e-Automation: Endless Possibilities

By: Cynthia M. Hollenbeck, Vice President SoftPLC Corporation

ABSTRACT

The successful manufacturing companies of the future will be those who implement Enterprise Control systems which encompass all levels in the organization - from the factory floor processes to the systems which service suppliers and customers. Control systems at each level are changing to meet users needs and to embody the tremendous base of technology being provided by the computer and software vendors. One technology that is a true enabler for Enterprise Control is the Java™ language, developed by Sun Microsystems, Inc., particularly in the need for data sharing between the different levels in the organization as they implement Enterprise Control systems.

This paper describes why a Java enabled factory automation controller can be the main component of a successful Enterprise Control system, by serving the needs not only of a single factory, but the entire organization, its e-commerce functions, and more. This web enabled controller can be accessed remotely to provide realtime information throughout the world, for reduced maintenance costs, or in situations where the controller is installed in hazardous/remote areas.

TABLE OF CONTENTS

© 2001 SoftPLC Corporation

INTRODUCTION

Traditionally, the structure of a manufacturing organization, whether it be a discrete manufacturing or batch/continuous process industry, can be thought of as being divided into three levels:

Level One

Direct interface to and control of process devices (I/O). Examples of controllers at this level include robots, drives, numerical controllers (CNCs), programmable logic controllers (PLCs), or other instrumentation.

Level Two

Devices whose purposes are to (a) make Level One controller operations more easily visible and accessible to machine operators, and to (b) extract realtime production data from Level One controllers for analysis, reporting, etc. Examples include operator interfaces, data acquisition systems, supervisory/cell control systems (DCSs/SCADA).

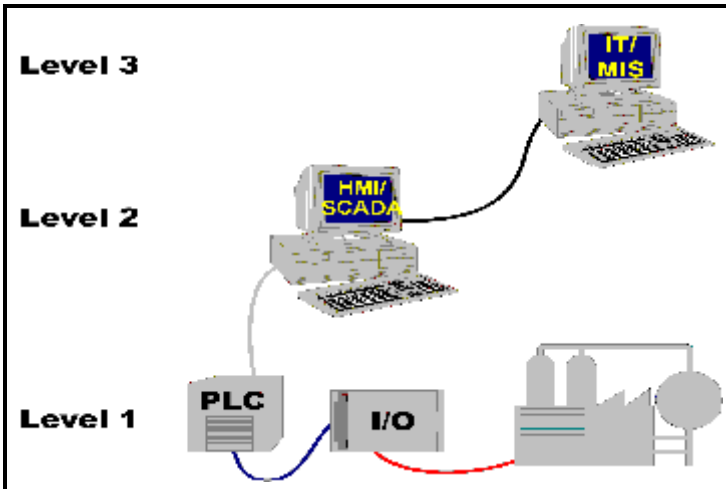
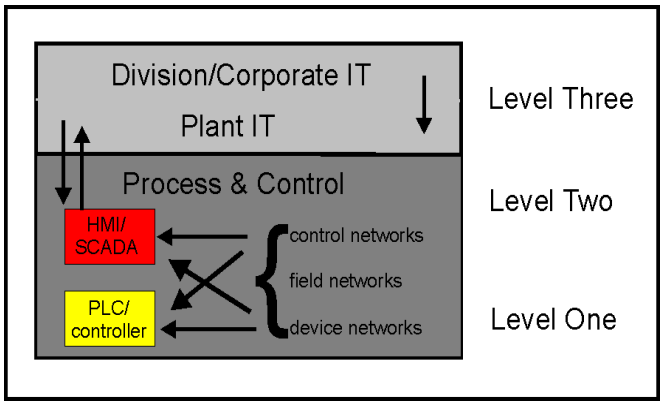


Figure 1
Traditional Controls Hierarchy of Manufacturing Organizations

Level Three

Business functions such as accounting, production scheduling, which are typically run on mainframe computers maintained by Information Technology (IT) or Management Information Systems (MIS) departments.



Each Level has been the responsibility of different groups in the organization, and exists separately due to different requirements. Business operates through the interaction of devices at each Level of the organization. The smoother and better the interaction, the better the business operates.

Figure 2
Business operates through interaction of Devices at each Level

© 2001 SoftPLC Corporation

LEVEL ONE CONTROLLERS

Level One controllers are installed on the factory floor manufacturing area and are subject to the harsh environments often found there. These systems are normally maintained by factory floor maintenance groups. From the original hand assembly processes to today's sophisticated microprocessor based automation systems, the requirement has always been that the systems must operate reliably in the environment ("24/7" uptime, 0% downtime), be easy to maintain, and provide deterministic, realtime control of the I/O. For these reasons, Level One controllers have traditionally been specialized proprietary devices.

Vendors of Level One controllers, such as PLCs, have all independently developed their proprietary technologies, rather than sharing development efforts or building to standards. Therefore, release of technology advancements have been slow, especially when compared to consumer markets (eg: audio/visual electronics).

