Published October 1, 2003. Distribution restricted to Sponsors until January 1, 2004.

AUTO-ID CENTRE

WHITE PAPER

Routes for Integrating Auto-ID Systems into Manufacturing Control Middleware Environments

Nirav Chokshi, Alan Thorne, Duncan McFarlane

AUTO-ID CENTRE INSTITUTE FOR MANUFACTURING, UNIVERSITY OF CAMBRIDGE, MILL LANE, CAMBRIDGE, CB2 1RX, UNITED KINGDOM

ABSTRACT

A key issue in developing Auto-ID technology is the requirement for a supporting information infrastructure. Unlike conventional automatic identification techniques such as bar-codes or RFID, which have remained limited to product-type identification only, the facility of instance-level product information in Auto-ID will lead to a multi-fold increase in the amount of data to be processed in real-time. How to handle such data effectively and economically will be crucial in developing the technology itself. Equally important will be the investigation of different routes along which this information passes – from sensing of tags to final use in business applications – so as to guarantee that only the right amount of information is retrieved at a particular location and that none of the useful information is discarded mistakenly.

Along the path of migration to fully Auto-ID compliant operations, there will also be a need to re-use relevant infrastructure that exists already in various application domains. In manufacturing control in particular, the use of bar-codes or RFID for product identification has been in place for quite some time and there exist a good understanding, both at the hardware as well as software level, for integrating product-type information within real-time control decisions. Similarly, modern industrial database systems have become increasingly sophisticated and easy-to-use for storing and retrieving large amounts of real-time, time-consistent information. A close scrutiny of such existing facilities could prove valuable to technology developers to effectively migrate to Auto-ID framework and could also prove useful to vendors for developing "out-of-the-box" design of manufacturing information systems.

We refer to the set of applications managing the interface between real-time sensory data and business information systems as "middleware". Focussing particularly on the middleware requirements in manufacturing information and control environments, we investigate some of the above issues in this paper. The three principle aims of our analysis are as follows:

1. To review the existing use of automatic identification techniques in manufacturing control environment by understanding the operations of two to three dominant control system products available "off-the-shelf";
2. To examine the capability of such systems for dealing with Auto-ID information and, in particular, managing the relatively high volume of real-time data generated on the production shop-floor;
3. To proposed and analyse different routes through which Auto-ID infrastructure could be effectively integrated within industrial data management and control environment.

Published October 1, 2003. Distribution restricted to Sponsors until January 1, 2004.

WHITE PAPER

Routes for Integrating Auto-ID Systems into Manufacturing Control Middleware Environments

Biographies

Nirav Chokshi
PhD Student

Nirav Chokshi received his Bachelor of Engineering in Instrumentation and Control discipline from Gujarat University, India in 1995. He then joined Indian Petrochemicals Corporation Ltd. (IPCL), India as an Engineer (Instrumentation). In September 1996, he then shifted to Reliance Industries Ltd., India as an Assistant Manager (Instrumentation). During his tenure at Reliance, he worked on various maintenance, shutdown and expansion activities at PVC Plant Instrument division. In October 1998, Nirav joined Institute for Manufacturing, University of Cambridge, UK, where he is a PhD Student under the supervision of Dr. Duncan McFarlane. His PhD thesis involved the study of Decentralized Coordination mechanisms in Continuous Process Optimisation and Control.
He is currently working in the UK with British Energy.

Alan Thorne
Laboratory Manager

Alan Thorne is the Program Manager for the Auto-ID Centre Lab at Cambridge University. Mr. Thorne graduated from Anglia Polytechnic University in Electronics and Control Systems and has a varied background in the field of Automation and Control. He has been involved in British Aerospace/IBM research projects as a systems engineer investigating flexible manufacturing systems on civil and military aircraft production. He has most recently been involved in projects relating to the development of novel AI-based control strategies.

Duncan McFarlane
Research Director Europe

Duncan McFarlane is a Senior Lecturer in Manufacturing Engineering in the Cambridge University Engineering Department. He has been involved in the design and operation of manufacturing and control systems for over fifteen years. He completed a Bachelor of Engineering degree at Melbourne University in 1984, a PhD in the control system design at Cambridge in 1988, and worked industrially with BHP Australia in engineering and research positions between 1980 and 1994.
Dr McFarlane joined the Department of Engineering at Cambridge in 1995 where his work is focused in the areas of response and agility strategies for manufacturing businesses, distributed (holonic) factory automation and control, and integration of manufacturing information systems. He is particularly interested in the interface between production automation systems and manufacturing business processes.

WHITE PAPER

Routes for Integrating Auto-ID Systems into Manufacturing Control Middleware Environments

Contents

1. INTRODUCTION

1.1. Background

Auto-ID technology is set to embrace the real world by providing a unique product identification mechanism that will allow manufacturers, distributors and retailers to seamlessly integrate their business processes across global supply chains. The key to this integration will be the provision of global Electronic Product Code (EPC™) for uniquely identifying individual products as well as a suite of networking tools for enabling the integration of product information within business decision-making and information sharing processes.

A key issue in developing Auto-ID technology is the requirement for a supporting information infrastructure. Unlike conventional automatic identification techniques such as bar-codes or RFID, which have remained limited to product-type identification only, the facility of instance-level product information in Auto-ID will lead to a multi-fold increase in the amount of data to be processed in real-time. How to handle such data effectively and economically will be crucial in developing the technology itself. Equally important will be the investigation of different routes along which this information passes – from sensing of tags to final use in business applications – so as to guarantee that only the right amount of information is retrieved at a particular location and that none of the useful information is discarded mistakenly.

Along the path of migration to fully Auto-ID compliant operations, there will also be a need to re-use relevant infrastructure that exists already in various application domains. In manufacturing control in particular, the use of bar-codes or RFID for product identification has been in place for quite some time and there exist a good understanding, both at the hardware as well as software level, for integrating product-type information within real-time control decisions. Similarly, modern industrial database systems have become increasingly sophisticated and easy-to-use for storing and retrieving large amounts of real-time, time-consistent information. A close scrutiny of such existing facilities could prove valuable to technology developers to effectively migrate to Auto-ID framework and could also prove useful to vendors for developing “out-of-the-box” design of manufacturing information systems.

1.2. Aims of the Paper

We refer to the set of applications managing the interface between real-time sensory data and business information systems as “middleware”. Focussing particularly on the middleware requirements in manufacturing information and control environments (see Figure 1), we investigate some of the above issues in this paper. The three principle aims of our analysis are as follows:

1. To review the existing use of automatic identification techniques in manufacturing control environment by understanding the operations of two to three dominant control system products available “off-the-shelf”;
2. To examine the capability of such systems for dealing with Auto-ID information and, in particular, managing the relatively high volume of real-time data generated on the production shop-floor;
3. To proposed and analyse different routes through which Auto-ID infrastructure could be effectively integrated within industrial data management and control environment.

Our analysis in this document is exploratory rather than normative. We identify several issues and strategies that can further guide the aforementioned integration process. The white paper s structured in the following way. In Section 2 we review the features of existing middleware in the manufacturing sector. In Section 3 we establish requirements for effective integration of Auto-ID data with manufacturing control environments, and in Section 5 propose a set of draft guidelines for managing the integration of an Auto-ID system into an existing industrial environment .

