Abstract: The modern anatomic pathology laboratory depends on a reliable information infrastructure to register specimens, record gross and microscopic findings, regulate laboratory workflow, formulate and sign out report(s), disseminate them to the intended recipients across the whole health system, and support quality assurance measures. This infrastructure is provided by Anatomic Pathology Laboratory Information Systems (APLIS), which have evolved over decades and now are beginning to support evolving technologies like asset tracking and digital imaging. As digital pathology transitions from “the way of the future” to “the way of the present,” the APLIS continues to be one of the key effective enablers of the scope and practice of pathology. In this review, we discuss the evolution, necessary components, architecture and functionality of the APLIS that are crucial to today’s practicing pathologist and address the demands of emerging trends on the future APLIS.

Key Words: laboratory information systems, anatomic pathology, LIS, APLIS, informatics, pathology informatics, digital imaging, whole slide imaging, WSI, cytology

(Adv Anat Pathol 2012;19:81–96)

Pathology, although not unique among the medical specialties in its need to manage and manipulate information, is unique in the sheer scale of data generated and handled. Over 70% of the content of a typical electronic medical record (EMR) is generated by our specialty, ranging in data type from the very simple (eg, simple structured numerical values generated by blood chemistry tests) to the very complex (eg, free-text diagnostic reports for surgical specimens, sometimes several pages long with embedded digital images). Given this, it comes as no surprise that (a) information overload is an especially persistent and difficult problem in Pathology and (b) attempts to coherently organize, present, and use said information on a real-time basis have given rise to a class of interrelated database-driven applications known as laboratory information systems (LIS).

Just as Pathology is broadly divided into Anatomic Pathology (AP) and Clinical Pathology (CP), LIS stereotypically come in Anatomic Pathology Laboratory Information System (APLIS) and/or Clinical Pathology Laboratory Information System (CPLIS) flavors. The division between AP and CP in the real world is not always clear-cut; for instance, Hematopathology, although under the purview of CP, is perhaps closer in spirit and practice to Surgical Pathology. In a similar manner, the distinction between APLIS and CPLIS is not always useful, as a single general purpose LIS may accommodate both AP and CP functionality, whereas a best-of-breed LIS may focus entirely on subspecialty-specific functionality instead.

In this day and age, even the simplest LIS is a vast conglomerate of software packages running on a combination of specialized and commodity hardware. In general, the LIS tends to be customizable; this has traditionally been their greatest strength and also their greatest weakness. Indeed, the interactions between an LIS’s capabilities and the unique needs of the end-user institution are often so complex that a discussion of the specific implementation of any given LIS is beyond the scope of this article. This review will focus on (a) the history of the LIS; (b) the fundamental components, architecture, and functionality of an APLIS; (c) subspecialty issues related to LIS functionality; (d) the role of the APLIS in digital pathology; (e) informatics challenges and obstacles for the future LIS.

A BRIEF HISTORY OF THE LIS

The idea of using computational technology to better manage laboratory data is almost as old as the history of general-purpose computers itself. In his seminal 1945 essay “As We May Think”, Bush briefly discussed the possible usage of an imaginary device called a “memex” (short for “memory extender”), easily recognizable as a forerunner to the modern computer, in the management of medical data. This thought experiment is significant not only because it is the first mention in the literature of the concept of an LIS, but also because it directly inspired the work of people like Douglas Engelbart, who would later go on to invent technologies that are not only fundamental to LIS and EMR, but without which computing as we know it today simply would not exist.

The increasing availability of computational technology throughout the 1950s and the 1960s gave rise to the first rudimentary LIS. One such LIS, built in the early 1960s as a collaboration between a company named Bolt Beranek Newman and the Massachusetts General Hospital, included time-sharing and multiuser techniques that would later be essential to the implementation of the modern LIS. Although the technology was genuinely impressive for its time, it also suffered from problems serious enough to preclude its use in a production environment. In the same timeframe, General Electric announced plans for a commercial hospital information system (HIS) of their own through a wholly owned subsidiary called MediNet; unfortunately, soon thereafter General Electric decided to abandon all of its computer initiatives, and MediNet was liquidated.
The first era of the LIS was thus dominated by abortive monolithic HISs that were everything to everyone, created by large technology corporations that had little to no knowledge on the workings of a hospital, let alone a health care system. There are a myriad of reasons why there were virtually no LIS success stories during this era, but 2 things stand out above all: (a) the lack of a proper programming and computing technology and (b) the lack of communication between the providers and the end users. Without a programming and computing environment that was both powerful enough for multiple simultaneous users and easy enough to promote rapid iteration of program design and implementation, a single program might take months to write, let alone debug. In contrast, without cooperation between the engineering teams that were creating these systems and the clinical teams that were trying to use them, the result was a product for which the end users had little use and even less buy-in.

The first problem would be explicitly tackled head-on by Pappalardo and Marble, who, in the mid-1960s, developed an advanced programming language known as Massachusetts General Hospital Utility Multi-Programming System (MUMPS; otherwise known as M). This language not only introduced programming concepts—such as interfaces for multiple simultaneous users and facilities for easier porting of database-driven programs from one instruction set architecture to another—that were startlingly ahead of their time, it also integrated a hierarchical system for persistent storage of data. In other words, MUMPS was one of the first (and certainly one of the most successful) hierarchical database management systems (DBMS) in computing history. However, the development team behind MUMPS was small, restricted to the use of relatively cheap commodity hardware, and required to work largely in very close collaboration with only clinical staff. As a result, development shifted from the hitherto- unsuccessful monolithic approach to a far more modular approach with smaller, production-focused design goals, and far more rapid iteration.5

The second generation of LIS in the late 1960s and the early 1970s was marked by rapid growth in technology and deployment. While these were almost universally implemented atop MUMPS, their implementation details were so different that interoperability was thought to be impossible. Part of this has to do with the nature of MUMPS itself: although it was both extremely advanced and extremely efficient with its computational resources, it had a number of serious flaws. First, as efficient as it was it still put a large strain on the computers of the time; for instance, the MUMPS interpreter alone took up half the available RAM on its initial target platform, the DEC PDP-7. Second, although certainly easier to use than assembly language, it was still difficult to learn and master. Thirdly, because it was the work of a small team that made its source code easy to obtain, many of the companies that used it also customized its fundamental properties, largely by adding proprietary commands; this led to a fragmentation of MUMPS, with individual institutions and companies supporting their own unique variants. Finally, there was no easy way for an end user to extract or analyze data from the MUMPS database without being a MUMPS programmer him/herself.4,6

