自己的查询操作，这可能致使不耐烦的用户在 C-Find 界面上一直不停地点击“取消”按钮。这倒是没什么，但的确不能起到任何效果；只不过是杀掉一个消息而已。

表 32 C-Cancel：取消较早前的命令消息

消息字段	标签	VR	值/描述
组长度	(0000,0000)	UL	从(0000,0000)值字段的结尾到 C-Cancel 消息的结尾的偶数个字节。
命令字段	(0000,0100)	US	0FFF
需要响应的消息 ID	(0000,0120)	US	需要设置为消息 ID(0000,0110)的值，这个字段在相关联的 C-Find-Rq 消息中使用。
数据集类型	(0000,0800)	US	0101

DICOM 标准的 PS3.7 部分提供了 C-Cancel 行为的更多细节，涉及“C-Cancel-Find”这样的消息。PS3.4 和 PS3.2 部分则称其为“C-Find-Cancel”，而其他部分提及相同的内容则使用“C-Cancel”。其实无所谓，朋友们，我们照样还是能理解的，不是吗？此外，我还没有提到 DICOM 标准中的最复杂部分吗？

7.5

影像设备工作列表（Modality Worklist）

影像设备工作列表（Modality Worklist (MWL)）SOP，记录在表 33 中，源自于 C-Find，但是用途完全不同。当影像设备技师准备开始工作室，他们非常希望了解当前待检查病人的列表，甚至希望病人数据已经导入到影像设备。这才是 MWL SOP 真正的用途：为影像设备预提取病人和扫描安排数据。

MWL 用在 DICOM 工作流程的最开始，甚至在图像存储或扫描之前。如图 43 所示。为了给影像设备提供扫描安排，MWL SCP 需要从什么地方获取这些内容。在多数情况下，扫描安排来自放射信息系统（Radiology Information System (RIS)），病人们会在扫描前先通过 RIS 系统进行扫描登记。RIS（或替代品，专门在临床场景使用）不是基于 DICOM 的，它们通常使用 HL7 标准，如我们将在第 14 章所见，因此要么通过 MWL 服务器要么通过一些额外的 DICOM 代理（broker）来将 RIS 数据转化为 DICOM 形式。

表 33 影像设备工作列表 SOP

SOP 类名	SOP 类 UID
影像设备工作列表	1.2.840.10008.5.1.4.31

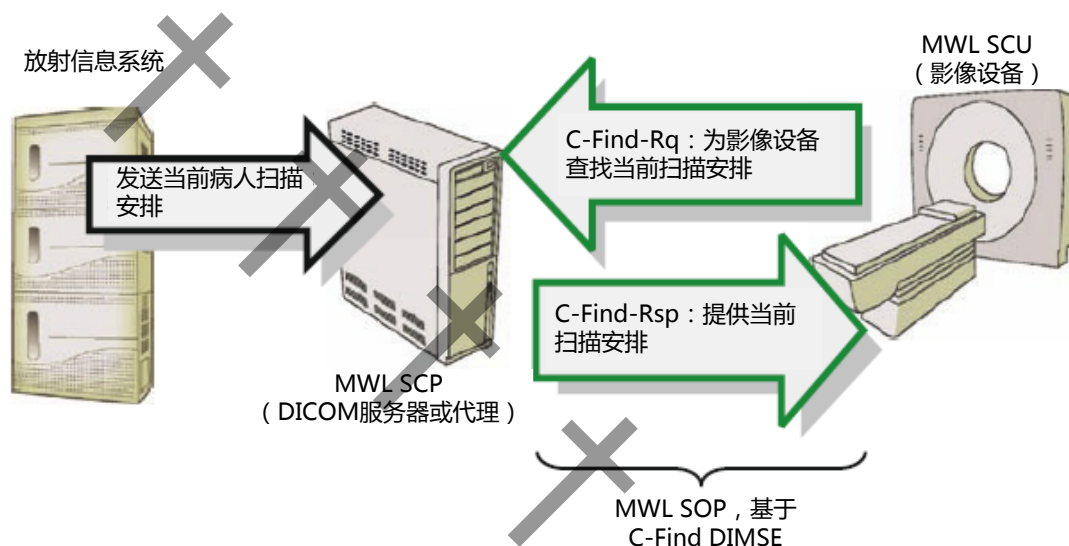


图 43 DICOM MWL 例子：基本的病人数据输入影像设备

The screenshot shows a software window titled "Modality Worklist, 2 records". On the left, there are input fields for patient information (First Name: Jane, Last Name: Doe, ID: 123456, Birth date: 5/1/1980, Sex: F), physician information (Requesting, Performing, Referring), and procedure schedule (Date: 9/5/2007, Time: 1:45:26 PM). Below these are fields for Station and Study (AE Title: CTSCANNER, Modality: CT, Accession: AN9876543, Description). At the bottom left are buttons for "Options...", "Add >>", "<< Edit", "Refresh", "Delete", and "Close". On the right, a table displays two records:

Patient Name	ID	Time	Modality	Access. #
Doe Jane	123456	05 Sep 2007 13:45	CT (CTSCANNER)	AN9876543
Sample Joe	517	05 Sep 2007 13:10	CT (CTSCANNER)	AN12345678

图 44 DICOM MWL：界面示例（AlgoM服务器，www.algom.com）

当病人数据从一个在 MWL SCP 上的 RIS 传出，它已经准备好让影像设备来获得工作列表。影像设备通常使用自动查询 MWL SCP，即配置一个提取当前工作列表的有规律的规则（每一分钟或两分钟）。在这些查询中，影像设备类型（如 CT、MR、超声）以及应用实体名（如 CTSCANNER1）常常用来在影像设备间识别检查目的地。影像设备技师也可以使用简单的查询界面来获得更个性化的安排更新，如图 44。

因此，MWL 解决了在影像设备上手工录入数据条目的问题，从源头上减少了人为错误和时间浪费。同样，MWL 的使用对于临床也很有意义；例如 MWL 也支持妊娠状态或过敏史等参数，这对于许多类型的放射检查来说是必不可少的。自动从 RIS 直接获取这些信息使整个临床过程变得健全和健壮。

7.5.1

The MWL IOD

MWL IOD 包含关于已安排的病人检查信息。在最简形式中，它包括下面这些最常用的元素，见表 34。注意(0040,0100)“已安排的检查步骤序列”元素在这里是采用 SQ VR 进行编码的（见 5.3.10）；因此，它包含一个项目(0040,0001)、(0040,0002)等的子序列。这个序列囊括了病人扫描的安排信息。当从 MWL SCP 进行提取时，序列匹配将它与特定的影像设备和时间间隔相结合。之前说过，这个列表是最简的，此外可能某影像设备会需要更多数据，具体内容将在影像设备的 DICOM 一致性声明中体现。如果某些特定属性不能够提供给影像设备，那么影像设备可能会取消 MWL 数据传输，坚持获得完整的属性列表。换句话说，如果你正在试图建立 MWL 连接，开始时一定要修改影像设备要求的属性列表并且确保这些属性都是可以提供的。

表 34 MWL IOD

(组, 元素)	名称
(0010,0010)	Patient Name 病人姓名
(0010,0020)	Patient ID 病人 ID
(0010,0030)	Patient's Birth Date 病人生日
(0010,0040)	Patient's Sex 病人性别
(0010,21C0)	Pregnancy Status 妊娠状态
(0008,0050)	Accession Number 检查号
(0032,1032)	Requesting Physician 提出检查请求的临床医生
(0008,0090)	Referring Physician's Name 相关的临床医生姓名
(0040,0100)	Scheduled Procedure Step Sequence 已安排的检查步骤序列
>(0040,0001)	Scheduled Station AET (name of the modality) 已安排的设备 AET (影像设备的名称)
>(0040,0002)	Scheduled Procedure Step Start Date 已安排的检查步骤开始日期
>(0040,0003)	Scheduled Procedure Step Start Time 已安排的检查步骤开始时间
>(0008,0060)	Modality 影像设备
>(0040,0006)	Scheduled Performing Physician's Name 已安排的执行检查医师姓名

7.5.2

The MWL DIMSE

MWL DIMSE 和咱们的老朋友 C-Find DIMSE 比起来没有什么新东西，这至少使得 MWL 实现起来比较简单(图 45)。因此没有 MWL-Rq 和 MWL-Rsp 表格出现在这里；它们与 C-Find-Rq 和 C-Find-Rsp 表格一模一样，见 7.4.3。唯一明显的变化是(0000,0002)中的属性值，现在这个值为了 MWL SOP 变为了 1.2.840.10008.5.1.4.31。

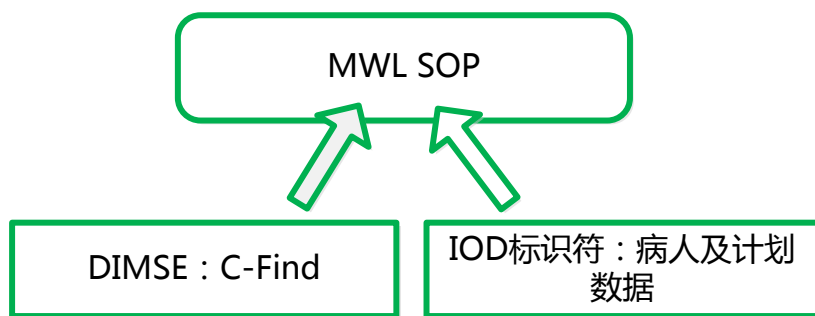


图 45 影像设备工作列表 SOP



7.6

基础的 DICOM 提取: C-Get

我们回溯几个章节,回想一下 C-Store 是怎么回事。如果你已经仔细地阅读了 DICOM Store SOP (用来传送 DICOM 图像) 的描述,你可能会因为一个问题而困惑:“我们怎么知道要如何存储?”的确, C-Store SOP 会一个接一个地传送预先识别的图像,但是在决定这些图像是什么方面没有提供任何帮助。

这太常见了,我们不用老是想图像选取。比如典型的情况是,将一个 CT 机设置为向一个数字归档发送所有新获取的 CT 检查。因为图像是 CT 机产生的,它清楚地知道那些检查需要进行 C-Store。但是如果我们想从数字归档中提取最新的病人图像又会怎样呢?

