

DEC @aGlance-Integration of Desktop Tools and Manufacturing Process Information Systems

1 Abstract

The DEC @aGlance architecture supports the integration of manufacturing process information systems with the analysis, scheduling, design, and management tools that are used to improve and manage production. DEC @aGlance software comprises a set of run-time libraries, an application development tool kit, and extensions to popular spreadsheet applications, all implemented with Digital's object-oriented Application Control Architecture (ACA) Services. The tool kit helps developers produce DEC @aGlance client and server applications that will interoperate with other independently developed DEC @aGlance applications. Spreadsheet extensions (add-ins) to Lotus 1-2-3 for Windows and to Microsoft Excel for Windows allow users to access real-time and historical data from DEC @aGlance servers. With DEC @aGlance software, control engineers and other manufacturing process professionals can use familiar desktop tools on a variety of platforms and have simple, interactive, and transparent access to current and past process data in their plants.

2 Introduction

At a chemical plant that has been producing nylon using the same process for over 35 years, the lead control engineer told an interviewer that what he likes about his job is that "it is totally different every day." [1] To an outside observer, the operation of a process plant, such as a refinery or paper plant, appears to be an unchanging flow of materials into a tightly controlled and repetitive process that produces a continuous flow of unvarying product-24 hours a day, 365 days a year. In reality, the operation of these plants is far more complex and challenging, involving constant adjustment to changing conditions, aging equipment, and variations in raw materials, as well as constant monitoring for equipment malfunctions.

The operation of a large process plant involves the functioning of numerous valves, switches, pumps, other actuators, and sensors measuring and controlling the levels, pressures, temperatures, and flows of various materials through a complex series of pipes, tubes, tanks, and vessels. In addition to detecting and managing failures in these components, a large proportion of the personnel in the plant is involved in process and product improvement. The personal computer or workstation and an array of sophisticated desktop tools allow data to be analyzed, visualized, manipulated, and explored in ways that support creative problem solving. Getting timely information about the process into the appropriate problem-solving tools is, however, difficult. This paper begins

with some background about manufacturing process information systems and the need for access to system data. The paper then describes the

Digital Technical Journal Vol. 5 No. 2, Spring 1993 1

DEC @aGlance-Integration of Desktop Tools and Manufacturing Process Information Systems

development of DEC @aGlance software and the choice and use of Application Control Architecture (ACA) Services to solve the problem of integrating independently developed applications in the manufacturing space.[2]

3 Background

In large manufacturing facilities, the production process is controlled through the use of advanced automation systems. These systems may track thousands of temperatures, flows, pressures, and levels and can drive hundreds of pumps, valves, and other actuators. To implement control strategies, such systems may compute large numbers of complex, dynamic control algorithms. Usually, additional systems measure various physical properties of the product, such as color, weight, viscosity, thickness, and moisture content. Supervisory control systems often coordinate parts of a complex process, as well as implement higher-level control and production strategies and keep historical records of key process variables.

The control of a large plant is usually implemented through strategies that allow the control problem to be divided into smaller parts, as illustrated in Figure 1. Each piece of the system is responsible for the control of a subsystem (e.g., steam generation and distribution, or cooling fluids), a part of the process (e.g., premixing, material storage, or reaction), or an area of the plant (e.g., packaging line, product stream, or finished goods management). Within each subsystem, there is typically a hierarchy of control. The lowest-level components control activities that require responses within less than a second to as much as one minute (direct control). The next level of systems control activities that require responses within less than a few minutes (distributed control). Above this level of response are systems that control activities that may not change for long periods or that implement control algorithms that involve measurements from more than one lower-level system (supervisory control). At the plant level, additional control systems may exist to implement control algorithms that reflect changes in the markets for products, market opportunities, and fluctuations in raw material availability and composition, along with the information about the process that is supplied by the lower-level systems (high-level control). Scattered among these levels may be various additional systems that schedule preventive maintenance, identify equipment failures, and advise on process improvements—all based on information about process from the other systems in the plant.

Distributed control systems include an operator console that consists of multicolor displays, push buttons, warning lights and buzzers, a touch screen or trackball, and industrialized keyboards with as many as a 36 special function keys. The displays allow an operator to oversee all parts of the process for which the operator is responsible. Typical displays show

recent trends of key variables and mimic diagrams showing the current state of the manufacturing equipment (e.g., valve positions and tank levels) and of the material flowing through the process. The keyboard and other input devices allow the operator to select displays, request reports, and modify

2 Digital Technical Journal Vol. 5 No. 2, Spring 1993

of

Desktop Tools and Manufacturing Process Information Systems

control settings. Response to problem or alarm conditions and modification of the process to change the product are effected through the console.

