Title: CASE STUDY FOR THE EVALUATION AND SELECTION OF MAN-MACHINE INTERFACE (MMI) SOFTWARE

Author(s): H. Nekimken, N. Pope, J. MacDonald, R. Bib @ B. Gomez, D. Sellon

Submitted to: Instrument Soc. of America, Chicago, IL
October 7-10, 1996
DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
Case Study for the Evaluation and Selection of Man-Machine Interface (MMI) Software

Howard Nekimken, Noah Pope, John Macdonald, Roland Bibeau, Ben Gomez, and Dustin Sellon,

Nuclear Materials Technology Division, Los Alamos National Laboratory.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer— or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
ABSTRACT

We evaluated three of the top man-machine interface (MMI) software systems. The main categories upon which we based our evaluation on were the following: Operator Interface; Network and Data Distribution; Input/Output (I/O) Interface; Application Development; Alarms; Real-Time and Historical Trending; Support, Documentation, and Training; Processing Tools (Batch, Recipe, Logic); Reports; Custom Interfacing; Start-up/Recovery; External Database; and Multimedia. We also present our MMI requirements and guidelines for the selection and evaluation of these MMI systems.

I. BACKGROUND

Over the past decade there have been numerous computer technological advancements, including the evolution from the 286 to the P6 personal computer processor. Software advancements have tended to lag behind the rapidly changing PC hardware. However, over the past decade OS/2, Windows 95 and Windows NT have joined UNIX as prominent, true, 32 bit, multitasking operating system software. Although typical office computing often does not require true multitasking, tasks such as man-machine interfacing benefit enormously from this computing environment. The man-machine interface (MMI) provides a graphical and intelligent interface between process operators and the process. A more complex MMI system will require multiple operations to be prioritized, even preempting unimportant operations, and will run simultaneous operations when necessary. It is not surprising that MMI software has changed dramatically in the past few years.

Selection of an MMI system for updating an old control system or developing a new control system should be completed carefully. A software evaluation should be carried out based upon the needs and requirements of your processes. However, there are several features and aspects that are important to most control systems. In this document we will present suggested guidelines for the evaluation and selection of MMI software. We will also provide some details of our own MMI evaluation and selection in the Nuclear Materials Technology Division at Los Alamos National Laboratory. The main categories upon which we based our evaluation on were the following: Operator Interface; Network and Data Distribution; Input/Output (I/O) Interface; Application Development; Alarms; Real-Time and Historical Trending; Support,
Documentation, and Training; Processing Tools (Batch, Recipe, Logic); Reports; Custom Interfacing; Start-up/Recovery; External Database; and Multimedia.

In 1992, North Carolina State University, PC Week Magazine, Sara Lee Corp., the Industrial Technology Institute, and others teamed up to evaluate what were then seven of the top MMI software systems. While this PC Week shoot-out provided an excellent summary of the software capabilities, the evaluation criteria differed significantly from the ones that were important to us, as determined by our own requirements. We performed our own evaluation using specifically tailored criteria with weighting that reflected our needs and requirements.

II. METHODOLOGY

A. Team Assignments

Because of limited time and resources, we chose to do complete evaluations of only three of the top MMI systems. Five people from our Process Control and Instrumentation team were chosen to be judges for this evaluation. Three team members were each assigned one software package. As part of this assignment, these individuals attended training courses for the software they were assigned to and arranged with the software supplier to have evaluation software systems shipped to us. Each of these three team members were assigned to complete certain identical process control, data monitoring, and data acquisition tasks with their MMI software. These application screens are listed along with a few of the required features for successfully completing these screens in Section II.B. The other two team members were assigned tasks for all three software packages: one individual was in charge of the computer networking aspects of the evaluation, whereas the other individual was in charge of the hardware interfacing aspects. Ultimately, the team members assigned software packages gave a presentation of their MMI software systems, answering any questions about the software and providing sufficient information to all the judges so that they could numerically rate the software in all the chosen categories. These categories are described in Section II.E.

B. Test System and MMI Application Screens

The test system used for the evaluations was an automated ion-exchange prototype system built as part of another project in one of our laboratories (Figure 1). An Allen-Bradley
PLC was used to acquire the data, while the MMI software systems were run on a Gateway 66 MHz Pentium computer running Microsoft DOS 6.20 and Windows for Workgroups 3.11. We developed the following application screens for each of the MMI software systems:

- **Main Menu.** Among the features required for completing this screen were the ability to easily move between screens and the implementation of user security.