基本 DICOM 图像提取是通过 C-Get SOP 实现的。抽象地说, C-Get 将 C-Find 和 C-Store 混合到一个服务类中,而整个过程会先使用类似 C-Find 查询来标识所需图像,紧接着通过一个 C-Store 来提取。就像用 C-Find 一样,我们在形成 C-Get 请求时会附加一个 IOD 标识符对象,这个对象中会包含图像搜索条件。当这个请求发送到 C-Get SCP 时, SCP 首先是用查询参数来查找图像,然后调用 C-Store 来将图像返回给 C-Get SCU,如图 46 所示。因此,一个 C-Get SCU (比如图 46 中我们的图像提取工作站) 必须也要作为 C-Store SCP 运行,这样才能接受返回的图像。如你所见,每个在 C-Get SCP 上符合条件的单个图像都囊括在一个独立的 C-Store 操作中,并且最终被发送到 C-Get SCU (C-Store SCP)。在发送过程中, C-Get SCP 也能够发送 C-Get 响应,响应中的状态设置为等待,用来告知 C-Store 子进程仍然在运行。在这些等待响应中, C-Get IOD 响应同样包含当前 C-Get 查询属性的匹配值。当所有存储操作被执行时, C-Get SCP 返回最终的 C-Get-Rsp, 它不包含 IOD 并且发出 C-Get 操作结束的信号。如果在整个链条中出现了问题,那么整个 C-Get 操作会取消,取消是通过 C-Get 状态字段中的一个适当的错误值来实现的。

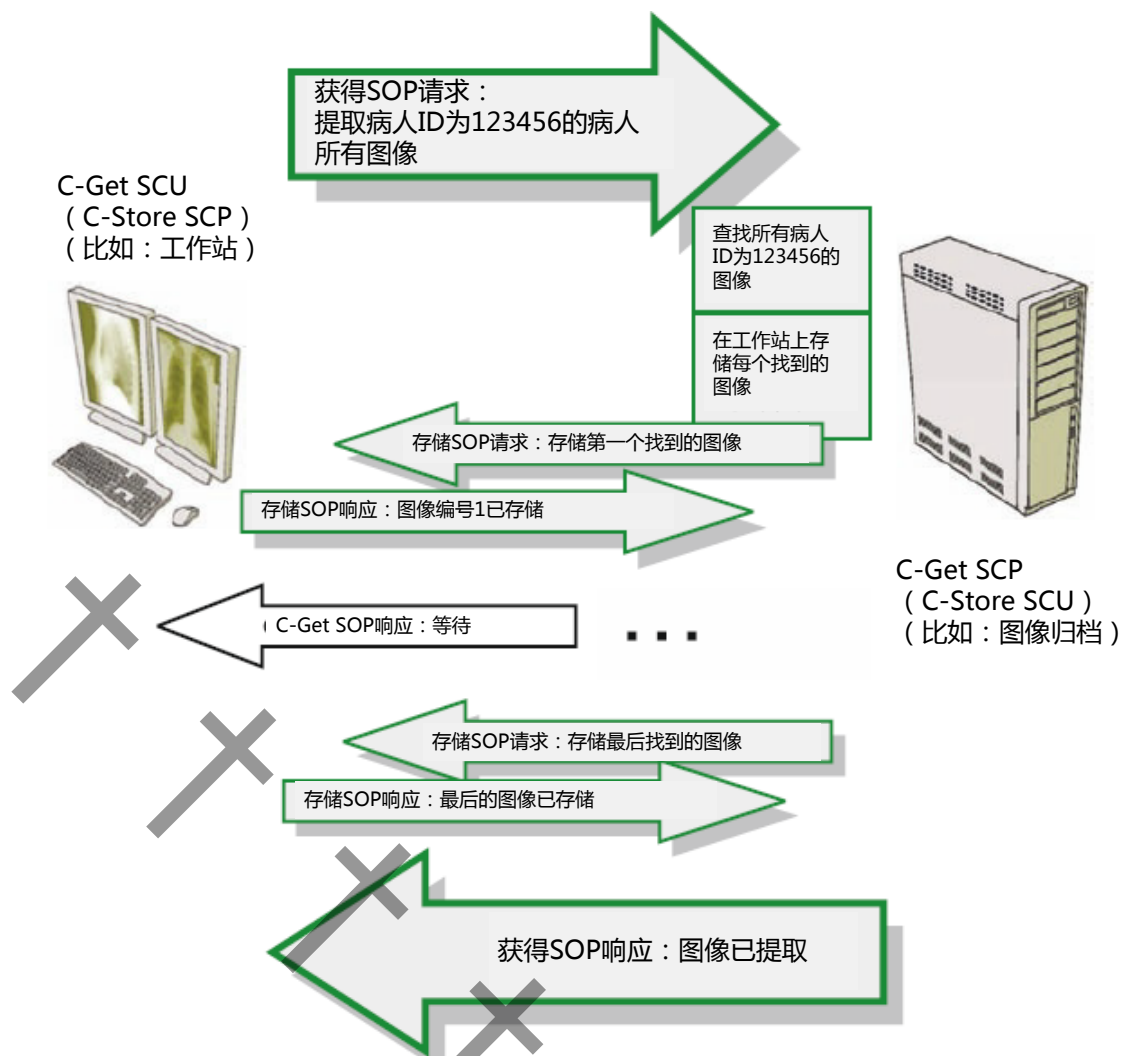


图 46 用 C-Get 服务来包装 C-Store 服务

表 35 C-Get SOP

SOP 类名	SOP 类 UID
病人根 Q/R 获取	1.2.840.10008.5.1.4.1.2.1.3
检查根 Q/R 获取	1.2.840.10008.5.1.4.1.2.2.3
病人-检查根 Q/R 获取（弃用）	1.2.840.10008.5.1.4.1.2.3.3

C-Get 从 C-Find 那里继承了三个提取根，如表 35。最常使用的是以检查为根，这意味着我们提取单个检查。一如既往，C-Get SOP 符合相同简单 SOP 布局，即一个 DIMSE 命令和 IOD 数据对象，如图 47 所示。

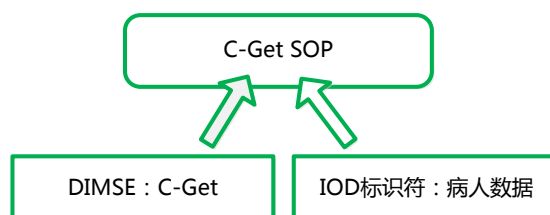


图 47 C-Get SOP

表 36 C-Get IOD 实例

(组,元素)	名称	实例	匹配 (见 5.3.2)
(0008,0052)	提取层次	STUDY	可以是“病人”、“检查”、“序列”或“图像”。这个元素定义了等级查询的层次
(0010,0020)	病人 ID	12345	单一值
(0020,000D)	检查实例 UID	1.2.840.1234567	单一值或列表匹配

7.6.1

The C-Get IOD

为了能够提取图像，C-Get IOD 负责传送查询属性。这和 C-Find IOD 很相似；在 C-Get SCP 结尾我们提供的属性将得到匹配。任何符合这些属性的图像都将返回给我们。因此，C-Get 的属性也必须符合 DICOM 病人-检查-序列-图像的等级查询逻辑，就像我们在 C-Find 中所说的那样。

如你在表 36 中所见，C-Find 和 C-Get IOD 之间的主要不同是：C-Get SCU 必须在提取层面提供用来标识一个实体的唯一键值。也就是说，为了替代 C-Find 的必须/可选的键值，C-Get 严格按照等级键值来执行，即为了提取图像 C-Get 会在当前层级和更高层级通过唯一的键值来识别图像。因此，C-Get 标识符 IOD 必须包含：

1. 查询/提取层级(0008,0052)：这定义了提取层级。
2. 唯一键值属性：这些属性中可能包括“病人 ID”(0010,0020)、“检查实例 UID”(0020,000D)、“序列实例 UID”(0020,000E)、以及“SOP 实例 UID”(0008,0018)。

C-Get 标识符 IOD 一定不能包括任何其他的可选项键值。在 C-Get 中所有唯一的键值属性必须有准确的值；即他们只能是单一的值匹配¹。C-Get SCP 通过一个标识符 (DICOM 数据对象) 针对每次匹配产生一个 C-Get 响应，这个标识符具有所有键值字段以及所有已知所需属性的值。²

C-Get 很好地实现了图像的查询和提取工作，但是很明显等级的层次-键值-唯一性模型具有很大的局限性，可以说在考虑用户友好性方面做得太差了。四个层级键值属性中只有一个——病人 ID，会在医疗界面上经常见到，PACS 用户会比较熟悉；而其他三个（检查、序列和 SOP 实例 UID）通常自动有影像设备生成，用户不慎了解。这三个属性由最多 64 个字符组成，永远不会显示给用户，也没有特定的意思，因此也从不会放在到人及交互界面上。这意味着当 PACS 用户想使用 C-Get 进行交互时，他们只能使用病人层次，必须要提取给定病人 ID 的所有图像。然而，病人层次的提取很少再放射领域出现；在放射领域，我们习惯于从最新检查开始并且喜欢在检查层次工作。此外，提取给定病人的所有图像可能产生数据量巨大的下载操作。

¹ 检查、序列和图像层次可能支持列表匹配，见 7.4.1。而病人 ID 必须是唯一的。

² 像往常一样，这里也有一个相关的 C-Get 行为，很少需要它或实现它，它是用来处理任何非等级提取和相关提取的。大多数 DICOM 一致性声明将包含一个警告：“SCP 将不能支持 C-Get 扩展协商”，这也说明了，对于 DICOM 厂商来说，高级特性通常不是很常用。