Process operators are responsible for maintaining the routine operation of a plant. Operators use the control system to change process parameters in order to produce different mixes or variants of the product, or to respond to an equipment failure by rerouting material around nonoperational process equipment.

To perform their functions, manufacturing plant production and engineering support personnel (e.g., control engineers, process engineers, production supervisors, production planners, maintenance supervisors, and manufacturing engineers) also need access to information in the control and supervisory systems. These professionals regularly access information contained in multiple manufacturing systems and have an occasional interest in particular measurements or parameters within other parts of the process. The functions of these manufacturing plant personnel include

- o Complex problem analysis and solution. Locating sources of product or process variation involves analyzing information from different parts of the process that may be under the control of different automation systems. Comparing the flow that exits one part of the process with the flow that then enters the subsequent part, for example, could disclose a faulty flow meter, a previously unknown temperature control problem, or a leak.
- o Product improvement. Improving product quality and consistency involves investigating how the product is affected by existing variations in the production process. For example, investigation may involve the study of a process variable that cannot be measured directly but can be calculated from the values of other process variables. Examining sets of variables over time and exploring possible relationships may result in discovering combinations of process variables that yield unexpected effects on product attributes.
- o Process improvement. Improvements in process yield and process reliability and reduction of waste and hazardous by-products may involve the study of historical data values from the process. Studying measurements obtained from multiple control systems may also result in process improvements.
- o Resource optimization. Usually, process plants are capable of producing different grades of product, as well as mixtures of end products. An oil refinery, for example, produces various grades of fuel oil and also home heating and lubricating oils, all from a single process. While the operators adjust the equipment to control the product mix, a process planner or production manager determines the best production schedule

based on customer orders and the efficient use of the process equipment.

Digital Technical Journal Vol. 5 No. 2, Spring 1993 3

DEC @aGlance-Integration of Desktop Tools and Manufacturing Process Information Systems

Process information is available to operators and engineers who are trained to work with the various control and management systems in the plant. Using proprietary tools for each system allows reports to be generated and specific types of analyses to be performed on the data contained within each of these systems. However, extracting the data from these systems to an engineer's desktop for analysis by generic tools, such as spreadsheets and statistical analysis packages, is difficult or even impossible. Lack of console- and tool-specific training is another obstacle to accessing process information.

Manufacturing Process Information Systems and Desktop Systems: Goals and Barriers

Production and engineering support personnel want to be able to use the desktop tools of their choice to explore and analyze data from manufacturing systems. Spreadsheets, simulation tools, report generators, visualization tools, statistical analysis tools, planning tools, charting tools, and graphic-generation tools have all become accepted parts of the array of computer-aided techniques and tools available to the contemporary knowledge worker. The interactive, easy-to-use graphical user interface, which can run on relatively inexpensive platforms under the complete control of the end user, has not only encouraged the wide use of these desktop tools but also enhanced their effectiveness. These tools stimulate professionals to creatively explore the character of large amounts of data and thus support the discovery of previously unexpected patterns and relationships.

The further an end user's primary function is from production, the more likely it is that such a user will want access to multiple systems. System interfaces, which may differ widely and are generally oriented toward production use, discourage users from making ad hoc inquiries into the system. Consequently, manufacturing system data may not be easily accessible to users of the many desktop tools available for such purposes as decision support, research, analysis, and simulation.

Today, the use of data from the manufacturing process in planning, reporting, and managing the operation of a plant is hampered by the difficulty in accessing the data from plant control and process information systems. It is typical for a production supervisor who needs data from a control system to request the data from a process operator. Once in hand, the data is then manually entered into a spreadsheet or other desktop tool for analysis. The results of the analysis often require entering new parameter values into the control system. This task is typically performed by another person, trained to use the control system, who transcribes the values from a hard copy of the tool's output. The process is time-consuming, costly, and error prone. Problem-solving activities are limited

to those that can justify the trouble and expense involved in simply accessing the data.

4 Digital Technical Journal Vol. 5 No. 2, Spring 1993

of

Desktop Tools and Manufacturing Process Information Systems

Existing Integration Efforts

The desire to use data from the control systems to analyze and improve the understanding and control of the manufacturing process has spawned a variety of efforts since the late 1980s. This work has attempted to ease the transfer of information between computing systems and control systems. However, the resulting products and standards are not oriented toward supporting ad hoc inquiries and, therefore, are not widely used.

Many currently available manufacturing systems may be connected to the plant network, but without standard higher-level interfaces, access to these systems remains limited.[3-7] Through such network connections, some manufacturing systems provide limited access to OpenVMS and/or DOS system users. However, the access is typically restricted to the use of unique, proprietary programming interfaces or to proprietary tools targeted at performing a manufacturing-related function, such as statistical quality control. Usually, interfaces are supplied only on a specific operating system or on limited versions of a specific operating system.