Additionally, the proprietary nature of the controllers means that the valuable factory floor data is not readily available to the other Levels in the organization. A single vendor has supplied not only the controller, but the I/O interfaces, the networks, and often even the Level Two controllers. Users of these controllers become locked into the vendor for all their needs, as the cost to change vendors or use multiple vendors simultaneously is high. Other limitations of proprietary controller include:

1. Proprietary networks require custom device drivers to access data
2. Fixed instruction sets
3. Limited I/O structures and other device interfaces
4. Limited choice of equipment from other vendors that can be easily interfaced (eg: for a PLC, a motion control system)
5. Difficult to extend capability without high cost (eg: adding memory capacity)

LEVEL TWO CONTROLLERS

Initially, Level Two controllers consisted of simple operator interface devices such as pushbuttons, LED displays or thumbwheel entry systems. As Level One controllers became more sophisticated, Level Two controllers have become extremely important to organizations as a means to interact at the factory floor with the Level One controllers, and also to extract the data from the proprietary controllers for use within the organization. This process required custom hardware and device drivers for the Level One network protocols known to automation engineers, but not IT personnel.

Level Two controllers sometimes are installed on the factory floor for machine operators to access Level One controller status or to modify operating parameters during production. Level Two controllers are also connected to the company network to send production data to Level Three systems. Based on the sophistication of the Level Two device, there may be sub-levels of proprietary HMIs and packaged software products within Level Two.

© 2001 SoftPLC Corporation

LEVEL THREE CONTROLLERS

Level Three systems work with the data necessary to develop production schedules (ie: sales orders, inventory levels, manufacturing capacity) and for accounting functions. The production requirements need to be communicated to Level One and the status of production at Level One needs to be communicated to Level Three. Successful organizations have found ways to accomplish the necessary communications.

TRADITIONAL CONTROLS HIERARCHY IS OBSOLETE

In the traditional hierarchy as described here, there are tremendous technical barriers to efficiently communicating data between Levels One and Three, and so the same information often exists separately at all three Levels! Level Two has usually been the "translator" between Level One and Level Three. Some of the problems that exist with this structure include:

1. Control systems at each Level must be maintained in parallel, and coordination between various departments is required.
2. The amount of data available between Levels is limited due to the difficulties in implementing and maintaining translation systems.
3. The initial development of systems and later modifications are expensive in terms of time and engineering costs.

With the many advances in transportation, communications technology, and the "global economy", traditional business models no longer are sufficient. Manufacturing organizations are rising to the challenges presented by these advancements, and making changes. These companies are thinking about integrated *Enterprise Level Control Systems*, rather than simply translating between the separate control Levels found in the traditional hierarchy.

At the same time, manufacturing procedures are also changing. Just-In-Time (JIT) manufacturing is being implemented to reduce inventory costs and the required size of manufacturing facilities. For JIT manufacturing to be successful, it is critical that the organization have realtime information flow between sales, purchasing and production.

Companies are also implementing strategies at Level Three to better manage resources and information flow, and to be positioned to react quickly to the changing business climate. Level Three systems which address these needs include e-commerce support, Supply Chain Management (SCM), Enterprise Resource Planning (ERP), Manufacturing Execution Systems (MES), decision support, data warehouses, and other database applications.

Also, we have adopted an "I need it now and I want it to be red" philosophy. Customers are no longer content with long delivery times and limited choices. Traditional production scheduling methods are becoming obsolete as businesses throughout the supply chain are implementing JIT and E-commerce themselves. The traditional hierarchy of information flow introduces delays in processing transaction requests that are unacceptable in today's world.

BRIDGING BARRIERS TO INFORMATION FLOW BETWEEN LEVELS

Traditionally, automation Control Technology (CT) focused from the machine operator DOWN to the manufacturing process while Information Technology (IT) focused from the machine operator UP to the business processes. Control systems for each Level were designed separately by different vendors in different industries,and implemented separately by different departments in the manufacturing organization. Also, the technologies used by each group were vastly different. Control Technology utilized proprietary controllers (eg: PLCs) whereas Information Technology utilized software programs on mainframe computers. Control Technology utilized proprietary networks such as Data Highway and Modbus whereas Information Technology uses TCP/IP Ethernet.

Getting data to and from the factory floor to the business level has always been difficult and expensive. This inability to access data directly from Level One in part explains poorer-than-expected performance of Enterprise Control systems that have been attempted to date. Level Two has become a middle ground, or “no-man’s land” used by both CT and IT trying to make the information transfer possible, while retaining the traditional separation of responsibility.

In recent years, differences between CT and IT have started to be bridged. Level One support of Ethernet as a control communication network, in place of proprietary networks means that standard network technology can be used throughout the facility. This means transferring data to Level Two becomes simpler and more cost effective, and eliminates specialized hardware interfaces and many of the custom device drivers formerly required.

