

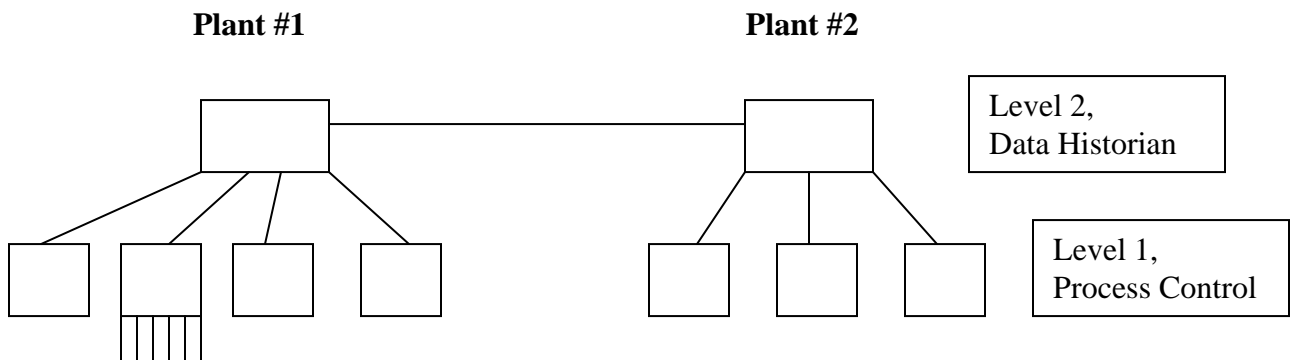
Eli Lilly and Co.'s Fermentation Process Control and Data Historian Computer System: A Model in "The Practice" of Implementing New Computer Science Techniques and Technologies

Copyright © by J. Alford and D. Clapp 2004
All rights reserved.

In the early 1970s, Eli Lilly and Co. had a dilemma. As one of the leading fermentation companies in the world, they realized that the fermentation process, (e.g. making penicillin), was as much art as science. Making antibiotics wasn't much more sophisticated than making wine. Ingredients were put into a tank which was then manually sterilized, manually inoculated with living cells, and then agitated and aerated for several days. The only automated feedback control was a pneumatic controller used for temperature. Data recording consisted of operators walking through the production floor, periodically recording values from gauges onto a clipboard sheet and hand entering a few values onto the paper manufacturing ticket.

Most of the parameters of real value to the fermentation culture (e.g. pH, dissolved oxygen, nutrient concentrations, etc.) were not monitored or controlled on-line. Further, Lilly scientists could not pursue their vision of an automated, well-understood, and controlled low-variability process, due to lack of appropriate tools and support systems. Sometimes, the best indicator as to how well a fermentation was progressing was the subjective analysis of an experienced employee looking through a tank site glass at the color and texture of the foam layer riding on top of the liquid broth.

Lilly scientists, engineers, and management decided they could, and should, do something about this. Since no commercial computer system & application software existed that could do the job, Lilly decided to develop their own system, based on a hierarchical architecture of HP minicomputers. The 1st layer of the system would consist of multiple process control computers, each capable of managing up to 10 fermentors. The 2nd layer would be a single computer/plant and contain the data historian.



Process Control (level #1)

Since disc technology was unreliable in the early 1970s, and near 100 % computer availability (uptime) was required, the Lilly team decided to create the process control portion of the system without discs, in a 32 K word (64 K byte) computer. This limited amount of memory had to include the operating system, configuration tables, and all application recipes. Note, in contrast, that today's PCs need over 30 times this much memory just to operate Microsoft Word.

The memory limitation led to the need to code the software in Assembler language. This language is very memory efficient as it has a one-to-one correlation with machine language (i.e. the computer's base manipulation of zeros and ones). As an example, the sending of a number to a printer required only a small (500 word) formatter in Assembler language. The same activity would have required a formatter several times larger had Fortran been used.