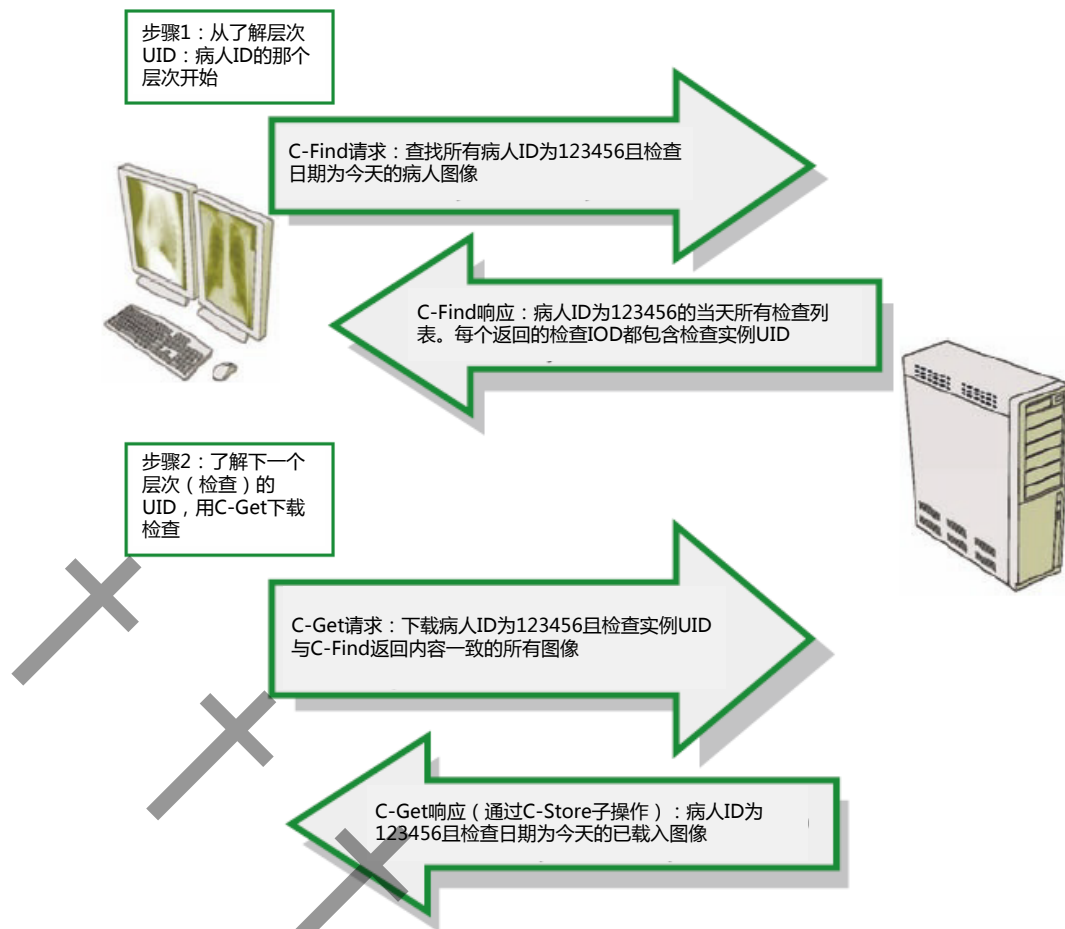


图48 C-Find和C-Get结合的实例：提取已知病人ID下当天产生的所有病人图像

表37 C-Get-Rq

消息字段	标签	VR	值/描述
组长度	(0000,0000)	UL	从(0000,0000)值字段的结尾到 C-Get-Rq 消息的结尾的偶数字节
受影响的 SOP 类 UID	(0000,0002)	UI	包含用于 C-Get 查询根的 SOP UID；因此使用以下三者之一： 1.2.840.10008.5.1.4.1.2.1.3（病人） 1.2.840.10008.5.1.4.1.2.2.3（检查） 1.2.840.10008.5.1.4.1.2.3.3（病人-检查）
命令字段	(0000,0100)	US	0010
消息 ID	(0000,0110)	US	消息的数字唯一 ID
优先级	(0000,0700)	US	下列选择之一： 0002（低优先级） 0000（中优先级） 0001（高优先级）
数据集类型	(0000,0800)	US	任何非 0101 的值

那么我们还能使用 C-Get 吗？当然可以，只要在使用 C-Get 之前先使用 C-Find。C-Get 真的不能支持可选键值匹配，但是 C-Find 可以。比如，任何系统都可以使用 C-Find 通过病

人 ID 和检查日期来定位最新的病人检查。当 C-Find SCP 响应这个查询时，还会返回一个 C-Find-Rsp IOD，这个 IOD 包含了 C-Get 需要的内容：下一层次的唯一键值 UID。比如，在病人层次上的 C-Find 会返回检查实例 UID。当这个 UID 变为已知，我们就可以考虑使用 C-Get 来提取图像了，见图 48。总之，这个方法在每个 PACS 中都会实现。用户首先使用界面来进行一个查询，并且在试图调取任何内容前总是先进行查找（C-Find）。当查找结果返回给用户时，查询结果中还会包含那些隐藏着的等级层次 UID，这很重要，因为 PACS 软件的后续 C-Get 下载需要使用它们。或者说，C-Get 是用来调取那些已知层次键值属性的图像的。

7.6.2

C-Get DIMSE

C-Get DIMSE 与 C-Find 及 C-Store 非常相似。C-Get-Rq 如表 37 所示。(0000,0800) “数据集类型”字段不是空(0101)，C-Get-Rq 消息后面紧跟着传送查询属性的 C-Get IOD。DIMSE+IOD 的组合从 C-Get SCU AE 发送到 C-Get SCP AE。C-Get SCP 首先接收 C-Get DIMSE 并识别它，之后使用 C-Get SCP AE 来查询它自己的本地数据库中相匹配的结果。当匹配结果找到时，C-Get SCP 通过 C-Get-Rsp 消息来响应，当然还补充有找到的 C-Get IOD。

Table 38 C-Get-Rsp

表 38 C-Get-Rsp

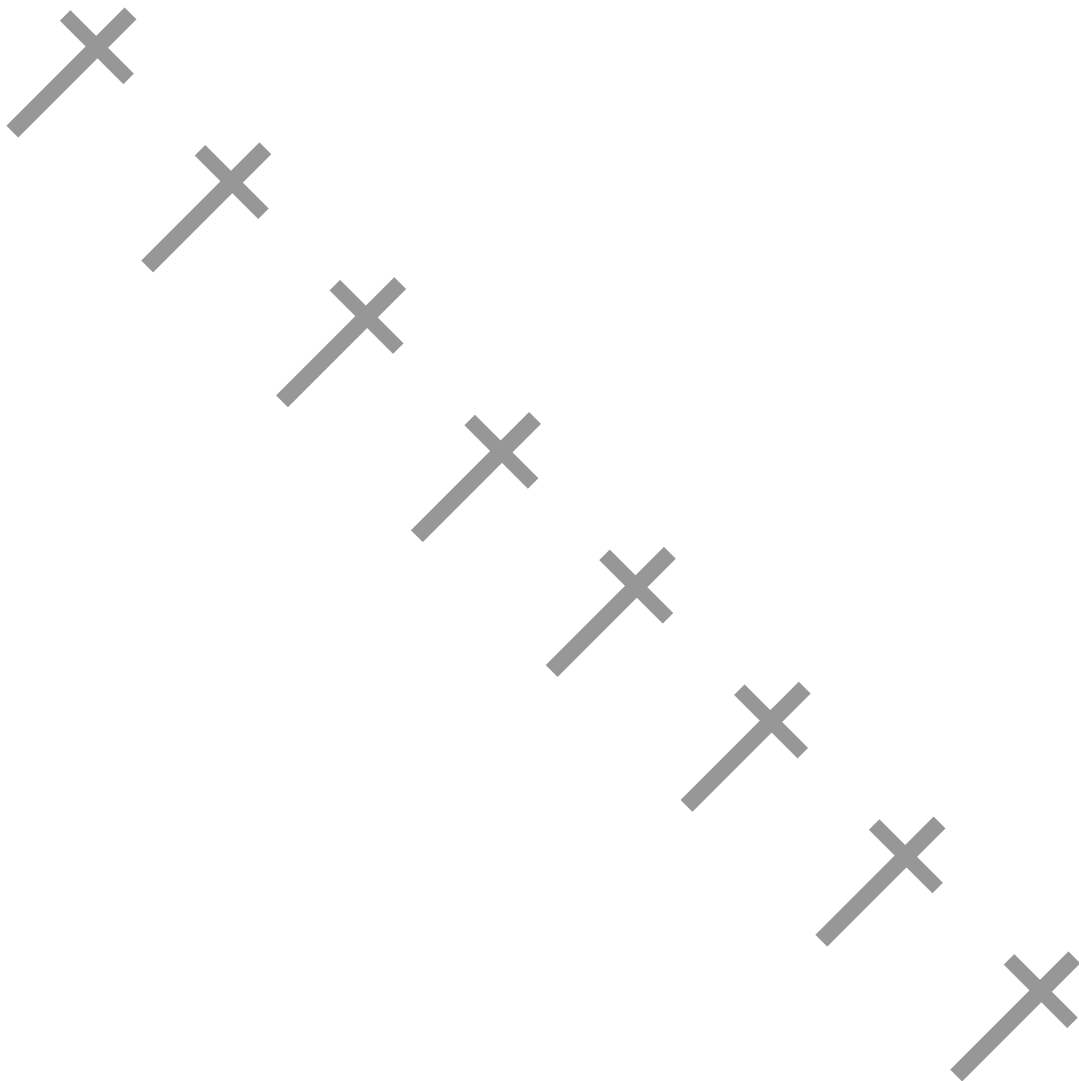
消息字段	标签	VR	值/描述
组长度	(0000,0000)	UL	从(0000,0000)值字段的结尾到 C-Get-Rq 消息的结尾的偶数个字节
受影响的 SOP 类 UID	(0000,0002)	UI	包含这个 C-Get 查询根的 SOP UID；因此使用以下三者之一： 1.2.840.10008.5.1.4.1.2.1.3（病人） 1.2.840.10008.5.1.4.1.2.2.3（检查） 1.2.840.10008.5.1.4.1.2.3.3（病人-检查）
命令字段	(0000,0100)	US	8010
被响应的消息 ID	(0000,0120)	US	将设置为消息 ID(0000,0110)字段的值，与相关的 C-Get-Rq 消息中一致。
数据集类型	(0000,0800)	US	如果反悔了成功匹配的 IOD，那么为任何非 0101 的值，
状态	(0000,0900)	US	0000（如果成功） FF00（如果等待） 或其他警告或错误值
剩下待处理的子操作数量	(0000,1020)	US	剩下待处理的 C-STORE 子操作数量，这些子操作由 C-GET 操作调用
已完成的子操作数量	(0000,1021)	US	已经成功处理的 C-Store 子操作数量，这些子操作由 C-GET 操作调用
已失败的子操作数量	(0000,1022)	US	已经处理失败的 C-Store 子操作数量，这些子操作由 C-GET 操作调用
产生警告的子操作数量	(0000,1023)	US	已经处理完但产生警告响应的 C-Store 子操作数量，这些子操作由 C-GET 操作调用

由于 C-Get 有一个嵌入其中的 C-Store 操作，因此表 38 中的 C-Get-Rsp 的最后四个属性常用来返回当前 C-Store 的执行状态。他梦告诉你有多少图像（单独的 C-Store 子操作）还没有传送；已经传送了多少；有多少失败了；以及有多少执行过程中产生警告。比如，如果在

C-Store 执行过程中，一个 C-Get SCP 返回“等待”消息。那么这些数字（子操作计数）表示整个图像提取的过程。

在大多数情况下，C-Get 警告涉及一类可容许的差错，不会导致整个 C-Get 失败（比如，当 C-Get SCP 不能支持可选的匹配参数时）。换句话说，如果 C-Get SCP 未能支持任何必须的参数，那么就会造成一个 C-Get 错误和失败。当所有提取结果都已经完成后，C-Get-Rsp 会传送最终的消息。如果每件事都没问题，那么这个消息将不会有任何附加数据（(0000,0800) 设置为 0101）；(0000,1020) 设置为零；(0000,1021) 包含所有已输图像的总计数；(0000,1022) 也会设置为零。

