Achieving standardization, connectivity, and compliance in process analytical technology

M. A. Power and W. M. Doyle

This paper outlines a new process analytical control software suite developed specifically to meet the established requirements for process analysis while providing a standard means for operating a wide range of analytical instruments, the connectivity required for networking, data archiving, and interfacing to other programs, and the safeguards needed for regulatory compliance.

1. Introduction

Process analytical technology (PAT) is both an old and a new field of activity. Chemical process analysis (to use the more historical term) has been an active field of endeavor for over 60 years. For example, online infrared (IR) spectroscopy was widely used in the synthetic rubber industry during World War II.[1] Within the past few years, however, infrared spectroscopy has seen rapidly growing application in the pharmaceutical industry resulting in the introduction of a number of new requirements.

Wide-scale applications of chemical process analysis in the petrochemical industries often involve the continuous processing of commodities such as gasoline, commodity feedstocks, or polymers. In such applications, a given instrument configuration — and often a given calibration — might be replicated numerous times at various plant locations. And each application generally involves only one type of instrument, most often monitoring a liquid (or at least molten) stream. The requirements of the pharmaceutical industry, however, are often more complex — involving short-run batch processes, samples that often are in powder or tablet form, and a wide range of different products.[2] Under such conditions, it is highly desirable for calibrations to be simple and generic.[3] At the same time, it is often necessary to integrate the results from disparate measurements in order to form a complete model of a complex process.[4] Another factor that distinguishes pharmaceutical process analysis from chemical industry experience is the need for compliance with various industry standards and regulatory requirements in the areas of good automated manufacturing processes (GAMP) and electronic signature integrity (21 CFR, Part 11).

Leading process analysts in the chemical industry have long recognized the need for a standard, flexible software platform capable of controlling diverse instruments while meeting the established needs for on-line process analysis.[5] These needs include sample system configuration and control, instrument control, extraction of process variable information from the raw data, real-time trending, system diagnostics, alarming, archiving of data, comprehensive historical data analysis, and remote communication using common protocols. The complex analytical needs of the pharmaceutical industry have made the case for standardization even more compelling while adding new requirements for regulatory compliance and data archiving.

The program outlined in this paper has been aimed at meeting the established requirements for process analysis while providing a standard means for operating a wide range of analytical instruments, the connectivity required for networking, data archiving, and interfacing to other programs, and the safeguards needed for regulatory compliance.

2. Requirements for a Standardized PAT Software Program

To those who have spent decades working in the field of process analysis, the distinctions between PAT and the more traditional world of process measurement may seem obvious. However, in our discussions with workers in the broad field of process automation, we are struck continually by the fact that

First published in the Spectroscopy supplement, *Process Analytical Technologies*, February, 40-50 (2006)

The authors are with Symbion Systems, Inc., 1451-A Edinger Ave., Tustin, California 92780. M. A. Power's e-mail address is mpower@gosymbion.com.