In some systems, it is possible to extract a table of data values into a file using a common representation and file format (such as Lotus Development Corporation's WK1) that can then be imported into a spreadsheet on an IBM-compatible PC. This technique obviates the need for hard-copy output and simplifies transcription but still requires that a specialist extract the data using proprietary interfaces. In addition, the data may need to be converted from string to numeric format to be usable within a particular spreadsheet.

The International Organization for Standardization standard Manufacturing Messaging Specification (IS9506 or MMS) addresses the problem of data exchange between applications and dedicated manufacturing systems (referred to in the standard as manufacturing devices).[8] Although some manufacturers of programmable controllers (that is, dedicated control systems that are primarily used in discrete manufacturing industries) offer MMS capabilities, the process industry manufacturers and their control system suppliers have not widely accepted MMS. Use of the standard has been perceived as expensive, inefficient, and oriented primarily toward the needs of discrete manufacturing. A committee of the Instrument Society of America (ISA) is developing a companion standard (ISA 72.02) to use with MMS in communicating with distributed control systems in process manufacturing.[9] An important aspect of this proposed standard is a data model that describes the organization and types of data in a distributed control system.

4 Requirements for Integration

Digital designed the DEC @aGlance architecture not to be a generic application integration mechanism but rather to support the integration of popular desktop tools with manufacturing process information systems. An application that complies with the architecture can be installed on any system within a network, run, and immediately exchange data with other compliant applications. Some key characteristics of the environment that helped to drive the architecture are

- o Multiple vendors. Although, MS-DOS personal computers are the most popular desktop environment, VAXstation, Macintosh, and UNIX workstations have a clear presence in particular departments and in certain large customer sites.
- o Multiple software developers. The applications to be integrated are products of many companies that build manufacturing systems and desktop tools. The software development groups in these companies focus on core application and human interface issues rather than on integration issues.
- o A large variety of desktop applications and user interfaces. Each class of desktop application has a different way of interacting with users. Spreadsheets, for example, have very different user interfaces from statistical packages and data visualization packages. Some applications have elaborate macro languages, whereas others are almost entirely graphically driven.
- o Multiple types of large networks. In the typical process manufacturing facility, large networks are already in place. While many plants use DECnet for their network, an increasing number of plants are choosing to use the transmission control protocol/internet protocol (TCP/IP), and some plan to migrate to Open Systems Interconnection (OSI) networks (including Digital's DECnet Phase V) from multiple vendors. PC LANs are also becoming popular.
- o Conservative computing strategies. Large manufacturing facilities cannot afford to halt operation to make major changes in their production-related computing systems and networks. Such facilities look to standards-based products as a way of achieving stability and of ensuring confidence in the longevity of a particular technology.

5 Architectural Issues

Simply stated, the problem that the DEC @aGlance architecture attempts to address is, how can a set of existing applications running on heterogeneous

platforms, distributed across a variety of networks, and developed by different vendors (with only peripheral interest in integration) be easily integrated? A good understanding of both the nature of the applications involved and how end users would use them if they were integrated is important for evaluating potential answers to the question.

6 Digital Technical Journal Vol. 5 No. 2, Spring 1993

of

Desktop Tools and Manufacturing Process Information Systems

The applications that we considered integrating can be divided into two groups: those that "own" manufacturing data, i.e., the manufacturing control systems, and those that are consumers of that data, i.e., the desktop tools. From the viewpoint of an end user, some aspects of the relationship between a desktop tool and a manufacturing control application must be considered in order to accomplish work goals. End users in this environment are primarily concerned about the manufacturing process, the equipment controlling the process, and the state of materials within the process. These users have little or no interest in such aspects as network topologies and protocols, operating systems, and byte ordering on different hardware platforms.

Some major concerns of the end user that the architecture should address are

- o The identity of the manufacturing control system. Generally, a large plant is controlled through the use of several control systems, each of which might control a part of the process, such as refining or packaging, or an aspect of the plant operation, such as steam distribution or waste reprocessing. A particular data point resides in a single manufacturing control system. The user should be able to specify precisely which manufacturing system is to supply the data values. The architecture should be capable of establishing a relationship with the specific application that owns the data of interest to the user. The end user should not have to specify either the network node, the operating system, or the hardware platform on which the application is running. Neither should the end user have to specify the network communication protocols required.
- o The length of the relationship between the desktop tool and the manufacturing control application. The relationship should be able to remain active for multiple transactions to allow end users to work interactively with desktop tools to explore possibilities. For example, end users may want to examine different data points or the same data point over various time intervals. Thus, usage of a desktop tool could involve multiple requests for data from a manufacturing control application. Establishing a relationship between applications over a network is time-consuming, and therefore establishing long-lived relationships would be advantageous. The ability to continuously monitor a set of points and have their values reported on a time or change basis is another desirable feature that would require the establishment of long-lived relationships.
- o Multiple access to the applications. Application relationships should not be exclusive. Each application should be able to have concurrent relationships with several partner applications. Each desktop tool may require data from several manufacturing systems, and conversely,

several users of desktop tools may need to access the same control system simultaneously. The relationships between desktop tools and manufacturing control systems is illustrated in Figure 2.