就行 C-Find。C-Get 可能会在任何时间点通过 C-Cancel 消息取消（见 7.4.4）一样。我们之前所知，鉴于 DICOM 创始人偶尔乐于使用多个名字来描绘同一事物，因此标准 PS3.7 中所提及的 C-Cancel-Get-Rq，别发愁，他的确就是 C-Cancel，我们的老朋友，我们曾在讨论 C-Find 的时候第一次提到它。



7.7

高级 DICOM 提取：C-Move

实际上，C-Move 和 C-Get 区别不大，只不过稍微复杂了点：你可以将图像移动到第三方。换句话说，C-Get 只能向发出请求的 AE 返回图像，而 C-Move 能够将图像发送（C-Store）到任何一个其他的 AE（图 49）。

在最简场景中，AE 可以要求将图像移动给自己，这就相当于进行了类似 C-Get 的图像下载。比如在图 49 中你可以将工作站 2 和工作站 1 想象成同一个设备。事实上，这是一个很常见的情况：这种处理方法在许多 PACS 中是很有好处的，因此 C-Move 常常用来实现类似 C-Get 的下载。如果工作站 1 将归档图像 C-Move 给自己，那么至少说明工作站 1 处于工作状态，它能够发出 C-Move 请求（C-Move-Rq）；它需要这些图像；且它应该能够接受他们。如果将这些图像发给一个真的第三方，而不是它自己，比如真的发给放在别处的工作站 2，那就确实不太容易了。因为，工作站 2 可能不会明确给予任何响应，它那时也许正好不可用，也有可能已经准备好来处理由归档发送过来的归档图像流，当然请记住我们所说的请求是由工作站 1 发出的。

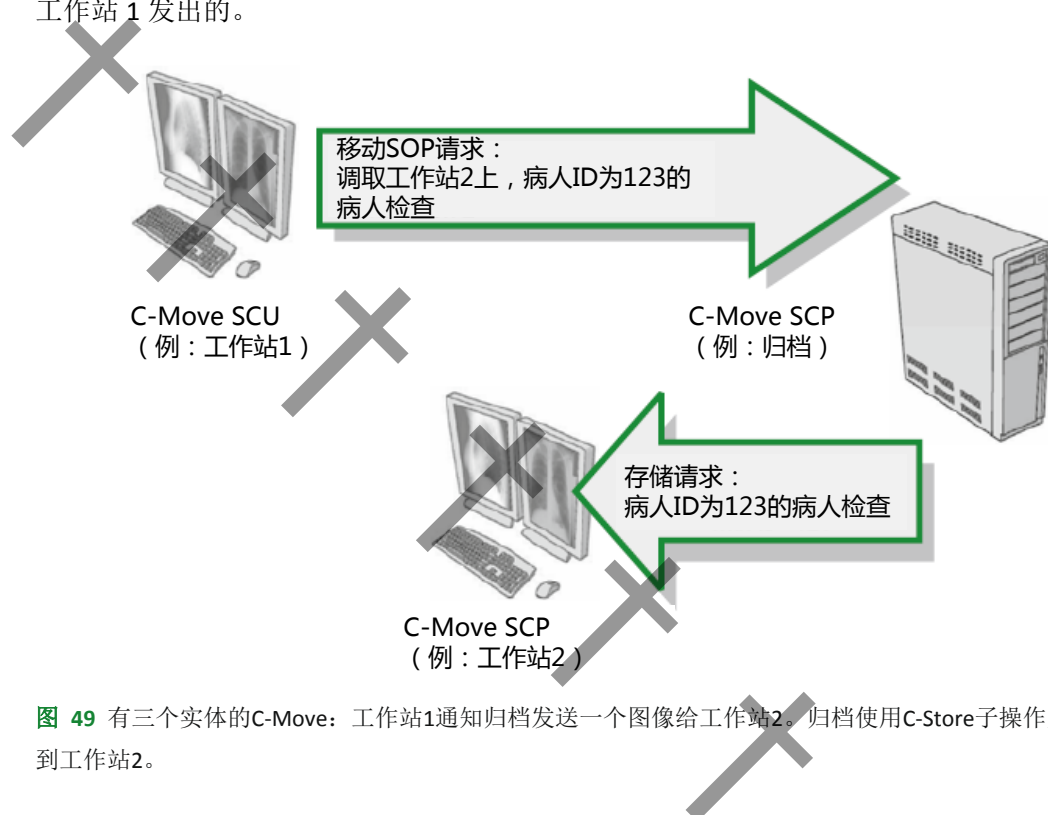


图 49 有三个实体的C-Move：工作站1通知归档发送一个图像给工作站2。归档使用C-Store子操作发送图像到工作站2。

因此这种复杂的、具有三个实体的 C-Move 通常用在比较容易控制的环境中——比如一个大型的 PACS 归档服务器群组。一个复杂的 PACS 归档会有许多用来存储图像的服务器，并且具有许多控制存储的处理节点。在这种状况下，当归档系统之外的一个用户请求调取归档图像时，其中一个归档服务器将会接收这个用户的请求，当它发现其他归档服务器存有用户所需图像时，它会要求那台归档服务器回传图像。对于用户来说，在整个过程中仿佛所有图像都是从一个实实在在的服务器下载的，其实是 C-Move 在背后起了作用。

C-Move 和 C-Get 的另一个不同点也是来源于 C-Move 中的三实体原则：C-Move 需要知道向何处返回图像。事实上，这个问题在 C-Get 中从未提及；C-Get SCP 总是向请求获取图像的实体（C-Get SCU）回传图像。从术语角度来说，C-Get 是在单路连接（这个有趣的话题会在第 9 章讨论）上工作的：如果我给你打电话，你会拿起听筒开始和我讲话，其实你回答

我时，并不需要知道我的号码。

C-Move 看起来更像是我给你打电话并且让你去给朋友 John 打电话。现在你需要找出 John 的电话号码并且判断你是否能够给他打电话。在 DICOM 中，你需要打开另一个连接去和 John 通话。许多问题会出现在你的这个复杂任务中。你可能找不到 John 的电话号码，你也可能拨错了电话。在 DICOM 中，之前图示中归档服务器的 AE 配置清单必须具有工作站 2 准确的名称、IP 地址和端口号，只有这样归档服务器才能成功的发送图像。

事实上，即使归档需要将图像 C-Move 回工作站 1(两个实体间的 C-Move，类似 C-Get)，那么它也需要知道工作站 1 的完整配置，并且用另外一路连接来回复。还是以我们打电话为例，我正在给你电话，你回答“你好，稍后我会给你回电。”挂上电话并且给我回电继续我们刚才的对话。如果你忘了我的电话号码，我们就卡住了。这很像那种希望节约手机费的策略，但是它准确地反应出 C-Move 的功能：它总是打开另一个连接（DICOM 连接）来返回它自己要的东西——图像。

撇开建立连接方面的业务不谈，C-Move 和 C-Get 真的很相近。当然，它也有三个 DICOM 查询/提取根，这三种查询都具有自己的 SOP（表 39）。C-Move 的 SOP 结构和别人没有两样（图 50）。

表 39 C-Move SOP

SOP 类名	SOP 类 UID
病人根 Q/R 获取	1.2.840.10008.5.1.4.1.2.1.3
检查根 Q/R 获取	1.2.840.10008.5.1.4.1.2.2.3
病人-检查根 Q/R 获取（弃用）	1.2.840.10008.5.1.4.1.2.3.3

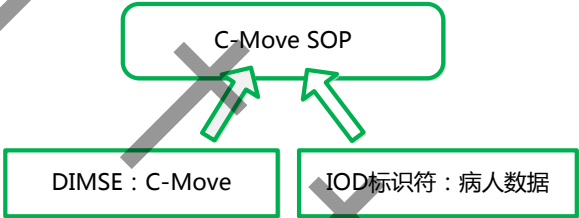


图 50 C-Move SOP

7.7.1

The C-Move IOD

如表 40 所示，一个 C-Move IOD DICOM 对象为图像传送查询属性；与 C-Get 一样，图像需要遵从“唯一层次键值”的提取方法。因此，就像 C-Get 一样，为了让 C-Move 请求更加实用以及为了支持可选查询键值，PACS 用户界面会让你先使用 C-Find 基于常用的原始查询悄悄地进行一次唯一层次键值的提取。当获得层次键值以后，我们就可以发起一次完整的 C-Move 请求了。这看起来有点费力不讨好了：正如 C-Get 介绍之初所说，C-Get 或 C-Move 实际上就等于(C-Find) + (C-Store)的混合体；C-Move 和 C-Get 里面使用的受限制（层次键值唯一）C-Find，实际上导致大家都需要在移动或获取之前首先识别键值，这就要求必须使用另一个常规的 C-Find 才能搞定。然而现实中图像传输之前，这种不同的 C-Find 操作确实大有好处：

表 40 C-Move IOD 实例

(组,元素)	名称	实例	匹配（见 5.3.2）
--------	----	----	-------------

(0008,0052)	提取层次	STUDY	可以是“病人”、“检查”、“序列”或“图像”。这个元素定义了等级查询的层次
(0010,0020)	病人 ID	12345	单一值
(0020,000D)	检查实例 UID	1.2.840.1234567	单一值或列表匹配

1. 它可以快速解答一个问题，我们所寻求的东西在 SCP 归档中是否存在。比如，如果“今日”的所有检查中没有病人 ID 为 123456 的检查，那么就没有必要把这些不存在的检查 C-Get 或 C-Move 到某些地方了。
2. 如果真的找到了一些东西，那么 C-Find 同样会返回这其中符合查询条件的到底有多少。比如，如果我们发现病人 123456 有多个检查（每个里面都有 2000 幅图像），我们确实应该好好想想我们到底想要下载哪些图像。载入图像是需要时间的，有时会用到相当长的时间。C-Find 相当于强迫你去考虑什么是我真正需要的以及我所能忍受的适度时长。在使用 C-Move 之前，你真的不用去担心 C-Find 的配置问题，PACS 用户界面会为你搞定一切的，它会首先出现并询问你想要移动或下载的内容。

7.7.2

The C-Move DIMSE

C-Move DIMSE 与 C-Get 也非常相似，但非常明显，它增加了一个移动目的地属性，如表 41 所示。事实上，DICOM 的 PS3.7 提出了另一个有趣的名字，“C-Get-Move 服务”，这只能说明即使在撰写 DICOM 标准时也非常容易将他们搞混。