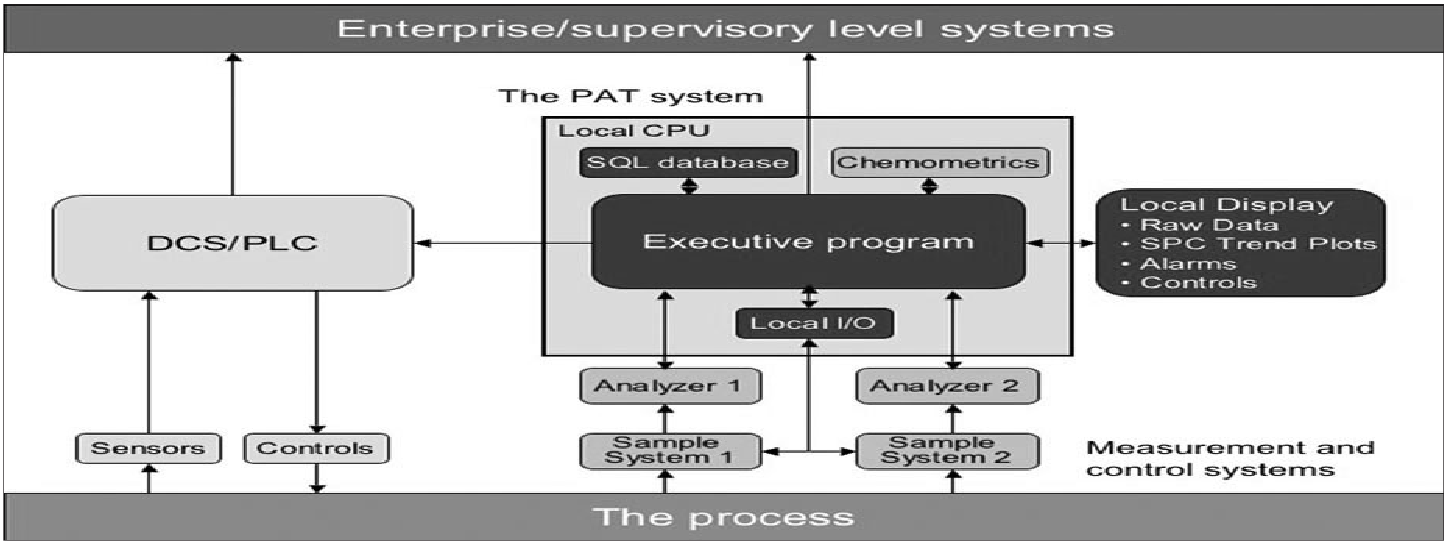


Figure 1. The role of PAT executive software.

this distinction is far from obvious to most people.

Figure 1 illustrates the role of a PAT executive software package in a comprehensive PAT system. The PAT system occupies the space between the manufacturing process and various enterprise data systems. It is in a parallel position to the data-gathering capacity of traditional distributed control systems (DCS), which are used universally throughout the chemical and pharmaceutical industries. However, there are fundamental differences between the two systems. The DCS obtains monovariate (scalar) information at a low data rate from sensors such as flow meters, pressures sensors, or resistance temperature detectors. Typically, each data stream will have a one-to-one relationship with a process variable such as pressure or temperature. In contrast, a process analytical system is designed to measure more complex and subtle variables such as chemical composition or the spatial distribution of various properties. Often it might measure several variables simultaneously. To do this, it will often employ one or more complex instruments such as near-IR spectrometers. The PAT executive software program, therefore, must control both a sampling system and one or more sophisticated instruments. It must handle the collection of a large amount of raw data. It must provide a means for extracting process information from the raw data. It must provide any required local user interface functions. It must provide for the storage of both raw and process data as well as the hand-off of data to other data systems. Finally, it must control the proper sequencing of all of these activities.

The analytical techniques employed in a PAT system have been used for many years in analytical laboratories.[6] However, there are a number of factors that need to be considered in choosing a particular analytical technique for on-line application. Among the more important are robustness, low maintenance, and suitability for continuous data acquisition. For many applications, these considerations tend to favor various forms of spectroscopy over the slower and more maintenance-intensive chromatographic techniques. In addition, one of the big attractions of spectroscopy is that it can gather very large amounts of information very rapidly. A typical spectrometer will record between 2000 and 4000 independent spectral points in a fraction of a second. Thus, a very important distinction between PAT and the traditional world of process automation is this high data rate – combined with the fact that the data must be analyzed in order to provide information about the process variables of interest.

The process of extracting process information from raw multivariate data is most often carried out by a chemometric software program. While a chemometric routine could be integrated into the PAT executive, this is generally not necessary or even desirable. Chemometrics is a highly specialized capability and several very capable programs already exist. In addition, most users already have strong preferences as to which program they prefer to use as well as libraries of existing calibrations. It is most important, therefore, for the PAT executive to provide interfaces to these existing programs so as to be able to integrate them into the overall measurement function.

In summary, the functions of the PAT executive include managing and sequencing the collection of a typically large amount of raw data, providing for the analysis of this data in such a way as to produce meaningful information about the process variables of interest, storing the results in a secure database, providing an appropriate local user interface, and interfacing to various other data systems. And, it must provide a complete audit trail, insuring that all the conditions under which the data were collected are stored permanently and are available for inspection. Simply stated, the overall task of the PAT software package is to integrate analytical technology seamlessly into continuous online measurement systems – in short, to convert sophisticated analytical instruments into smart sensors.