- **Ion-Exchange Valve Configuration.** Among the features required for completing this screen were ease and flexibility in drawing and manipulating objects and the simple implementation of screen animation.

- **PID Control.** Among the features required for completing this screen were the ability to present PID parameters simply and an ease of operator input to change PID setpoints.

- **Trending.** Among the features required for completing this screen were the ability to trend data both in real-time and historically, the ability to easily follow trended data, and the ability to easily move within the trending windows.

- **Maintenance.** A main feature required for completing this screen was ease in examining a variety of parameters, including the ability to view parameters to assist in troubleshooting the PLC and MMI systems.

- **Alarms.** Among the features required for completing this screen were simplicity in reporting a history of alarms and the ability easily to acknowledge alarms.

Other miscellaneous required features included the development of totaling liquid flows for overall volume determinations, demonstration of the use of audio messaging as part of an alarm condition, ease in the configuration of the I/O driver(s), ease of networking a local station to remote stations, and ease in generating reports.

### C. Requirements for MMI Software

There are a few requirements that must be available from an MMI package before it can be considered for use. MMI systems we evaluated had to meet the following requirements:

- The software must be reasonably priced. Although the determination of whether a software package is reasonably priced is somewhat subjective, the cost should be comparable to that of its competitors.
The operating system must allow multiple programs to run simultaneously (i.e., it must multitask). Several modules or programs typically must be run simultaneously in MMI software systems (e.g., trending, I/O driver, alarms, networking, etc.). Most operating environments or systems except DOS can run programs “simultaneously.” If a particular operating system has already been chosen by your company, an MMI version for that operating system must be available.

- The appropriate workable I/O drivers must be available as required by site installation(s).
- The company that developed the MMI software should be financially sound, be a leader in the market, and provide effective support. This requirement reflects a need to choose an MMI software system that we (you) believe can be used for several years in the future.

D. Numeric Evaluation

Thirteen main categories were ultimately chosen to be part of our software evaluation (see Table 1). For each main category we had at least two subcategories, usually more. Categories were numbered according to their scoring “weight” (i.e., category 3A, which had a weight of 3, was weighted more than category 2A, which had a weight of 2) as specified by team members according to perceived importance. Each judge gave each subcategory a score from 0 to 4, where 0 is equivalent to an F grade and 4 equivalent to an A grade. These subcategory scores were then averaged to yield an overall main category score. This score was then multiplied by the category weight. The total score was then determined by adding all the weighted scores from the thirteen main categories. Each main category will be discussed in some detail below. A complete listing of subcategories for each main category can be found in the Appendix 1. The categories and their corresponding weights were based on our needs and requirements. Each company’s evaluation categories and weights should be determined based upon their own needs and requirements.

E. Category Descriptions

Performance

Performance was originally one of our main categories, however due to limited time and resources we were unable to complete extensive testing necessary to evaluate MMI system
performance. Given sufficient time and resources performance testing is recommended. There were eight subcategories pertaining to system performance. Once there is a data query to the MMI system, the response time should be fast and each query should allow for a large number of data points. In general, the number of points processed per second and the frequency data can be polled should be as high as possible. Overall response requires a rapid response from the I/O driver(s) and if applicable the data networking system. The speed of screen refreshing is also a factor in the overall system responsiveness. The computer resource requirements of an MMI package are also very important considerations. This includes how much system memory (both low and high memory) is used and required by the MMI, how much hard disk space is required and used by the MMI, and if applicable, how much and what type of network resources are required and used.

3A: Operator Interface

There were ten subcategories pertaining to the operator interface. The graphic user interface should allow an operator to easily move between application screens and within an application screen. For example, “point-and-click” technology allows the operator to easily open and close valves and change numeric analog outputs. The ability to print out displays and trends is also important to the operator. Again, the ease of executing these printouts is important. The manner in which process information is displayed on the screen is important. An operator should easily be able to determine the status of process variables. Some tools that can make this simpler include graphics animation (e.g., flashing objects, changing colors, moving and overlaying objects) and pop-up messaging of events. Support for various user interface devices is important (e.g., mouse/trackball, touch-screen, and point-and-click technology mentioned above). The ability to use certain keys to execute various commands (i.e., macros typically using the function keys) can simplify operations.

Security is important from an operator viewpoint. Some aspects of an application screen need to be protected by the application developer so that an operator does not inadvertently make a change that could create serious problems. Useful features include the ability to protect various parameters and screens via security controls and the existence of multiple levels of user security. It is also useful for the operator to be able to monitor data trends and process screens.
simultaneously either on the same window or on different windows that can be observed simultaneously on a monitor display. Displaying status information on the screen provides the operator with valuable process-related information.

