

# Software/Products in Process Control

E. H. Bristol

When I first worked in Process Control, Foxboro was a highly esteemed manufacturer of mechanical (with some electronic) process control instruments. It was beginning to enter the digital, and separately the systems, era. But mostly it understood what constituted high quality design and products (as sort of a specialized market, Process Control, Apple®).

Electronics stretched this understanding with the beginnings of producing (potentially standardized) things out of standardized parts. The evolution of the digital system and control software furthered the use of standardized functions, practices, and implementation, and seemed to signal an era of far more flexible application design.

But as the applications people were reaching more end application mastery, the original product oriented engineers and managers were faced with a new confusing technology, out of their control. Of course, the traditionalists had great difficulty with the whole concept of systems.

In a note at the time I argued that Systems was about applying Mass Production to One-of-a-Kind application problems. For Process Control, both disciplines centered<sup>1</sup> in software. And software itself was getting out of hand in the engineering world at large.

Applications work contrasts with product design in working with a standard practice that is relatively more standardized and relatively more quickly carried out. The corresponding attitudes are naturally compatible with managers, knowing

no better and desiring an efficient and predictable business.

Most process control businesses originate serving a particular application focus, tending to preserve the corresponding emphasis. Foxboro had overcome this, in time developing a body of designers and managers who appreciated the opportunities from high quality design and the different timing requirements of this activity.

Process Control is really not a business, but the interface between two potential businesses: the Processing business and the Controller business. Its participants, ideally, have to understand the processing, operational, and control technology issues, making it an opportunity and playground for individuals with the needed breadth.

But the whole is best served when these participants preserve their interfacing role. The resulting Process Control/Controller combination then constitutes an area with enormous opportunity for innovation and farsightedness, contrasted with the clunky hardware and blind copycat-ism seen at the trade shows and necessarily underpinning it. This opportunity is centered in software.<sup>2</sup>

Software is naturally implemented as a layering of functions. The challenge for Process Control is to recognize opportunities for layering appropriate software on top of the inherent clunkiness, that give the various levels of user real control over their function. Over the years, I saw several ways to recast a “Controller” business in terms of different levels of software:<sup>3</sup>

---

<sup>1</sup> For the traditionalists, a product based largely on someone else’s hardware and only proprietary software was confusing. But this represented an extension of mass production to low volume systems: first at the level of the hardware applied, using the proprietary software, to the low volume process control and the software configured to the actual one-of-a-kind application.

<sup>2</sup> Of course nothing prevents unique hardware or mixed hardware/software products where the distinctive hardware design augments novel software function. Like the iPod (or the Celestron SkyScope).

<sup>3</sup> The dependence of Process Control software on the process I/O has tended to restrict the basic software to be proprietary and deeply linked to a hardware product. I see this as analogous to the traditional Apple® approach.

- As special control technology design/implementation algorithms, EXACT® being a typical example. The opportunity here was a novel solution<sup>4</sup> to a recognized problem surpassing the flawed copycat standard solutions.
- As control configuration/implementation (language) systems. Initially these were proposed to model traditional analog or logical control diagrams. But there evolved hybrid combinations of different diagram forms or mixes of real time BASIC languages with block forms. I see the ICL/Idiom work as breaking away from such old models and providing a language system best fit to the application issues.
- The proposal for Alarm Analysis Tools represents a different kind of specialized extension of Process Control function to make life easier for the related user, through a novel rethinking of a traditional function.

A number of small control consultancies sell design software tool products as well, but these are distinct from the kind of large control supplier role that I intended.<sup>5</sup>

If software is key, trying to make high quality software play a large and strategic part in process control may seem quixotic. Highly sophisticated software like ICL and the Alarm Tools may seem beyond us.

Nevertheless, during the developmental digital era, Foxboro produced, through itself, and with consulting help, pioneering high quality control software: The PCP88 (mid 60s), besides being the first dual supervisory/control/redundant computer system, had a distinctive early real time FORTRAN.

The FOX1 (early 70s) had a pioneering real time disk-operating LP program and an extremely high quality real time supported FORTRAN. The FOX2-10 and FOX3 represented evolutions in two well-thought-of computer systems based on real

time control BASICS. Other Process “Controller” companies had analogous offerings.

So why can’t these companies do this now. First, this good stuff was mixed up with all the difficulties inherent in the software world of the time (and even yet). Second, an application driven world operates on the wrong time scale and business model for this kind of activity.

It expects projects to be done directly to a short plan in a comparatively short time with a team built for the project. Third, a management style, more remote from the “old days” and more attuned to the same organization favored by the application teams.

High quality products of any kind requires high level management involvement. It also requires engineering continuity, where the designers have the opportunity to develop their thinking over several evolutions of a given class of product, and time to “research” novel solutions. And support for any necessary specialists.

The old management knew this. In the 70s Foxboro had to learn this again after related mistakes. Remaining old-line engineers were able to exert influence which lead to this kind of mutual cooperation (as well as a fellows position).

More recently managements not so inclined, together with engineering groups inclined to keep management out of their world, created entirely different kinds of companies.

But EXACT®<sup>6</sup> would never have succeeded in the ‘80s unless we had a top management who not only learned to understand its benefits, but understand them well enough to appreciate them. As an external example, Apple’s success is clearly intimately tied to the connection of its top manager with its top technologists.

---

<sup>4</sup> But one which had been developed, and even field tested, twenty years before. Suppose that we had implemented it earlier.

<sup>5</sup> Microsoft shows that software doesn’t need to mean small.

<sup>6</sup> In the larger scheme of things EXACT may seem a small thing. But it had a very important role in reestablishing the company’s reputation for special expertise at a time when the “larger scheme of things” was breaking down.