DEC @aGlance-Integration of Desktop Tools and Manufacturing Process
Information Systems

- o The data model. Applications should agree about how to reference data and about data types. Within the context of this environment, a relatively simple data model exists in the draft standard ISA 72.02. Data should always be converted to types appropriate to the local system and to the application. A spreadsheet user should not have to manually convert strings into numeric values.
- o The user interface. Application integration should not require the use of any particular desktop user interface, such as the X Window System or DECwindows software, or even the existence of a windowing system. Also, the user interface of the manufacturing data application should be of no concern to the desktop user.

6 Usage Model

To help us understand how a user might go about employing the capabilities that we were considering, we developed a simple usage model. We based the model on the scenario that an end user makes a series of ad hoc inquiries into the state of a process. We assumed that the user was familiar with the manufacturing process but not necessarily expert in all the details of the process. The user would know, for example, what the major areas of the plant were called and what functions they performed but might not know the internal reference identifier of every flow meter in each control system. We focused on how the user of a spreadsheet tool might reasonably expect to proceed to get data into a spreadsheet and how services that we might provide could aid in exploring the data.

The information within a manufacturing system consists of the many parameters and measurements that the system uses to monitor and control the process. Generally, this data is organized into blocks, each one related to a particular part of the process, such as flow, level, temperature, or pressure. As the typical data block in Figure 3 illustrates, every block has a unique name or tag that can be used for reference purposes.

In control systems, tag names are assigned as part of the configuration. Large plants use a naming convention to ensure the assignment of unique tag names to the thousands of blocks spread throughout the plant and over several control systems. In addition to the tag, the block contains attributes such as the parameters of the control algorithm, measured input values, unit conversion algorithm identifiers. The data model proposed by the ISA 72.02 committee describes seven types of blocks, each with a standard set of attributes with associated names and data types.

This usage model allows a user to easily determine the tag names recognized by a particular manufacturing system. To examine the data values associated with a specific tag, the user needs to know the valid attributes. (All

blocks do not have the same attributes, e.g., an analog loop control block has more attributes than a simple digital monitoring block.) Once the tag names and their valid attributes are known, the user can inquire about current values as well as historical values.

8 Digital Technical Journal Vol. 5 No. 2, Spring 1993

of

Desktop Tools and Manufacturing Process Information Systems

The use of operating prototypes, including simulated servers and a simple spreadsheet, advanced the development of the usage model. The prototypes were shared with potential end users and application developers at customer visits and industry trade shows. Feedback obtained from demonstrations and discussions of the usage model helped expand and refine the services.

7 Architecture

The DEC @aGlance architecture defines two kinds of applications, a set of services for accessing data in the control systems, a data specification model, and some basic types of data. The application classes are (1) manufacturing data servers and (2) clients. Typical manufacturing data servers are the manufacturing control system applications. Typical clients include desktop tools such as spreadsheets and statistical analysis tools, as well as production planning, production scheduling, and other production management applications. An application may be a client in relation to one application and a server in relation to another.

A data point is specified to DEC @aGlance applications by the name of a server, a tag name, and an attribute name. A data point has a current value and may also have historical values (if the manufacturing system has a historian capability). A current value is the most recent available value of a parameter or measurement within the system. A historical value is a value that the data point had at some time in the past. A historical value is specified by the name of a server, a tag name, an attribute name, and the time associated with the value.

The services defined by the DEC @aGlance architecture fall into one of four functional categories: configuration information, data value exchange, monitoring, or management. Each service defines an operation that may be requested by one application of a partner application. The services defined are not necessarily the same functions that an end user requests.

Configuration Information

One service is defined for requesting the tag names that the server finds in the control system's database. An additional service returns a list of attribute names that are defined for a specified tag name or a list of tag names.

Data Value Exchange

Services are defined for reading and for writing current and historical data point values. For current values, services support reading or writing either a list or a table of data point values. A read or write list request specifies pairs of tag names and attributes. A read or write request for a table of data point values specifies a list of names and a list of

attributes. The table of data points consists of all tag names paired with their corresponding attributes. Both the list and the table requests can be used to read or write a single data point, collapsing to either a list or a table of one data point.