It was not until the 1970s and 1980s that solutions to these problems would begin to emerge. With the advent of the relational database model and commercial DBMS that used this model, came a standardized syntax for data manipulation known as Structured Query Language (SQL). Technology companies like Intel and IBM embarked on relentless improvements in their semiconductor fabrication technology, resulting in the rough doubling of available computing power every 1 to 2 years, now known as Moore law. Standardized and highly portable programming languages designed for ease of use and power both—like Pascal and C/C++—emerged and were embraced by industry. Intel's x86 instruction set architecture was born and became a force in the consumer space during this timeframe.7

This era ushered in the third generation of LIS. Exponential increases in computing power meant that computationally expensive but user-friendly relational database management systems (RDBMS) could be used instead of MUMPS. Moreover, interchange standards such as Health Level 7 (HL7) were born and adopted during this time, although true “plug-and-play” interoperability remained (and still remains) elusive. At this stage, the LIS had become fundamental to clinical laboratory practice, and governmental regulations (such as CLIA ’88) began to influence both the capabilities and the security measures included in the commercial LIS. SQL provided a standardized manner of manipulating clinical data, meaning that for the first time business analytics and business intelligence techniques could be applied in a coherent manner.8

In the present era, the modern LIS has been able to leverage information technology (IT) and data networking—largely thanks to the popularity of the World Wide Web—which are ubiquitous and dominant. Powerful web-based and database-centric Rich Internet Applications (RIA) has changed the way we interact with our computers, and web-driven data formatting technologies like eXtensible Markup Language (XML) have fundamentally changed our approach to LIS and EMR interoperability.9 The drive for interoperability in between LIS is stronger than ever, and new technologies like whole slide imaging (WSI) are beginning to change the way that AP is practiced.10 Now more than ever, patients are demanding the kind of just-in-time access to their clinical records that were hitherto only given to medical staff. The amount of data generated by the LIS has increased dramatically, and the rate of increase will only accelerate as time goes on; this mirrors the exponential rate of data usage increase that we have seen for the average user of the Web. This, combined with a slowing of Moore law, has led to a reevaluation in some industries (but not yet in the EMR or LIS industries) of the continued utility of RDBMS, with several prominent companies pioneering so-called “NoSQL” databases that abandon the linked-table based approach of the relational database model in favor of more computationally efficient but less standardized approaches.11 Concurrently, a promising new paradigm known as “cloud computing”—in which the LIS and its associated data exist on a cluster of remote (Internet-connected) servers administered by a third-party—has emerged. At the same time, new security challenges have appeared; identity theft is a clear and present danger, and computer system intrusion has reached an all-time high. We have yet to see how the LIS will adapt to these new challenges in our increasingly networked world.

**COMPONENTS OF AN APLIS**

An APLIS can be described using a stack metaphor, with hardware at the bottom of the stack and the LIS application software at the top of the stack (Table 1). The higher a layer’s position on a stack, the more abstracted it is
from the layers below. Take for example a user who prints out an AP final report: the APLIS application pulls the relevant data from the DBMS, assembles a printable document, and signals the operating system (OS) to print said document. The DBMS pulls the relevant data through SQL commands that are hidden from the end user by the APLIS application, but it does not concern itself with how to specifically send the electrical signals to read data from a hard drive, or write data to it. The OS has low-level code that, through special software packages known as “drivers,” can receive input from (in the case of the hard drive) or write output to (in the case of the printer) hardware devices, but neither the APLIS application nor the DBMS need to know the specifics of how those drivers work. Finally, the hardware interfaces with the drivers (and through them the OS) through “firmware,” which is something akin to “software written in hardware”—that is to say, a hard-wired control package that converts the low-level software commands of the OS to electrical signals that directly cause the hardware to function in a specific manner.12

### Hardware

Generally speaking, the “hardware” of an APLIS consists of every physical element that interfaces electronically with the APLIS application in one way or another. This includes, as a bare minimum:

- The computer(s) on which the APLIS application resides (server)
- The computer(s) on which the APLIS database resides (server)
- Computer(s) for laboratory staff and pathologist use (client)
- The basic input/output devices of the computers
  - Keyboards, mice, monitors, etc.
  - Document scanners
  - Digital cameras
- Printers (paper, labels, tissue cassettes)
- Network hardware
  - Cables
  - Routers

Depending on the sophistication of the APLIS involved, other pieces of equipment might also be interfaced, including:

- Barcode scanners
- Gross pathology examination stations
- H&E autostainers
- Whole slide scanners

Figure 1 presents a schematic diagram of a simple APLIS setup. In this client-server architecture setup, a central APLIS server is networked to several peripheral computers that may be located for example in the grossing area, histology laboratory, in the pathologist’s office, or to computers attached to devices/instruments. For example, the gross pathology computer integrates a barcode scanner and specimen label printer, both useful in accessioning specimens. In contrast, the histology computer is attached to a barcode scanner (for scanning in tissue blocks), a slide printer (for creating barcoded glass slides), and perhaps a WSI scanner (for digitizing glass slides). The pathologist’s computer (workstation) may be attached to other devices such as a microscope camera (for capturing static images to be put in a final report). All of these individual computers feed different kinds of data into the APLIS server, and use different data that the APLIS server provides. For instance, the histology computer may query the APLIS server for a specimen’s accession information (which would previously have been entered into the APLIS at the gross pathology computer), and would feed the WSI data into the APLIS server (which would later be queried by the pathologist’s computer). The LIS itself is also typically networked to other external information systems such as the HIS, where inbound patient registration data are received through an Admission-Discharge-Transfer (ADT) interface, or to downstream systems where outbound data from the LIS gets transmitted such as patient pathology reports to an EMR or billing codes to a billing and accounts receivable system.

In the last decade, the maintenance, upgrade, and configuration of the traditional APLIS computer has moved away from laboratories to centralized institutional information services/IT departments. Several large health care organizations now host their information in large server farms that enable storage and backing up of information at a more reasonable cost with offsite backup.

