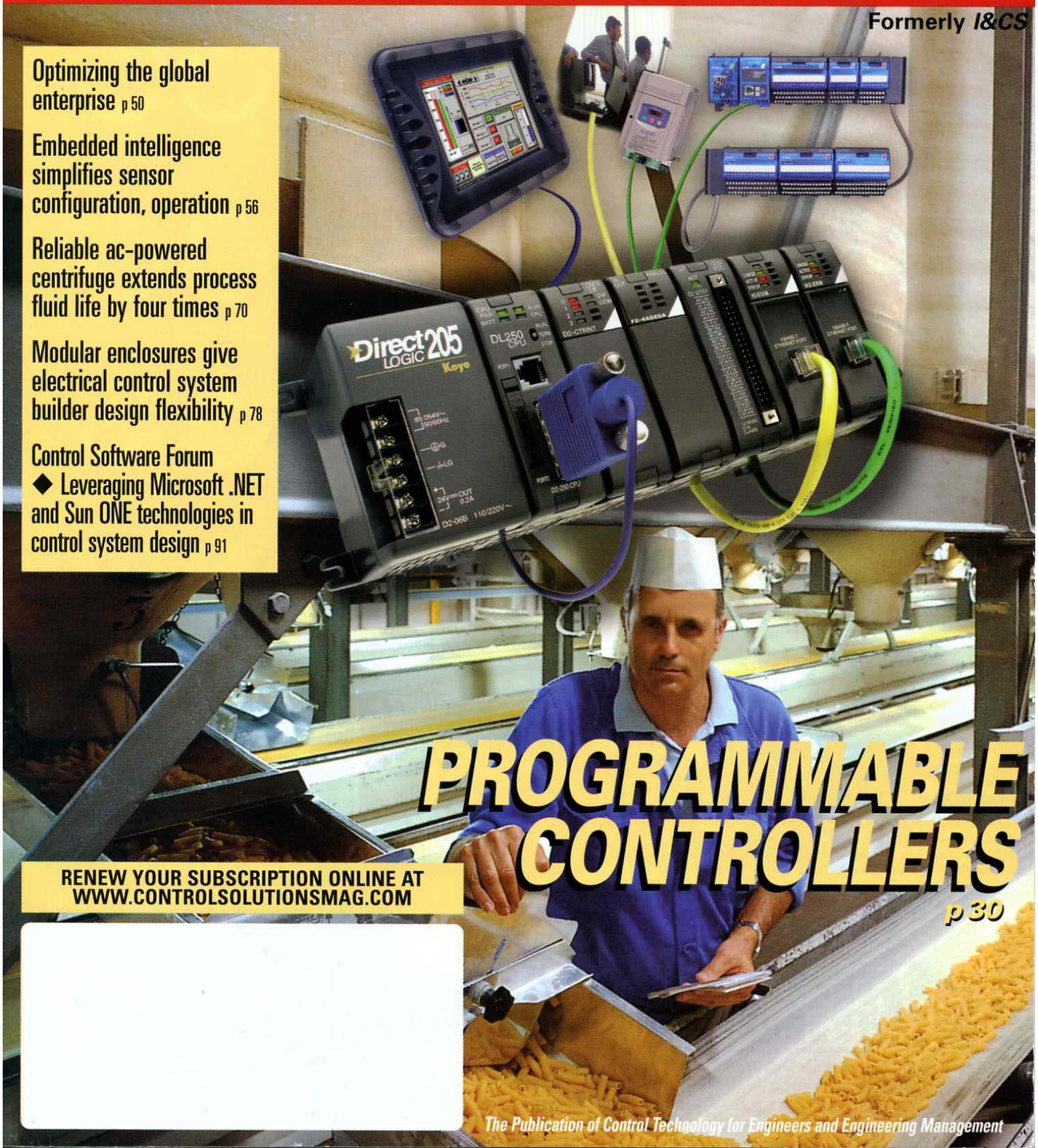
April 2001

PennWell

Control Solutions

Formerly *I&CS*



PROGRAMMABLE CONTROLLERS

The Publication of Control Technology for Engineers and Engineering Management

Leveraging Microsoft .NET and Sun ONE technologies in control system designs

Derek Spratt, P. Eng
Chairman & CEO,
Intrinsyc Software, Inc.
Vancouver BC V6C 1G8

In January 2000, Steve Balmer became Microsoft's CEO and proclaimed a new era for the company and its customers based on a *software-as-a-service* (or application service provider [ASP]) business model. This was a bet-the-company strategy for the world's largest and most successful software firm. While most control engineers, who largely are involved in real-time programming projects, probably skipped Steve's inaugural address, they should have taken notice when five months later, Microsoft .NET™ was born. In February of this year, Sun responded to Microsoft's market entry with its own ASP architecture called Sun ONE™. Collectively, the titans of the software industry have weighed in with all-encompassing platforms for *anywhere* and *anyplace* computing. And yes, all of this will have a profound affect on control system designs.

Both the Sun and Microsoft business models are based on the XML communications protocol, and, along with open standards such as SOAP and Java, create something called *web services platforms*. The applications that developers write, the services they wish to provide for their target environment, and the embedded devices that form the end-points of the network they will be deploying, will all work together as a platform to provide access to the information their customers need at anytime and anyplace via the Web. It may sound like a consumer technology, but it is really the first comprehensive architecture for the post-PC era, and it will ultimately impact control systems as much as 32-bit OSs like Windows, UNIX, and Linux have today.