2. MANUFACTURING CONTROL MIDDLEWARE

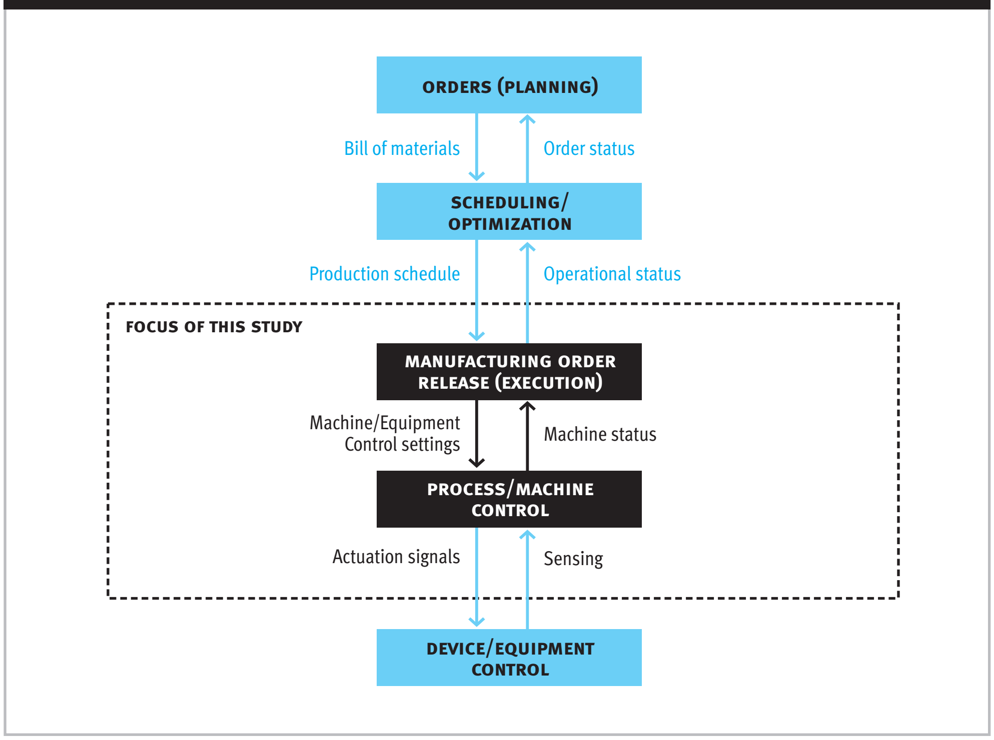
2.1. Information Architecture

Computer systems have been used to manage manufacturing operations for more than three decades. Early systems were custom made with software/hardware developed for a particular manufacturer's operating style. The introduction of Computer Integrated Manufacturing (CIM) in the late 80's led to the development of several generic architectures that allowed large-scale manufacturing information systems (MIS) made available off-the-shelf in market. Modern design of such information systems also aim to provide capability for integrating real-time production data available on shop-floor with the business planning and management functions at an enterprise level.

Figure 1 depicts a schematic of information flow within a typical manufacturing information system. MRPII/ERP systems, CAD/CAM systems, and industrial controls are prime examples of well-recognized components of manufacturing information systems. The focus of discussion in this document is mainly on the middleware environment – the interface between real time sensing and MRP2/ERP syetems – and hence encompasses the real-time control and manufacturing execution (MES) functions. (We refer the interested reader to a previous Auto-ID Center White Paper [2] for a detailed introduction to the different elements in a manufacturing business information environment.)

Figure 1

MANUFACTURING CONTROL HIERARCHY



The term Manufacturing Execution Systems (MES) in particular represents a collection of functions that enable optimisation of production activities from order launch to finished goods. Using recent production data, the MES guides, initiates and responds to plant activities as they occur in an optimal and effective manner. In fact, MES in itself encompasses various sub-functions as shown in Figure 2. The figure also identifies three key functions where the product type information is extensively used.

Our interest in this document is to understand how real time product information available on the shop-floor can be routed through real-time control systems to the MES level and in particular to the three sub-functions identified in Figure 2.

Figure 2:

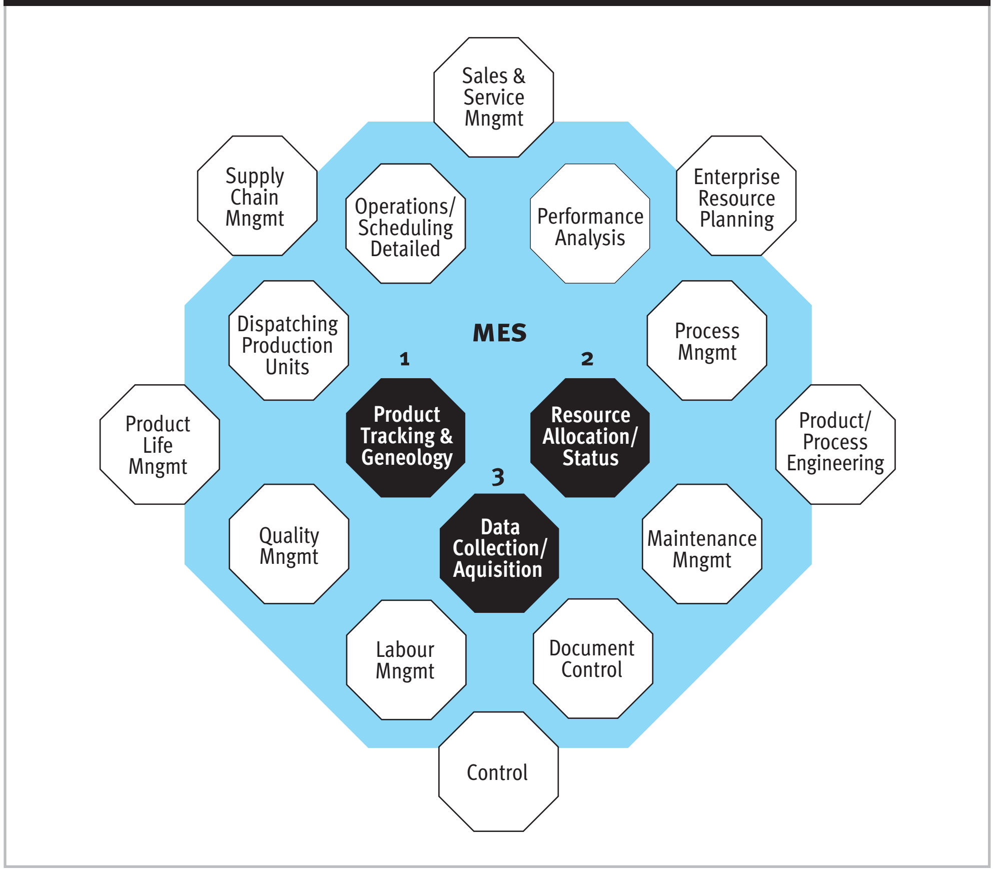
1 Monitoring the progress of units, batches, or lots of output to create a full history of the product

2 Guiding what people, machines, tools and materials should do, and tracking what they are currently doing or have just done

3 Monitoring, gathering, and organizing data about the processes, materials, and operations from people, machines or controls.

[Source: MESA International, 1997]

MESA MODEL



2.2. Dominant Commercial Products and Their Trends of Development

In recent years, numerous commercial products have appeared off-the-shelf in market that can provide industrial control and MES functions shown in Figure 1.