表 41 C-Move-Rq

消息字段	标签	VR	值/描述
组长度	(0000,0000)	UL	从(0000,0000)值字段的结尾到 C-Move-Rq 消息的结尾的偶数字节
受影响的 SOP 类 UID	(0000,0002)	UI	包含用于 C-Move 查询根的 SOP UID；因此使用以下三者之一： 1.2.840.10008.5.1.4.1.2.1.3（病人） 1.2.840.10008.5.1.4.1.2.2.3（检查） 1.2.840.10008.5.1.4.1.2.3.3（病人-检查）
命令字段	(0000,0100)	US	0021
消息 ID	(0000,0110)	US	消息的数字唯一 ID
优先级	(0000,0700)	US	下列选择之一： 0002（低优先级） 0000（中优先级） 0001（高优先级）
数据集类型	(0000,0800)	US	任何非 0101 的值
移动目的地	(0000,0600)	AE	设置为目的地的 DICOM AE 名称，在目的地 AE 上将执行 C-Store 的子操作。

新近出现的（相比 C-Get）移动目的地(0000,0600)属性规定了图像需要移动的目的地，当然移动是通过 C-Store 子操作（前例中的工作站 2）实现的。当 C-Move SCP 接受了请求，

他就会进行一下工作：

1. 基于附加标识符 IOD 中的唯一层次键值，C-Move SCP 就可以在他自己的数据库中找到匹配的图像了。
2. 对于每个图像来说，C-Move SCP 将分别发起一个 C-Store 子操作，请求在移动的目的地 AE 上存储图像。需要注意的是，在 C-Move-Rq 中将只提供 AET。根据这个名称，C-Move SCP 会在它自己的目的地列表中找到其他的 AE 参数（IP 地址、端口号）并在它与目的 AE（前例中的工作站 2）之间打开第二个连接。
3. 如图像被存储在目的地 AE 上一样，C-Move SCP 通过暂缓发送的 C-Move-Rsp 来告知发起请求的 AE（C-MOVE SCU），就像 C-Get 一样。尤其会告知这些所包含的 C-Store 子操作的计数（成功、失败、警告、错误）。
4. 当向移动目的地 AE 的 C-Store 图像传输工作结束时 C-Move SCP 会向发起请求的 C-Move SCU（或者如果有失败，那么会发出一个报错 C-Move-Rsp，见表 42）发出最终成功的 C-Move-Rsp。

表 42 C-Move-Rsp

消息字段	标签	VR	值/描述
组长度	(0000,0000)	UL	从(0000,0000)值字段的结尾到 C-Move-Rq 消息的结尾的偶数字节
受影响的 SOP 类 UID	(0000,0002)	UI	包含这个 C-Move 查询根的 SOP UID；因此使用以下三者之一： 1.2.840.10008.5.1.4.1.2.1.3（病人） 1.2.840.10008.5.1.4.1.2.2.3（检查） 1.2.840.10008.5.1.4.1.2.3.3（病人-检查）
命令字段	(0000,0100)	US	8021
被响应的消息 ID	(0000,0120)	US	将设置为消息 ID(0000,0110)字段的值，与相关的 C-Move-Rq 消息中一致。
数据集类型	(0000,0800)	US	如果反悔了成功匹配的 IOD，那么为任何非 0101 的值，
状态	(0000,0900)	US	0000（如果成功） FF00（如果等待） 或其他警告或错误值
剩下待处理的子操作数量	(0000,1020)	US	剩下待处理的 C-STORE 子操作数量，这些子操作由 C-MOVE 操作调用
已完成的子操作数量	(0000,1021)	US	已经成功处理的 C-Store 子操作数量，这些子操作由 C-MOVE 操作调用
已失败的子操作数量	(0000,1022)	US	已经处理失败的 C-Store 子操作数量，这些子操作由 C-MOVE 操作调用
产生警告的子操作数量	(0000,1023)	US	已经处理完但产生警告响应的 C-Store 子操作数量，这些子操作由 C-MOVE 操作调用

还是老样子，C-Move SCU 可以使用 C-Cancel 来取消正在进行的 C-Move 操作。

7.7.3

C-Move vs. C-Get

由于 C-Get 和 C-Move 具有相似的功能，因此值得我们花些精力来仔细区分一下。

C-Move 必须要知道它自己的目的地是一个很好的安全特定。如果某个邪恶的黑客试图盗取一些图像，那么他将会一败涂地。因为，他那邪恶的工作站并不在 C-Move 的目的地列表中（C-Get 也许返回）。只向已知目的地发送图像绝对是 C-Move 的一个优质特定，它切实地提高了安全性。此外，DICOM（在 PS3.7 中）谈的更深入些：

“如果忽略需要建立两个连接的实施，那么在多数的环境下，C-Move 是一个更简单的选择。C-Get 服务的使用反而不会广泛的使用。C-Get 可能只会是一些特殊的情况下使用，比如系统不支持建立多路连接。在本标准的这个版本中还是保留了 C-Get，以便向后兼容标准的先前版本。”

或者说，C-Get 实际上已经过时了，只是为了向后兼容才保留下来，这难道是真的吗？

称 C-Get 过时，或者认为单路连接模型真的比更先进的双路连接模型逊色实在是对此部分 DICOM 的一种误解。采用单路连接模型的 C-Get，使用户远离了一个最严峻的 DICOM 问题：动态配置通讯设置（每个实体都需要提前知道可以联系谁和如何联系）。当你只能期待你电话簿中记录的人打来电话时，生活似乎变得索然无味了，不是吗？更何况，任何事情都会改变，电话簿也是需要更新的。

在今天的放射领域，动态的东西是非常受人欣赏的，因此在 DICOM 世界中，那些所谓的优势和安全特性其实代表着缺乏动态的功能性。

用 C-Get：你可以进行远程放射诊断；你可以在世界上的任何地方连接你的 DICOM 设备；你可以使用移动终端；你还可以随便重建你的网络连接，而不用像使用 C-Move SCP 那样需要顾及你电话簿的更新。

用 C-Move，你被限制在一个非常静止的世界中。甚至一个小小的祸事（比如，你的系统管理员常常喜欢更改 IP 地址）就会导致整个 DICOM 网络的故障。在通讯的世界中，任何新的链接或者连接都需要增加维护工作并且造成额外的花销，这意味着事情将很容易夭折。PACS 安全性需要在功能和能力的基础上经受打击，而不是在复杂性上经受打击。

单路连接 C-Get 模型增加了另一个自由度：可在防火墙下正常运作。看看以下非常典型的场景：你需要从远端装置读取图像。你知道他们的 AE 设置，你能够连接他们，你甚至能够查询他们，但是当你试图使用 C-Move 提取图像时，一切都将失败。原因很简单：C-Echo（用作连接性验证）、C-Find（用作查询）和 C-Get（用作提取）是在单路连接上工作的。当你在防护墙保护下（比如你的医院，甚至你那安装了防火墙的电脑）初始化一个单路连接时，多数防火墙会允许建立连接；因为向外的请求通常被认为是友善的。Ranger，C-Move-Rsp 暗示远端系统用另一个、向内的连接来联系你，者将被防火墙阻止。从这个角度来看，C-Get 比 C-Move 还要安全，因为它不用在防火墙上钻个洞洞，能够和你的安全策略（VPN 或更好的策略，但是它们通常要耗费更多的时间）合并共处、另一方面，C-Move（前面提到过）更适合用在复杂的、自包含的、多服务器的 PACS 归档上，这个环境下图像存储和处理负荷分布在多个归档计算机上。在这种控制良好、封闭的环境中，变化是很少的，因为它就是使用这种单一、不可分的机制；它的目标就是静态。既然如此，C-Move 为内部的 PACS 功能性带来了许多益处。“C-Move 主内、C-Get 主外”看起来才是最高效的组合，请利用两种图像传输协议来实现最可靠和高效的 DICOM 方案吧。

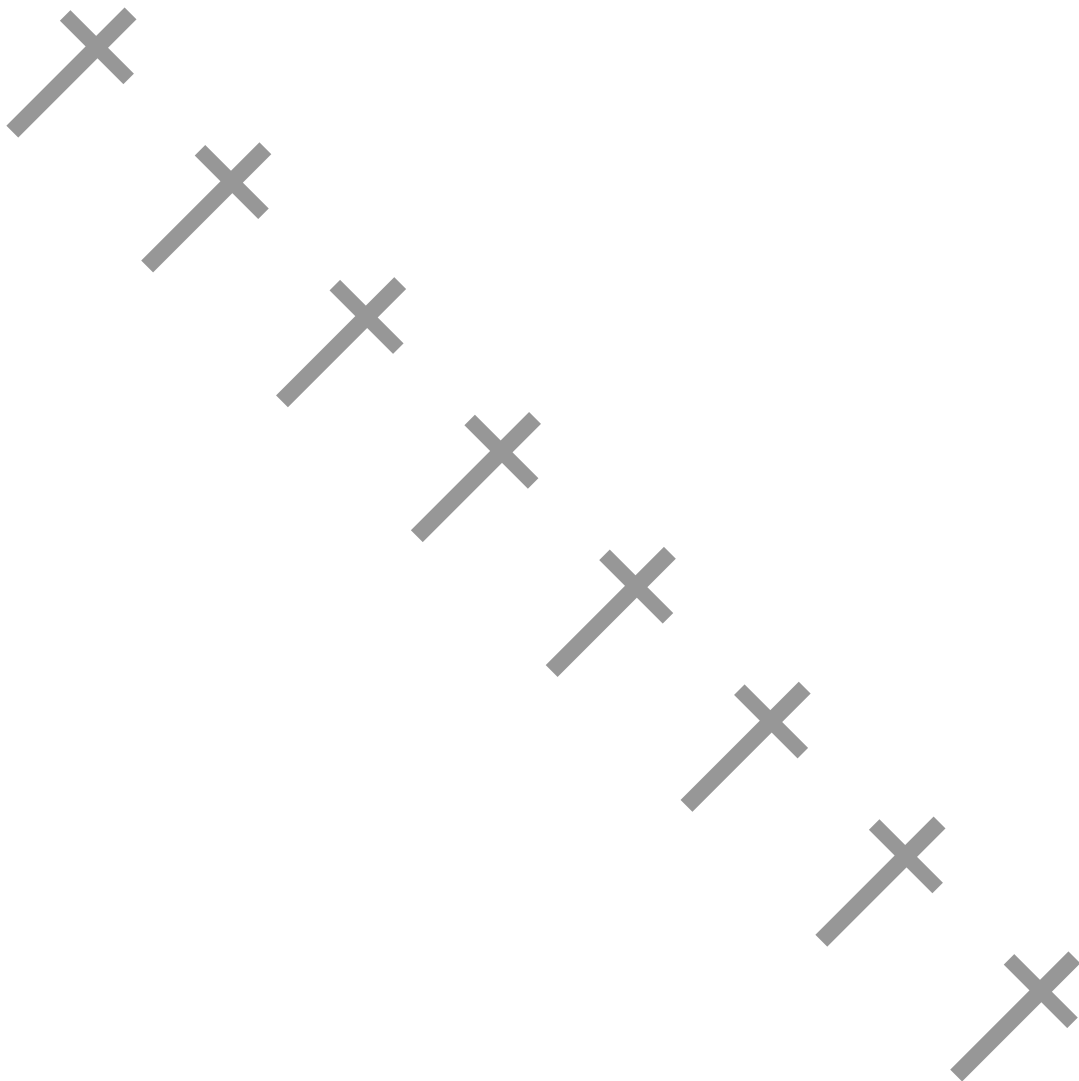
7.8

DICOM Ping, Push, and Pull

Practical DICOM jargon used by field engineers and PACS support specialists differs from that of the DICOM standard. You will not likely hear any references to C-Echo, C-Get/C-Move, or C-Store. Instead, they talk about DICOM ping, DICOM pull, and DICOM push. These are the same:

1. DICOM ping: sending a signal to another AE using the DICOM protocol to determine whether it is DICOM-connected. This is DICOM C-Echo (Verification SOP).
2. DICOM pull: pulling (retrieving) images from another AE. Corresponds to sending either a C-Move-Rq or a C-Get-Rq.
3. DICOM push: sending (pushing) the images to another AE (opposite of DICOM pull). Corresponds to C-Store.