One of the most important aspects of the operator interface is the availability of on-line help. If an operator does not know how to complete a certain operation within the MMI software, on-line help can be invaluable. Looking up how to do something in a manual and calling one’s supervisor are much less efficient modes. It is advantageous to have some sort of self-teaching tool or program available when an operator is learning how to use a particular MMI software system. This is an excellent way for an operator to become familiar with the software without fear of doing something terrible to the actual process.

3B: Network and Data Distribution

There were ten subcategories pertaining to network and data distribution. Ideally, the MMI software should have the ability to distribute data via different protocols (e.g., via NetBIOS and transmission control protocol/internet protocol [TCP/IP]). NetBIOS is a peer-to-peer protocol introduced by Sytek and IBM in 1984. NetBIOS now runs on ethernets, token rings, ARCNETs, Starlans and interfaces to IBM, TCP/IP, XNS, OSI, and IEEE 802.2 protocol stacks. NetBIOS is currently the portable standard for network application providers. The reasons for NetBIOS’s success are its simplicity and history of reliability. TCP/IP is a peer-to-peer protocol which is well tested, well established, and is the standard for open system interconnections. Computer systems worldwide use TCP/IP to communicate because TCP/IP provides the highest degree of interoperability, includes the largest number of vendor systems, and runs over more network technologies than any other protocol. For example, the Internet is basically run on the TCP/IP protocol. TCP/IP is used by DOD, DOE, and NSF computers as well as many commercial and educational computers located around the world. TCP/IP is an ~15-year-old well-established technology that can operate over low-cost cables. If total bandwidth is kept under 10%, the probabilistic nature of CSMA/CD (carrier sense multiple access/collision detection) does not seem to affect plant applications. However, if an ethernet system runs at more than 80% of its capacity, performance may degrade because of the number of collisions and retransmissions that occur. A serious consideration for a TCP/IP network is security.
requirements. Since network stations broadcast their frames simultaneously to all other stations on the network, other stations therefore can receive and copy any transmitted data.

The ease with which process data can be configured for distribution throughout a network is very important. The client/server capabilities of networked data when using the MMI software are also important. These capabilities relate to how seamlessly and easily data, application screens, and tags are updated across the network. Ideally, only the I/O server systems will physically have the MMI database on their hard drive. Dead-band network filters are useful because they can be used to regulate data distributed throughout the network, thereby helping to minimize the amount of network resources required to distribute this data.

Network speed and performance are important because they determine how quickly and reliably process data is distributed throughout the network. The size of the network is a factor in determining the speed and reliability of the data distribution system. How local and remote MMI stations interface to the MMI databases is also important. This interface also affects the network speed and reliability. There are various remote station capabilities that can be very useful, including distributed data storage (a possible mode of data backup) and processing and the use of remote graphics. Remote execution can add flexibility to an MMI system, although this feature should be used sparingly. One important aspect of data distribution is combining data from multiple I/O machines. If an MMI system can combine data effectively, processing within a plant can be overseen and monitored much more easily.

3C: Input/Output (I/O) Interface

There were seven categories pertaining to the I/O interface. The MMI software should provide a variety of I/O drivers, including support for common I/O hardware such as Allen-Bradley and Modicon PLCs. In addition to this, the MMI system and operating system should be capable of running multiple I/O drivers simultaneously. The I/O driver should be easy to configure and should allow access to all the features provided by the data acquisition hardware device (e.g., PID blocks, block transfers). The rate of transfer of data between the MMI graphical system and the I/O data acquisition device should not be what limits the graphics screen update rate.
These drivers and/or the main MMI software should provide troubleshooting tools to help solve data-acquisition-related problems. Similarly, if there is a driver communications watchdog system (a means for continuously monitoring the I/O driver status) available, operators and system developers can readily keep track of the status of various I/O drivers and devices. Finally, the MMI system should provide support for user-defined device drivers. In some cases, less common data acquisition hardware must be used. The MMI vendor should make it as easy as possible to develop a communication link between these devices and the MMI software.

3D: Application Development

There were fourteen subcategories pertaining to application development. The MMI system should handle a variety of data types, including strings, integers, floating-point numbers, and digitalists. The level of computer knowledge required to develop an application should be minimal. The application developer should not have to be an advanced computer programmer or a computer scientist. The amount of time required to learn how to develop an MMI application should be minimal.