At the real-time control level, an important issue in the design of manufacturing control systems has been the deterministic and reliable character of operations. With the growth of networking technology however, it has become possible to distribute the control functionality directly on the shop-floor and hence closer to the manufacturing operations where the actual events occur. The early developments at MES level remained more-or-less isolated from the underlying real-time control. In recent years however, a lot of pure-play MES software products have been acquired by larger process control and automation vendors. For instance, almost all large process automation vendors (e.g., Siemens, Rockwell, Invensys, and Honeywell) currently have their own MES solution offerings that directly link with their real-time control products. These vendors have had better opportunities to position themselves in the MES market since they have detailed knowledge of the process automation and equipment area within organisations. Additionally, some of the other growing trends of development within manufacturing information systems could be summarized as follows (adapted from [12])

- **Integrated MES Suites:** Since the introduction of a standardised model for MES environments – the so called MESA model [1] in Figure 1 – many vendors have moved towards developing integrated MES suites or frameworks (e.g. Wonderware' FactorySuite, Rockwell's FactoryTalk, Siemens's SIMATIC IT framework) which contain many of the building blocks of MESA model shown in Figure 2. Such an approach removes the inconsistency due to "islands of automation" and enables seamless integration between different modules.

- **Acceptance of Product and Process Description standards:** Key international standards such as ISA-S95 (for integration of enterprise information systems with factory floor), ISA-S88 (for standardizing process recipes and procedural control) and IEC-1131 (for programmable control logic definition) have increasingly become norms for operations of most MES and real-time control products.

- **Integration of Business, Execution and Real-time Control levels:** Vertical integration of MES solutions with ERP at a higher-level and the real time control at lower level is becoming a key driving force for allowing consistent and efficient use of information.

Additionally, concepts such as OPC (OLE for process control) and Fieldbus networking have also gained major interests in providing open and standardized means of communication. The Microsoft platform and its flagship .NET technology are also being increasingly adapted to support the internet-enabled control architecture [12]. Summarizing, there is a growing focus in manufacturing information systems to follow the suite of decentralized, vertically integrated and open, standardized character in future.

2.3. Use of Product Identity Information in Manufacturing Control

A number of functions within the real-time control and MES domains can directly benefit from product identity information being made available on the shop-floor in order to enable them make better control decisions. The following list gives a sample overview.

1. Product Tracking and Genealogy
Perhaps the most relevant application that requires product identity information, the product tracking function provides visibility to where products and WIPs are at all times and their disposition. Such information as, which machine is working on WIP and at what time, which components or raw-materials are used and who are their suppliers, what are the ongoing lot/sublots, etc. could be analysed.

2. Automated Storage and Retrieval
Automated Storage and Retrieval (AS/RS) involves maintaining the WIP and finished goods' inventory levels. Important issues involving product identify information may include, sorting of incoming parts; allocating storage space for prioritised products or orders; routing parts to right locations during storing and retrieving procedures; and maintaining an account of inventory at all times.

3. Product Routing, Dispatching and Flow Control
Material handling involves transferring of products from one location to another. Products may have different aggregation or containment relationships. For large-scale transfers, multiple product types may share same routes or transfer mechanisms, in which case switching of product routes should happen as per product recipe steps. In discrete manufacturing, dispatching of products between machines becomes crucial when machines are process bottlenecks; more important products (i.e. products with tight deadlines, costly storage conditions) should be routed with higher priority. Product ID and related information could enhance many of these control functionalities.

4. Process Recipe Management
Products may have different "recipe" requirements in terms of sequences of instructions, for machining, assembly or disassembly, packaging, palletising or intermediate storage. The issue especially applies to automobile industry, but also to semiconductor and electronics industries, where cars or computers would be assembled with different components and would need separate procedures for doing such. Mixed packaging of items in consumer package goods is also a key application for product identity information.

Other potential applications of product identity information include forecasting of raw-material consumption, statistical process control, estimation of resource usage, and alarms and trigger management etc. For further discussion on such applications, refer to [5,6,13,2,3].

2.4. Previous Uses of Automatic Identification in Manufacturing Control

In order to understand how product identity information has been integrated previously into manufacturing execution and control environments, we now review the use of automatic identification for control purpose, as simple as a simple proximity switch, has prevailed for many years. The three principle forms in which automatic identification and data collection (AIDC) technology has been used previously can be categorized as: (i) discrete sensing, (ii) bar-codes and (iii) RFID systems. Other techniques such as optical vision or acoustic sensing have received little attention then in some specific applications.

1. Discrete Sensing
These include sensing mechanisms that detect the presence of product on a specific location. Examples range from a mechanical or proximity switch to more sophisticated infrared or laser sensors allowing primitive coding of items using a multi-bit numbering scheme.

2. Bar-codes
Bar-codes have been used in conjunction with discrete sensing for applications such as, (i) tracking or sorting of products on conveyors; (ii) identifying raw-materials used in product assembly, (iii) assessing work-in-process inventory, etc. Unlike retail applications, there has been a growing trend to use more sophisticated coding mechanisms such as alpha-numeric characters or 2D mechanism in order to allow storing larger amount information on printed labels. See Tompkins et.al. [14] for a detailed overview of various bar-code standards previously employed within industrial applications.

The usual concerns of "accessibility" and "line-of-sight" for bar-codes in retail applications also have equal significance in manufacturing environment. The use of bar-codes makes it necessary that only a single item can be read at a time. The items must be arranged in a fashion that bar-codes could be located right in front of the scanner. Moreover the speed at which scanning can be accomplished also limits the speed of conveyors or items moving along material handling systems. Harsh manufacturing environments such as chemicals or biological conditions may also hamper the bar-code labels thus making them inaccessible.

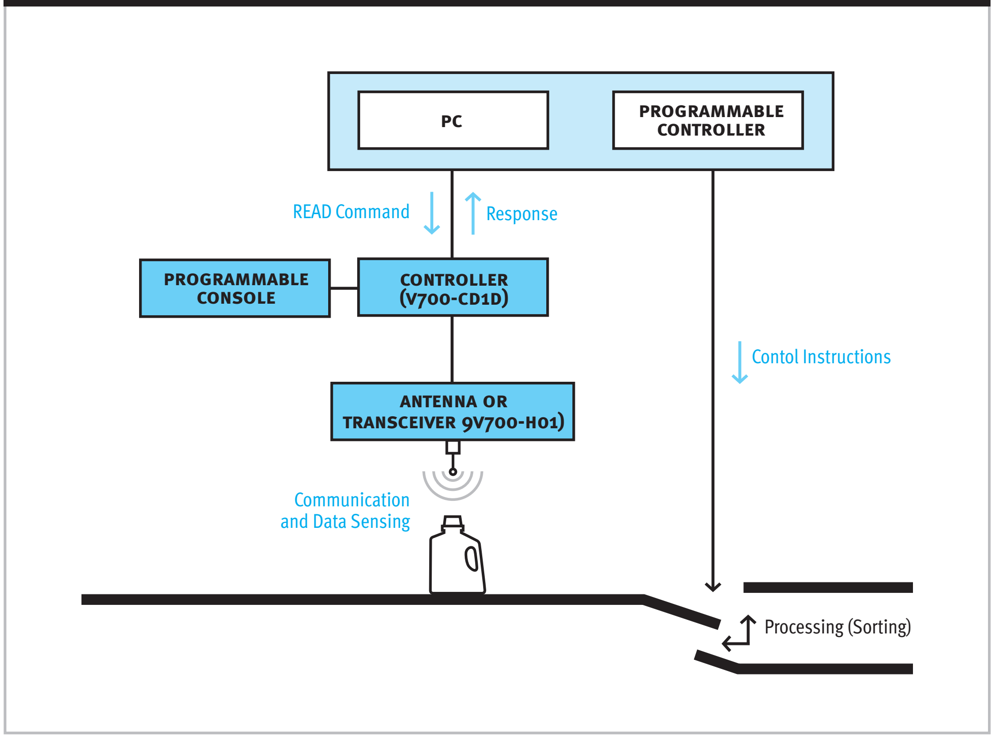
3. RFID
Use of RFID in industrial control has existed for over fifteen years, in most cases as a counterpart of bar-codes. Much similar to retail application, the use of RFID sensing in manufacturing control has been expected to alleviate the concerns with bar-codes due primary to the availability of remote sensing mechanism. This also enables scanning multiple tags or filtering of tag reads according to specific tag patterns.