DEC @aGlance-Integration of Desktop Tools and Manufacturing Process Information Systems

By using the DEC @aGlance services to get lists of tag names, attribute names, and data point values, and the name of a server, an end user can generate a wide range of ad hoc queries without knowing much about the control system in advance. A common data point attribute is the descriptor, which characterizes the function of the data point, e.g., south tank level. Thus, it is a fairly straightforward task to use DEC @aGlance services to build a list of tag names and descriptors that provide a basis for further inquiries.

The services for historical data values are defined to deal with tables of historical values for a list of data points. Historical data service requests specify a list of tag name and attribute pairs and a time specification that is applied to all the data points. The time specification consists of a start time, a time interval, and the number of intervals for which values are to be returned.

Monitoring

Monitoring is useful for reading the values of a set of data points at intervals in time or when a significant change in value occurs for any of the data points. A graphical display program can run on a desktop system and make minimal use of the network and computing resources while maintaining an accurate representation of what is occurring in the manufacturing process. Monitoring could also be used to update a spreadsheet at regular time intervals or whenever a particular process variable changes.

No standard definitions exist for what constitutes a significant change in value. Definitions supported for various systems include (a) detection of change outside of a specified range or "dead band," (b) change by more than some percentage of the previously reported value, and (c) change by more than some percentage of a fixed value. Therefore, the service is defined to support monitoring and reporting of changes on a time basis or on some other basis that is specific to the data server application. Whenever the requested monitor condition is fulfilled, the data server application uses a monitor update service to send the new data point values to the original client application. Since the server initiates monitor update requests, the usual relationship between the client and the server is temporarily reversed.

Management

Connection management services are provided to establish a connection, to terminate a connection, and to test a connection.

8 Implementation Considerations

Using existing networking and application integration technologies to implement the DEC @aGlance architecture was important both in terms of reducing development efforts and improving compatibility with existing environments. Technology used in the implementation had to provide as many as possible of the capabilities described in the architecture while imposing minimal restrictions on the end-user operating and network environments and on the developers of the applications. In addition, it was desirable that the underlying technologies offer capabilities that could support future enhancements to the DEC @aGlance architecture.

The DEC @aGlance architecture allows an existing desktop tool to be integrated with existing manufacturing control systems, as shown in Figure 4. The architecture effectively combines the functional capabilities of the desktop tool for analysis, visualization, computation, etc., with the capabilities of the manufacturing control system for monitoring and controlling a manufacturing process. The individual applications were, of course, originally designed and written without any knowledge of each other's existence. Therefore, to facilitate integration efforts, implementation of DEC @aGlance software should localize and minimize required changes to the applications.

A network protocol such as DECnet, the transmission control protocol /internet protocol (TCP/IP), or one of the local area network (LAN) protocols could have provided the network services required for DEC @aGlance's interapplication communications. However, this approach lacks a mechanism for locating servers on the network, requires DEC @aGlance to support the multiple network protocols that exist in the manufacturing environment, requires DEC @aGlance to include data type conversion between application platforms, and necessitates the development of monitoring and management tools unique to DEC @aGlance. A better approach is to use an existing product that is available on an appropriate set of platforms, supports an appropriate set of networks, and already solves these problems.

A remote procedure call (RPC) mechanism appears to have many of the capabilities that the DEC @aGlance architecture requires. RPC mechanisms provide for location of a partner or server application, and they provide data type conversion and reliable network services. The RPC model of application integration, however, is actually more appropriate for the distribution of a single application across multiple systems in a network. This use implies a simple, static relationship between the parts of an application: one part is always a client that requests the execution of a procedure, and the other part is always an RPC server that executes the procedure and returns the results. In such a relationship, each request generates a single response. This model would be poorly suited for supporting the DEC @aGlance monitoring service. When DEC @aGlance was

being developed, no commercially available RPC implementation ran on the key platforms, the OpenVMS and Microsoft Windows environments. Furthermore, no one had announced their intention to produce a portable implementation

DEC @aGlance-Integration of Desktop Tools and Manufacturing Process
Information Systems

that would be available on the wide range of platforms that we considered important for future versions of DEC @aGlance software.

Digital's ACA Services was chosen as the basis for implementing DEC @aGlance software because it implements an application integration model that closely matches the requirements of the DEC @aGlance environment. ACA Services supplies many capabilities required of the integration mechanism including

- o Abstraction of functions from implementations
- o The ability to encapsulate existing applications
- o Location of partner applications on a variety of networks
- o Establishment and management of reliable, long-lived communication links
- o The ability to easily add new applications to the system
- o The ability to easily install new versions of existing applications in the system
- o The correct handling of data type conversions between heterogeneous systems
- o Commercial availability of portable interfaces on OpenVMS, Microsoft Windows, Macintosh, and a wide variety of UNIX platforms from multiple vendors

The class hierarchy capabilities of ACA Services allow the creation of new combinations of applications integrated to provide new capabilities without additional coding. Thus, a new class of server can be defined to offer the capabilities of a DEC @aGlance data server as well as additional capabilities. The older DEC @aGlance servers would actually provide the DEC @aGlance services while, transparent to the client applications, the new server would make the new capabilities available.