At Level Two, PC’s running SCADA/HMI systems can send data from the factory floor directly to Level Three computer programs without the need for custom software. Technologies that have aided this change include:

1. Common database formats - SQL, Oracle

2. Open systems platforms - Today’s extremely powerful PC’s have replaced mainframes performing many of the Level Three functions, allowing common hardware and software between Levels Two and Three. Also, software interfaces such as DDE, OPC/DCOM, and CORBA are making data transfer between Levels Two and Three easier.

3. Client/Server technologies - The ability to distribute functionality across a network of computers has reduced the maintenance, complexity, and cost of previous systems. File servers, print servers, database servers, and communications servers all help to keep resources centrally located and available to all users, which greatly reduces chances of error in duplicated data existing in multiple systems or that data won’t be backed up properly.

PC-BASED CONTROL AT LEVEL ONE

Computer based Level One control was another logical outcome of the tremendous advances in computer hardware and software technology. Just as at Level Two, the use of a PC at Level One makes a great deal of sense. PC-based controllers allow CT to reduce cost and complexity while increasing performance, by moving away from proprietary devices to "Open Architecture" systems. Benefits of PC-based controllers include:

1. A single vendor is no longer required to supply controller, I/O interfaces, and networks. Users can easily integrate components from multiple vendors, selecting the "best of class" for their machines from each category product.
   a. A wide variety of standard I/O networks have evolved, with products being available from hundreds of vendors in support of the I/O network. Examples include Profibus, DeviceNet, Interbus, and Fieldbus.
   b. Standard ethernet networking is supported.
   c. Motion controllers, drives, robots, and other equipment can be easily connected to these standard networks.

2. Controller capabilities are no longer fixed.
   a. Controllers support multiple programming paradigms, not a single proprietary language.
   b. Users can develop their own custom functions.
   c. Virtually "unlimited" memory and disk capacity, and later expansion can be done by the user simply adding computer memory or disk space. This allows controllers to support today's requirements for increased data storage and larger programs.
   d. Controller performance specifications can be changed by the user through selection of CPU speed, memory capacity, communications ports, etc. based on system requirements rather than working within the parameters of what the vendor provides as in the past.

3. Users can select hardware based on environmental conditions (eg: temperature, dirt). Desktop PCs can be used for laboratory or clean room environments, industrialized PCs can be used for standard manufacturing conditions, and military specification or NEMA rated PCs can be used for extreme conditions, outdoor installations, or hazardous locations.

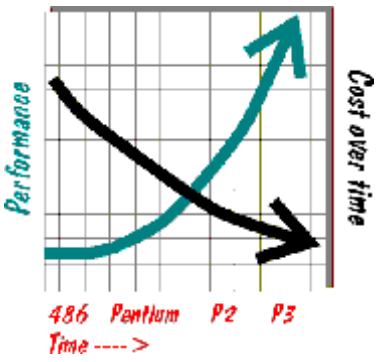


Figure 3
Control Technology Performance vs Cost Comparison

Just as in the desktop PC market, industrial PC prices are decreasing while at the same time computing power and capacity are increasing. The PC market is extremely large, especially when compared to the Level One controller market.

a. Lots of competition between PC vendors means lower costs of equipment, with good customer service.
b. Shared technology standards means new technologies are released faster through competitors building on each others advances, such as in ethernet networking, CPU speed, disk technologies.

4. The most important benefits of PC-based control are in networking and data exchange capabilities.
   a. PC controllers use standard networks as opposed to proprietary networks and have the ability to quickly take advantage of new technologies (eg: 100MB and soon 1GB ethernet).
   b. Networking cost is reduced by removing proprietary hardware and software. For example, even when they do support standard network interfaces, proprietary controller vendors charge an average of 5000% markup on network interfaces and equipment!
   c. Common cabling and hubs can be used. Separate physical networks maintain desired security and high speed required at each Level. Network bridges easily interconnect factory floor networks and IT level networks.
   d. Communications between Level Two and Level One are also much simpler and less costly. Without memory or computing power limits, more realtime data can be managed by the Level One controller, then passed to Level Two computers, rather than burdening Level Two with the entire task.

Once the Level One controller becomes an open architecture device based on computer industry standards, other exciting opportunities for change become available:

a. Direct communication between Level Three and Level One becomes possible, as the systems now share common/compatible data file formats, and data exchange layers within the software make custom device drivers unnecessary. In fact, Level Two becomes unnecessary for data translation functions, and can become simply a window to the process for machine operators or production personnel.

b. Realtime data can easily be made available to remote programs through the internet or company intranets for MES and E-commerce functions.