Key applications of RFID apart from those of bar-codes include, (i) tracking and tracing of aggregates such as pallets or unit loads, (ii) storing machining instructions on tags to ensure consistency between different machining stations (in automobile applications), and (iii) ease of managing AS/RS systems, etc.

Figure 3 below depicts an example illustration of interfacing a RFID reader with a programmable logic controller or a personal computer responsible for implementing a control routine. The tag readers routinely supply product identity information back to PLC, based on which PLC decides which control actions should be taken and how it should be implemented. Note that, the same infrastructure could also be used for a bar-code based sensing mechanism.

Figure 3

USE OF RFID TAGS IN REAL-TIME CONTROL OF SORTING CONVEYOR



Both bar-code and RFID readers normally supply tag information as an ASCII string. This has led to interfacing either type of readers directly with the ASCII modules of programmable logic controllers (PLCs). Detailed tag information could then be extracted by using "string" processing instructions within ladder-logic or by using "structured text" language. Some vendors also support the use of higher-level language such as BASIC (e.g., Omron) or C/Java (e.g., Rockwell Automation) for such purposes.

A survey of industrial RFID tools suggests that the requirement of deterministic and reliable communication has led RFID vendors to provide with standardized communication tools such as the use of Fieldbus protocols (e.g. Devicenet, Profibus) for communicating between RFID readers and PLCs. Another trend, apart from the usual ASCII interface, has been to provide a separate hardware module that fits into the backplane of PLCs (e.g. EMS, Balogh and Omron RFID systems). Such modules connect to more than one reader and provides PLCs with the product identity information in a structured memory form that can be directly accessed within ladder-logic or similar programmable instructions.

Unlike retail applications however, there has been little thrust observed so far in industrial RFID applications for reducing the cost of RFID tags as against the possibility of incorporating increased amount of information on tags and/or to increase their read range by means of battery-powered operations. In an automobile application for example [15], such tags would carry product assembly information about which parts are to be fitted on the car body when the body would pass through a specific assembly operation. The RFID system would sense the tag and subsequently interact with real-time controllers to drive the control actions accordingly. In a packaging goods application, it was also suggested that the tags be re-used once the pallets or products move out of the warehouse.

Since the conventional use of RFID has been limited to product-type level information only, the amount of information generated in any such application is expected to remain bounded by the variety of products handled at a time. Moreover, the conventional strategies for dispatching products along the production routes do not advocate mixing of multiple different products along production life-cycle. That means, in a typical production cycle, WIP or finished items would pass in lots or sublots and would observe exactly similar processing operations within each lot or sublot. As a result, detecting the identity of aggregates, e.g. pallets or unit-loads, would automatically guarantee that all items within such aggregates have features similar to those that the pallet as a whole is ought to possess. This logic partly explains an observation why the use of RFID technology in manufacturing control has remained sporadic in nature compared to bar-codes. Perhaps this is also the reason that the RFID technology has received much less developments in terms of its integration within industrial control environments, for instance, the provision of facilities for processing and filtering of tag information, etc. has not as yet appeared to be a major issue.

2.5. Benefits from the Auto-ID Technology

The instance-level information available in the Auto-ID system has several potential benefits including the capability to link product identity (in the RFID form) with network technology that will allow retaining the product life-cycle information throughout a global supply-chain. Moreover, the global standard for the Electronic Product Code (EPC™) will enable that such life-cycle information be made available in a standardized form for use in any specific application. In manufacturing control, the standards would lead to increased ease in maintaining the product tracking and genealogy related information as well as the dynamic use of process recipes in manufacturing of products or product orders.

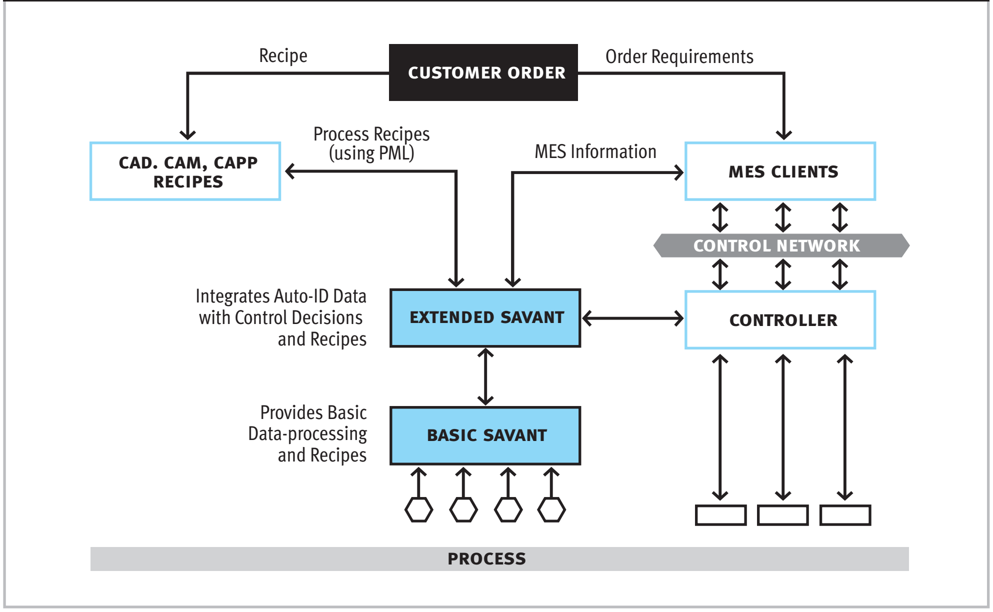
The Auto-ID Centre's perusal for standardized read/write protocols across the entire information channel, i.e. tag-to-reader and reader-to-host (which can be Savant™ or other modules) enables that a highly open communication infrastructure be developed for integrating relevant components from different makes of tags, readers or higher-level data processing modules.

3. REQUIREMENTS FOR INTEGRATING AUTO-ID INFRASTRUCTURE WITHIN MANUFACTURING CONTROL

Figure 4 depicts an outline of integrating the Auto-ID infrastructure within a manufacturing control environment. The key component among others therein is the role of so-called Extended Savant™ [18], which forms a bridge between the product identity information available from basic Auto-ID infrastructure (i.e. tags, readers and basic Savant™) and the operations of manufacturing control systems including the process recipe information and the control requests from MES clients and real-time control level.

Figure 4

INFORMATION FLOW WITHIN AN AUTO-ID INTEGRATED CONTROL ENVIRONMENT



In this section, we examine a set of different requirements in reference to the design and integration of Extended Savant™ component. Our aim is to understand what an "efficient integration" of Auto-ID technology means for manufacturing control. This set of requirements also lead to a set of different routes that an engineering team can consider in developing an Auto-ID integrated control system. Note that the selection of an appropriate route has several implications on the operations of Auto-ID integrated tools, and an appropriate level of care must be taken about which route suites to a specific application being developed.

We in particular analyse these requirements from the point of view of two different perspectives:

1. Software/hardware development perspective
2. Information processing and data management perspective

Each of these perspectives is discussed in detail in the following sub-sections together with issues for efficient integration. We also review relevant literature therein to discuss potential design approaches that could be used.