ACA Services has been selected as a major component of the Object Management Group's (OMG) Object Request Broker, which in turn has been selected as a part of the Open Software Foundation's (OSF) Distributed Computing Environment (DCE). ACA Services is designed to be independent of the type of network that provides the interapplication communications services and currently works over both DECnet and TCP/IP networks, the networks most commonly found in manufacturing environments. Therefore, applications using ACA Services need not be concerned about network communications.

ACA Services is supported on the OpenVMS, Microsoft Windows, Macintosh, and SunOS operating systems, the most often used platforms in this application space. In fact, ACA Services is the only application integration mechanism currently available on all these platforms. Moreover, ACA Services supports the kind of asynchronous services required by DEC @aGlance.

12 Digital Technical Journal Vol. 5 No. 2, Spring 1993

of

Desktop Tools and Manufacturing Process Information Systems

Although it provides many important components of the required integration service, ACA Services does not completely solve the integration problem. ACA Services is a tool intended to be used to integrate applications; it does not define the data model nor does it define the set of services that applications are to provide. Application integrators are expected to define (1) the classes of applications that provide sets of services, (2) the services, and (3) the meaning and type of data to be exchanged by applications using the services.

9 DEC @aGlance Software: The Tool Kit and Add-ins

As shown in the DEC @aGlance component diagram in Figure 5, DEC @aGlance software uses ACA Services as a basic application integration facility. Above ACA Services, DEC @aGlance adds definitions of a class of manufacturing data server applications (servers), a set of definitions of the services provided by the servers, and definitions of the data reference model.

ACA Services provides a general capability to integrate sets of applications. DEC @aGlance software provides a set of routines that are specifically designed to simplify the implementation of the set of services that DEC @aGlance supports. For server applications, DEC @aGlance software supplies a set of callback points, as well as callable routines for declaring callbacks, filtering strings, and supporting monitoring activities. For client applications, DEC @aGlance software supplies a set of callable routines for requesting each of the defined services, as well as callback points in support of monitor updates.

The DEC @aGlance server library also supports a test connectivity capability used to verify that an interapplication relationship can be established to the server application. This capability simplifies the diagnosis of problems encountered during both server development and client-server installation.

To reduce dependence upon properly written server code, the test connectivity capability operates entirely within the library. Thus, once a server calls the DEC @aGlance initialization routine, and if the server is still running, this service should function properly in response to requests from DEC @aGlance clients. Proper functioning includes verifying the installation and configuration of the network and of the ACA Services and DEC @aGlance run-time components of the systems on which the client and server applications reside.

Software add-ins, i.e., extensions, for two popular spreadsheet applications, Lotus 1-2-3 for Windows and Microsoft Excel for Windows, are also DEC @aGlance products. These add-ins allow users of the spreadsheets to request data from manufacturing data servers by means of

the spreadsheets' macro facilities. The add-ins provide a dialog box to guide untrained users through the process of constructing a DEC @aGlance macro. Once built, a macro can be executed one or more times, modified if necessary, and saved in a worksheet for reuse at some other time.

DEC @aGlance-Integration of Desktop Tools and Manufacturing Process Information Systems

Tool Kit

The tool kit was developed to encourage the rapid and successful development of DEC @aGlance applications by third parties. Successful applications are those that interoperate with other DEC @aGlance applications upon delivery to a customer site with no additional coding, no application recompilation, and no application rebuilding.

The key components of the tool kit are

- o A DEC @aGlance client or server library
- o Example code
- o ACA Services definition files for the DEC @aGlance class and methods
- o Simple test facilities
- o The DEC @aGlance Programmer's Guide[10]

The ACA Services definition files contain the information required to define the manufacturing data server class and the services that members of the class support. Supplying the definitions in this form ensures strict consistency among all server and client developers with regard to these definitions. The routines in the DEC @aGlance client and server libraries use these definitions. The DEC @aGlance libraries contain all the code required to establish and maintain an ACA Services session.