实际情况下，场地工程师所使用的 DICOM 行话



7.8

DICOM Ping, Push, and Pull

实际工作中，场地工程师和 PACS 从业人员使用的 DICOM 行话和 DICOM 标准所述有所不同。没有 C-Echo、C-Get/C-Move 或 C-Store，通常是以 DICOM 连接验证（DICOM ping）、DICOM 推送（DICOM push）和 DICOM 获取（DICOM pull）来代替。其实他们没有本质区别：

1. DICOM 连接验证：使用 DICOM 协议发送一个信号给另一个 AE，来探测它是否已经建立 DICOM 连接。其实就是 DICOM C-Echo（校验 SOP）。
2. DICOM 获取：从另一个 AE 获取（提取（Retrieve））图像，相当于发送 C-Move-Rq 或者 C-Get-Rq。
3. DICOM 推送：向另一个 AE（与 DICOM pull 相反）发送（推送）图像。相当于 C-Store。

设计 DICOM 推送是为了对图像进行自动路由（比如，从影像设备到归档）。DICOM 获取则主要用于在用户选择他们想看的内容时所使用的用户界面上，随后用户就可以获取（提取）图像了。从 DICOM 的角度来说，“获取”（C-Move/C-Get）只不过是启动了一个指向你计算机（AE）的“推送”（C-Store）。本质上，获取是一个聪明的推送方法，因为你可以选择让远端应用程序将什么内容推送给你。

当你打开一些 DICOM 软件的界面时，你会看到一个用来在你的 AE 和远端应用程序之间测试连接性的 ping 按钮或测试连接按钮。这个按钮很容易让人困惑，它到底表示可以执行一个简单的 TCP/IP 连接测试（检测两个计算机是否能够在网络上互相看到）呢，还是表示执行了一个 DICOM C-Echo（校验各个 AE 能够与其他 AE 进行 DICOM 通讯）呢？后者隐含了已经验证前者，反之则不行；因为即使 TCP/IP 验证通过，也有可能出现 DICOM 连接性验证失败。千万不要低估了 DICOM 正确连接的验证（C-Echo）功能。一定要向你的厂商确认你的系统界面上有切实可用的 DICOM C-Echo 功能。

7.9

牛人的工具箱

C-Echo、C-Find、C-Store 以及 C-Get/C-Move 协议在所有 DICOM 实施过程中会特别常用。事实上，大多数 DICOM 设备和软件实际上实现的就是这些 SOP，除此之外再没有其他什么了。这很合理，如果你可以校验 DICOM 连接性，并且可以在 DICOM 实体间查找和传输数据，那么你的系统就已经具备了 DICOM 能力了。

同样，如果你参与了 DICOM 软件开发，那么请记住如果用对恰当的 SOP IOD 和属性，你可以从有限的服务中发掘出许多功能来。在 DICOM 界面中，首当其冲的最受欢迎功能就是在 DICOM 自由文本字段中执行关键词查询。比如，检查描述（DICOM 数据字典中的元素（0008,1030））的查询，这个功能在教学图像归档和临床历史图像归档中经常使用。

我们来更近距离地瞧瞧一个实例吧。比如你想要查找所有检查描述中具有“aneurism（动脉瘤）”或“endoleak（支架周围渗漏）”关键词的最近检查。如果支持的 C-Find 属性列表中包含(0008,1030)，那么你可以设置(0008,1030)的值为“aneurism\endoleak”并在 C-Find-Rq 消息中使用这个值来查找你需要的检查。事实上，考虑到所有潜在的词性变化，你甚至可以使用“*aneurism*\endoleak*”这个值来确保所有“aneurisms”和“wejustdonottypespacesendoleaks”也能够被查到。现在你知道了，如果能够用 C-Find 找到一些东西，那么你就应该能够使用 C-Get 或 C-Move 来提取它。简而言之，有了这些 DICOM 协议和充足的属性，即使有等级层次查询/提取的限制，但是你仍然可以非常灵活地进行 DICOM 数据管理。

7.10

匹配的应用角色

在我们下结论之前，让我们再看看 DICOM 通讯中的 SCU-SCP 概念，其中 DICOM 通讯我们在 7.2 中已经介绍过了。有计算机背景的朋友们愿意将 SCU 和 SCP 与众所周知的客户端-服务器 (client-server) 模式联系在一起，这个模式通常用在计算机网络和应用程序方面：SCP 作为服务器；SCU 作为客户端。这可不完全正确。

SCP-SCU 只不过是角色而已。只要当时条件允许，这两个角色允许任何兼容的 AE 来使用；角色就是特定的应用程序在特定的时期为其他特定的应用程序而产生作用。比如（图 51），我们 CT 机的数字归档（典型的服务器）SCP，在发送图像给另一个 CT 图像存储 SCP（比如工作站）时，实际上起到的是 CT 图像存储 SCU（典型的客户端）的作用。在几乎所有 DICOM SOP 中，SCU 发起通讯而 SCP 答复它；因此，我们不能简单地认为他们就是我们常见的 C/S 模式。

再举个例子，PACS 归档应用程序将 DICOM 视为它自己内部的语言。为了在它自己的数据库中找到一幅图像，它将发出一个 C-Find-Rq 给自己并且将会使用 C-Find-Rsp 来响应这个请求，因此它既是 C-Find SCU，同时又是 C-Find SCP。

另一个常见的想法是人们认为某中程度上 SCP 比 SCU 更高级，而 SCU 只不过是 SCP 的子功能。并不会总是这样的。SCP 和 SCU 甚至常常在功能复杂性方面成为对手，而且做好一个并不能说明能做好另一个；再者，它们实实在在地是两个完全独立的角色，任何 DICOM 设备都可以调用它们。因此，对于任何特定的 SOP 类，每个 DICOM 装置都可以是 SCU、SCP、两者都是或都不是。DICOM 装置的 DICOM 一致性声明会在这方面提供准确的说明。

下面进入我们最后要谈论的内容。SOP、SCU、SCP 是任何 DICOM 实践中最为必不可少的要素。只有准确地定义了 SOP、SCU、SCP 才能知道 DICOM 实体是怎么交互的，处理些什么，以及需要通过 DICOM 网络向其他实体提供些什么信息。从最大的 CT 机到最小的 DICOM 文件浏览器，SOP 为这些 DICOM 兼容设备定义了 DICOM 规范。因此，任何由设备厂商提供的设备或应用程序的 DICOM 一致性声明只不过是所支持的 SOP 列表以及他们的 SCU/SCP 角色而已。传统上说，好好读一读一致性声明是很重要的，但许多 PACS 管理员真的忽视了它。事实上，多数人根本不会去读它，而是过度信赖 DICOM 厂商，常常去问：“它能够在我这儿正常运转，对吧？”

真没这个必要，因为：

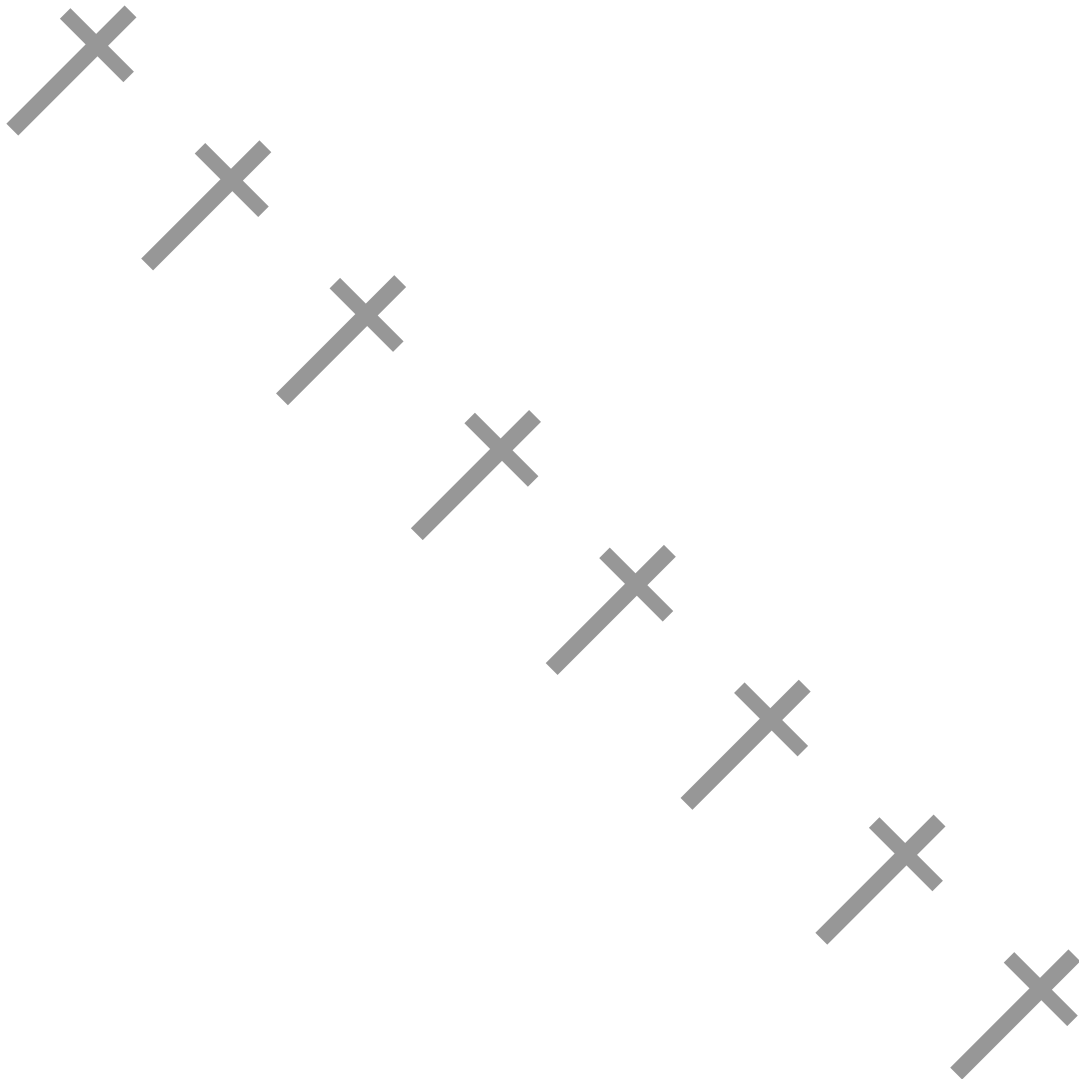
1. 如果你买了一个不能作为 DICOM 校验 SOP SCP 的装置，那么你将永远不能从另一个 DICOM 设备对这个装置进行 DICOM 连接性验证。它将不会对发来的验证消息有任何的回应，就像装置不存在或已经断开网络连接一样。
2. 如果你买了一个能够作为 MR 存储 SOP SCP 的归档，那并不意味着它能够作为 CT 存储 SOP SCP，因此你可能不能将你的 CT 数据传送到这个归档中。
3. 如果你有一个工作站可以作为 C-Find（查询）SCU，但是不支持 C-Find（查询）SCP，那么你将永远不能从另一台工作站或归档来对该工作站进行查询。只有查询 SCP 才会理会查询请求，并为另一个装置提供响应。
4. 如果你买了一个在进行 C-Store 和 C-Move 图像传输时不能进行图像压缩的 DICOM 远程影像诊断服务器，那么你就不能发送压缩图像；甚至不能让你的远程影像诊断业务正常运转。