3.1. Software/Hardware Development Perspective

Given the basic Auto-ID infrastructure (namely Edge Savant™ [18] and readers) in Figure 4, the first obvious need for integrating Auto-ID technology within manufacturing environment would be to develop an appropriate form of Extended Savant™ module. Extended Savant™ could potentially be implemented in various combinations of software and hardware forms. The four critical requirements that any such implementation will need to satisfy apart from the usual concern of open, standardized nature of resulting implementation, can be listed as below.

1. Ease of deployment and use
2. Speed of response
3. Reliability, Fault-tolerance and the Deterministic Character of Operations
4. Distributability of Software/Hardware Modules

Ease of Deployment and Use

Ease of design and deployment refers to how quickly end-users can build an Auto-ID integrated control system, both within an existing plant as well as in a green-field plant. End-users will want that the software tools available in the market provide suitable built-in library of data processing modules that could be simply connected together in order to bring the system up and running. Table 3 in Appendix A gives a taxonomic classification of several processing modules that would normally be used in a manufacturing control application. Note that this issue relates to the Basic and Extended Savant™ modules where developing different processing modules such as filters, loggers or queues from scratch would become a critical bottleneck in the design procedure.

Recently, there has been also a growing trend for developing purpose-built control solutions for specific industrial application domains, e.g. packaging or material handling operations within specific industries. This trend may also apply to control solutions integrating Auto-ID. For example, the manner in which product identity information is used in a semiconductor industry will be significantly different compared to its use within an automobile or consumer packaged goods application. A tailored-made design and configuration of Extended Savant™ for targeted application could significantly reduce the design effort.

Finally, it will be necessary that the Auto-ID infrastructure is implemented in a reconfigurable form. Scenarios such as machine failures, adding a new machine or a new class of products etc., which all need reconfigurable control design will inevitably also need reconfigurable sensing and data processing means within Auto-ID infrastructure. Against this design-time reconfigurability, the run-time or dynamic reconfigurability will equally be important especially for both types of Savant™ sub-modules, but also for the reader operations. An example to this is the dynamic tuning of filters within basic Savant™ or readers for a change in the product orders or process operations. Different physical configuration of products may lead to different read responses that readers may receive and subsequently the amount of data processing that may have to be performed. The same reconfigurability argument also carries over to continued operation of Auto-ID infrastructure amidst variations in the mix of products and the product dispatching strategies to be used.

Speed of Response

The amount of information generated in an Auto-ID based control application will be a number of times larger than the conventional discrete sensing or even the use of bar-codes or RFID. Processing such large amount of information will need not only efficient hardware infrastructure but also the procedures or algorithms used in their implementation. For instance, a string processing filter can be implemented in (a) a pure hardware form, (b) a mnemonic language or (c) a higher-level programming language (e.g. C, Java). Of course, while a higher-level language affords flexibility or ease of development, it may not yield the data processing throughput that the former two options would endeavour. It is thus quite likely that the processing modules which are required in almost all circumstances, such as filtering of tag patterns or the removal of mysterious reads, etc. are implemented in a hard-wired form, while the other less used modules in a flexible, programmable form. As discussed subsequently, the speed of response can also be increased by optimising the information channel, i.e. the amount of information, its route and the timeliness for its reaching to destination.

Distributability of Software/Hardware Modules

Within an industrial application, the number of readering points can initially be expected to be in the range of hundreds if not thousands. We note that designing such a large information infrastructure will inevitably require a modularised design of sub-components within integrated control architecture.

Modularity by distribution is one potential approach. Two different types of distribution are possible particularly for the Basic and Extended Savant™ modules: Horizontal and Vertical. The horizontal distribution refers to how Auto-ID components are distributed across plant-wide operation. Different possibilities may include cell-level or even PLC-level distribution of Savant™ and other data storage modules. The vertical distribution on the other hand refers to how the information channel from sensing of tags to the final usage of data actually operates in synchrony with the control hierarchy. Different possibilities here would be to develop a hierarchy of Savant™ modules, much similar to the edge and internal Savant™ concept discussed earlier in the Auto-ID literature. Such a hierarchy would normally operate in parallel to the physical aggregation hierarchy (namely, workstation, cell, line or train and plant) as well as control hierarchy (as in Figure 1). Note that the vertical distribution indirectly also refers to horizontal distribution as we show later in the next section.

Reliability, Fault-tolerance and Deterministic Character of Operations

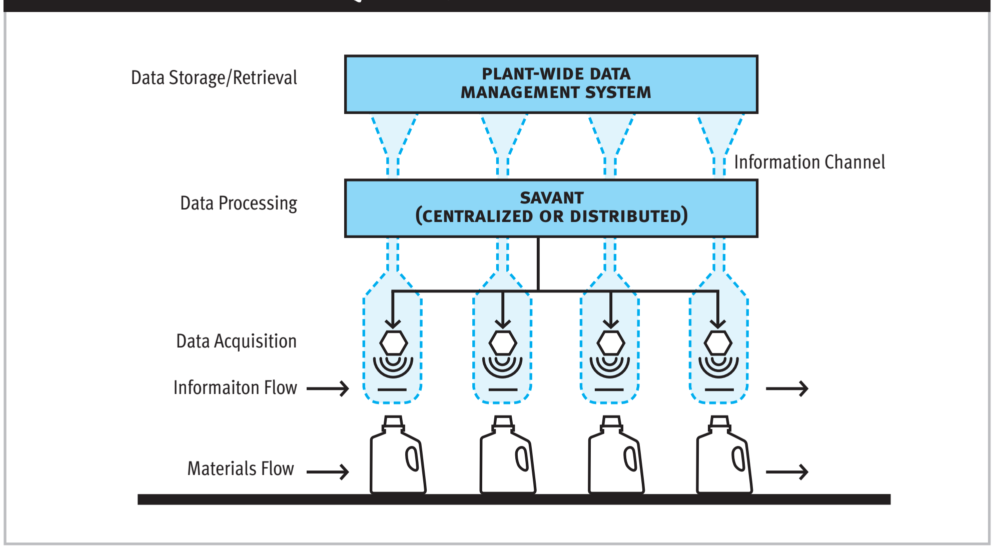
Unlike retail or supply chain logistics applications, the industrial operations must operate in an entirely reliable and deterministic manner, not only at the real-time control level, but also at the MES or the planning and scheduling level. This determinism of operations will become a particular issue in tag sensing because the slotting or the back-scatter protocols used are themselves of non-deterministic nature. Although UHF tags provide better accuracy of tag reads in this respect, the back-scatter protocol used therein does not guarantee such accuracy when tags are moving and/or new tags arrive during when a particular read-cycle is ongoing. The argument also applies at the higher-levels, i.e. for reader-to-host communication. The use of Ethernet protocol for such communication is not preferable considering its non-deterministic character. The use of fieldbus protocols has become standard in recent years in the industries, and it will be crucial that Auto-ID vendors adapt their infrastructure to make best use out of fieldbus protocols.

3.2. Data Management Perspective

The different components within the assembled Auto-ID infrastructure lead to a multi-tier hierarchy of information channel as depicted in Figure 5 below. This entire data management system can be viewed as a large-scale database in which the information is sucked from the shop-floor and inserted at appropriate locations and time. Note that the figure represents an upward-flow of information only; how such information is utilized for control purposes (as shown in Figure 4) is not depicted therein. The top-most tier of data storage represents the function of data historian within MES operations (e.g. Wonderware's IndustrialSQL server).