3. Elements of an Approach to Standardization, Connectivity and Compliance

In considering the design of an optimum PAT executive platform, we realized that a crucial issue was finding the right balance between the flexibility required for the rapid development of a wide variety of individual methods and the operational simplicity needed to enable the validation of each method. To accomplish this, we adopted an approach that employs syntax specifically tailored to the needs of process analysis. By combining this syntax with a PAT-specific user interface, we can provide ample flexibility within a structured framework uniquely suited to process analysis. Furthermore, by employing a modular, hierarchical command structure, we can minimize the number of commands required while standardizing the means for communicating with diverse instruments, analysis routines, and enterprise data systems. The key elements of this approach are as follows:

PAT-specific syntax. The challenge in developing appropriate command syntax is to provide the flexibility required to meet diverse needs while limiting the number of user commands to a specific set that can be easily validated. To accomplish this, first we designed a set of generic commands which cover the categories of operations common to process analysis. Then we combined this with a modular approach in which a given command can accomplish a variety of tasks by substituting the appropriate arguments. These arguments refer to various drivers and other modules containing specific instructions. The final step was the provision of a set of scripting assistants to enable the commands to be written automatically.

Database storage. The use of an SQL database, as opposed to flat file storage, is an essential element in meeting each of our three major goals. Furthermore, the detailed design of the database structure is also critically important. It must be organized in such a way as to provide standardized tables for a wide variety of specific information while providing links between measurement configurations (that is, data collection and analysis methods), user information, and both collected and processed data. Database structure is thus an important key to standardization. It fosters connectivity also by providing a standard format for exchanging information with instruments, chemometrics programs, data historians, and other systems. Finally, a properly designed database structure can provide a secure audit trail with a complete record of the time, date, author, and description of each operation performed.

PAT-specific GUI. Our overall goal of providing ample flexibility to meet any process analysis requirement combined with the simplicity of operation needed for compliant routine application is fostered by combining a PAT-specific command syntax with a set of standard operating windows (actually programs) corresponding to the major PAT functions. In approaching the development of our user interface, we found that the various functions that were needed fall into four fundamental categories which could be assigned to a set of four windows. These are as follows:

- The manual window serves as a standardized instrument operating screen, enabling the user to operate one or more instruments and I/O devices in real time. This window can be used, for example, during the initial set up and testing of the various components of a system and for collecting calibration data.
- The set-up window provides overall control of system operation. It includes the tools for configuration design (method development) as well as the means for setting various system preferences and user permissions. In particular, the scripts that control both system operation and access to various features during routine operation reside in the set-up window.
- The run-time window is the only window normally available during routine on-line operation. In its standard configuration, it displays any combination of the following three types of information (as determined by preferences established in the set-up window): current and recent multivariate information such as spectra, time-dependent (SPC) trend plots, and tabular statistical information. It provides for the input of certain operator commands, as determined by the appropriate preferences. The run-time window can also be customized to provide the specific displays and operator input functions required by each individual application.
- The historical window provides access to the SQL database. It allows the user to recall and display data sets by means of any combination of attributes such as instrument, operator, time and date, or process location. Once displayed, the data can be outputted in various formats or copied to the manual window for further manipulation and analysis.

Virtual run-time capability. In developing PAT applications, it is important to be able to test and compare various approaches and assumptions. To do this, we needed the capability to operate a given method in a virtual fashion in which data is recalled from the database rather than being obtained from a currently operating instrument. This "replay" mode is essential, for example, in developing and testing chemometric methods.

4. Implementation

The approaches outlined above have been implemented in two closely related sets of programs referred to as Symbion-DX and Symbion-RX (Symbion Systems. Inc., Irvine, CA). The primary difference between the two is that Symbion-RX includes a number of safeguards and limitations designed to aid in compliance with various standards and regulations in the pharmaceutical industry. The sections below will focus on the aspects of these programs that are especially pertinent to standardization, connectivity, and compliance.

Syntax. As noted, a major goal of our program has been the development of syntax that would enable the specific needs of process analysis to be met with a minimum number of relatively simple commands. To accomplish this, we adopted an approach that translates the typical control objects into a standard set of hierarchically organized commands. These allow transparent control and fluid transport of data from one object to the next. The specific code related to the function of a given object is contained within that object, allowing the high level script to remain general and modular. The result is efficiency, economy, and transparency. As an example, consider the following three commands:

"Set x1 [ControlInstrument Instrument ScanType #Scans OptParams …..]"