忽视、误解或错误配置 SOP 将带来大量的 DICOM 和 PACS 问题。一些装置可以通过增加一些软件模块来实现 SCU/SCP 的功能（供应商会向你收费的，通常是几千美金这个级别）。但是一些装置不能够打补丁，只能被更换（当然还是要花钱）。如果你正在参与任何 PACS 的管理和采购工作，那么一定要在 SCU/SCP 的支持成度方面特别留意。

真实的案例：往归档打印？

DICOM 厂商对这些规则的遵守程度完全是另一个问题。有一次，我们花费数小时试图配置一个 CR 装置，希望它能够向数字归档发送它自己的 X-Ray 图像。这个 CR 只有两个选项：发送图像到一个归档和发送图像到一个打印机。在尝试了所有可能的（甚至不可能的）归档配置后，CR 仍无法正常工作。因此我们诉诸于我们的最后一招：给 CR 厂商的场地支持打电话。

那家伙来了以后，看了看我们之前忘我的努力成果，咧嘴笑了笑，然后告诉我们，所有其他厂商的数字归档（包括我们的）必须以打印机的形式添加到这台 CR 上！因此，我们唯一能做的就是不得不杀入归档的说明书中查找 CR “打印机”的段落。的确，设计这个 CR 的家伙脑子实在成问题，但是几分钟后我们的“归档-打印机”收到了来自 CR 的图像。



第 8 章

DICOM SOP：进阶

附录 A.2 中看似无止境的表格提供了所有当前 SOP 类的列表。所有这些 SOP 类均来自 DICOM 标准 2007 版 PS3.6。表格中大部分的 SOP 都属于我们前面讨论过的存储 SOP；它们提供了对不同影像设备类型的支持。还剩下的那些其他的 SOP 值得我们特别注意一下，因为你可能会在你的 DICOM 一致性声明中使用它们。

8.1

存储承诺（Storage Commitment）

存储承诺 SOP 是常规的存储 SOP 类之外的 SOP，它的作用是保证设备接受存储用的图像，并且向别人承诺已经存储了它们，即明确地对数据的妥善保管负责。比如，一个简单的工作站能够执行一个常规的 CT 存储 SOP 类，仅仅就是能够接收 CT 图像并显示出来，用过之后就立即删除这些图像。然而，数字归档则可以支持长期（承诺的）存储并且典型地支持存储承诺 SOP 类。

虽然物品存储的真实承诺严格地说是个管理问题，但是如今越来越多的 DICOM 设备提供存储承诺的功能。在 DICOM 中，存储承诺只不过是另一种数据传输协议罢了。如果某人能够登录到一个存储设备并且删除某些数据，或者更糟，如果一个服务器的自动删除功能可以在磁盘空间不足时清理图像，那么 DICOM 承诺 SOP 根本没有用处。

各种 DICOM 软件包会实现不同的私有承诺策略，通常会允许锁定某些检查（因此他们将无法被删除），诸如标记为不可删除。但是显然这并不能保证数据的安全保管，并且已经是 DICOM 范围之外的内容了。在任何真实的存储承诺实施中，你的最佳方案应该是每日备份你的 DICOM 存储以及限制所有未验证的数据访问（尤其是应将它们放在单独的房间）。在管理层面上，这样可以降低丢失的可能性降到最低。由于这个原因，因此在 AE 的设置中存储承诺 SOP 通常会被忽略，而且即使程序可以支持也很少被启动。

8.2

二次获取（Secondary Capture）

SC 存储 SOP 是在数年前引入 DICOM 标准的，因为一个简单的事实，数字图像数据不仅来自于影像设备，而且来自于其他图像源。比如，数字图像的获取可能是在你扫描（数字化）平片时、扫描报告时、影像设备图像（比如，从原始的断层图像生成三维图像）后处理时、或者即使你只是将工作站上的抓图放在 PACS 数据库中（图 52）。

极具讽刺意味的是，PACS 处理 SC 图像时最常见的问题是识别它们的影像设备。我来举个例子。比如你在 CT 机上扫描了一个病人，希望进行灌注检查并且希望产生灌注彩色图谱，就像图 52 所示。显然，存储这些彩色图谱的最佳方法是将它们归类到产生它们的设备上：将它们追加到原始 CT 检查上。但是灌注彩色图谱图像显然不是 CT 图像，并且它们不能满足 DICOM CT 图像 IOD：它们不是黑白的；它们没有 CT 扫描方法；它们没有亨氏灰阶等。简而言之，如果你将你的灌注图谱的检查设备列为 CT，那么你就麻烦大了；你系统（比如 CT 图像存储）中一些 CT 处理功能迟早会因为使用非 CT 数据而造成故障。

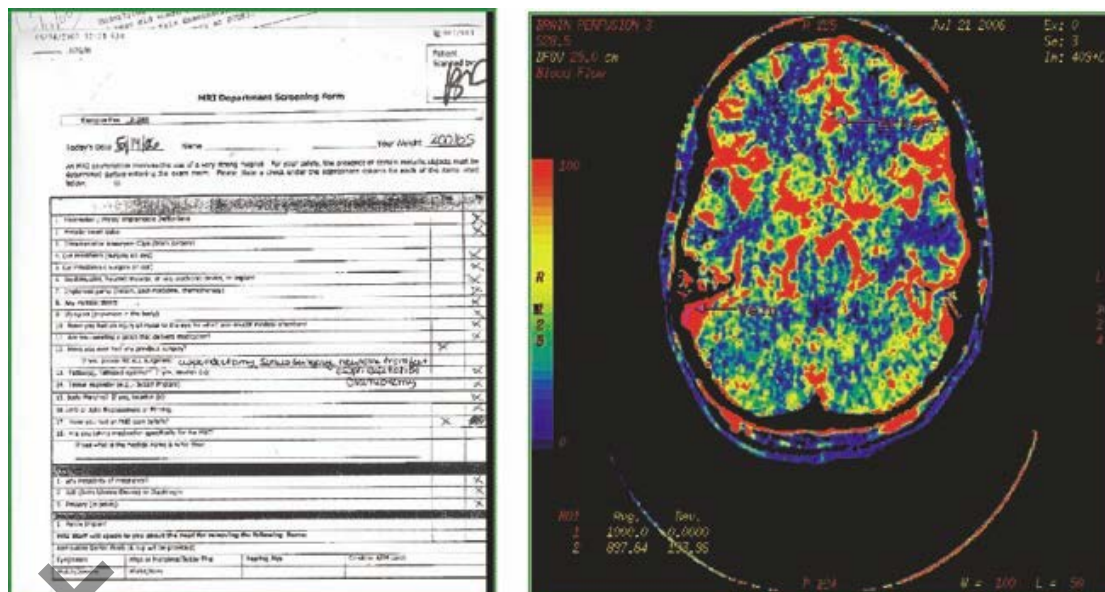


图 52 二次获取的实例：扫描的文档、后处理图像的抓屏

唯一的解决办法就是将二次获取图像的检查设备列为 SC，对于我们来说，这样可能会从一开始就让这类事情变得合情合理了。但不幸的是，此时你可能已经闯进了另一个麻烦：比如你想要向其他工作站或归档传送你的 CT 检查（其中包含一些 SC 图像）的时候。在大多数简单的 DICOM 实现中，你需要调用 CT 存储 SOP。CT 存储 SOP 将负责传输所有 CT 图像，但显然不包括 SC 图谱！因为 CT 检查中的 SC 图像，不是 CT 图像，CT 存储 SOP 将完全忽略其传输。放射医师读取检查时，也会因为在 CT 工作列表中看到 SC 图像而感到迷惑。

真的就没有办法了吗？当然有：保证你的 DICOM 实施支持 SC 存储 SOP，而且更重要的是，保证其在针对任何特定设备传输图像时都必须检查是否具有 SC 图像。我们在第 9 章将会看到，在传输数字检查过程中，DICOM 能够使用多种存储 SOP；如果每次都需要使用 SC，那么你的 SC 图像是绝对不会被遗漏的。实时上，你可以测试一下你即将购置的 DICOM 应用程序在 SC 处理方面是否可靠。你可以取一些图像样本，无论它们的原始检查属于什么影像设备，只要具有 SC 图像就行，你将图像发给新的 DICOM 程序便可知它是否能够正常处理。

8.3

结构化报告（Structured Reports）

很长时间以来，DICOM 只处理数字图像，并且将其按照影像设备类型严格分开。如果放射检查总是从原始数字图像开始，那么复杂的后处理、显示和分析这些后续产品就要分成更加多种多样的数据类型了，而且每一个类型可能都需要一种特定的连接来处理。在早期的 DICOM 中，在处理这些诸如文档的非图像数据时，最流行的方法是将它们转化成图像，就像 SC 中的抓屏一样。目前仍然在使用类似功能的 PACS 厂商数量足可以让你咋舌。但显然，这将破坏原始图像格式和属性。比如，你可以用关键词来查询文字报告，但是你不能用存储成图像格式的扫描报告来进行查询。