---

**TABLE 1. APLIS Components Presented As An Integrated Stacked Information System**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>APLIS application</td>
<td>The software interface to the end user. Usually programmed for a specific operating system and almost always programmed for a specific DBMS. Has user interfaces for data entry and manipulation.</td>
<td>Cerner CoPath Cerner PathNet Orchard Harvest LIS SSC SoftPath Microsoft SQL Server mySQL PostgreSQL Oracle Database MUMPS</td>
</tr>
<tr>
<td>DBMS</td>
<td>A specialized software package for the persistent storage and manipulation of data. Currently the vast majority of these use the relational model, and implement a SQL interface. In the LIS high performance is not as important as high reliability, requiring certain tradeoffs to be made.</td>
<td>Microsoft Windows Mac OS X Linux</td>
</tr>
<tr>
<td>OS</td>
<td>The fundamental control program through which the end user interacts with a computer. There are different operating systems that are suitable for different niches; it is far more common for Linux to be the OS of choice for a server than for a personal computer, for instance.</td>
<td>Server computer Client computer Barcode scanner Label printer Slide printer H&amp;E autostainer</td>
</tr>
</tbody>
</table>

The layers on the top are closest to the end user, and the layers on the bottom are closest to the raw hardware. APLIS indicates Anatomic Pathology Laboratory Information System; DBMS, Database Management System; LIS, Laboratory Information System; OS, Operating System; SQL, Structured Query Language.
This collectivization of hospital resources has resulted in a need for advance planning, prioritization, and justification of allocation by pathology departments when compared with other clinical departments. With the introduction of digital pathology—most notably WSI—the storage and network bandwidth needs of AP are set to dwarf the needs of any other medical discipline.\textsuperscript{13}

It should be noted at this point that while the kind of hardware found in servers is generally more powerful than what is found in same-generation client computers, that gap has become extremely narrow. Presently, a high-end personal computer will use components that are virtually indistinguishable from that found in a low-end or midrange server, and indeed the overall progress of technology has been such that the relatively humbly provisioned smartphone of 2011 is more powerful than the most powerful supercomputer in the world circa 1960. This has allowed for an architectural paradigm shift: while in the past an LIS would run in its entirety on a single extremely powerful mainframe that represented all the computational power and storage capacity that the system had, the present-day LIS has become much more distributed by design (with clusters of powerful servers providing an infrastructure for many only somewhat-less-powerful clients).\textsuperscript{14}

With this increasing computational power has come increasingly fine-grained control of processes. Barcoding...
and radiofrequency identification (RFID), for instance, are both technologies that (a) are primarily concerned with the identification of unique physical objects and (b) were only made possible by modern computers. They have uses in automatic identification and data capture applications like the automation of grocery store checkout and tracking of mail packages, and have become so successful in these applications that at least 1 barcode can be seen on the packaging of the vast majority of items sold in the world. Technically speaking, it should be possible to use these technologies to enable finely granular tracking of specimens and slides across the AP workflow; indeed, most laboratories have at least limited forms of such tracking in place. However, a fully tracking-enabled APLIS necessitates workflow changes and other (hardware, software, and social) challenges that have thus far proven difficult to crack.15

Operating Systems and Related Software

OSs are the major point of human-computer interaction, and exist 1 layer above the hardware of the computers. They define the end user experience to such an extent that a computer without an OS is considered useless. Some OS’s are text-based; others sport a graphical user interface. There are hundreds of different OS’s that are tailored for hundreds of different roles, but as a general rule all OS’s can be divided into 2 categories: those built primarily for “frontend” use—that is, interaction with a human being—and those built primarily for “backend” purposes—databasing, web serving, storage, networking, and other largely automated processes that only rarely require human intervention. Most modern OSs can fall under both categories at the same time, but are optimized toward one end or the other. For instance, Mac OS X is a frontend OS, yet has a variant (Mac OS X Server) that is more tuned for backend use. Similarly, Linux is primarily a backend OS, yet can be used as a frontend OS with the addition of graphical user interface packages.

In the realm of hospital IT—and therefore APLIS—Microsoft Windows is overwhelmingly dominant in frontend space. The laboratory staff and the pathologists are most likely to interact with this OS, and as a direct consequence APLIS frontend applications are usually written for this OS. With the increasing popularity of the World Wide Web, however, web browsers have emerged as capable application platforms in and of themselves, leading to a class of so-called “Rich Internet Applications” (RIA) that deliver advanced functionality—much of which had previously been thought to be impossible to achieve over a network—straight from the web browser.

Although no current installable APLIS delivers its functionality as a full-fledged RIA, APLIS vendors are increasingly using web technologies, making it likely that APLIS of the future will become increasingly OS-agnostic on the frontend, relying instead on the web browser as the presentation layer. There are also vendors who operate under the so-called application service provider model, in which a laboratory can rent an entire web-based LIS without having to deal hardware at all. For a detailed discussion of the promises and the challenges of these approaches, please refer to section Database Management Systems.

One must also keep in mind that every peripheral device attached to a computer requires a specialized piece of software known as a “driver” to function properly. Drivers provide an interface between the OS and a peripheral device, converting high-level OS tasks into low-level commands that the firmware (the basic input-output system of the peripheral device) can interpret, and then turn into electrical signals that perform the actual requested action. Drivers are extremely OS-specific—not only must they be written for individual OS’s, but most often they are not compatible between different versions of the same OS. A device driver that happens to be written for Windows 3.1, for example, will not work in Windows 7 (or vice versa). Although the current dominance of Microsoft Windows XP in hospital IT makes this a relative non-issue at present, Windows XP will no longer be supported by Microsoft—ever should a critical flaw be found—as of April 8, 2014.16 After this date, it will not be feasible for any hospital to use Windows XP without opening itself up to unacceptable security risks.

Finally, consider the case of a laboratory with an older generation (legacy) APLIS that, though robust for its time, is now unable to perform all the tasks required in the modern day and age. Although replacing the LIS is certainly an option, it is also possible to extend the functionality of an existing LIS through pieces of third-party software collectively known as “middleware.” These fall into 5 broad categories:

1. Lab operation improvements—such as third-party software components that support LIS operations (eg, Microsoft Word and Excel, Crystal Reports), data transmission (eg, Forward Advantage (which manages faxes) and LabDE (which automates capture and transfer of data in lab reports)), storage solutions (eg, HP StorageWorks EVA8000), support for virtual applications, legacy apps, and web-based cloud management (eg, VMWare, Citrix).
2. Workflow improvements—such as instrument middleware, tracking solutions, remote system monitoring (real-time, web-based dashboards), digital image management (eg, Apollo PathPACS).
3. Quality improvement—such as quality assurance (QA) programs written for platforms like Alitosoft Insight or IBM Cognos. This group of middleware serves to mitigate the fact that the LIS has traditionally been weak at collecting, manipulating, and displaying data.
4. Service improvements—such as patient and client services that give customers easy access and connectivity to the LIS (through web portals, for instance). Outreach connectivity tools from this class of middleware include products like Lifepoint, Atlas, Initiate Exchange Platform, 4Medica, Halfpenny, CareEvolve, Blue Iris eLaborate, and JResult-Net.
5. Revenue improvements—such as interfaced billing management tools.