During the early 1970s, computer memory was very different than it is today. Memory, known as magnetic core memory, consisted of iron core donut elements located at wire intersections. Consequently, execution speed was slow and cost was high. Typical computer MIPS (millions of instructions executed per second) was about 0.5. This, in itself, was a challenge to the design team as they desired to develop a system that could process all fermentation device inputs (measurements) and outputs (control valves) and execute all algorithms every second. This was desired as several process control functions (e.g. generation of alarms) require a fast response by the computer. Also, some algorithms (e.g. PID controllers and totalizers) assume that a consistent, frequent, and periodic update of measurements is occurring.

These requirements helped define the need for Lilly to develop their own "real-time" operating system to insure that appropriate "interrupts" to the executing software could take place as needed.

If this were not ambitious enough, the objectives of the team also included:

1. use of relative I/O addressing so that an application written for one fermentor could work, without modification, for any fermentor in any Level 1 computer.
2. the ability to change any aspect of an application recipe on-line; *i.e.*, without having to recompile code, stop the process, and download edited code.

Data Historian (level #2)

The Level 2 computer part of the system was developed and implemented on an HP minicomputer with 32K of main memory and was the repository for all fermentation process data collected by the level 1 computers, captured in “flat files.” The level 2 computer was also the off-line home for Level 1 applications, known as recipes. The level 2 computer was the environment in which permanent edits were made to Level 1 recipes and from which recipes would be compiled (by a Lilly-developed compiler) and loaded into the Level 1 computer at the beginning of a fermentation.

No requirement existed for near 100 % uptime for the data historian so the team was comfortable in including use of a hard disc as a peripheral device. For longer-term data storage, a 9-track tape unit was also interfaced to the computer. The availability of a disc permitted use of HP’s operating system, so this part of the overall system did not have to be custom developed.

Additional objectives of the Level 2 system included:

- 1. The ability to record and present data by manufacturing lot #**
- 2. The ability to generate trend plots of process data using relative time (rather than calendar time). A batch process, such as fermentation, requires that the data be plotted as time (e.g. hours) since the beginning of the batch lot and not as the calendar time of day. It is noted that this functionality did not start to appear in commercial historians until 20 years later, and some still are unable to do this.**
- 3. The ability to transfer data files from one plant site to another. This would permit scientists to compare data from different fermenters at different plants (presumably making the same product) on the same plot. This would prove to be of great benefit in scaling up and transferring processes from pilot plants to manufacturing plants and in troubleshooting production problems.**

Early development and implementation

In a period of about 4 years during the mid- to late-1970s, a team of about 8 engineers and scientists developed this system and implemented it in a major fermentation pilot plant and in one of the company’s manufacturing facilities. It met all the objectives, as summarized above.

Of special note is that recipes were developed as a combination of 10 different types of building blocks, somewhat analogous to building structures with Tinker Toys[®]. Each block is configured and managed as an independent object, which has

“pointers” to other blocks. Examples of block types are: measurement input, electronic output, message output, data record output, calculation, timers, timed schedule changes and conditional schedule changes. Recipes mimicked today’s spreadsheets in how they organized, labeled, linked and coded the individual blocks. Extra spare unconfigured blocks could be included in the recipe to enable new functionality to be added on-line.

Once configured, compiled, and loaded into the Level 1 computer, the software still allowed on-line access to each of the blocks and all its attributes and permitted on-line changes. The requirement for on-line changes was visionary and ambitious, but proved countless times to be of very high value; i.e.:

Logic errors could be corrected on-line, without having to wait until the next recipe loading for the next manufacturing lot.

“Learning” that occurs during a process run could trigger immediate manual modification of the control/alarm strategy or to computational algorithms. This is important in developing biological processes since they are not as well understood and are more variable than non-biological processes. Further, the cycle time for bioprocesses is often 1 week or more, so losing a run due to uncorrectable software problems can cause the loss of product as well as much valuable time.

Finally, not having to download recipe changes in the middle of a manufacturing lot avoided losing the current values of software totalizers, the current output of PID controllers (i.e. valve positions), etc.

During this time, the team took advantage of new computer technology as it became commercially available.

