Radiology Informatics in Residency: Developing a Resource for Training in Radiology Informatics

Nathan M. Cross MD, MS, CIIP
University of Washington, Department of Radiology
https://www.nathancross.net/
nathan.cross@gmail.com
No Financial Disclosures
Introduction

Imaging informatics is a critical component of residency training in radiology, attested by the ABR's Core Exam Guide. While not a typical part of medical school or residency education, imaging informatics has revolutionized the field of radiology and is crucial to our specialty. The ABR identifies several key components of modern radiology informatics including PACS, RIS (Radiology Information System), and HIS (Hospital Information Systems) as well as the standards that support these technologies. There are curricula for imaging informatics, such as the imaging informatics professional (IIP) offered by the Society for Imaging Informatics and Medicine, but the material is not tailored for the radiology resident or their examination. The creation of a concise web-based overview of radiology informatics systems and the supporting standards furnishes residents a glimpse into the technology they depend on daily, provides a structure for ongoing learning, and is a component of diagnostic radiology competency.
The Site

Each of these five topics was reviewed at the level of a second-year radiology resident on a publicly accessible web server with accompanying diagrams. To provide a conceptual framework, basic principles of computing are explained in simplified form and the role of technology standards is emphasized. The current medical informatics ecosystem is illustrated in addition to the tools crucial to integrating radiology systems.
Digitization and Data

We are about to discuss digital data: how it is stored, moved, and manipulated. It is important to understand the fundamentals of digital data and how computers use long strings of 1s and 0s to perform complex tasks.

Binary data "bits" are the fundamental quanta of digital data and have two states ‘1’ and ‘0’. A collection of bits (101101110110...) would be unintelligible unless the computer or reader had a standard way to organize the bits. If this didn’t exist, each programmer or device manufacturer would come up with their own way of organizing their bits and devices or software couldn’t interoperate.

<table>
<thead>
<tr>
<th>BYTE 1</th>
<th>BYTE 2</th>
<th>BYTE 3</th>
<th>BYTE 4</th>
<th>BYTE 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIN</td>
<td>0100</td>
<td>0100</td>
<td>0100</td>
<td>0100</td>
</tr>
<tr>
<td>HEX</td>
<td>044</td>
<td>049</td>
<td>043</td>
<td>04F</td>
</tr>
<tr>
<td>ASCII</td>
<td>D</td>
<td>I</td>
<td>C</td>
<td>O</td>
</tr>
</tbody>
</table>

Bits are infrequently used alone since they only represent 2 states (1 or 0). More frequently they are grouped into groups of 8 which is referred to as a “byte”. A clever computer scientist determined that if a byte was 8 bits, then 4 bits should be a nibble! Arranging bits in groups of 8 is convenient for computers because it is a power of 2 which computers are able to work with easily.

To count in binary is not dissimilar from counting in decimal: 00000000, 00000001, 00000010, 00000011, 00000100, 00000101, 00001111. This sequence spans the values from 0 to 255 in decimal notation, and represents the 256 individual values that can be stored in a byte. Bits and bytes are referred to throughout your life. You may have an 8bit display at home and a 16bit PACS monitor at work, referring to the Number of bits used to store each value of red, green, blue. The more bits that are used to store each value, the greater the number of individual values. And the greater Number of shades of color.

Humans frequently find groups of 8 1’s and 0’s to be a bit of a headache and thus sometimes interpret them in base10 which is more familiar to us, or in base16 which is more compact.

To convert 8 bits to a base10 value the first bit represents 2^0, the second bit 2^1, the third 2^2, etc. (Table <$table:Table(bit_convert)>). Thus if we have the nibble 0b1101 where ‘0b’ is simply a prefix to indicate this is a binary number the base10 value would be 8 + 4 + 0 + 1 = 13. If we extrapolate this to a larger collection of bits such as a byte we would see that the largest number a byte can hold is 255. More frequently groups of bytes are represented in hex (standing for hexadecimal). This is a base16 number scheme where each digit ranges from 0,1,2,3...A,B,C,D,E,F. If you look at the examples you will realize that the maximum value of a nibble is 15, and so each nibble (0.1.2.3.10.11.12.13.14.15) can be easily represented by a single hex digit 0...F. Thus hex becomes a very efficient way to represent a large number of nibbles which are often grouped as a pair since a great deal of computing is done in bytes or multiples of bytes.
References

• Oosterwijk, H. (2014). PACS Fundamentals (2nd ed.).