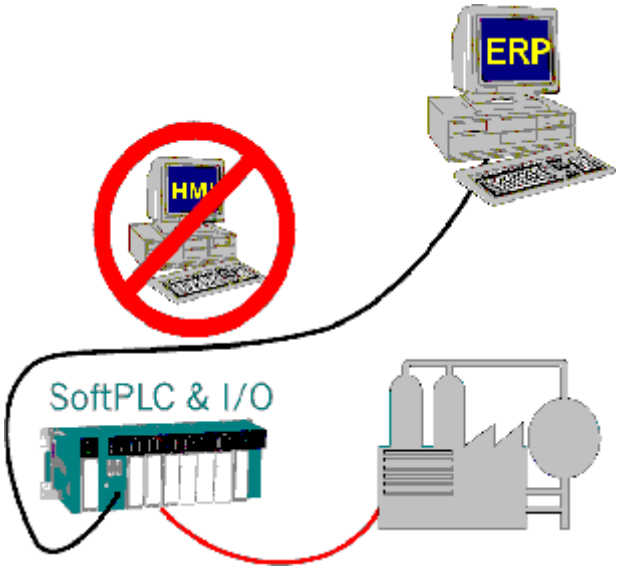


Figure 4
Translation of data between Levels One and Three can become unnecessary with PC-based controllers

JAVA™ AS AN e-AUTOMATION ENABLER

A key enabling technology for the software industry as a whole has been Java™. Java is a language, a computing platform used to build programs. Java language programs are built from compiled source code which is then run by an "software engine" called a Java Virtual Machine (JVM). The Java language has a number of benefits over other languages:

1. Built-in Client/Server support, with zero-cost client administration.

2. Powerful built-in communications capabilities. Java programs are "Network Ready", including support for TCP/IP based intranets and the Internet.

3. The compact size of Java program code also supports its use in both client/server and network based systems.

4. Supports native database connectivity via JDBC (Java DataBase Connectivity) to SQL, Oracle, etc.

5. Java offers multi-platform support so code is "Write-once, Run-anywhere". This means programs do not need to be re-engineered to run on different hardware systems. The same source code can be used on PCs, Macintosh systems, mainframes, and other systems, thus reducing development cycles, ensuring code consistency across platforms, and makes the programs available to a wider audience.

6. The Java language allows software vendors or internal development resources to deliver solutions faster. Java program developers can achieve higher productivity through features such as incremental compilation, a large number of available third party source code classes, and a huge "open source" code base. In addition, there is tremendous support within software development communities, particularly at the IT level and companies such as IBM, HP, Sun, and Borland for Java development tools.

As stated earlier, IT has evolved in recent years to meet changing customer demands. Corporations have been scrambling to update their systems by implementing MES, ERP, PDM, SCM and E-commerce. Implementation of these systems has been made possible particularly due to the Internet and Java language.

Current uses of this technology exist at, or between Levels Two and Three. Extensive use of Java has made these sophisticated IT systems easily possible. Examples include:

1. Interoperability between packaged software programs
2. Remote viewing control
3. Information wrapping
4. Intelligent agents for distributed intelligence
5. Web browser based data access via internet/corporate intranets

© 2001 SoftPLC Corporation

Level Two systems are also evolving through these technologies to meet changing customer demands. Some common examples are found in operator interface systems. Through client/server technology, a client operator interface (via Intranet or Internet) can display Level One realtime data. The Level Two system is on the TCP/IP network, and connects to a number of Level One controllers on a separate network, whether it be ethernet or a proprietary network. Web based video cameras ("Web-Cams") allow remote viewing of live video over the network in Level Two program screens.

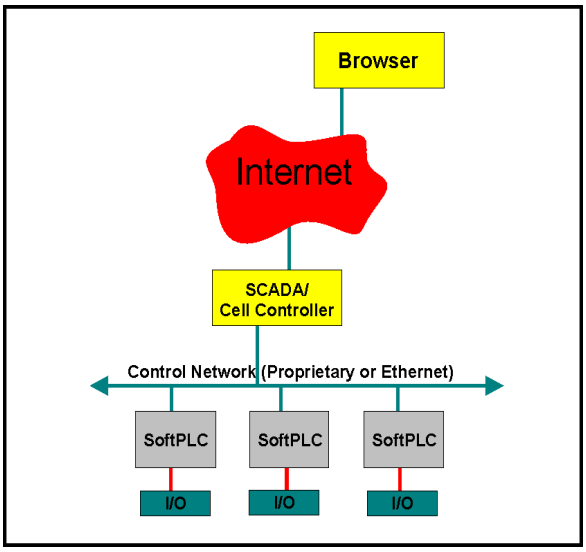


Figure 5
Level Two Systems can provide realtime data over Intranets/Internet

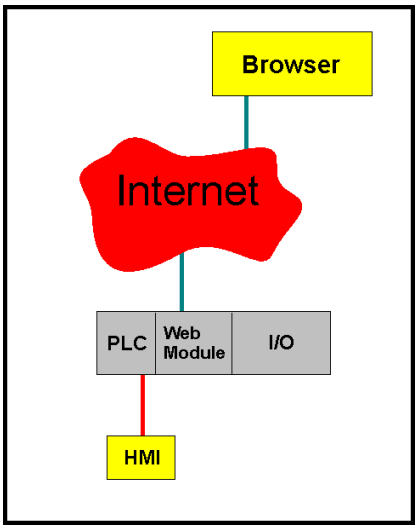


Figure 6
Some Level One Systems provide realtime data through "Web Interface Modules"

Level One control technology is also evolving due to Java and Internet technologies:

1. Web interfaces now exist for Level One controllers.
   a. Some proprietary controller vendors offer web interface hardware modules that are http servers. These can be programmed so that realtime data can be made available to an HTML page, which allow a web browser to be used as an operator interface.
   b. Some PC-based control vendors offer embedded web server capability in their control software to perform the same function.