"Set x2 [RunPrediction Predictor CalFile $x1 #]"

"Plot RTTag structure $x2 Concentration Name Units Color"

These three standard commands are all that are needed to operate an instrument, predict the value of a process variable based on the data obtained, and plot the results as a function of time. The first command is used to operate any instrument. The identification of the specific instrument model is substituted for the argument "Instrument" shown in green. This argument then directs the program to a driver that contains all of the information needed to operate the selected instrument. After the instrument argument, the command contains a series of placeholders for additional arguments whose number and function are determined by the instrument driver. In the case of a spectrometer, the first two of these are generally "Scan Type" and "Number of Scans". The rest of the arguments are often optional and can be used to set other operating parameters that are exposed by the instrument in question. In some cases, the "optional parameters" can refer to a default operational method and installation directory, as in the following example:

"set x1 [ControlInstrument Foss Background 8 {OM = XDS, install_dir = C:/Vision}]"

The second command in the group of three listed previously simply runs a prediction using a program (whose name is substituted for the word "Predictor") and a designated calibration file. The "Predictor" can be a chemometrics program such as Pirouette (InfoMetrix, Bothell, WA), Simca QP (Umetrics AB, Sweden), or the PLS Toolkit (Eigenvector Research, Inc., Wenatchee, WA), or it can be a user-generated math routine. In any case, the command operates on the data contained in the structure x1 produced by the first command and outputs a prediction contained in the structure x2. The argument "#" designates the selected output in the case of a multiple component prediction.

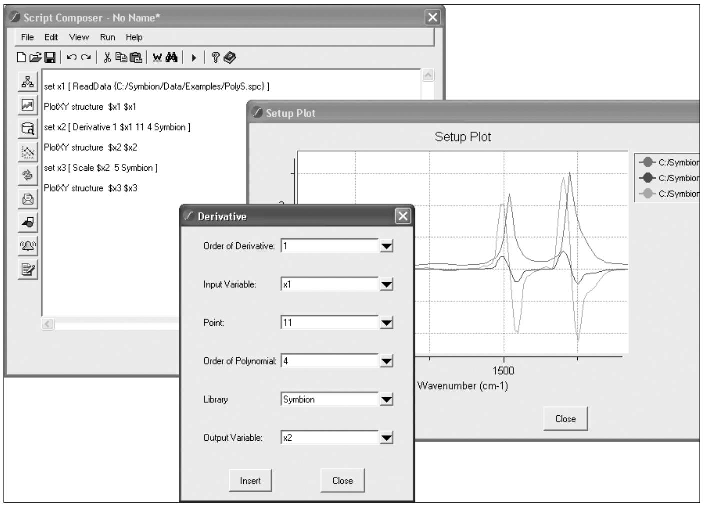
The third command simply plots the predictions contained in x2 as a function of time and applies the indicated plot type, component name, units, and trace color. In the case of a multiple component prediction, the second and third commands, listed earlier, are simply replicated, with the "x" structure names and the component "#'s" incremented appropriately.

By using this modular approach, we have been able to condense all of the control, analysis, and connectivity operations commonly encountered in process analysis into a total of about two dozen commands. These are augmented by a set of optional math routines that can be used for preprocessing data. The issue of "validation" then comes down to validating a relatively small number of commands plus a set of specific instrument drivers and interfaces to chemometric and other software programs.

Variable format. In the previous discussion, we have referred to the variables x1, x2, and so forth as "structures" rather than "values". In fact, these "structures" comprise both numerical data and attribute information. For example, in the first command given above, the variable x1 contains numerous objects, each associated with a particular associated datum or attribute. For instance, the data block xvar contained within x1 holds all of the x-axis data returned from the instrument. Similarly, the attribute block, ylabel, contained within x1 includes the y-axis label attribute returned from the instrument.

Composing scripts. As illustrated by the previous example, our modular syntax provides a set of commands that are simple and transparent. These commands standardize the functions and the input and output relationships between analytical instruments, analysis applications, and industry standard I/O systems. In essence, we convert these instruments, applications, and systems into "black-box' objects that have pre-defined function, input, and output arrangements. The result is rapid applications development.