An important aspect of developing an application is configuring the MMI database. Important factors in working with the database include the ease to configure and modify the database, the ease with which this information is transferred to other MMI workstations, and the documentation for this database. The ease with which the MMI database can be checked and maintained will have an impact on the overall maintenance of the entire MMI system. When drawing screens, how the screen is developed is important. Support for graphic properties, functions, and operations such as cut and paste, scaling, rotating, symbols, fonts, color, graphics resolution, undo, grids, and access to bitmaps all combine to help determine the difficulty in developing an application screen. The availability of libraries of predefined symbols and the ease of creating and saving user-defined symbols are crucial to the development of application screens.

Control languages are sometimes required to complete certain (often mathematical) operations. Scripts, which are preconfigured functions or commands, are sometimes available for this purpose. The triggering and/or initiation of internal or external procedures are also
important. Triggers can be time (e.g., periodic or scheduled) or event (e.g., change of state, physical I/O, or operator interface events) based.

Evaluation of the graphic user interface was discussed in reference to the operator interface. The graphic user interface is also important in application development in terms of the ability to maneuver about a screen using point-and-click technology and the utility of on-line help (to save time in developing an application). It is useful to be able to employ data entry validation, which assists users in changing analog output values. Critical features of an application should be protected by security. Flexibility in implementing security (e.g., the use of different security levels) is useful because different aspects of a screen or application may require different levels of security.

The ability to document an application and the ease with which to complete this documentation are important. Developing quick reference application information should not require accessing external word processor software. Finally, the MMI’s tag system is very important. Key factors to consider when evaluating the tag system are the maximum number of tags allowed and the maximum number of characters allowed for the tag name and the tag description.

3E: Alarms

There were eleven subcategories pertaining to alarms. A critical MMI system requirement is that it provides real-time monitoring of alarm conditions. A useful feature is the ability to set multiple alarm limits for a given parameter. The availability of different alarm categories (groupings such as critical and urgent) is also important. Multiple alarm limits and groupings used together provide significant flexibility in configuring and using alarms. The ease with which alarms are configured (just like tags, etc.) is important to effectively implement process alarms.

Alarm messaging is an important tool to assist operators in monitoring process operations. Alarm notifications are valuable at remote workstations. Alarm messaging, acknowledgment, and clearing (or resetting) are all important features on a local MMI station. On a remote station, these features may or may not be useful depending on the process requirements. In the initial stages of implementing an MMI system, it is useful to be able to
modify alarm limits on line. Once the behavior of certain process values are understood better, alarms may need to be modified. Logging alarms, both in real time and historically (hard copy and on disk), are crucial in an MMI system. Ideally, the MMI system will provide flexibility in where and how alarms are logged.

The option to take some sort of automatic action (including an automatic remote program call), triggered by a parameter's alarm condition, can enhance the options available when an alarm condition exists. One excellent way to provide more information about alarms is to incorporate multimedia (i.e., WAV files). Voice messages can be more descriptive than normal alarm messages. Different audible alarms can also be used to improve alarm recognition.

3F: Real-Time and Historical Trending

There were ten subcategories pertaining to real-time and historical trending. A good trending system should have the capability of monitoring multiple variables at once. This capability should include the ability to observe simultaneously multiple trending windows and multiple parameters over a given time span.

The MMI system should be able to plot trends of virtually any MMI database element. Often it is desirable to plot expected, modeled, or planned data on the same chart as the actual data. Ideally, this can be done directly within the MMI system; however as an alternative, MMI data can be sent to another application (e.g., a spreadsheet) for comparison purposes. It is desirable to be able to extract data to external applications as easily as possible.

There should be features which allow the operator to easily maneuver within a trend chart. Some of the more important maneuvering features are (1) being able to pan through data, (2) the ability to easily zoom in and out, and (3) the use of a cursor to look at values of different parameters at different times. Real-time data is often more easily viewed than historical data. It is important that the MMI system allow the operator to look at both types of data as easily as possible. The MMI system should also make it as easy as possible for realtime and historical trending charts to be configured. Finally, the maximum number of data points that can be trended is important. Certain processes may have many parameters (both analog and digital) that are important enough to monitor as a function of time (at least during certain segments of a process).
3G: Support, Documentation, and Training

There were fourteen subcategories pertaining to support, documentation, and training. The frequency of upgrade releases is important. Upgrade scheduling and the types of improvements made in the upgrade affect the overall usefulness of an upgrade.