Database Management Systems

In the simplest sense, a “database” is a persistent collection of data in digital form, organized to model information of interest (such as patient data). Given this minimal definition, a text file with the names of one’s friends qualifies as a database, as does a collection of all the clinical tests performed by a hospital’s central laboratory in a week. The term “database management system” refers to the software that is used to manage the database and its data structures. Therefore, while “database” and “DBMS” are often used interchangeably, they are actually 2 very different things: the former refers to the raw information, whereas the latter refers to the program that manipulates...
the information. As an illustration, if a spreadsheet holding experimental data values can be called a database, then the program used to edit that spreadsheet (ie, Microsoft Excel) would be the DBMS. DBMS are so central to LIS that there are some who argue that an LIS is nothing more than a fancy wrapper around a DBMS. Although this is not an entirely fair assessment of the situation, it does rather forcefully highlight (a) the fundamental role that the DBMS plays in the LIS and (b) the fact that DBMS technology had to be invented before the LIS could even be contemplated, let alone implemented.

Every DBMS is defined by its “model.” A model specifies what the DBMS can and cannot do, and how the DBMS can go about doing what it does. There are many models, some of which are obsolete in general-purpose DBMS, but all of which today live on in one widely used form or another. The 4 most common database models the pathologist is likely to encounter are:

- **The flat model**—single 2-dimensional tables stored in individual files
  - Most common, but become unwieldy with large amounts of data
  - Examples:
    - Microsoft Excel spreadsheet
    - Tab-separated (TSV) or comma-separated values (CSV)

- **The hierarchical model**—data as a “tree” of interconnected “nodes,” in which “parent” nodes can have multiple “children,” but each child node has only 1 parent
  - Was dominant among first-generation and second-generation LISs, but was largely made obsolete by the relational model
  - Currently experiencing a resurgence in popularity thanks to web-based technologies like XML and HTML
  - Examples:
    - MUMPS
    - XML

- **The relational model**—2-dimensional tables linked to each other by way of special “key” values
  - Dominant among current-generation LISs
  - Has a standardized syntax known as SQL that allows for some measure of interoperability and interchange
  - A high-level approach that sacrifices computational performance for usability
  - Examples:
    - Microsoft SQL Server
    - Oracle Database

- **The dimensional model**—a specialized form of the relational model that uses 3-dimensional instead of 2-dimensional tables
  - The model of choice for data warehousing and data mining
  - Allows for high-level analysis and decision support frameworks
  - Built atop relational databases
  - Examples:
    - Altosoft Insight
    - IBM Cognos

Although it is beyond the scope of this article to provide a comprehensive review of DBMS, we should note that the relational model is currently dominant, with the hierarchical model being (until recently) of primarily historical interest. As noted in section A Brief History of the LIS, MUMPS was the programming language (and thus the DBMS) of choice for the LIS until the relational model came into being in the 1980s. The universality of SQL makes the relational model (and its adaptation, the dimensional model) uniquely suited to today’s growing needs for (a) true interoperability between LIS and (b) data warehousing and data mining for QA and business analytics purposes. In contrast, the recent popularity of XML and XML-based technologies has led to a resurgence of interest in the hierarchical model. XML has been adopted by health information interoperability standards organizations like HL7, and it seems reasonable to assume that the usage of XML will only increase in the years to come. That being said, at present the vast majority of current APLISs use the relational model, and will continue to do so for at least the near future.

APLIS Application

If the DBMS is considered the “heart” of the LIS, then the application layer probably represents the “face.” This is the layer that the end user (eg, pathologist, technologist) directly interacts with, and whose experience is largely impacted by the user interface design. As LISs have grown in complexity and functionality, application layers have accordingly grown increasingly complex and cumbersome; this has been exacerbated by the fact that different end users of the LIS often use entirely different subsets of an LIS’s functionality. Although most LIS vendors have provided a partial remedy by way of allowing some measure of customization in the application layer’s human-computer interface, the problem remains a thorny one. At the very least, most modern APLISs have been forced to present different user interfaces for the purposes of specimen accessioning, histology (including stain/recut order entry), transcription, billing, and signout.

The APLIS application layer can be presented to the user in several ways:

1. As an installable desktop application
2. As a virtualized application
3. As an RIA or “web portal”
4. Through a text-based terminal

Methods 1 and 2 are currently the most common, although method 3 is gaining in popularity. Method 4 is primarily of historical interest, although it should be noted that there are some commercially available LISs—mainly on the CP side—that still operate in this manner. Methods 1 and 4 are conceptually the simplest, whereas methods 2 and 3 bear some explanation.

Virtualized applications reside on the server, but are presented to the end user as if they were desktop applications. This is made possible by a class of programs known as “virtualization applications,” sold by companies like VMWare and Citrix. These applications behave in a manner not dissimilar to streaming video, in that output is sent on a just-in-time, as-needed basis by the virtualization server in response to input that is sent on a just-in-time, as needed basis by the virtualization client. Virtualized computing is more resource-intensive and puts a much greater strain on the server than any of the other methods described.
include:

than several smaller ones. Advantages of this architecture budget on a single extremely powerful machine rather it therefore made sense for organizations to spend their an era when computational power was extremely expensive

In contrast, the LIS as RIA,""web portal"" (as exemplified by vendors who offer their LIS as an application service provider) is a relatively new idea. In this method, the functionality of the LIS is exposed through a set of webpages that are viewable by any modern browser. This is attractive for obvious reasons: the system becomes client OS-agnostic, and the end user is not required to install any additional software (especially important if the end user might not have administrative privileges for his or her work machine). However, because the system is exposed to the Internet, there are inherent issues with security and data ownership. Consider a situation where an entire LIS is implemented for a hospital by a vendor. Because the end-user interface is being delivered through a web browser, it is now possible for data to be stored “in the cloud” (ie, on clusters of servers on the Internet, some of which might not even belong to the hospital) as opposed to being stored on private secure hospital servers. In a situation like this, it is difficult to properly audit the system’s security, and even more difficult to ascertain who actually owns the data being transmitted.

APLIS Architecture

APLIS architecture relates to the combined hardware and software setup of the devices within the laboratory network. Historically, the LIS used “hub and spoke” mainframe architectures wherein the storage and processing of data was done centrally at a mainframe computer and information was displayed on peripheral “dumb” (ie, “without processing capability”) terminals.19 This architecture was born during an era when computational power was extremely expensive and programmer time was relatively cheap in comparison; it therefore made sense for organizations to spend their budget on a single extremely powerful machine rather than several smaller ones. Advantages of this architecture include:

- The ability to limit major maintenance and updates to a central mainframe computer.
- Consistency of information display on terminals distributed across the whole network.
- Security monoculture: only a single system to defend, meaning that all resources allocated to security could be spent more effectively.