SR SOP 的目标是创建复杂的结构文档，在其内可以混合和组织文本、不同的图像以及其他的数据类型。SR 的原始目标是为了处理报告，但是现在它具有了处理任何复杂的混合数据文档的能力。SR 支持基础的编码条目应用和标题分级树，所有的内容都将被编排在条目和子标题下。考虑到 SOP 实例（比如图像、波形、或其他 SR 文档）被严格限制在主体文本树的叶子上（最低的子标题）。这个结构简化了作为 SR 文档的常规文本报告的编码；而且

它也简化了他们展示。

因此，了解 SR 对象的最简单的方法是将它想象成一个包含文本和参考图像的文档，甚至包含一些如录音这种更复杂的数据。文本可以用段落、子段落等结构来进行组织；而图像可以是原始检查图像或重建图像——如图 53 中实例所示。

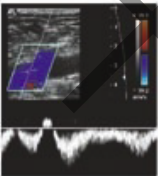
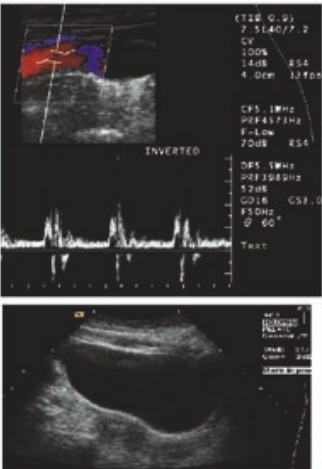
Smith, Joanne, OB Exam 1/20/2008—Fetus A				
Patient Info				
Name:	Smith, Joanne	Exam Date:	1/20/2008	OB
Patient	123456 1/1/1975	Procedure:	Obstetrical Ultrasound	Hx: G: F
ID: DOB:	Houston, Sam, M.D.	Exam Site:	Inpatient	Trm: (2)
Age: Ref.	31 yrs	Plurality:	5	Pre: (1)
Phys:	Graham, David	Fetus:	A	Ab-I: (0)
		LMP:	5/12/2006	AB-: (0)
				S: (0)
				Ect: (0)
				Multi: (0)
				Liv: (1)
Measurements			Computations	
cm	GA	± wks	Selected GA	
BPD:	8.1	33w3d [±1.36]	GA:	33w4d [+2.0wks]
FL:	5.6	34w0d [±1.28]	Method:	LMP
HC:	28.5	34w0d [±1.23]	EDD:	2/16/2006
AC:	27.5	33w3d [±1.31]	Sono GA	
HL:	5.7	33w1d [±1.36]	GA:	33w3d
TCD:	2.8	25w4d [±2.40]	[+1.02wks]	
Doppler			Method:	BPD, HD, AC, FL
UASD: 1.5			Selected FW	
			Gms:	2235
			Lbs:	4 lb 13 oz
			%:	23
			Method:	BPD, HD, AC, FL
Placenta:	Placenta is Anterior and Grade II. There is no evidence of a placenta previa.			
AFI:	18 [65%]			
David Graham, M.D. Thank You For This Referral				
Images				
				
Indications for Sonography				
Abdominal pain-Unknown etiology				
Decreased fetal movement Diabetes				
Evaluate for cervical incompetence				
Multiple gestation Vaginal bleeding				
Observations				
Fluid Volume: Normal				
Fetal Lie: Transverse Back Up				
Fetal Size: Normal for dates				
Fetal Growth: Within normal limits				
FHR: 136				
Sonographer: Jane Smith, RDMS Next				
Exam Scheduled in 2 Days				

图 53 结构化报告的实例

在最极端的形势下，DICOM SR 还可以用来显示书籍。在最简单的情况下，SR 可以包含如超声图像上的一次测量信息（Hussein et al. 2004a）。此外，考虑到 SR 的安全性，可以在其上进行数字加密和数字签章（见 A 部分. 4, PS3.15）。

相对来说在 DICOM 标准中，SR 是比较新的内容，所以较老的 DICOM 装置/软件可能无法处理它们。比如，我们曾见过一个 PACS 工作站试图使用打开数字图像的方法来打开 SR DICOM 对象，最终导致打开失败和处理终止。我们可以期待，随着 SR 的被更广泛使用，PACS 厂商将至少更加重视它们。

另一方面在过去几年间，SR 已经经历过了爆发式的流行。主要原因是他们的多影像设备支持能力。如果说所有 DICOM IOD 都是基于影像设备的，那么可以说 SR 不是，SR 允许你

来混合和匹配不限数量的各类检查、扫描和文档。因此，在许多情况下 SR 已经成为比通用的 DICOM 数据更加复杂的数据，并且某些放射设施已经开始使用基于 SR 的工作流（替代基于传统影像设备的工作流）。考虑到结构化报告的优点和复杂性，它带来的挑战和益处是均等的（Hussein et al. 2004b; Batchelor 2006）。

整个 SR 的概述已经超出了本书的范畴。如果你正在查找关于 SR 的更深层次的内容，那么我向大家推荐一本由 David Clunie 所著的书《DICOM 结构化报告》（DICOM Structured Reporting, Clunie 2000），或查看 DICOM 标准的补充部分 23。

8.4
封装的 PDF（Encapsulated PDF）

就像有存储二次获取图像的情况一样，我们很快就会发现 DICOM 需要存储不只是来自影像设备的图像数据。在 1993 年，Adobe 公司发布了便携文档格式（PDF），是一个应用广泛和功能强大的文档交换工具，并与各类应用程序进行了很好的集成。

许多医疗实践都习惯使用 PDF 格式，因为对于 DICOM 来说，几乎所有东西都可以放在 PDF 格式中。许多文档（比如检查报告）扫描到 PDF 中比放在普通图像中更好，因为 PDF 对页面顺序和扫描图像的压缩方面可以提供更好的支持。录入（非扫描的）的报告通常也会存放在 PDF 中。相比普通的问恩，这些文本很难被修改，它们可能是经过电子签章的并且具有其他安全保护（PDF 支持丰富的安全特性），它们可能有复杂的格式，并且可能包括图像就相当于 SR 的作用。由于这些 PDF 包含文本内容，因此可以使用关键词来进行查询，这就相当于建立了一个可查询的报告数据库。简而言之，如果你在文档处理工作流程中还没有享受过 PDF 的好处，那么我建议你去调研一下这个工具所能提供给你的丰富功能。不管你是否采用 DICOM，PDF 可以对你的文档管理甚至图像管理等工作提供坚实的基础。

由于这些原因，DICOM SOP 1.2.840.10008.5.1.4.1.1.104.1，封装的 PDF，允许你将 PDF 文件作为 DICOM 对象进行存储。因此，一个 PDF 文档会补充一些 DICOM 特定的信息模型，比如病人数据，来提供所有必须的 DICOM 属性（表 43）。

就像 SR 一样，PDF SOP 还没偶在 DICOM/PACS 系统中广泛一个用。如 PDF 这种复杂的数据格式需要各种复杂的软件工具来进行显示，并且需要能够熟练操作它们。本质上，你需要将一全套 Adobe Acrobat 专业版软件与你的 DICOM 应用程序集成。从 DICOM 供应商的角度来看，继续使用只支持影像功能的 DICOM 系统会更简单一些；把所有无影像设备类型的数据存到普通的二次获取图像中就可以了。而且，复杂的文档意味着会用在复杂的任务上，大多数放射科都愿意将其 DICOM 和文档 workflows 尽可能的保持简单。PDF 的需求自然只有在涉及多医院和多诊所的文档分发时才会体现它的优势；所以在 DICOM 和 PACS 中存储 PDF 的能力早晚会派上用场的。

表 43 封装的 PDF IOD 模块

IE	模块	用途
病人	病人	M
	样本标识	U
	临床试验题目	U
检查	通用检查	M
	病人检查	U
	临床试验检查	U
序列	封装的文档序列	M
	临床试验序列	U

设备	通用设备	M
	SC 设备	M
封装的文档	封装的 PDF 文档	M
	普通 SOP	M

然而，一些商用 PDF 插件已经具有在 Adobe Acrobat 上打开 DICOM 文件甚至直接向 PACS 发送或从 PACS 接收他们的能力。封装的 PDF 格式设置了一个有趣的范例，用来描述在不需要对 PDF 和 DICOM 任何一方进行调整的情况下，所谓“外来的”文档格式（比如 PDF）是如何嵌入 DICOM 对象的。显然，封装的主意也会用于其他格式，并且可能成为 DICOM 设计者研究和探索的新领域。

8.5 硬拷贝打印

大量的打印（Print）、硬拷贝（Hardcopy）和图像盒 SOP（Image Box SOP），弃用的和在用的，都是为将 DICOM 图像打印到胶片上而准备的。在许多方面，这些 SOP 意味着让你向后兼容那些 PACS 出现之前的世界（大约是 1980），那时胶片时唯一的医疗图像存储介质。尽管在当前的 PACS 中，胶片仍然有其恰当的位置：当没有其他可用时，它会担当可靠备份的角色。如果你的 PACS 网络挂掉了；如果你需要发送一个检查给一些尚未连通的边远地区；如果你信守“我就是讨厌数字化”的誓言，那么胶片仍然会陪伴你左右。在你的归档和工作站上保留胶片打印 SOP，对于上述理由和一些其他的情况来说还是非常有用的。

然而，在日常的放射工作中，胶片的使用正在日益减少；这就是 PACS 革命的唯一目标。软拷贝 SOP（Softcopy SOP）一直在向传统的硬拷贝领域渗透。因此如果你想要让你的数字化工作流更加灵活，那么你最好确认一下你买的 PACS/DICOM 产品（包括简单的 DICOM 浏览器）支持 DICOM CD/DVD 的刻录；支持在外部存储（硬盘，闪盘）中保存 DICOM 文件；以及支持各类标准的多媒体图像格式，比如 TIFF、BMP、AVI。这种在标准 DICOM 网络之外导出和交换 DICOM 数据的做法显然已经非常主流，很适合当代的数字工作流。更常见的做法是用它代替了胶片和古老的磁光碟（MOD，许多 DICOM 设备制造商仍然支持这种介质，只要那几年间设备厂商采用过这种介质。）与奇异的 MOD 驱动器不同，CD/DVD 刻录功能通常都会具备。在规模较小的机构，DVD 通常作为长期备份的解决方案，因为它的确比较便宜而且用起来很简单。

最终，在 DICOM 兼容性方面数字媒体比胶片（见第 10 章）强得多。CD、DVD 和闪盘可以提供更加全面的存储方案以及更加多样的安全选项。没什么可惊讶的，看看为什么许多医院和检查组织现在都要求使用 CD/DVD 来代替胶片用作图像交换（作者在美国，与天国国情略有不同），你即可了解原委了。