On-site engineering support should be available if a customer is in need of assistance beyond phone customer service (such as helping develop an MMI application). The quality of customer service is crucial. Several factors affect the overall quality of customer service including (1) the hours the service is available, (2) the speed of response, (3) the friendliness of the support people, (4) the support people’s product knowledge, and (5) whether a call tracking system is employed. A bulletin board service (BBS) and/or a world wide web (WWW) page should be available to easily upload or download files. A user group and/or user conferences are important, because they should give present or perspective customers a chance to discuss the product more objectively and to trade user tips.

The quality of documentation is another very important consideration. Documentation should be easy to read and use, be complete, be easy to access, and include up-to-date information. Training is a vital aspect of customer support. There should be a separate, skilled training staff and the training materials should be effective learning tools. Other factors, such as the frequency of training classes and the availability of other training modes (e.g., video training), should also be considered. The time required to learn how to use the MMI software effectively, often determined by how much training course work is required, should be considered in an MMI software evaluation. The software company should allow the option of training on-site.

2A: Processing Tools (Batch, Recipe, Logic)

There were four subcategories pertaining to processing tools (batch, recipe, logic). A useful MMI tool is the ability to trace alarms in an effort to determine process faults. The flexibility of the recipe management and batch control systems are important. Some of the important features for these MMI subsystems include the quality of documentation, the ability to edit recipes on line, the option to use audit trails for recipe changes, the ability to store libraries of recipes, and automatic upload or download of recipes based on a time or event. The ability to
execute operations based on timed intervals can provide flexibility (such as, implementing user-defined shifts). The availability of counting functions is also useful. Some important counter features include semi-real-time counting operations, increment and decrement counting, and the use of preset and limit values.

2B: Reports

There were five subcategories pertaining to reports. The ability to generate reports automatically based on time or event triggers is very important. It is useful to be able to print graphics as part of a report. An important feature is to create whatever report layout is desired. The MMI software should have the ability to send generated reports to a printer or to archive them as ASCII files onto a disk. Another useful feature is the ability of external applications to access data for the purpose of creating reports.

2C: Custom Interfacing

There were three subcategories pertaining to custom interfacing. The MMI software should allow user programming with scripts or some simple programming language or languages that have debugging capability. Another useful feature is support of common computer languages such as C, BASIC, Pascal, and FORTRAN. Finally, an important evaluation factor is program execution speed: is a slower interpretive mode all that is available or is there a faster compiled mode.

2D: Start-up/Recovery

There were eight subcategories pertaining to start-up/recovery. It is vital that the MMI system start up in a proper and safe configuration after a power failure or reboot. Retention of last known values can be important if something happens to the MMI workstation (e.g., a power failure) in the middle of a process run. Using the last known values should make it easier and safer to continue a process run at a later time. The time required for the MMI system to come back on line after a computer reboot (e.g., after a power failure) can be important. If a qualified operator wants to restart the MMI software for any reason, this task should be relatively easy to accomplish and should not require a complete reboot.
The MMI system should have the capability to log system errors, which is useful when troubleshooting the system. In addition to the ability to log errors, it is also useful to be able to display system status information at the same time as process information. Often, when the MMI system is starting up, it is advisable for certain modules to be started before other modules. The system should provide the flexibility to change the order by which modules are started. Finally, it is useful to be able to receive network alarms that report if a workstation fails or never starts.

2E: External Database

There were two subcategories pertaining to external databases. The MMI system should provide interfaces to various external databases, including SYBASE, ORACLE, and dBASE IV. MMI system support for distributed database reads and writes in real time and especially in block format is also useful.

1A: Multimedia

There were two subcategories pertaining to multimedia. The MMI system should have the ability to call a sound file, typically a WAV file, based upon a time or event. The system should also be able to import and export graphics files, such as TIF, DXF, and BMP files.

IV. SUMMARY

We have outlined how we carried out our MMI software evaluation and provided some guidelines that should be useful in other companies’ evaluations. Most importantly, the final details of a company’s own software evaluation should reflect their own control system requirements and needs. The final MMI selection may also involve other factors such as price or something less tangible like the system people in your organization are most comfortable with. In a few cases the best choice may actually be the selection of more than one MMI system.