Server Applications

A server application built with the tool kit has three major components: an initialization section, the control system-specific section, and the DEC @aGlance section. The initialization section simply declares the server's name to the DEC @aGlance application, declares a set of callback points, and enters a dispatch loop. The server name is the name that client applications can use to interact with this server. The callback points are the code entry points to which DEC @aGlance dispatches in response to the receipt of service requests from the client applications. For a server, callback points exist for the following services:

- o Get a list of tag names
- o Get a list of attribute names
- o Get a list of data point values

- o Get a table of data point values
- o Put a list of data point values
- o Put a table of data point values
- o Get a table of historical values
- o Put a list of historical values
- o Register a monitor request
- o Cancel a monitor request

14 Digital Technical Journal Vol. 5 No. 2, Spring 1993

of Desktop Tools and Manufacturing Process Information Systems

- o Initiate a session
- o Terminate a session
- o Execute a server-specific request
- o Terminate the server

The control system-specific section consists of code modules that execute calls to the control system application programming interface (API). These modules have to convert parameters to and from the DEC @aGlance format and the control system-specific format. The entry point of each module is declared as a callback point during initialization.

In addition, callable routines are provided for sending monitor updates and for session management. The DEC @aGlance section of the server is contained entirely within a library of callable server routines. This section handles all interactions with ACA Services, including server registration and session management. It also handles the dispatch of incoming requests to the callback routines and a number of housekeeping tasks for which each server developer would otherwise have to develop and implement solutions. The DEC @aGlance section also responds to test connectivity requests.

Almost all vendors of manufacturing systems have applications that execute calls to the control system API, but such applications are typically driven off a command language or menu interface. Conversion of these applications to a DEC @aGlance server is relatively easy; some vendors have created a simple DEC @aGlance server in as little time as one day.

Client Applications

The typical DEC @aGlance client application is built on an existing desktop tool. Desktop tools provide a user interface for performing some class of generic function such as decision support, statistical analysis, quality control, or production scheduling. Other types of applications that could make use of process data, such as report generators, batch schedulers, and maintenance tracking systems, can also provide the basis of DEC @aGlance client applications. Adding DEC @aGlance support to an existing tool allows the user to treat data from DEC @aGlance manufacturing data servers like data entered manually or from other data sources.

A DEC @aGlance client application incorporates the DEC @aGlance client routine library, which provides callable routines for initialization and for each of the following DEC @aGlance services:

- o Get a list of tag names

- o Get a list of attribute names
- o Get a list of data point values
- o Get a table of data point values
- o Put a list of data point values
- o Put a table of data point values

DEC @aGlance-Integration of Desktop Tools and Manufacturing Process
Information Systems

- o Get a table of historical values
- o Put a list of historical values
- o Initiate a monitor request
- o Cancel a monitor request
- o Initiate a session
- o Terminate a session
- o Execute a server-specific request
- o Terminate the server
- o Terminate the client

In addition, support routines help monitor updates.

To support the DEC @aGlance monitoring capability, a client application must have some server characteristics. Once a monitoring request has been initiated, the server issues monitor update requests when the monitoring condition is satisfied. The monitor update requests are received by the client application using the same callback mechanism that the server uses when servicing client requests.

A typical client calls the DEC @aGlance initialization routine and then continues to perform its normal functions. When a DEC @aGlance service is requested through the user interface or other mechanism, the application simply formats the request and calls the appropriate DEC @aGlance service request routine. Upon completion of the routine, status (and if requested, data) is returned from the server application. If data is returned that is to be further processed by the client application, the application moves the data to its workspace in preparation for additional processing.

DEC @aGlance Lotus 1-2-3 for Windows and
Microsoft Excel Add-ins

Whereas most manufacturing control systems provide a callable library that allows the development of applications that access the data in the system, some desktop tool applications have mechanisms that allow for extension of their capabilities in the field. Spreadsheet applications such as Lotus 1-2-3 and Microsoft Excel support the use of add-in modules to add external functions and external macro capabilities. Add-ins for these two spreadsheets are available as DEC @aGlance software products.

With the add-ins, spreadsheet users can access most DEC @aGlance services and thus can

- o Fill a range of cells with a list of tag names from a server
- o Fill a range of cells with a list of attribute names associated with a range of tag names in a server
- o Fill a range of cells with a list of data point values

of Desktop Tools and Manufacturing Process Information Systems

- o Fill a range of cells with a table of data point values, as shown in Figure 6
- o Write a list of data point values to a server
- o Write a table of data point values to a server
- o Fill a range of cells with a table of historical values for a specific time interval
- o Write a list of historical values

The interface for the add-ins was designed to support ad hoc inquiries. A dialog box guides the end user through the process of supplying the appropriate parameters for a selected function. Where appropriate, defaults are suggested based upon the previous inquiry.

