Advancing technology in laboratory instrumentation along with higher-throughput processes are generating massive volumes of analytical data – even terabytes per project. The challenge is to automate and integrate laboratory operations and procedures wherever possible in order to manage and process these growing data volumes more effectively. A LIMS (Laboratory Information Management System), like any other information system, becomes more responsive and productive when it is integrated with other systems and applications in the laboratory environment. Integrating LIMS with analytical instruments not only automates laboratory functions but also provides data-sharing benefits that add value to the original investment.

LIMS implementation tends to be a high profile undertaking, and effective instrument integration can provide a positive and visible early milestone. Achieving this integration, however, demands not just specific product functionality but a close partnership between the LIMS vendor and the customer.

Traditionally, instrument integration products were designed for single PC...
operation and struggle with today’s net-
worked operations, leaving a clear need
for a more flexible, ‘scalable’ solution.
Alternatives are emerging that employ
advanced technology and architecture
that provide considerable more effi-
cient and effective integration of LIMS
with laboratory instrumentation.
Instrument integration systems
acquire data automatically, with the
primary goal of eliminating errors in
transcription and reducing the amount
of time needed to manually write or
type results. Most systems in the phar-
maceutical industry were implemented
to ease compliance with regulatory pro-
tocols, such as 21 CFR Part 11. Further
development has meant that some sys-
tems offer instrument connectivity that
involves very little coding to be
installed and, more importantly, the
fault tolerance necessary for 24-hour production environments.

Systems are usually capable of con-
necting to most instruments that have
either an output transmitting ASCII or
that can export data as an ASCII file.
Also, each PC that runs the instrument
integration software is able to connect
multiple instruments so that the num-
ber of PCs that can be connected to the
LIMS is effectively unlimited.

costly specifications
But these types of solutions, often
termed ‘thick client’ solutions by infor-
matics vendors, tend to fall short in a
number of areas and can struggle, par-
ticularly in laboratories in a high
throughput environment. They were
not originally designed to handle huge
volumes of data from the latest versions
of highly sophisticated instruments.
Scalability of hardware can also
rapidly become a problem. Each PC
dedicated as an instrument workstation
needs a full system installation — capa-
ble of data acquisition, processing and
interchange to the LIMS, while also
providing the client interface for the
user. This inevitably requires a costly,
high specification machine at each
workstation. Also, logistical problems
may be encountered when integrating
instruments ‘down the wire’, in which
case a high specification PC would need
to be within the radius of the maximum
cable run from each instrument.

In a laboratory with 20-30 instru-
ments spread across a number of loca-
tions, providing the administration and
security for all workstation PCs can be
a major undertaking. Configurations
and upgrades of any individual ele-
moment of the software can require a full
installation of the application, includ-
ing the necessary configuration and
subsequent validation.

To cover for events such as hard disk
failure, an administrator would require
a back-up strategy for each workstation.
Maintaining system security and elimi-
nating risk of data loss becomes a sig-
nificant management challenge – occu-
pying resources that could be more
productively allocated elsewhere.

new approaches
A more flexible and scalable instru-
ment integration solution would incor-
porate an architecture that separates the
collection and processing of instrument
data into separate software compo-
nents. In situations where high levels of
data processing and modern, reliable
networking are required, these software
components could be deployed in a
distributed configuration. For many
global organisations, server-based data
processing is the preferable option.

A new approach to integrating LIMS
with instrument integration includes a
PC that holds a series of instrument
collectors with the workstation and
instruments. The function of these col-
lectors is configured at the central
server, which instructs each to look for
either a data file from a network or hard
drive or a stream of data via an inter-
-facing port.

For example, the server can be con-
figured to direct a collector to listen for
a stream from a specific port, or to be
ready for a particular stream of data
and to stop at the termination of the
stream. In addition, configuration of the
server controls the scheduling of the
transfer of the instrument data from the
collector. No processing whatever
need be carried out at the
instrument workstation.

As analytical instru-
ments evolve with new
technology, collectors may
be added to support differ-
ent types of data output.
For instance, access to data
via XML (eXtensible
Markup Language) based
files are becoming more
prevalent within pharma-
caceutical QA/QC laborato-
ries. Connecting these
instruments to the LIMS
could be as simple as installing a
new collector.

An architecture utilising a central
server within a dedicated, secure, IT
department-controlled strong-room leads
to management efficiencies in a number
of areas. For example, it provides a single
location for updates, for configuration of
all parsing scripts (which extract the data
from the instrument output) and map-
ping scripts (which correlate the data
from the instrument to the correct fields
in the LIMS).

The single central server also enables
the back-up routines and security to be
centrally administered and provides
the ability to build in hardware fault
tolerance, redundancy and back-up at a
single location. In traditional archi-
tecture for LIMS/instrument integration
architecture, all these critical issues
would need to be addressed at each
workstation individually.