Unfortunately, for all its advantages, this architecture also has some major disadvantages, which include:

- In a high-user setting, even the power of a mainframe can be overwhelmed.
- If the mainframe goes down, the entire system becomes unusable.
- Dumb terminals are no longer made; client computers can run terminal emulation software instead, but this ignores the fact that clients have impressive computational capabilities of their own.
- Security monoculture: if 1 system is breached, then the entire LIS is breached.

In contrast, client-server architecture is illustrated in Figure 1. This is the dominant architecture used with the current APLIS, and is projected to remain dominant in the future. In this architecture, end users interact with “thick client” computers—each of which is more powerful than the most powerful mainframes of 3 decades ago—that run the APLIS application layer as standalone programs that interface over the network with the servers on which reside the DBMS. Advantages of this architecture include:

- The ability to tap into the computational power of modern desktop computers.
- The continued benefit of centralized data management and manipulation.
- Distributed computing resources, meaning that it is possible for client computers to operate at least temporarily in “offline” mode even when the servers go down.

However, this approach has disadvantages of its own:

- Increased complexity of design.
- Large amounts of data traversing networks, necessitating heftier network resources.
- The overhead of maintaining the client computers.
- Security: having to support multiple systems instead of just one.

There is a variant of client-server architecture known as “thin-client” architecture, in which client computers are deliberately outfitted with a minimum of computational resources, and instead the relevant applications are virtualized from the server (section Database Management Systems). This variant brings back many of the advantages of the mainframe architecture, but also unfortunately brings back many of the disadvantages of that architecture. Recent advances in hardware virtualization technology (eg, AMD’s AMD-v; Intel’s VT-x and VT-d) have mitigated some of the performance disadvantages of the thin-client architecture, but problems of scalability remain.20

Finally, laboratories may choose to use the services of web-based (cloud) LIS vendors. These arrangements enable the laboratory to use a web-delivered portal to a vendor-provided LIS that allows them to perform functions similar to an on-site installation. The advantages of this system include lower installation and maintenance cost, especially for smaller practices. Disadvantages include storage of data on off-site servers, inability to truly audit security, and limited ability for customization. Largely due to the data ownership and security issues, cloud LIS is not presently considered a viable option for many practices, including ours. Although the challenges surrounding cloud-based technologies are real, the possibility that these challenges might someday be overcome is also real. This allows us, at the very least, a tantalizing glimpse of a future in which all LISs may exist in a secure and standardized cloud, allowing for truly transparent interchange of medical data across organizations.51

APLIS ELEMENTS

There are 3 fundamental components found in any LIS, whether it be an APLIS and/or a CPLIS:

- Dictionaries
- Worksheets
- Interfaces

Dictionaries

Dictionaries, otherwise known as “maintenance tables” and “definition tables” are data tables in the LIS
Worksheets are also known as “logs” or “work orders.” They define specimen flow and data flow through the laboratory, most often by defining a day’s or shift’s work for a given area. For instance, a histology log would indicate for the histotechnologists, which cases are required to be embedded, which tissue sections to perform, and which stains to perform. In contrast, a pathologist’s work list would tell a pathologist which cases he or she has yet to sign out. The structure and format of, as well as the data elements within, these worksheets are dynamic, and are constantly being updated electronically depending on crosslinks with other worksheets and dictionaries. For instance, a dictionary might define a case as being “overdue” when it is of a certain Current Procedural Terminology code that denotes a low-complexity specimen, it has no further histology pending, and it has been on a pathologist’s work list for a certain number of days. At this point, this case would meet criteria to appear on an “overdue cases” worksheet, and would programmatically be added thanks to these crosslinks.

### Interfaces

Interfaces are software and hardware connections that allow for interchange of data between otherwise incompatible systems. LIS interfaces come in three broad varieties:

- **Application interfaces**
- **Interface engines**
- **Instrument interfaces**

Application interfaces are interfaces to other computer systems, most commonly a hospital’s EMR. These interfaces are crucial, as this is how a patient’s ADT information, as well as his or her demographic statistics, can be discovered by the LIS. It is through these interfaces that computerized order entry occurs, as well as result reporting and transmission of billing codes. In contrast, interface engines are a specialized case of application interface—to be more precise, they are an amalgamation of many application interfaces into one, with a single input and a single output application programming interface. Interface engines are a major engineering undertaking, but when successfully implemented they reduce the complexity of the system by reducing the number of individual interfaces needed for multiple systems—each system need only be interfaced to the interface engine, rather than to all the other systems.

If application interfaces are software-to-software links, then instrument interfaces are best described as hardware-to-software links. Hardware instruments—like automated immunostainers—are interfaced with the LIS either directly or through a translational software layer known as “middleware,” and are able to write output to and/or take input from the LIS programmatically. Given the automated nature of the CP laboratory, it comes as no surprise that these are integral in CPLIS. However, with the increasing prevalence of barcoding and RFID technology in AP, instruments as disparate as cassette engravers, slide labelers, and immunohistochemistry strainers are now routinely interfaced to-software links. Hardware instruments, like the rest of the LIS, must be implemented individually. Although LIS vendors typically have “off-the-shelf” interfaces for the most common analyzers found in laboratories, it is important to recognize that because each laboratory’s dictionary is unique, there is no such thing as “plug-and-play” in the world of the LIS. As a result, all interface software must be carefully installed and tested on both the instrument and the LIS, with rigorous validation before go-live on a production server.

### APLIS Functionality

The functionality of an APLIS can be divided into 3 phases:

- **Preanalytic**
- **Analytic**
- **Postanalytic**

**Preanalytic Phase**

Although with a CPLIS the vast majority of all orders are handled by a computerized order entry system, for the APLIS the order entry is still largely manual and dependent upon paper—often with handwritten requisitions.
Electronic order entry interfaces for AP are not commonly implemented for several reasons:

- There are no specific dictionary-driven tests; specimens are generically called “surgical pathology” or “cytopathology” or “autopsy” specimens.
- AP orders require more information as compared with CP orders; a blood specimen for chemistry testing can simply be specified by checking off the proper box for the desired testing, whereas an AP order would ideally contain information like an organ of origin, a specified location, and relevant clinical information.
- A single order may encompass several parts from several different organs; this is not something that the CPLIS has to commonly deal with.
- AP specimen collection is inherently procedure driven (as opposed to CP specimens, which often require nothing more than a blood draw); for instance, a surgeon might initially order for an AP specimen but be either unable or unwilling to collect the specimen due to operation complexity or patient instability (leading to accession numbers with no specimens, in the worst case).