In spite of the simplicity inherent in our command structure, we recognized the need to further simplify the writing of scripts so as to minimize the learning time required for full-scale application development. To do this, we developed a "script composer" function with "scripting assistants" corresponding to each of our standard commands. An example is given in Figure 2. In this simple illustration, we have recalled a spectrum from memory, performed a Savitsky-



```
set x1 [ ReadData {C:/Symbion/Data/Examples/PolyS.spc} ]
PlotXY structure  $x1 $x1
set x2 [ Derivative 1 $x1 11 4 Symbion ]
PlotXY structure  $x2 $x2
set x3 [ Scale $x2  5 Symbion ]
PlotXY structure  $x3 $x3
```

Order of Derivative: 1

Input Variable: x1

Point: 11

Order of Polynomial: 4

Library: Symbion

Output Variable: x2

Figure 2. The Script Composer, a typical scripting assistant, and the "Setup Plot" window.

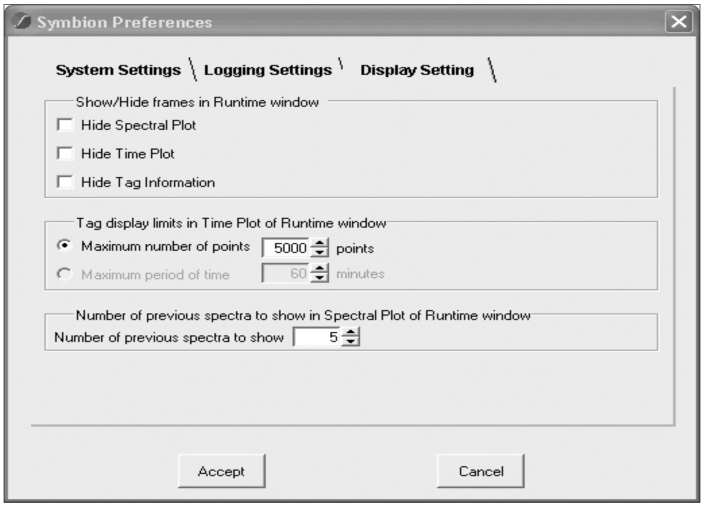
Golay first derivative, scaled the result by a factor of five and plotted all three spectra. The "derivative" pop-up displayed allows the user to select the values of various parameters from a set of pull-down menus. The user then presses the "insert" button and the program writes the appropriate line of code – in this case, the third line shown in the "script composer" window. If the user then selects "execute in Symbion" from the "run" pull-down, the results will be plotted in the "setup plot" area. Any errors in the script are reported in the "message" area of the "setup" window.

Once a set of commands has been written and tested using the script composer, it can be pasted into either the initialization script or the run-time script area of the Setup window and saved to the database as a new measurement configuration. The database entry will include a full audit trail, including time and date and the identity of the author. Any subsequent modifications to the configuration will also be tracked.

Run-time operation. The operation of the system during on-line application is controlled by the set of "configuration" scripts loaded in the initialization and run-time areas of the setup window. The initialization script controls those operations that are only required once at the beginning of a run. These may include the setting of certain valves or acquiring an instrument background. The run-time script cycles continuously during routine operation until an "off-line" command is received.

In addition to containing the operating configuration scripts, the Setup window provides access to a number of additional administrative functions such as user management, database management, and system preferences. For example, Figure 3 shows the page of the "system preferences" window that controls the appearance and function of the standard "run-time" window.

The standard run-time window includes three primary display areas. These correspond to current data (typically spectra), time-dependent plots, and statistical "tag" information. Any or all of these can be hidden by checking the appropriate boxes in the above system preferences window. In addition to the standard displays, additional, custom displays can be popped up over the run-time window to provide for alarms, operator inputs, or any other functions specific to a particular application.