10 Summary

DEC @aGlance software has been specifically designed to make it easy for users of desktop tools to access, explore, and analyze data from distributed control systems, supervisory control systems, and other common systems used to run manufacturing processes. An analysis of the information environment and the ways in which end users want to access the data led to the refinement of the architectural requirements. The analysis also led to the decision to use ACA Services as the appropriate mechanism for integrating desktop and manufacturing control applications. The creation of a usage model and rapid deployment of prototypes were instrumental in the analysis. To promote widespread availability of plug-compatible applications that use DEC @aGlance, a developer's tool kit was created. The tool kit contains libraries of DEC @aGlance routines that both simplify and encourage proper and consistent usage of ACA Services to integrate DEC @aGlance applications.

DEC @aGlance add-ins for the popular spreadsheet programs Lotus 1-2-3 for Windows and Microsoft Excel for Windows were developed also. With the add-in, users can interactively explore data in plant manufacturing control systems from within a familiar spreadsheet, as well as write reusable worksheet macros for performing repeated tasks like report generation.

11 Acknowledgments

The author gratefully acknowledges the contributions of the members of the DEC @aGlance development team: Judie Dow, Bob Harrison, Nick Miller, Ramesh Swaminathan, Patrick Taber, and the lead developer, Charlie Trageser. I would also like to thank Steve Dawson, for introducing our group to the problem and generally educating me about the process manufacturing

environment; Chuck Kukla, for introducing me to his research on how people work in manufacturing and for his work with customers and control vendors that helped lead to the design of the product; Jim Thompson, for pushing and pulling all the strings that it took at every stage of the effort to

DEC @aGlance-Integration of Desktop Tools and Manufacturing Process
Information Systems

bring the concept to a marketable product; and Mike Renzullo and Alan Ewald
of the ACA Services Development Group, for their support.

12 References

1. This quotation was taken from the transcript of an interview conducted by C. Kukla et al., who have published the results of their study in "Usability Turning Technology into Tools," *Designing Effective Systems: A Tool Approach*, P. Adler and T. Winograd, eds. (New York, NY: Oxford University Press, 1992).
2. DEC ACA Services System Integrator and Programmer's Guide (Maynard, MA: Digital Equipment Corporation, Order No. AA-PFYUA-TE, 1992).
3. CM50N User Manual, Order No. CM11-320 (Phoenix, AZ: Honeywell Industrial Controls and Automation, 1991).
4. Computer/Highway Interface Package (CHIP) User Guide, Part No. D001093X012 (Marshalltown, IA: Fisher Controls International, Inc., 1987).
5. AIM Connectivity Software User's Manual (Houston, TX: W. R. Biles and Associates, Inc., 1992).
6. S/2 SCADA System Description, Document No. SD2.0001 (Dallas, TX: Texas Instruments, Industrial Systems Division, 1988).
7. PI System Plant Information System Technical Overview (San Leandro, CA: Oil Systems, Inc., 1990).
8. Manufacturing Messaging Specification, ISO/IEC 9506 (Geneva: International Organization for Standardization/International Electrochemical Commission, 1990).
9. Manufacturing Messaging Specification: Companion Standard for Process Control, ISA 72.02 (Research Triangle Park, NC: Instrument Society of America, 1993).
10. @aGlance Programmer's Guide (Maynard, MA: Digital Equipment Corporation, Order No. AA-PQB8A-TK, 1992).

13 Trademarks

The following are trademarks of Digital Equipment Corporation: DEC, DEC @aGlance, DECnet, DECwindows, Digital, OpenVMS, and VAXstation.

IBM is registered trademark of International Business Machines Corporation.

Lotus and 1-2-3 are registered trademarks of Lotus Development Corporation.

Macintosh is a registered trademark of Apple Computer, Inc.

Digital Technical Journal Vol. 5 No. 2, Spring 1993 19

DEC @aGlance-Integration of Desktop Tools and Manufacturing Process
Information Systems

Microsoft, MS-DOS, and Excel are registered trademarks and Windows is a trademark of Microsoft Corporation.

SunOS is a registered trademark of Sun Microsystems, Inc.

UNIX is a registered trademark of UNIX Systems Laboratories, Inc.

14 Biography

David Ascher Dave Ascher joined Digital's Industrial Products Software Engineering Group in 1977 to work on the DECDataway industrial multidrop network. Since then, he has worked on distributed manufacturing systems as a developer, group leader, and technical consultant, and as an architect of the DEC @aGlance product. As a principal software engineer, Dave leads an effort to develop DEC @aGlance service offerings. He holds a B.S. in psychology from City College of New York and a Ph.D. in psychology from McMaster University, Hamilton, Ontario.

=====
Copyright 1993 Digital Equipment Corporation. Forwarding and copying of this article is permitted for personal and educational purposes without fee provided that Digital Equipment Corporation's copyright is retained with the article and that the content is not modified. This article is not to be distributed for commercial advantage. Abstracting with credit of Digital Equipment Corporation's authorship is permitted. All rights reserved.
=====