Taking PC theft as an example, in a
workstation-based application a full
installation of the application would be
required on a replacement PC plus a
full configuration determining from
where data is to come and in what for-
mat (instrument file, COM-based
instrument, etc). Reconfiguration of all
the parsing and mapping scripts would also
be necessary.

This would either result in signifi-
cant down time, or require laborious
manual transcription of results while
waiting for the instrument to be recon-
figured.

Contrast this scenario against a solu-
tion using scalable architecture, where
a replacement preconfigured PC can be
quickly brought online. Only the col-
lector component needs to be installed,
along with any local parsing and map-
ping configurations, for the system to
be up and running again.

Once the server is directed to the new
collector or series of collectors, instru-
mment capture can be restored immedi-
ately, furthermore, savings can be made
since only low-grade PCs would be
required in this new architec-
ture.

Modern LIMS/instrument
integration solutions have been
designed to address the data
management and security requirements
of today’s laboratory, such as compli-
ance with 21 CFR Part 11. Organisations
may configure different roles and groups
for individuals inside and outside the
lab, assign appropriate functionality and
system menu options to groups or spe-
cific end-users, and prevent unautho-
rised users from accessing specific areas
of the system.

Integrated security, password protec-
tion, audit trail and electronic signatures
allow multi-user access, while ensuring
each piece of information is accessible
only by authorised personnel.

The trend for larger LIMS vendors
Informatics is toward developing and implementing their own instrument integration solutions as alternatives to separately sourcing third-party products. This approach, taken by Thermo Electron Corporation with its flagship LIMS, SampleManager, provides additional value to existing and future installations. A standard user interface is utilised for all types and brands of instruments, providing consistency throughout the lab and the organisation, reducing training requirements and accelerating implementation. Automating the transfer of results from analytical instrumentation to SampleManager is eliminating manual typing of results with its associated transcription errors.

Now, even in very high throughput environments, analytical information from sophisticated instrumentation can be delivered accurately and quickly to laboratory management and other authorised decision-makers, increasing productivity in the laboratory and lowering the cost of ownership for LIMS.

Contact
Colin S. Thurston is LIMS product manager, Thermo Electron Corp, based in Altrincham in the UK.
T +44 161 942 3000
info@thermoinformatics.com
www.thermo.com/informatics

Microsoft .NET technology is a popular framework for scalable computing architecture and the ‘componentising’ of software solutions. This technology offers a building block approach to application development utilising XML, enabling interconnectivity as well as integration to larger applications over the Internet. The attributes of XML as a basis for a data file standard have been well documented and it is enjoying widespread acceptance both as a data interchange and storage format, including within the regulated industries. XML is a public domain, platform-neutral data formatting standard. It offers an application-independent way of representing data, however rich, using plain ASCII text.

These technologies provide a ‘future-proofing’ quality to the LIMS/instrument integration solution. New collectors, for example, can be embedded in the software by an instrument vendor, thereby simplifying integration with the LIMS.

By using architecture based on components rather than single larger applications facilitates customisation of collectors without affecting the whole solution. In a regulated environment, this offers significant savings in the extent of validation required to demonstrate compliance with GxPs.

By using Microsoft .NET within such an architecture, the instrument integration system can be presented to the user in different ways, via a Windows32 client or web pages. The ability for a chemist to configure instruments and review the status of instruments using PDA technology is another possibility.

By using a Web service interface to store analytical data in a LIMS as part of this architecture, the processing components can be isolated from variations in multiple LIMS implementations; an important consideration with large pharmaceutical manufacturers operating on a global basis. In brief, a Web service is simply an application that can be delivered as a network service and integrated using standard Internet technologies. Much of the strength of the Web services concept lies in the fact that it combines a simple-to-understand format with proven and well-established communications technology.

Web services are built using a group of XML standards, thereby providing complete platform independence. This allows an organisation to deploy the Web service on the server platform of its choice, and to use the Web service from any application written in any programming language. A Web service also allows for the reuse of existing business rules within the LIMS. Organisations that have standardised on a LIMS for their laboratories will also have configured the LIMS differently for specific labs – r&d and production environments, for example. Should one of the LIMS be upgraded in this situation, no changes would be required to the instrument integration functionality. This simplifies deployment of new LIMS features, reduces validation efforts, and allows much greater flexibility than conventional LIMS deployments.

LIMS users, in fact users of any software application, will be familiar with how existing applications can mysteriously stop working when they install a new application onto their PC. This is due to conflict between the .dll files in the two applications. By isolating applications, .NET eliminates this conflict. This means that when installing a new collector, for example, at the instrument workstation there need be no concern when using .NET component-based architecture.

This not only minimises impact on PC performance, but also reduces the amount of validation required in regulated environments.