Enterprise, embedded systems converge

These web-services-platform system architectures are the blueprints for tomorrow's seamless enterprise-to-embedded system networks. A key requirement for many next-generation control system devices will be the incorporation of networking and network management technologies that are compatible with Microsoft .NET and Sun ONE platforms. The very basis of these technology thrusts is to address the issue of how to best connect smart networkable devices into Internet-based enterprise systems.

In the last few years, the traditional proprietary world of embedded systems design has been moving rapidly towards open standards, and control system design, in particular, is an important beachhead for this revolution in networking beyond the PC. Control systems are being targeted by Microsoft and Sun at the outset of this ASP computing revolution.

The initial leverage of desktop PC and enterprise technologies and standards into the embedded space was driven mainly from a cost-of-development perspective. Code-reuse and sourcing of third-party solutions surrounding time-to-market dynamics drove most of the demand. But increasingly, communications are at the heart of next-generation device design decisions as they are plugged into the larger world of computing.

Internet and Moore's Law are key

This Microsoft .NET and Sun ONE end-to-end convergence of enterprise and embedded computer systems is mainly driven by the pervasiveness of the Internet in business applications, and the relentless progress of Moore's Law.

Moore's Law has enabled the cost-effective use of powerful, highly-integrated, embedded 32-bit CPUs and off-the-shelf 32-bit OSs, such as Windows and Linux, to replace proprietary OSs that were tailored to the constraints of previous generation 8- and 16- bit microcontrollers. With the same computing horsepower under the hood of next-generation embedded devices as their enterprise-based cousins, there will soon be no technical barriers to allowing smart devices to be intelligent and communicating equals in tomorrow's ubiquitous computing paradigm—which means that they must be architected to work as part of the ASP paradigm.

Challenges in control systems

In some important respects, control system developers have more significant communications challenges to tackle than what has been required of desktop PC or enterprise systems developers. Ultimately, control system designs must include

additional communications intelligence to perform auto-discovery of their network service/function and self-installation, as well as remote management capabilities that allow for automatic software updates, network-based fault tolerance, remote monitoring, and control.

Networks of desktop PCs and servers at the enterprise level have always been accompanied by small armies of IT professionals to help with their installation and support. In control system environments, there are not only many more potential network nodes to install and support, but the network installation must also address the demands of five 9's reliability, fault tolerance, and remote installations—which are concepts generally foreign to the desktop PC environment. These are some of the thorny issues that ASP technology vendors and ISVs are aiming to address with a range of new products that are under development for Microsoft .NET and Sun ONE platforms.

Total cost of ownership

Perpetrating yesterday's desktop PC-based IT management practices (or worse, a proprietary approach) in a next-generation control system environment is far from an ideal strategy. The total cost of ownership (TCO) of developing, deploying, and maintaining networks of smart devices must be addressed when designing any new control system. One of the best ways for developers to ramp up quickly is to contact ASP technology vendors and ISVs to discuss their plans and needs.

When developers review their next generation-control systems requirements, they should try to ensure that everything from basic software installation to configuration and maintenance is network based and automated. Also they should consider adding the ability for a device to auto-install itself on the network and intelligently link into enterprise data management systems. Last but not least, they should seriously consider adopting XML and SOAP or Java technologies as well. This will help keep them and their customers *future proof* in the coming ASP computing revolution.

Bridging technologies

There is no certainty that a single dominant technology will unify the embedded and enterprise worlds despite the recent introductions of Microsoft .NET and Sun ONE platforms. Java and Enterprise JavaBeans (EJB) hold critical mass with many ASP-based enterprise systems, and UNIX and Linux have a significant and growing installed base on servers, Windows completely owns the desktop PC space, and at the embedded and control system level, there are many proprietary legacy OSs and protocols that must continue to be supported. What is a developer to do?

TCP/IP, HTML and XML are likely to remain core technologies far into the future, and both Microsoft .NET and Sun ONE platforms extensively leverage these networking standards. Common distributed computing protocols such as EJB, DCOM and CORBA are also projected to remain popular. For these reasons, deploying third-party bridging technologies that can provide the necessary translation between competing platforms and protocols, and the use of other agnostic platform-independent networking protocols, should form an important part of control system designer's wares as the ASP platform battles are fought for dominant market share.

A call to action

Control system OEMs and developers face a significant risk by waiting on the sidelines during this ASP revolution. Yes, there will continue to be uncertainties in terms of which technology, or combination of technologies, will form a dominant position in this space, but the likely result will be a hybrid—that is, a mix of old and new, and an acceptance of the concept of designing-in support for major networking, network management and distributed computing paradigms in new product offerings.

There are very real benefits to embracing this revolution for OEMs, developers, and their customers. The ASP model isn't just promotional spin, it's a critical step forward in the journey towards an end-to-end computing reality.

About the author

Derek Spratt founded Intrinsyc Software, a developer of networking and network management technology solutions for OEMs, in 1996. He graduated from Queen's University in 1983 in electrical engineering, and has held senior management positions with Nexus Engineering and Motorola. He can be reached at dspratt@intrinsyc.com.