One such change was the shift from iron core memory to semi-conductor memory during 1974, when the HP M series computers were put on the market. Previously, 8K words of memory would consist of a magnetic memory stack consisting of an 8” x 8” x 3” piece of hardware. Today, this amount of memory, plus much more, is contained in a single tiny chip. The shift in memory technology also permitted computer speed to increase from 0.5 to nearly 2.0 MIPS.

Significant improvements in throughput resulted from implementing microcode, which HP permitted and supported in a practical way to customers. This was driven by a continuing attempt to have the computer check and deal with many of its process control functions every second. Analysis of the executing software revealed that a particular sequence of 150 Assembly language instructions (an on-line interpreter in computing absolute software addresses based on recipe-block relative-address pointers) was being invoked about 2000 times per second. By moving this routine into microcode, the 150 instructions were replaced by one new

Assembly instruction. The previous 150 instructions were put into a microcode environment which operated with a faster clock, was “read only” (instead of slower “read-write” capability), and had many additional registers available to it for mathematics algorithms. This change, developed and implemented with about 15 person-days of effort, dramatically improved computer throughput.

Improvement in recipe size came about because of a concept known as memory mapping. Initially, the HP minicomputer contained only 32K words of memory, more than half of it required for the operating system, configuration, and other utilities. This left only about 12 K available to share among all the 5-10 fermentors interfaced to a Level 1 computer. With memory mapping, 12K of memory could be made available to each fermentor, since the operating system typically processed only one recipe at a time.

The 1980s

The system had proved so successful in late 1970s implementations, that the early 1980s saw its replication in all remaining major Lilly fermentation pilot-plant and production facilities, including one in England. Six plants and over 200 fermentors, ranging in size from 150 L to 200,000 L, were involved. Said another way, the business process in approving the first fermentation automation projects in the 1970s required significant time and energy in developing the economic justification. By the mid 1980s, this part of budget preparation was no longer needed. The benefits of automation were proven and fermentation automation was now a corporate expectation.

The availability of this system then prompted an avalanche of application development, new control loops, on-line analytical systems, semi-automated bioreactor sterilization and broth sampling (see insert photograph for a view of a modern-day fermentor). One example was the development of on-line process mass spectrometry in which fermentation gas streams were analyzed for oxygen, carbon dioxide, and other component concentrations. The Level 1 computer interfaced to these mass spectrometers, performed automated calibrations and contained the algorithms that calculated 10 new fermentation culture and bioreactor performance parameters, not previously available on-line to technologists. These included culture oxygen uptake, culture carbon dioxide evolution, respiration quotient, oxygen mass transfer coefficient, culture heat evolution, and estimates of cell mass and substrate concentrations. This on-line analytical system was replicated into all of Lilly’s major fermentation facilities, as were several other technologies associated with the overall automation system.

The dramatic increase in the number of parameters to be logged and the desire to keep the data available “on-line” as long as possible drove a recommendation to pursue data compression. This was accomplished by implementing the well known “boxcar backward slope” method within the Level 1 computer and then “pushing” the resulting data records up to the Level 2 computer for storage in the data

historian. This approach helped to minimize traffic on the inter-computer communication data highways while ensuring that all process control events were captured. It is interesting that the Lilly fermentation computer system differs from commercial vendor process control systems in that, in commercial systems, process data is “polled” by the data historian and compression exists at the location of the data historian. This “polling” approach has been known to sometimes overload data highways so that operations like updating operator consoles is significantly slowed down. Also, since data recording can be no more frequent than data polling frequency, a rapidly occurring event (e.g. a momentary power loss or measurement spike) can be missed. Lilly fermentation automation personnel still believe their approach is the better of the two methods.