Figure 5

INFORMATION CHANNEL FOR ACQUIRING SHOP-FLOOR INFORMATION



For efficient integration, it will be necessary that the Auto-ID tools are provided with suitable means for managing the entire data transfer in an effective and time-consistent fashion. In this respect we consider the following three issues as the major requirements for data management systems:

1. Optimality of Information Pattern
2. Distribution of data management
3. Real-time Storage and Data Retrieval Procedures

3.2.1. Optimality of Information Pattern

The biggest challenge in using Auto-ID within a manufacturing environment will be to acquire large amounts of real-time information and to process it across different levels of information hierarchy. Ensuring optimality of the information pattern will be crucial to guarantee that the information does not pass through unacceptably long and slow routes which are prone to malfunction. Some options for this optimisation can be listed as follows:

- **Tag Data Filtering Early in the Information Channel**
  On the path from tag sensing to final usage of data, it will be quite likely that the amount of information that is finally stored at the data storage level will be much less than the amount that has been sensed. This is due primarily to various filtering operations involved along the path. Filtering of unwanted information much early in the channel could reduce the information processing overhead being passed to the higher-levels. The same logic also applies to selection of an optimum value of the frequency or time-duration at which information is accessed from the shop-floor. For instance, in an automated storage application, a reader may not need to communicate with tags as often considering that items would largely remain steady for longer periods of time.

- **Pull vs. Push type Data Delivery Mechanisms**
  Three different types of data delivery mechanisms can be used within a typical data management environment such as manufacturing information systems [8]. These are pull, push and hybrid. In the pull mode, the transfer of data from data servers to clients is initiated by a client pull. When a client request is received at a server, the server responds by locating the requested information. The servers must be interrupted whenever the data is required by clients. The information that clients can obtain from a server is limited to when and what clients know to ask for. Alternatively, in a push mode of operation, the transfer of data from servers to clients is initiated by a server push in the absence of any specific requests from clients. The servers decide themselves which data would be of common interest to clients, and when to send them to clients. A server can disseminate information to either an unbounded set of clients (random broadcast) or a selective set of clients (multicast) who belong to some categories of recipients. The hybrid mode of data delivery combines the client-pull and server-push mechanisms. One such approach, called continual query approach, would be to combine pull and push modes: namely, the transfer of information from servers to clients is first initiated by a client pull, and the subsequent transfer of updated information to clients is initiated by a server push.

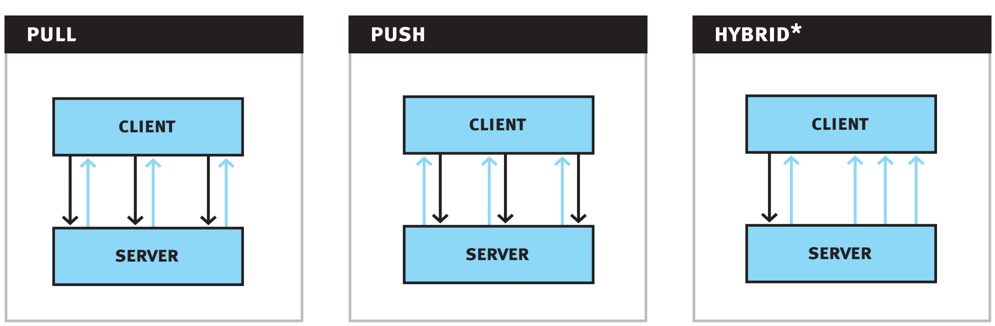


Figure 6: Pull, Push and Hybrid data delivery mechanisms

Pull: ↓ Request ↑ Data
Push: ↓ Acknowledge ↑ Data
Hybrid: ↓ Initial Request ↑ Data

* Client-pull and Server Push

In their normal or conventional mode of operation, we can see that the industrial MES or data management systems operate in either a pull mode or the hybrid mode as stated above. The clients here can be any of the MES functions while the servers are the information suppliers, such as readers or Savant™ modules within Auto-ID infrastructure. Normally, the clients would initialise Savant and/or readers with which types of tags or information they would like to receive; subsequently these servers would periodically or conditionally pass the information to clients. Although the pull-mode of operation would work satisfactory when the objective is to simple access the information on shop-floor, there are cases where it may not be optimal in a closed-loop control environment. An alternative push-mode of operation would be a better option, particularly when the servers could be programmed with some useful mechanics that allow them decide what information they have observed, and where such could be utilized for control purposes. A simple example to this is the design of sorting conveyors, where a reader senses some tag information and notifies it to a focussed set of recipients (which could be an another reader, a localized Savant™ module or a control function) about which items have just passed through the reader. Note that in a pull-mode of operation, such information would only be routed to these recipients when the higher-level clients next receive the information and operate on it during the next routine cycle. At a real-time control level (i.e. PLC operations), the delay in this closed-loop processing of information could become excessive similar to what has been observed in a lab-scale demonstrator being developed at Cambridge Auto-ID Centre.

– **Use of Tags with Memory (Class II Tags)**
The use of tags with memory (i.e. Class II or higher class tags) has not been advocated in the Auto-ID literature so far due primarily to their potential high costs. In a manufacturing control environment however, this higher cost of Class II tags could be justified if such an option can reduce the information overhead on the network. Class II tags could be particularly used for holding PML data relating to tracking of aggregates, such as pallets or unit-loads, and could potentially be re-used once items leave the factory premises. The important advantage of Class II tags comes in their capability to store state information for specific item being tracked, and hence representing a saving in terms of communicationss. This implicit object oriented character could help reduce the large amount of information transfer over the network that could otherwise become bottleneck in case of Class 0 or I tags. On the other hand, an important concern linked to Class II tags is related to localization of information that could only be sensed when tags passes through readers. Unlike to that a networked storage of information provides with a global access of tag information without worrying on repeated accessing of tags.

By combining both, we see that a hybrid option comprising Class II tags together with the networked storage of information would be a sensible option for control environment. The tags could potentially provide a redundant storage of the same information that can also be made accessible over network. A client needing the product identify information can then optimise the route from which it can access the product information. For instance, for the lower-level, real-time control loops, one could potentially access the information directly from tags without requesting the same from a centralized database server, while for higher-level clients such as MES functions, which operate on more than one items at a time could use the networked storage. Effectively, the hybrid option can be aimed to minimize the information overhead and provide with a better distribution of information that a purely networked storage of information may not. And, it is quite likely that such benefits could balance the increased costs of tags.

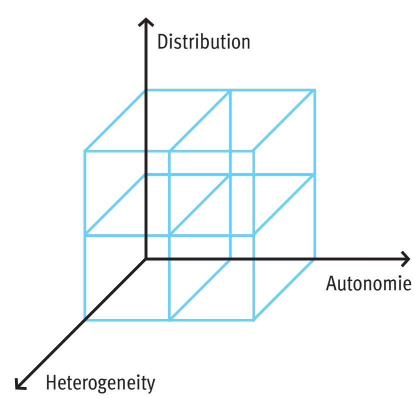
3.2.2. Distributed Data Management

The concept of distributed databases is not new. In manufacturing control however, it apparently has received far less attention so far. It is however quite likely that the explosion of information generated by the use of Auto-ID will need some means of employing distributed databases within manufacturing domain. Özsu et.al. [8] give a comprehensive overview of the distributed database technology from the perspective of commercial tools (e.g. Oracle, PostgreSQL, SQL Server). The authors classify three dimensions in which distributed databases may be put together (see Figure 7): Autonomy, Distribution and Heterogeneity. The distribution dimension deals with distribution of data among multiple sub-databases, each associated with a separate division or localized environment. Autonomy on the other hand refers to the distribution of control. It indicates the degree to which individual DBMSs can operate independently. Autonomy depends on a number of factors such as whether the component systems exchange information, whether they can independently execute transactions, and whether one is allowed to modify them. While heterogeneity refers to hardware heterogeneity and the differences in networking protocols to variations in data managers, which include data models, query languages and the transaction management protocols. As we show subsequently in the next section, applying distributed DBMSs in Auto-ID based control would be useful, both to guarantee increased speed of response as well as to improve reconfigurability of the Auto-ID information channel.