Show/Hide frames in Runtime window

□ Hide Spectral Plot

□ Hide Time Plot

□ Hide Tag Information

Tag display limits in Time Plot of Runtime window

● Maximum number of points 5000 points

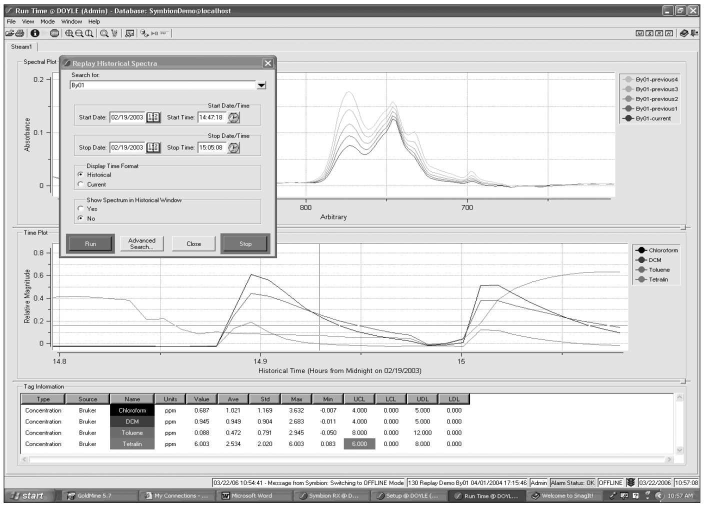
○ Maximum period of time 60 minutes

Number of previous spectra to show in Spectral Plot of Runtime window

Number of previous spectra to show 5

Figure 3. The Display Setting page of the System Preference window. These settings govern the appearance of the standard Run-Time window.

Virtual run-time operation (replay mode). The replay mode can be used with any online configuration. It disables the control instrument command and replaces it with a stream of previously acquired data recalled from the database. This mode of operation is extremely important for use during applications development. For example, it allows chemometric calibrations and other data processing algorithms to be compared using a given set of data. An example of its operation is shown in Figure 4.



Search for: By01

Start Date/Time

Start Date: 02/19/2003 Start Time: 14:47:18

Stop Date/Time

Stop Date: 02/19/2003 Stop Time: 15:05:08

Display Time Format

● Historical

○ Current

Show Spectrum in Historical Window

○ Yes

● No

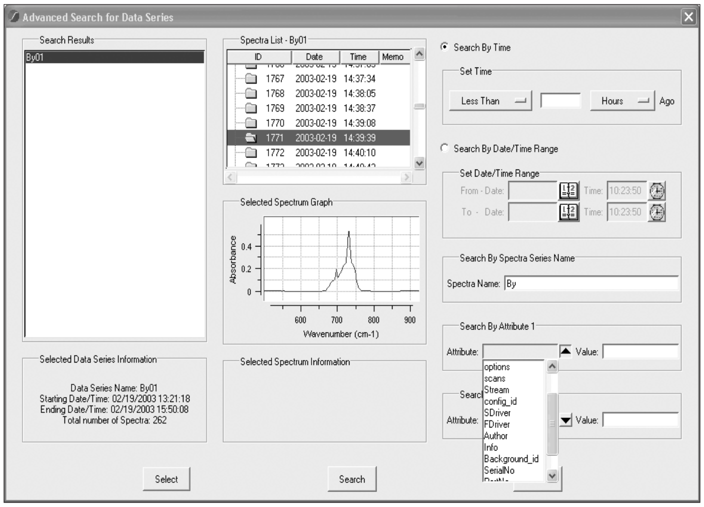
Type	Source	Name	Units	Value	Ave	Std	Max	Min	UCL	LCL	UDL	LDL
Concentration	Bruker	Chloroform	ppm	0.687	1.021	1.169	3.632	-0.007	4.000	0.000	5.000	0.000
Concentration	Bruker	DCM	ppm	0.945	0.949	0.904	2.683	-0.011	4.000	0.000	5.000	0.000
Concentration	Bruker	Toluene	ppm	0.088	0.472	0.791	2.945	-0.050	8.000	0.000	12.000	0.000
Concentration	Bruker	Tetralin	ppm	6.003	2.534	2.020	6.003	0.083	6.000	0.000	8.000	0.000

03/22/06 10:54:41 - Message from Symbion: Switching to OFFLINE Mode | 130 Replay Demo By01 04/01/2004 17:15:46 | Admin | Alarm Status: OK | OFFLINE | 03/22/2006 | 10:57:08

Figure 4. The standard Run-Time display showing an example of Replay operation.

Locked-down applications. After a method has been developed using the various tools available through the Setup window, it usually will be locked down so as to operate only in a predefined way. This typically involves establishing permissions allowing an operator to only access a dedicated version of the run-time window and to operate only a specific measurement configuration. In some cases the allowed version of the run-time window will simply be the standard version with one or more of the standard display areas hidden. In other cases, all of the standard displays may be hidden and a new custom screen popped up over the blanked out display.

Database operations. As noted earlier, the use of a SQL database and, above all, the development of a well thought out, comprehensive database structure are essential to achieving all three of our major goals; standardization, connectivity, and compliance. In our design, a single structured entry in the database – corresponding to a single measurement – can contain an unlimited number of discrete pieces of information. These are organized into three general categories: user information (such as permissions and user attributes), data (both raw and processed data and various attributes), and control information (for example, measurement configurations). The general management of database functions is accessible through the Setup window to an administrator having the appropriate permissions. Access to data stored in the database involves a different set of permissions and is provided through the Historical window. The standardized organization of the database facilitates searching by any combination of attributes such as time and date, instrument ID, operator, or process stream. The Advanced Search window is shown in Figure 5. The results of a typical search are shown in Figure 6.