2. Some PC control vendors combine Level One and Level Two functions on the same PC. These products offer the functions of both levels, usually with a common database. Using similar technologies to the previous Level Two translation functions, these systems can exchange data with Level Three systems. However, there are serious reliability concerns regarding using desktop computer hardware and operating system platforms (usually Windows) as Level One controllers, which as stated earlier, need to have zero downtime.

3. SoftPLC is an open architecture controller that is leading the evolution of Level One control through the use of Java technology. SoftPLC includes a JVM, which supports Java programs running in the SoftPLC controller, or for servicing Java programs running in another system on a network.

   a. An embedded web server is one available program.

   b. SoftPLC's JVM supports requests from remote Java Applications on clients through a Java Application Program Interface (API). Example uses of this technology are described later in this paper.

   c. SoftPLC overcomes the objections to PC-based control through its internal architecture. SoftPLC runs on a dedicated CPU with minimal hardware requirements for hardware reliability, and utilizes an embedded Real Time Operating System (RTOS) for software reliability (eg: no Windows).

   d. SoftPLCs software architecture and capabilities provides tremendous connectivity between SoftPLC at Level One, and Level Two and Three systems.

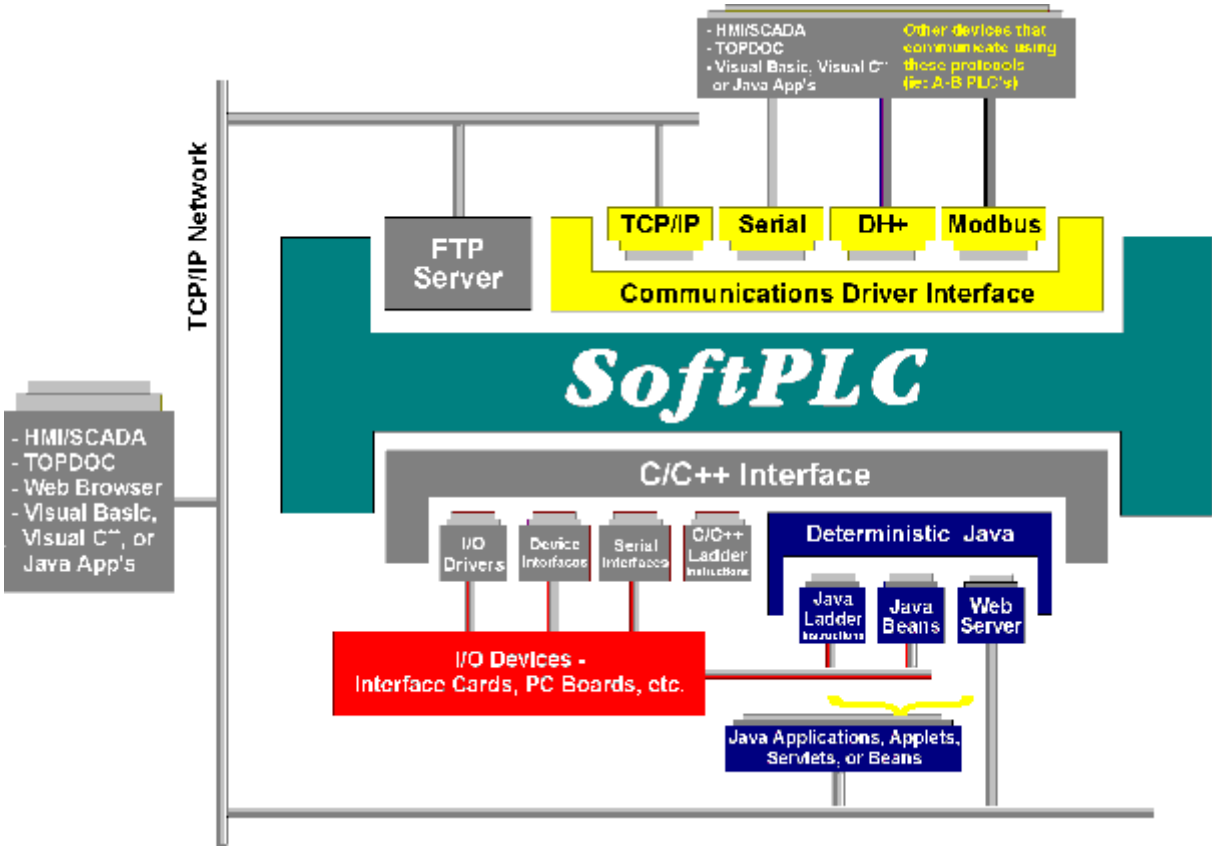


Figure 7
SoftPLC Internal Software Architecture

© 2001 SoftPLC Corporation

CLIENT-SIDE JAVA VERSUS DETERMINISTIC JAVA

However, Java was not initially designed for realtime, deterministic Level One systems but instead for Level Two and Three client-side programs. Therefore, subset versions of Java, such as Embedded Java and Deterministic Java were created. These reduce the bulk of full client-side Java, removing the Abstract Windowing Toolkit (AWT) classes and using realtime "garbage collection" schemes (a garbage collector is Java's way of managing computer memory allocation and recycling unused memory blocks). Deterministic Java is ideal for Level One systems as it provides Control Technology vendors the technology needed to provide reliable, real-time control and connect to Enterprise solutions:

1. Small memory requirements mean programs will run on small, embedded hardware systems.

2. Powerful error/exception handling allows program developers to identify and eliminate bugs early in the development cycle, and other features of the language prevent memory leaks and protection violations common in C/C++ based programs.

3. Built-in security and client/server support makes it ideal for operations in a distributed networked environment.

4. Java language is a vehicle which allows direct data exchange between Level One and Level Three through the built-in connectivity of Java DataBase Connectivity (JDBC).

SoftPLC utilizes a deterministic JVM, which does incremental garbage collection, making any Java tasks it runs deterministic. Additionally, SoftPLC itself only allows Java tasks a user specified slice of time during its control scan, thus ensuring the controller performance is also deterministic.

JAVA TECHNOLOGIES SUPPORTED BY SoftPLC

Manufacturing organizations can use the Deterministic Java technology, as found in SoftPLC, in many ways:

1. CT programmers can create their own Function Blocks (instructions) that run in-line with the ladder logic control program. Java language's native capabilities are often a better programming paradigm than traditional control languages (such as Ladder Logic) for some tasks, such as data manipulation or calculations. Custom function blocks, which look like a ladder logic function to the operator, allow automation designers to encapsulate special functions so that factory maintenance personnel can use the function, but can't modify the algorithm. A minor disadvantage of this feature is that the algorithm, as a compiled block, can't be updated without restarting SoftPLC, whereas true ladder logic provides for online programming changes. An example function block is the "Send Email" instruction, which can send an email message based on process conditions, embedding current realtime data into the message text.

© 2001 SoftPLC Corporation

2. Modlets are Java programs that run in SoftPLC independent of the ladder logic control program. There are two primary reasons CT and IT personnel find Modlets useful:
   a. Java language's capabilities are a better programming language than control languages (such as Ladder Logic) for some systems. Through Modlets, different paradigms can be implemented, such as State Logic.
   b. Often there is a need for non-process event driven tasks related to data handling or calculations that are better separated from machine control logic. Examples include: (i) an embedded web server and (ii) production scheduling and performance data transfer between SoftPLC and Level Three database. In the latter example, this data transfer is not the responsibility of factory floor electricians and the logic that reads inputs or controls outputs is not the responsibility of the IT department - so keeping the two types of software separate helps maintain the system integrity.

3. Servlets are Java programs that run in a web server. Servlets are tightly integrated with the web server task and can provide dynamic web page content. Servlet code is clean, object oriented, modular, and amazingly simple to develop. Servlets also provide for data security, which allows restriction of realtime data access to only selected users and/or selected data registers. This security is extremely important if the Level One control system is to be made available on the Internet. Other advantages to Servlet technology are:
   a. Servlet invocation is highly efficient. Once loaded, Servlets stay in RAM.
   b. Code can be updated remotely without restarting SoftPLC.
   c. Nothing needs to be pre-installed on the client computer (other than a browser).

In addition to the above SoftPLC resident functions, SoftPLC's Java API RemoteDatatable Class can be used to provide realtime data to Java Applications executing on client systems. With these Java Applications, data transfer can be extremely fast (eg: 2000 values per second).

1. Java Applets are Java programs that are loaded from a web server, then run in the client web browser. An advantage to Applet technology is that there is nothing to pre-install on client computer - the applets reside in the SoftPLC and are loaded over the network by the browser when the web page using them is invoked. Applets, although they take a "relatively" long time to load the first time, can then provide extremely fast updates (eg: can read or write 2000 values per second). A good example of the use of Applet technology is when a web browser is used as an operator interface.

2. Java Applications are programs which are pre-installed on a Level Two or Three computer. Java Applications are also supported via SoftPLC's Java API. They load faster than Applets, and do not need to run in a browser. Java Applications are not confined by a browser's security manager and have full access to the realtime data in SoftPLC. Examples include operator interface (HMI) screens or programs that transfer data between a database in a Level Three system and SoftPLC.

© 2001 SoftPLC Corporation

Running the Java program in the client system, in contrast to running it in the SoftPLC itself as described above, offloads CPU resource requirements from SoftPLC to the client computer. This may be important in machines needing very high speed control, or when the data from many controllers is being used in the same Java program. This technology, under SoftPLC client-side Java, uses a non-secure protocol, and requires a live permanent TCP/IP connection, which most factory IT people do not want to allow through their firewalls. This lack of security can be a disadvantage in some situations, therefore, client- based Java technologies are more easily implemented on Intranet versus Internet enabled control systems.