Many other applications were also developed for the Level 2 computer, several of which involved “mining” the nuggets of gold (i.e. information and knowledge) out of the growing historical fermentation database. Selected enhancements included:

1. An electronic interface to other computer systems such as Level 2 computers at other plant sites and to the laboratory assay computer systems. e.g., The data in the lab assay computers were electronically copied, transferred, and then automatically reformatted and added into the Level 2 fermentation process data base. This permitted users to view their continuous process measurements (e.g., pH and dissolved oxygen) and their off-line assay results (e.g., penicillin) on the same plot, almost immediately following the completion of the lab assay. This increase in centrally-located, easily-viewed, fermentation data (even from home) allowed technical-service employees to be even more informed as they considered on-line changes to the process. It also facilitated post run-analysis of the process and root-cause analysis of any deviations.

2. An application to generate the sample bottle labels to be used for fermentation broth samples that were collected manually.

3. An application to periodically summarize the status of the manufacturing plant and produce a plant status report for each oncoming shift of operators.

During the late 1980s, the Level 1 process control system was upgraded to make use of commercial improvements in computer science technologies. This included shifting from the in-house developed operating system to HP’s RTE-A operating system. Data logging and other software utilities were rewritten from their original Assembly language into Fortran and Pascal. Coincident with this change was the addition of data buffers to provide temporary storage of process data in the event the Level 2 computer was unavailable. This replaced a previous, more cumbersome system where 3rd party Tandberg tape units were interfaced to both Level 1 and 2 computers to provide a manual version of this contingency data-capture functionality.

The process control (Level 1) system was interfaced to a large variety of “front ends”, which typically participated at some level in control-panel information,

operator interface, signal conditioning, and even control. In some installations, the Level 1 computer did it all, but more typically it was a shared environment. e.g, the Level 1's limited memory could not support a sophisticated operator interface, so complementary "front ends" often picked up some of this functionality. Examples of "front ends" included Foxboro Interspec, Opto, Allen Bradley PLCs, Moore, Honeywell, and Taylor single-loop electronic controllers, and Fisher Rosemount's Delta V system. Some systems used combinations of the above.

The 1990s

During this decade, the primary focus of upgrades involved the Level 2 data historian computer. During this time, inexpensive gigabyte disc-based data storage became available. This permitted keeping historical data immediately available for access and analysis for up to several years (vs. a few months previously).

The most important change came about in the recognition that engineers and technical service employees were spending great amounts of time reviewing data plots for currently running fermentations to ensure that they were running OK. They would analyze these plots with known mental "rules of thumb" – i.e., looking at plot noise, slopes, minimum and maximum values, points of inflection, etc. In many cases, these time-varying plots were manually compared to the same variables from historical runs. Most of the time, the fermentations were operating normally and so no corrective action on the part of operators, scientists, or engineers was needed.

Lilly decided to interview the plant operation experts (operators, engineers, and scientists) and capture their expertise in analyzing process information. This information was then organized into "if then rules" and other forms of knowledge. This was then coded into a real-time expert system (G2 from Gensym) and interfaced to the Level 2 data historian. The resulting system would look at all the data coming into the data historian (in real time), analyze it with respect to the now captured and coded "rules" in the expert system and make its conclusions available to CRT consoles and to paging systems. This saved nearly 50 man-hours of manual data review/plant/month. Manual routine data monitoring by technical service folks was dramatically reduced or discontinued. Instead, operations and support employees were alerted via a pager when the expert system determined an abnormal situation existed, and only then, began reviewing data and other computer generated results.

In the initial validation of this expert system, a novel Level 2 computer application was developed which allowed the playback of complete sets of data from historical manufacturing lots in a compressed time frame (e.g. a data set representing a 1 week fermentation could be serially fed, in chronological order, to the expert system in 10 minutes). It would then be determined if the expert system automatically drew

the same conclusions as technical service personnel manually reviewing the same data.