The horizontal and vertical distribution of Basic and Extended Savant™ modules discussed in the previous sub-section automatically leads to distribution of their internal databases. A similar distribution will be necessary at the data storage and retrieval level in Figure 5. Moreover, any such distribution should be perceived from the data management angle as discussed by Özsu et.al. [8] and depicted in Figure 7. Issues such as concurrency control, data localization and communication tradeoffs etc. would be crucial to such an analysis.

Figure 7

IMPLEMENTATION ALTERNATIVES FOR DISTRIBUTED DBMS



3.2.3. Real-time Data Storage and Retrieval Procedures

The relational model of DBMS is considered unsuitable for real-time applications in a manufacturing control environment, primarily due to the very fast and large amount of data generated within manufacturing applications. Moreover, the conventional SQL language does not directly support temporal or time-series data. These limitations obviously multiply manifold when the instance-level Auto-ID information is required to be considered.

The concept of real-time DBMS as already used by many industrial vendors (e.g. Wonderware's IndustrialSQL server) would be a key necessity for storing and retrieving such large information. The RIED (Real-time In memory Event Data) module in the Savant™ reference implementation also considers such means in dealing with Auto-ID information.

Stankovic et.al. [10] and Ramamritham [9] discuss in detail the different issues involved in designing real-time database systems. Stankovic differentiates between two different types of timing aspects in data management: temporal and real-time. The **temporal** databases deal with information containing timestamps, while the **real-time** databases concern with timing constraints associated with database operations (e.g. deadlines for completing transactions). The temporal DBMS need not necessarily operate in real-time. To guarantee such, it is necessary that efficient scheduling mechanisms are employed for sequencing of prioritised transactions, i.e. transactions with deadlines together with the amount of data being stored or retrieved. Stankovic et.al. also discuss different options for implementing real-time aspects in database design, including the use of in-memory storage of data, different buffer management policies, and the disk storage scheduling.

As pointed out above, many of the improvements as suggested here have already been in place in some of the industrial DBMS systems, and an end-user wanting to develop Auto-ID infrastructure can significantly benefit from studying different such features already available within their existing implementation of data management tools.

4. DIFFERENT ROUTES FOR AUTO-ID INTEGRATION

In this section we draw on the different perspectives described in the previous section to illustrate a few possible integration routes. The configurations highlighted here may help control system developers as well as system integrators to understand the issues such as where the information is originated, how such information flows within the system, which specific configuration is optimal in terms of design effort, how the information channel can be best maintained subsequently, etc.

4.1. Horizontal Distribution of Savant™ Functionality

The first important issue in the Auto-ID to manufacturing execution system integration process is to implement the Savant™ functionality (both Basic and Extended) as a part of plant-wide control design. Savant™ can be distributed both horizontally (along physical, plant-wide dimension) as well as vertically (along the information hierarchy). In this subsection we first discuss the horizontal distribution. The vertical distribution is discussed in the next subsection.

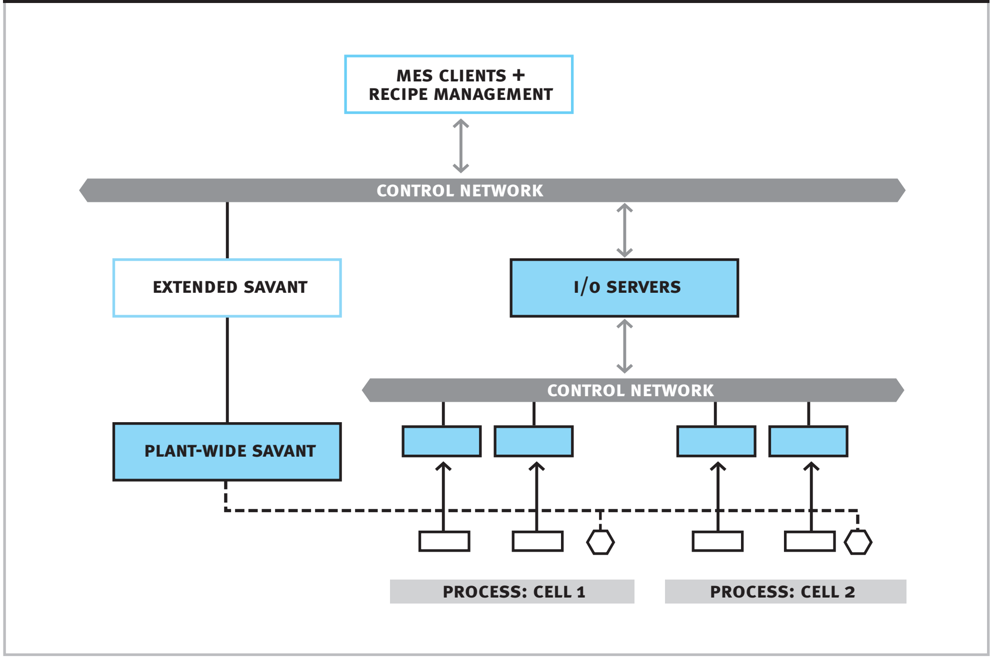
We consider three principle options: (i) centralized implementation, (ii) cell-level distribution and (iii) PLC-level distribution. The third option primarily is further divided into two different designs as discussed subsequently.

4.1.1. Approach 1: Centralized Plant-wide Design of Savant™

As advocated in the Auto-ID literature so far, this design as shown in Figure 8 represents the most straightforward means of putting an Auto-ID system in-use for manufacturing control. All Auto-ID components are implemented more or less independent from the rest of the control system and are centralized together at a single location. The information is primarily acquired from readers and processed by the basic and extended Savant™ modules. This data is then integrated with control operations by extended Savant™ together with the higher-level MES clients.