EXAMPLE SoftPLC e-AUTOMATION INSTALLATIONS

1. Extremely popular is the use of a web browser as a Level Two system. Previously, Level Two systems needed to be purchased for each workstation where the realtime data was to be made available. Additionally, development of web pages can be done by almost anyone without the specialized training required by traditional Level Two systems.
   a. Realtime production information can be available via web pages to anyone on the factory network, allowing financial managers to see summary data, production managers and maintenance personnel to remotely view machine status, etc.
   b. Operators can easily send production parameters to the controller through a web page for functions like recipe selection, enter new setpoints, etc.
   c. Full spreadsheet capabilities (eg: Microsoft Excel functions), including graphing, are available in client/server based spreadsheet products. The Server-side resides in the SoftPLC as a Servlet, which performs the live data retrieval and formula calculations. The Client-side resides in the browser system and is used to display the realtime data, allowing the user to move around the spreadsheet, etc.

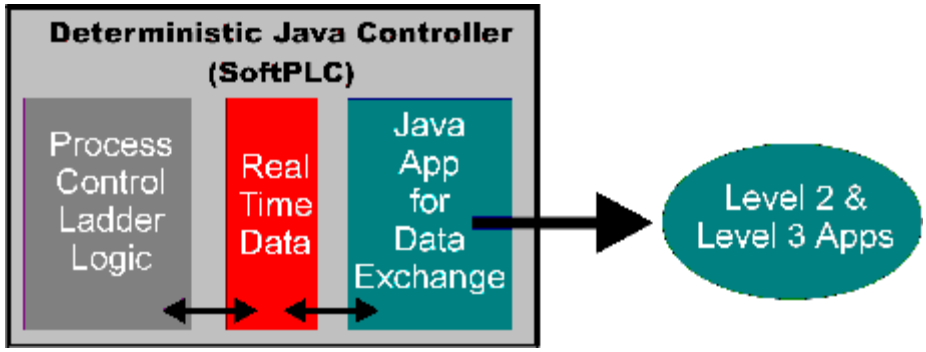


Figure 8
SoftPLC can provide a direct link between the realtime factory floor data and the Level Three Systems

© 2001 SoftPLC Corporation

2. SoftPLC can provide a direct link between its realtime factory floor data and the Level Three scheduling, accounting, and analysis systems. The controller logic can even be segmented so the manufacturing logic and the data exchange logic are separate and can be easily managed by both CT and IT personnel.

    a. SoftPLC logs data during processing, and periodically sends it to an SQL database in another department, eg: accounting or research engineering. Shareware SQL interface software that uses JDBC is available.
    b. SoftPLC can read production requirements from Oracle database scheduling system and select recipe and quantity to run for each product, and reports current status back to the production monitoring system, again using JDBC.
    c. SoftPLC sends realtime data via a proprietary protocol to Level Three computers. Custom protocols are easily implemented in Java.

3. "Send Email" capability makes it easy for SoftPLC to initiate data exchange based on process conditions.

    a. Production and uptime/downtime reports can be emailed to factory managers at the end of each shift.
    b. Process alarms can be sent to alphanumeric pagers or cell phones of maintenance personnel or production managers.
    c. Machine operators can push a button that causes an email message to be sent to a forklift driver's display which indicate the parts needed at that station.

4. SoftPLC can also receive commands and/or data from remote locations anywhere in the world via email messages. Simple text or numeric messages, or more sophisticated commands using technologies such as XML, can change SoftPLC's operating data. For example, a remote site could send new contents of a recipe, a production schedule, the specific data registers to be logged or other such non-realtime data.

5. Placing the SoftPLC controller on an intranet or the Internet for remote maintenance, remote status access, e-commerce functions offers a number of benefits not previously possible without tremendous cost.

    a. An operator interface based in a web browser, can be remotely located from the controller for installations such as utilities, pumping stations, pipelines, offshore drilling, mining, or in hazardous areas.
    b. e-Commerce example: An OEM (Original Equipment Manufacturer) has an Internet link to the machine's SoftPLC which allows them to access how many hours a machine has been run and for which type product, then automatically generates a sales order to ship his customer replacement parts, along with an invoice (eg: blades for a sawmill).
    c. OEMs can view real-time machine status over internet link to help troubleshoot problems without need for travel delays or costs.

6. Users can store logic documentation, operator manuals, technical manuals, bill of materials, drawings, etc. directly in the SoftPLC. As a PC-based system, memory storage is not limited as it has been with proprietary systems. For years, CT personnel have struggled with trying to maintain documentation associated with the control system, and version control of the documentation which is often changed frequently by a number of maintenance personnel. Some SoftPLC users have even stored video clips of how to set up/use the machine in the SoftPLC. All these can be made easily accessible from a web page, which also resides in the SoftPLC.