We decided to upgrade our old MMI system to one we considered to be in the mainstream that used current technology, possibly even state-of-the-art technology, had a strong support department, and was sold by a stable mainstream company. We wanted to choose a system we could count on in the future. An important aspect of our final selection of an MMI package turned out to be price and the availability of local support.
Figure 1. Automated ion-exchange system.
Table 1. Main Evaluation Categories

<table>
<thead>
<tr>
<th>Category Number</th>
<th>Category Name</th>
<th>Category Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>3A</td>
<td>Operator Interface</td>
<td>3</td>
</tr>
<tr>
<td>3B</td>
<td>Network and Data Distribution</td>
<td>3</td>
</tr>
<tr>
<td>3C</td>
<td>Input/Output (I/O) Interface</td>
<td>3</td>
</tr>
<tr>
<td>3D</td>
<td>Application Development</td>
<td>3</td>
</tr>
<tr>
<td>3E</td>
<td>Alarms</td>
<td>3</td>
</tr>
<tr>
<td>3F</td>
<td>Real-Time and Historical Trending</td>
<td>3</td>
</tr>
<tr>
<td>3G</td>
<td>Support, Documentation, and Training</td>
<td>3</td>
</tr>
<tr>
<td>2A</td>
<td>Processing Tools (Batch, Recipe, Logic)</td>
<td>2</td>
</tr>
<tr>
<td>2B</td>
<td>Reports</td>
<td>2</td>
</tr>
<tr>
<td>2C</td>
<td>Custom Interfacing</td>
<td>2</td>
</tr>
<tr>
<td>2D</td>
<td>Start-up/Recovery</td>
<td>2</td>
</tr>
<tr>
<td>2E</td>
<td>External Database</td>
<td>2</td>
</tr>
<tr>
<td>1A</td>
<td>Multimedia</td>
<td>1</td>
</tr>
</tbody>
</table>
Appendix 1. Evaluation Subcategories for Each Main Category

3A: Operator Interface
   Graphic user interface
   Adequate operator printouts
   Process information displayed on screen
   Security
   Trends and bar graphs
   User interface devices
   Programmable keys (macros)
   On-screen help
   Status information on screen
   Self-teaching program for introduction to program

3B: Network and Data Distribution
   Multiple network protocols
   Ease of configuration of network software
   Client/server
   Dead-band filter for network data
   Speed and performance of distributed software
   Size
   Interface of local and remote modules to real-time database
   Distributed system architecture
   Local and remote process execution
   Data from multiple I/O machines combined

3C: Input/Output (I/O) Interface
   Interface to multiple devices
   Multiple simultaneous I/O drivers
   Configuration
   Speed of data acquisition
   Debugging
   Communications watchdog
   User-defined device drivers
3D: Application Development

- Storage and retrieval of many data types
- Level of computer knowledge required for application development
- Time required to learn
- Ability to configure database components
- Screen definition
- Standard libraries
- Control languages
- Initiation of internal or external procedures
- Graphic user interface
- Data entry validation
- Security
- System configuration documentation
- System maintenance
- Tag system

3E: Alarms

- Real-time monitoring for alarm conditions
- Multiple alarm limits—exceptions and warnings
- Priority grouping (critical, urgent, etc.)
- Messages and displays
- On-line modification of alarm limits at operation
- Corrective action initiated on alarm
- Alarm notification at local and remote workstation
- Alarm logging
- Multimedia alarms
- Program call execution
- Ease of configuration

3F: Real-Time and Historical Trending

- Multiple variables monitored and tracked
- Trends plotted
- Multiple trend windows
- Method of displaying actual values at a given time
- Maneuvering within trend charts
- Method to superimpose planned data and actual data
- Ease of historical data examination
- Ease of extraction to other programs
- Number of data points
- Ease to configure
3G: Support, Documentation, and Training
Frequency of upgrade releases
On-site engineering support availability
Quality of customer service
Overall quality of documentation
Documentation easy to read and use
Documentation completeness
Documentation up-to-date information
Training
Time required to learn package (number of courses)
Customer site training available
Quality of training materials
Separate training staff
Bulletin board service (BBS) available
User group available

2A: Processing Tools (Batch, Recipe, Logic)
Fault diagnoses for alarms
Recipe management/batch control
Timed events and logic
Counting functions

2B: Reports
Automatic report generation
Graphics
Freeform layouts
Report archiving and printing
Access of data for external applications to create reports

2C: Custom Interfacing
User programming and scripts
Computer languages
Program execution

2D: Start-up/Recovery
Recovery from power failures
Log errors
Last known values retained
Time required to come back on line
Ease of manual restart
Display of system status
Start-up timing/sequencing issues
Network alarming on system failure
2E: External Database
Interfacing to external databases
Distributed database reads/writes

1A: Multimedia
Stored sound
Digitized imagery: TIF, GIF, EPS, etc.