Coincident and complementary to these new applications was the evolution to a more intelligent remote alarming system. As with most industrial plants, Lilly fermentation plants originally generated many low-level alarms (i.e. by comparing a single setpoint to a single current value of a measurement), an approach that led to many nuisance alarms. By shifting most of the alarm system to the real-time expert system, logic could be created which simultaneously looked at many pieces of information and could, therefore, draw more intelligent conclusions about what was going on; then the system would page the appropriate person or plant utility area, depending on the specific situation. This would include paging scientists at home, if needed. This system resulted in a 67 % reduction in total alarms generated and eliminated most of the nuisance alarms. The ability of this system to immediately contact the right personnel as soon as an abnormal situation or a drift towards an abnormal situation was detected also ended up contributing to a several percent increase in plant yield and a double digit reduction in process variability. These benefits resulted from earlier detection of “poor” performing fermentations, giving plant personnel more time to intervene and “save” or improve many of them. The use of real-time expert systems for these kinds of applications was then published in the literature, received both internal and external vendor national awards, and was replicated into all of Lilly’s major pilot plant and production fermentation facilities.

Epilog

The success of the Lilly fermentation computer system has been the result of a great many factors, both technical and managerial. In the context of this article, the computer science and IT folks working on the project certainly point to:

- 1. Having worked with commercial vendors (e.g., HP) that have a long track record of business viability and longevity and that produce products that are leading-edge, flexible, and permit some customer customization (e.g. microcode, and interface to multiple 3rd-party products).**
- 2. Having worked with a commercial vendor that provides upwards compatibility with new computer models. Lilly retained this requirement in their own software development as well, so that throughout the life of the system (> 25 years), new system upgrades could always run previously developed applications, including process-control recipes.**
- 3. Having developed a system that is an almost perfect fit to the application area. The system met all customer requirements. Most commercial systems cater to many different application areas and so must compromise in many ways. No one customer group gets all it wants.**

4. **Having essentially identical systems in different plants, including the pilot plants. This has greatly facilitated fermentation process scale-up and product tech transfer activities and has minimized the number of support and development personnel required, corporate wide.**

From a managerial perspective, a key activity was an annual one-week workshop that brought together the key support employees from each plant involved with the fermentation computer system as well as representative customers. In this forum, key issues were discussed and resolved, and a development action plan was agreed to, with assignments made to appropriate individuals. In this way, an individual at a plant could work on a piece of software development that would not only apply to his/her own plant, but, because all sites used a similar system, could be replicated to all sites. This greatly leveraged the contributions of individuals and kept all sites on the same wavelength, working together as a corporate team. This also provided a day-to-day network where individuals could bounce ideas off one another and in which individuals at one site could help back up another site.

As the year 2000 approached, Eli Lilly and Co. found itself once again with a dilemma. DCS and PLC vendors had evolved their systems over the years so that they were certainly capable of handling fermentation processes. Further, regulatory agencies (e.g. FDA) were adding new expectations to the validation of computer systems used in the pharmaceutical industry (audit trails, electronic records and signatures, etc.) so that significant continuing computer system development would be needed to meet these new requirements (and other situations such as the year 2000 date-code issue). So, should Lilly do what Lilly does best (make medicines) or should they allocate valuable people resources replicating what commercial process control vendors can do. Put in these terms, the answer was clear, as Lilly is now slowly migrating to commercial automation systems in fermentation, a process that will take several years.

Regardless, this system represented a great example of how a core group of company scientists and engineers could partner with a reliable computer hardware vendor and create a system that could adapt to new technology, remain upwards compatible, and simultaneously add much bottom-line benefit to a company's operations. Its documented results include improved process control, significant fermentor productivity increases, process variability reduction, major capital (plant expansion) avoidance, accelerated product development time (due to greater understanding of the process by engineers and scientists), enhanced root cause analysis of deviations, and the saving of many production fermentations and pilot plant experiments that would otherwise have been lost. Indeed, the magnitude of these benefits was directly influenced by "the practice" of implementing new computer science techniques and technologies as they became commercially available.

The corresponding author for this article is:

Joseph S. Alford, Ph.D, P.E.

Engineering Advisor

Eli Lilly and Co.

[Alford Joseph S@Lilly.com](mailto:Alford_Joseph_S@Lilly.com)

Tel: (317) – 276-7653

Fax: (317)– 276-1403

This article was written in response to a request by CAST, following Dr. Alford's receiving the AIChE Computing Practice Award at the AIChE National meeting in November of 2003.