7. Data logging has traditionally been a Level Two function that uses a great deal of network bandwidth to accomplish. In a SoftPLC system, a web browser can trigger a servlet that dynamically creates a file with the current and/or historical data in it (the file format is flexible, eg: CSV or zip). These files can be created in RAM memory and then transferred to another system while SoftPLC is running. By having the Level One system archive its own data, only one transfer over the network is required versus continual polling, again allowing for remote data access not previously possible due to limited bandwidth on networks other than those dedicated to the Level One control system, or an internal Intranet.

SUCCESSFUL ENTERPRISE LEVEL CONTROL SYSTEMS

Successful Enterprise Control Systems can now be achieved. Combining existing Java-based IT with embedded Java-based controllers, such as SoftPLC, overcomes one of the last hurdles to full Enterprise Level control systems, namely, direct data exchange between Level Three and Level One. These systems bring CT and IT together to meet corporate goals, rather than have them continue to battle at Level Two over who will do which part of the development and maintenance work.

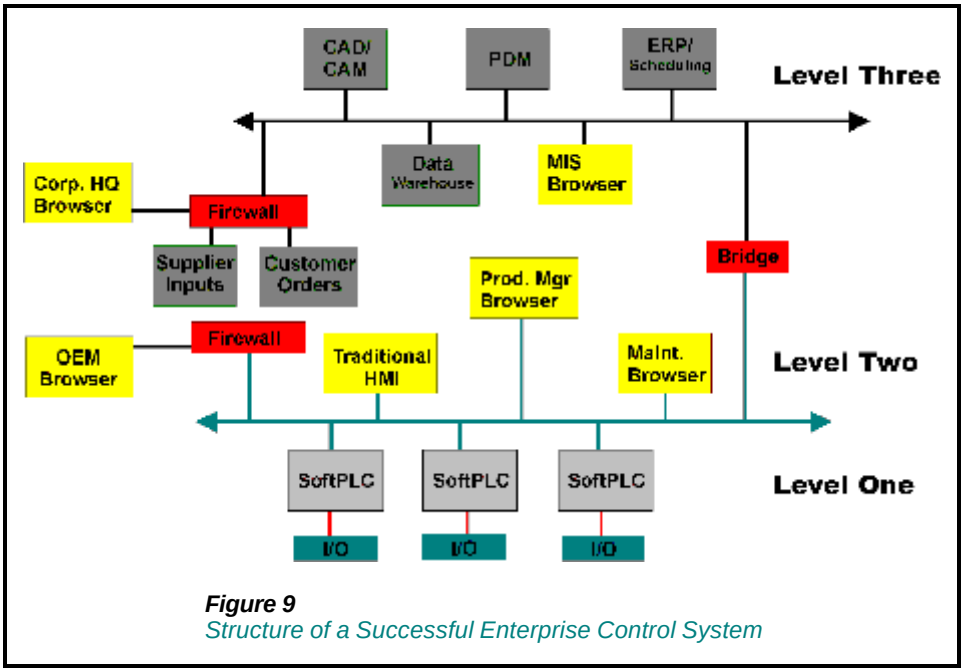


Figure 9
Structure of a Successful Enterprise Control System

© 2001 SoftPLC Corporation

There are other benefits to Java based Enterprise Control Systems:

1. Java development tools can be used to create Level Two (SCADA/HMI), as well as control programs at Levels One and Three. Many of these tools are free or very low cost, and most do not have runtime license fees like traditional Level Two systems. Several do not require much actual Java programming knowledge (Visual Editors) which lowers the specialized knowledge and training required to program controls and HMI.

2. Java can move the control industry closer to having cross compatible development environments, since the same development environment can be used to program embedded Java controllers as well as the associated operator interfaces.

3. Java could move the control industry closer to having cross compatible program code. Control programs may someday be able to move between control hardware from different vendors.

4. Common development tools, code and knowledge can reduce the cost of control products at all levels. Vendors won't need to invest as many resources to develop new products, and new control product features will be created by other industries and companies (as we have seen happen in commercial software markets). This reduced vendor cost will also lower the entry barrier to the control vendor market, which will increase competition, and lower the cost of products to customers.

5. Control systems are more tightly integrated into the entire Enterprise control scheme, rather than being proprietary islands of automation. A web browser can be the standard configuration and maintenance tool for all devices, at all levels of the organization. This is good not only for customers, but also vendor through reduction in support and service requirements. Also, this will reduce the number of proprietary technologies such as drivers and protocols, which often cause integration problems, support and maintenance issues, and increase costs.

Some proprietary controls vendors who have a lot to lose are trying to stop these open architecture trends by introducing new proprietary communications networks, protocols, I/O systems, etc. These proprietary technologies will be nothing but a distraction to the total domination of the control industry with information based technologies, such as Java. The Internet has taught us that having information readily available is important, and is achievable with the technology we have today.

SUMMARY

Java is clearly exciting technology, but Java by itself will not reduce the cost of control systems, control system complexity, control system development times/effort or machine downtime. However, information technology and tools based on Java, integrated with a good deterministic Java based control product like SoftPLC, will solve many of the existing problems in e-Automation and Enterprise Control Systems, and also provide organizations a path for future and continued success.