The first interaction the APLIS usually has with a specimen is at the time of its receipt in the AP laboratory, usually with a printed requisition. Once the case is received, a human is required to manually accession the case during which the APLIS assigns it a unique accession number, and related information from the requisition is entered into the APLIS. In multipart cases, each part is entered and documented separately.

There are 2 data fields in particular that are important at this stage: the “part type” and the “part description.” The part type is chosen from the possible specimen types that have been built into the APLIS part type dictionary, and cannot be entered as free text. Other data fields—including fee codes and histology protocols—can be auto-populated given the part type is chosen from the possible specimen types that have been built into the APLIS part type dictionary, and cannot be entered as free text, usually by way of dictated description. Text templates for commonly processed specimen types (eg, colon polyp biopsies) exist, and there have been some successes with speech-to-text recognition software at this stage.

The final product of the first part of this phase is the so-called “gross report,” which consists of a description of the specimen, how it was dissected, what was seen macroscopically upon dissection, and an alphanumeric list (key) designating what tissue went into each cassette. This key becomes of importance to the APLIS at this point, but in the absence of truly effective natural language parsing algorithms that could render a gross report into machine-understandable form, it is currently impossible for the APLIS to programmatically extract that information. Cassette engravers may be interfaced with the APLIS to keep track of how many tissue cassettes were made per case, but do not provide meaningful information on the kind of tissue that went into the cassette. As such, tissue cassette designations must usually still be entered into the APLIS by hand. Gross specimen digital images are commonly acquired during grossing, and some APLISs have modules to accommodate and manage these images (Fig. 2).

In histology (or the cytology laboratory), the APLIS supports the workflow of slide preparation by leveraging the part type dictionary to trigger predefined protocols for sections to be cut and stains to be applied. Slide labels are autogenerated based on the tissue cassette data previously entered by the gross prosector. Specimen tracking and barcoding are both being increasingly used in this phase, with the LIS providing the ability to update specimen status and location based on the scanning of a barcode or, less frequently, an RFID-enabled tag. Some LISs have gone as far as to autogenerate barcoded and labeled slides at individual histotechnologist stations at the time of individual case microtomy, leading to a reduction of case misidentification and an improvement in histotechnologist efficiency. Once the slides have been created and are ready for distribution, the slides are paired with an autogenerated “working draft” (so-called case assembly), which is templated in an LIS dictionary and includes requisition data, the patient’s demographics and relevant clinical history, the gross description of the specimen, any interoperative consultation diagnosis (eg, frozen section), and the patient’s past AP reports.

Once the case has reached the pathologist, the analytic phase nears its end. If at this point the pathologist needs to order recuts, special stains or immunohistochemical stains before making his or her diagnosis, the histology interface has tools for computerized order entry. Final pathologic diagnosis is, like gross examination, largely a free-text affair, usually involving transcription of a pathologist’s dictation. Just as in gross examination, other options include selection of predefined templates or quicktext for frequent diagnoses (eg, tubular adenoma) by either dropdown list or coded text entry, and/or speech-to-text conversion by voice recognition software. Once the final diagnosis has been entered, the case is marked as “final” in the APLIS, and thence placed on the pathologist’s queue (worklist) for final edits and electronic sign-out. Billing and diagnostic codes are often entered automatically at this point based on part type, stain orders, and sometimes rudimentary natural language processing of the final report. Final case sign-out consists of an electronic signature that flags the case as being complete, and incapable of being further modified. This action also triggers the transmission of the final report through an application interface (or an interface engine) into a downstream system such as the clinician’s EMR.
Because of the fact that so much of the preanalytic phase data are stored as narrative free text, it is not as easy to analyze the data in an APLIS as it is in a CPLIS. Not surprisingly, there is an increasing push toward the deconvolution of text-based pathology reports into more structured synoptic reports that contain discrete data elements (Fig. 3). Use of synoptic checklists (Fig. 4) makes reporting efficient (easy to use), uniform (standardized among surgical pathologists), and complete (contains all required data elements, such as those provided by the CAP cancer checklists). Using synoptic dictionaries laboratories can customize their synoptic checklists to incorporate data elements important for their practices. Several LISs offer synoptic reporting modules. Alternatively, third party vendors that facilitate synoptic reporting are available that readily interface with the LIS. Synoptic reports containing discrete data element better facilitate QA and research initiatives.

Postanalytic Phase

Pathology reports can be automatically reported to clinicians in a variety of manners. In the case that the ordering clinician belongs to the same hospital/institution as the pathologist, pathology reports are sent electronically through an HL7 message to a reporting interface that sends the report to the hospital’s EMR. In the case where the ordering clinician is not part of the same hospital, some LISs may enable “auto-faxing,” in which the final report is automatically faxed to the clinician using the relevant fax number found in that clinician’s person dictionary entry. It is also possible (and increasingly popular) for a clinician to be directed to an online portal, in which he or she is given

FIGURE 2. An example of an Anatomic Pathology Laboratory Information System integrating gross and microscopic digital photograph management capabilities.

FIGURE 3. Final synoptic pathology report showing an example of a urethra specimen with urothelial carcinoma that contains several key scientifically validated data elements. Data headings are in gray, discrete data elements are in black, and free-text areas are in italics. Note that (a) the free text is quite limited and (b) even in free-text areas, there are dictionary rules that make sure only valid data is written (in this case, the rules would be “must include a unit of measurement” and “must be a number”).
direct electronic access to relevant final reports. There is significant customization in the generation of AP reports, as most LISs offer both extensive text formatting options and the capability to insert logos and digital images. In any case, laboratories are required by law to ensure that the electronic display of their results in a downstream system is an accurate representation of what resides in their LIS.

Amendments and addenda are a fact of life in the AP realm, and as such the APLIS must be flexible enough to handle these postanalytic reporting events. Most LISs offer dictionary-based capabilities to enumerate the reasons for an amendment or an addendum; this is desirable for both documentation and business analytics purposes. When an amendment or an addendum is made, the LIS must clearly report that an amendment or an addendum is present, and keep an audit trail of the changes made.