Search Results: By01

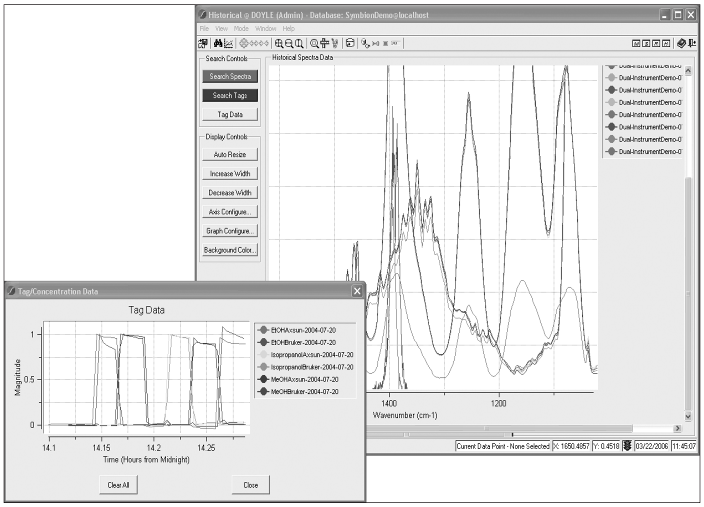
ID	Date	Time	Memo
1767	2003-02-19	14:37:34	
1768	2003-02-19	14:38:05	
1769	2003-02-19	14:38:37	
1770	2003-02-19	14:39:08	
1771	2003-02-19	14:39:39	
1772	2003-02-19	14:40:10	

From - Date: Time: 10:23:50
To - Date: Time: 10:23:50

Spectra Name: By

Data Series Name: By01
Starting Date/Time: 02/19/2003 13:21:18
Ending Date/Time: 02/19/2003 15:50:08
Total number of Spectra: 262

Figure 5. Database search window.



Current Data Point - None Selected | X: 1650.4857 | Y: 0.4518 | 03/22/2006 | 11:45:07

Figure 6. Database search results showing both spectral and processed time dependent information.

5. Conclusions

We have demonstrated an approach to achieving standardization and connectivity in a software environment that fosters compliance with industry and regulatory standards governing validation and security.

Standardization is achieved by means of a uniform database structure for all raw and processed data, measurement configurations, and user information, by employing a standard set of PAT-specific windows, and by the use of modular syntax which enables a limited number of transparent PAT-specific commands to perform all of the common functions encountered in process analysis. This syntax treats all instruments, I/O devices, chemometric packages, and other software programs as black boxes that fit into predetermined slots in the command structure. All of the specific arguments required by these devices and programs are determined by separate modular drivers.

Connectivity is provided by the availability of interfaces to numerous industry standard analytical instruments, I/O devices, and chemometric software packages. In addition, the development environment provides for the inherent capability of interacting with protocols such as TCP/ IP, OPC, ActiveX, and COM.

Compliance is aided by systematically controlling user access, by the provision of a comprehensive audit trail, by employing a limited command set which has been fully validated, and by providing the means for individual methods to be locked down so as to further limit their function to only those operations required for the particular application.

The net result of all of these measures is the provision of a software package that can serve both as a highly flexible application development tool and as the overall executive for controlling all aspects of a PAT system. It controls the instrumentation and sample systems, sequences all events, and supervises the flow and processing of data. In short, it manages the measurement function, performing all of the functions needed to mold a complex analytical measurement system into a smart sensor of chemical or physical composition.

References

1. R. Robert Brattain, The Spectrum 75(10), 1-7 (1999).
2. Steve V. Hammond, Process Analytical Technology 2(4), 15-18 (2005).
3. Jill Parris et al., Spectroscopy 20(2), 34-41 (2005).
4. John Higgens et al., Process Analytical Technology 2(5), 23-29 (2005).
5. David Marrow, Exxon Chemicals (Baytown TX), private communication.
6. Angelo De Palma, Pharmaceutical Manufacturing July/August, 29-32 (2004).