THE APLIS AND DIGITAL IMAGING

AP is a predominantly image-based specialty, yet we have traditionally been slower than our colleagues in Radiology in adopting digital imaging techniques.24,25 Recent advances in WSI have raised the possibility of an all-digital AP workflow, but such an all-digital workflow is not projected to be in widespread use for at least the next few years. Comprehensive literature already exists pertaining to the state of digital imaging in pathology,10 so in this section we will instead focus on how the LIS when integrated with images can be—and has been—an enabler of digital pathology. There are two main aspects to consider:

1. The APLIS as an image management system.
2. The APLIS as a “digital cockpit” for signout.

The APLIS as an Image Management System

Traditionally, nondigital photography (Polaroids, Kodachromes) has been used in both gross and microscopic pathology for both diagnostic and teaching purposes. This kind of usage has extended itself very naturally into the digital realm, with many practices exclusively using digital cameras to take pictures of gross and microscopic specimens. As digital photography becomes more prevalent, however, it becomes necessary for there to be some way to manage the growing repository of digital imaging data.

Let us consider the uses of an APLIS as an image management system throughout the stages of the digital imaging process:

- Acquisition refers to the process of creating the digital image itself. Although some interchange standards (eg, TWAIN) exist that can facilitate this process, it is by no means “plug-and-play.” Not many pieces of imaging hardware and software (and almost no WSI scanners) are currently integrated with LIS. This is problematic because an end user is more likely to take (and integrate into reporting) photos if the functionality to quickly take a snapshot of the relevant case is available within the workflow of the LIS itself, rather than the alternative (in
which the end user has to go to a separate application to take the image, save it, and then import it into the LIS.

- **Storage** refers to the specific manner in which the digital image is stored—both on physical media and in the database of the LIS. There are 2 approaches to consider: an image management module as an integral part of the LIS, or a separate image management system that automatically feeds images into the LIS (Fig. 5). Both approaches have their advantages and disadvantages:

  - **Integral image management** means that the user will never have to leave the LIS, and (perhaps more importantly) that the image can be more easily manipulated in the setting of the LIS (Fig. 2). Images can be kept in a gallery for internal use (documentation purposes, etc.), or copied into final reports. At time of acquisition, the LIS can also record image metadata into its database, including but not limited to the date the picture was taken, the location where the picture was taken, and the user who took the picture. Disadvantages of this approach include (a) the fact that image-editing tools are restricted to what the LIS explicitly supports (either as native tools or through a TWAIN interface); (b) that it is more difficult for the end user to directly access the raw image data; (c) if the LIS goes down, so too do all digital images in-system; (d) that the file format in which the images are stored may be proprietary, hindering interoperability.

  - **Separate (modular) image management** can accomplish everything that integral image management

---

**FIGURE 5.** Integral versus modular image management. Integral image management in gray, modular image management in white. APLIS indicates Anatomic Pathology Laboratory Information System.
As WSI—and with it, a pure digital workflow for AP—becomes more prevalent, it is likely that we will see the LIS integrating additional image management features, especially with relation to the rich metadata that can be embedded in these very large imagesets. However, in order for a pure digital workflow for AP to become a reality, there is an important user interface problem that APLISs will have to surmount first: the concept of the “digital cockpit.”

As any anatomic pathologist knows, it is impossible to truly sign out a glass slide in isolation. Relevant case data—including the operative note, the gross report and its associated images, older surgical pathology cases from the same patient, and the patient’s clinical notes—is often crucial in the delivery of the correct diagnosis. Traditionally, this has involved multiple sets of glass slides and multiple pieces of paper for the pathologist to keep track of. The need to manage and keep track of this information is not lessened by the addition of an all-digital workflow: indeed, such a workflow only emphasizes the point that the image data by itself is not enough.

As such, there is growing interest in the specific way that an APLIS might present the available data on a case to the pathologist. This is a difficult problem in user interfaces, and most current solutions rely on at least 2 monitors—one to display the WSI, the other to display case and clinical data—for information display purposes. It is not known what form the “optimal” digital cockpit for AP will take in the future, but given the history of the LIS and of imaging systems in medicine, we are confident in stating that it will only happen with full participation from practicing anatomic pathologists in collaboration with user interface researchers and vendors.

**CYTOPATHOLOGY**

Cytopathology presents unique challenges to an APLIS, but also presents unique opportunities for analysis. The workflow of cytopathology is unlike that of surgical pathology, in that the prepared slides are first sent to cytotechnologists who screen the slides for the pathologist. As such, APLISs therefore allow for separate fields for screener impressions and final diagnosis. Cytopathology requires an assessment of whether the obtained specimen is adequate (satisfactory or unsatisfactory), a primary interpretation (negative, atypical, suspicious, positive), as well as a final diagnosis; this must be designed into the APLIS. Gynecologic and thyroid cytopathology has a codified diagnostic terminology (The Bethesda System), which must be taken into account when designing dictionaries. Indeed, one of the great benefits of the APLIS is that it gives you the ability to enforce these mandatory report entries before a case can be signed out, ensuring compliance and standardization. With the Bethesda System for Pap smears and now for thyroid lesions, diagnosis in cytopathology is becoming increasingly standardized. Although this adds complexity in terms of additional dictionaries to be created, it also allows for increasing amounts of cytopathology data in an APLIS to be held as discrete data elements instead of free text. This fact allows for easier statistical analysis of current cytopathology diagnostic data, and has wide-ranging implications for cytopathology data mining. Furthermore, in certain circumstances a particular diagnosis (eg, ASCUS for a Pap test) might lead to reflex testing (eg, high-risk HPV); this, too must be taken into account when designing dictionaries.

There are screening and performance indicators that are important per CLIA ‘88 that must be considered. Provisions for setting up the maximum workload for individual cytotechnologists is one such consideration: United States federal law requires that cytotechnologists manually document the number of slides screened in each 24-hour period,
and the number of hours spent screening each day. It is illegal to screen more than 100 slides per 8-hour periods or 12.5 slides per hour. The LIS could keep track of these things, and lock out individual cytotecnologists once their limits have been reached. Another consideration is the practice of rescreening: at a minimum, in the United States a 10% random rescreen of negative Pap stains and a rescreening of a specific percentage of negative “high-risk” cases is mandatory. This, however, neglects the fact that different users might require different rescreening measures. For instance, a cytotecnologist fresh out of training may require a higher rescreening ratio than a cytotecnologist with 30 years’ experience under his or her belt. The LIS can be instrumental in setting up these individual thresholds. Furthermore, the LIS can also be used to ensure that each cytopathologist—and the practice as a whole—is doing appropriate rescreening.30

The LIS could also be used to automatically flag cases, which would traditionally be described as “high risk,” either because of previous history or current history of abnormal signs, symptoms, and/or pathologic findings. This can involve natural language parsing of free-text fields like “clinical history” and “case description” with the goal of searching for suspicious text strings like “history of LSIL”; it can also involve the presence of previous cases such as a Pap smear that was diagnosed with LSIL last year. By using an algorithm (tuned to the specific laboratory) on all gynecologic cytology cases, the LIS can automatically alert the pathologist to cases that fall under its “high risk” criteria, thus improving both patient care and turnaround time.30

DATA WAREHOUSING AND DATA MINING

Compared with paper records, electronic data has tremendous advantages, including ease and efficiency of qualitative and quantitative data analysis, standardizing and structuring the reporting of results, rapid transmission of information, efficient integration and consolidation of multiple health records, and timely financial transactions (ie, billing). Electronic data storage also requires far less physical storage space. Moreover, multiple users may remotely access electronically stored information.

In data warehousing, data are continually extracted from production sources, copied, cleaned, transformed, catalogued, and made available for purposes such as data mining and decision support. Data warehouses are attractive because they:

- Maintain data history even if the production sources do not.
- Integrate data from multiple source systems that may be mutually compatible with one another, providing a central enterprise-wide view.
- Eliminate inconsistency and anomaly in enterprise data by applying a consistent code and metadata model.
- Allow for complex, processor-intensive decision support systems to be run without affecting the production environment.
- Allow for processor-intensive data mining and online analytic processing to be run, again without affecting the production environment.

In contrast, data mining refers to the act of having the computer automatically analyze large quantities of data to identify meaningful, statistically significant patterns. Common examples of such data mining at work include the Bayesian e-mail spam filters that by now are standard with every new e-mail account, as well as the way in which chess-playing computers “learn” how to become better at chess. With algorithmic advances in computer science like neural networks, genetic algorithms and support vector machines, data mining has become increasingly sophisticated and increasingly sensitive. Large-scale clinical trials are often mined for data by health organizations and pharmaceutical companies alike, and the conclusions drawn often give new direction to research and health policy.31

In the realm of APLIS, data mining is often used for the purposes of QA and tissue bank support. Using data mining techniques, it is possible to calculate, for instance, how long one’s pathology department takes on average to sign out a breast biopsy, or to flag unusual cases for review. It is also possible to run complex analyses on ranges of cases; for instance, one could take all the cases of a certain kind of bankable tumor, see how many were actually banked during a certain time period, and see how that compares to how many were not.

The difficulties surrounding the usage of data mining in AP largely revolve around 2 unfortunate facts:

1. Diagnostic terminology changes over time.
2. Almost all gross and final reports are handled as free text.

The first is the relatively easier of the 2 problems to tackle: simply create a dictionary that correlates older terms with newer ones. In contrast, the second is more difficult. While natural language parsing and analytic technology has made great progress since its inception, it is still impossible to create an algorithm that can extract adequate context from a pathology report. Although some branches of pathology have standardized their diagnostic terminology, others have not, and the wording of final reports is left to the discretion of individual pathologists. In this milieu, it becomes very difficult to properly mine the data in an APLIS without a large expenditure of time and effort. As steps are taken toward synoptic reporting and even synoptic grossing, however, data mining is becoming gradually easier and more fruitful.32,33

Some final points to consider:

- Electronic tracking of QC/QA indicators can be done either within the LIS and/or by exporting data from the LIS (eg, using common spreadsheet software or business analytics software like Altsosoft Insight or IBM Cognos).
- Most software solutions have fields where QA comments can be entered. Specimen rejection incidents and labeling errors should be documented, and regular reports run to monitor specimen rejection frequency by the clinician’s office. Periodic reports can be run to list occurrences and identify trends with any particular physician office sending specimens to the laboratory.
- Queries based upon the text entered (ie, natural language search) are possible with many current LISs. However, something as simple as typographical errors may interfere with searches based on text alone. This can be avoided by utilizing common built-in software features such as a spell check and automated comments in the LIS.

CONCLUSIONS

The APLIS is now an integral part of the AP laboratory and the hospital at large. Beginning as primitive in-house specimen registration and billing programs, they have evolved into complex, integrated information systems that are capable
of fine granular tracking of specimens, administration of laboratory workflow, automation of billing and coding, and retention of laboratory records used for patient care and overall laboratory performance improvement. Some APLIs integrate image management software packages similar to those used by radiologists, and still others are beginning to experiment with the integration of WSI.

Amidst the triumphs, there have also been many challenges. Although there is an increasingly great push toward interoperability in between disparate systems, true “plug-and-play” has remained elusive. Barcoding and RFID tagging have become increasingly prevalent, but few institutions have unlocked the full potential of these tracking technologies. The advent of WSI offers many exciting possibilities, but current issues of LIS interoperability and logistical feasibility remain. Although the APLIS has become more feature rich and faster running on ever cheaper commodity hardware, these systems are not easy to implement, operate, maintain, or even upgrade/replace. A great amount of standardization work lies ahead of us, especially in the realm of interfaces. This area suffers from the same problem that has plagued all medical interoperability standards: existing interoperability standards merely focus on the structure of the message, not on the format of the message itself. It is not too dissimilar from telling 2 people that their letters to each other can be interoperative because they each have an opening paragraph, one or more body paragraphs, and a closing paragraph, despite the fact that one only speaks Swahili and the other only speaks German.

Nevertheless, this is a time of great excitement and opportunity for the contemporary APLIS. As we enter the digital decade of personalized medicine, our clinicians and patients will demand greater access to integrated AP, CP, and molecular data. We are already seeing clinicians asking for microscopic images to be attached to pathology reports and comprehensive theranostic summaries, and this trend will only continue. The amount of data that exists in the APLIS is vast, but it has hitherto been difficult to fully mine that data because so much of it exists only as free text; as natural language processing technology advances, our data mining will become more effective, allowing for new theories to be explored and new conclusions to be reached. The patterns embedded in this data, and hopefully within pixels of stored digital images, will continue to inform treatment decisions as they have in the past, but the possibility—and the hope—exists that we will become much more agile at discerning them. Barcoding and RFID tracking of AP specimens will become standard-of-care, much as barcoding of CP specimens is now de rigueur. This, in combination with the usage of WSI in routine diagnostic work, will mark the end of an era where slides could be lost and cassettes misplaced, and the beginning of an era where fine-grained tracking of cases is the rule, not the exception. As the APLIS becomes increasingly image-centric, automated image analysis techniques will be used on an increasing basis. Data from these analyses will be recorded, data warehoused, and data mined, leading to advances in computer-aided diagnosis. The possibilities are endless.

REFERENCES