Figure 8

APPROACH 1: CENTRALIZED PLANT-WIDE DESIGN OF SAVANT

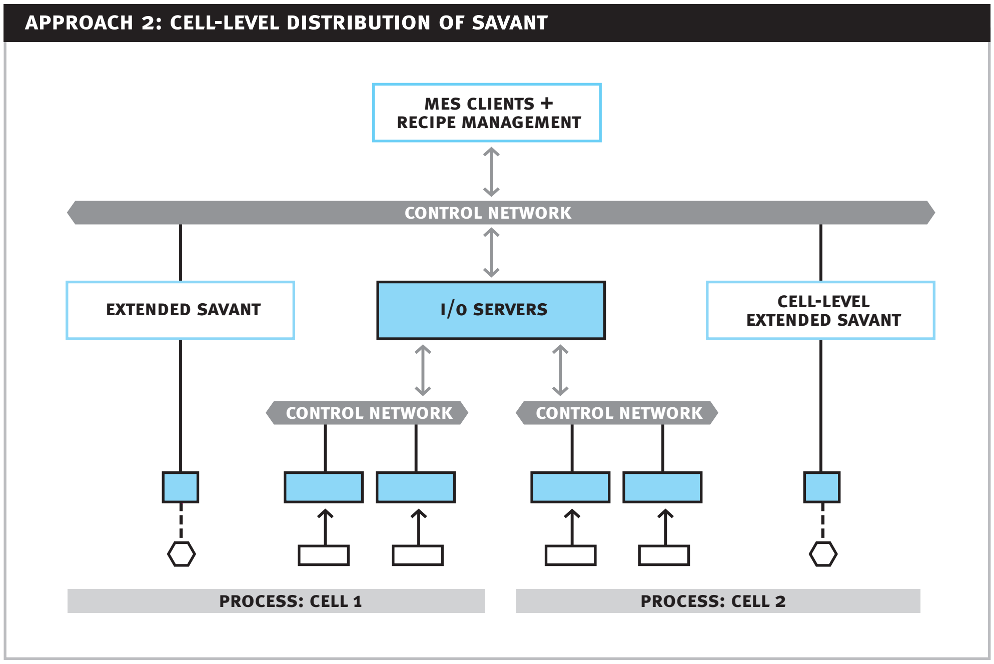


4.1.2. Approach 2: Cell-level Distribution of Savant™

When the amount of information to be acquired grows, it is advisable that the Savant™ module in the previous option is distributed into different component servers, each serving a separate section or cell of plant operations, e.g. assembly, packaging, material handling etc. Figure 9 depicts the resulting configuration. It is clear from the figure that the suggested distribution should provide increased flexibility and ease of deployment of integrated design compared to a purely centralized implementation.

Note that such distribution need not be reconfigurable considering that cell-level Savant™ modules may depend on other Savant™ modules for tag or recipe information. Two different remedies could be considered here to retain the reconfigurability: (i) store all or at least the common information between multiple, localized Savant™ modules on a centralized location (possibly as a part of MES data historian) so as to allow all localized modules to operate independently, or (ii) to provide appropriate means of data delivery mechanisms between localized modules so as to ensure that the information channel remains easily adaptable (see description of pull vs. push mechanisms in the previous section).

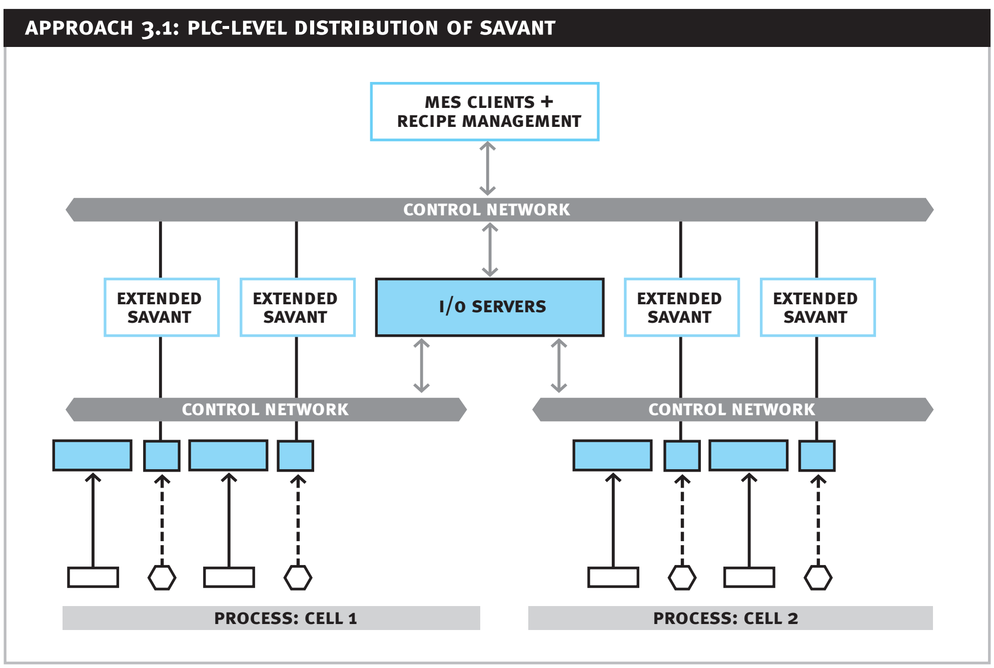
Figure 9



4.1.3. Approach 3.1: Machine-level Distribution of Savant™

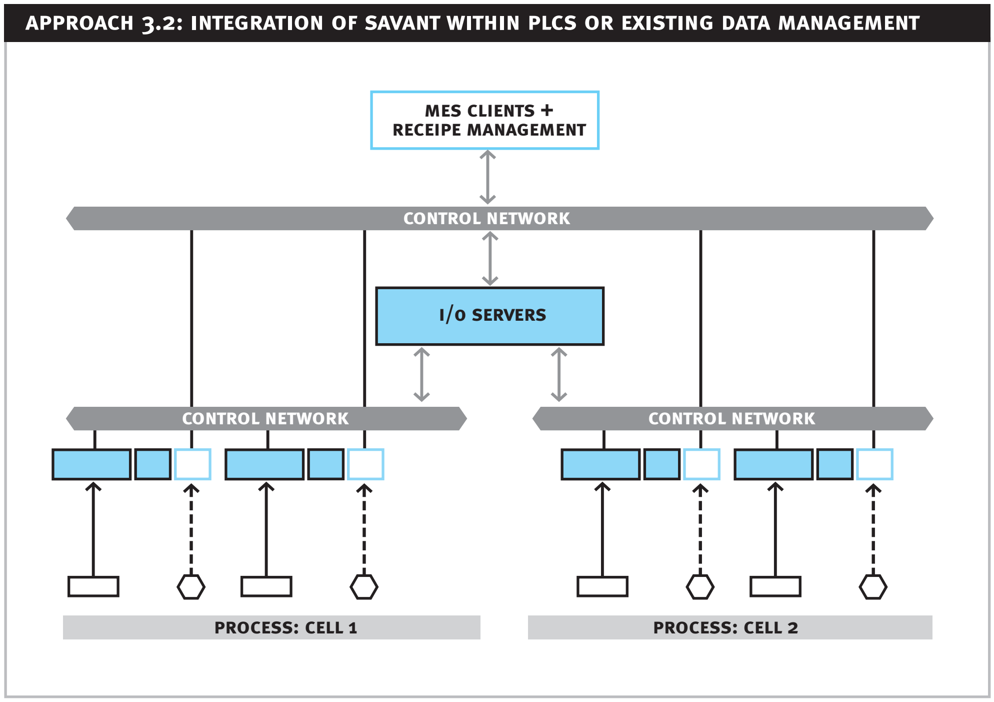
The cell-level distribution of Savant™ could be further enhanced by distributing the functionality at PLC level as shown in Figure 10. Each PLC may now have associated with a separate basic and ex tended Savant™ module that co-ordinates the Auto-ID information (namely tag sensing and the recipes) with the real-time control operations. Note that the issue of reconfigurability becomes much more important here considering that localized Savant™ modules may remain heavily linked with each other and at least within each cell operation. It is clear that this direct implementation of Savant™ next to PLC should lead to a significant increase in speed of response compared to both previous options.

Figure 10



4.1.4. Approach 3.2: Integration of Savant™ within PLCs or Existing Data Management Tools
A significant ease of design can be achieved in the previous case if the localized Savant™ modules are directly embedded within PLCs themselves. The resulting configuration is shown in Figure 11. Much similar to the RFID interface cards supplied by many vendors (e.g. EMS, Balogh), the new module should provide direct integration of Auto-ID information and the real-time control code.

Figure 11



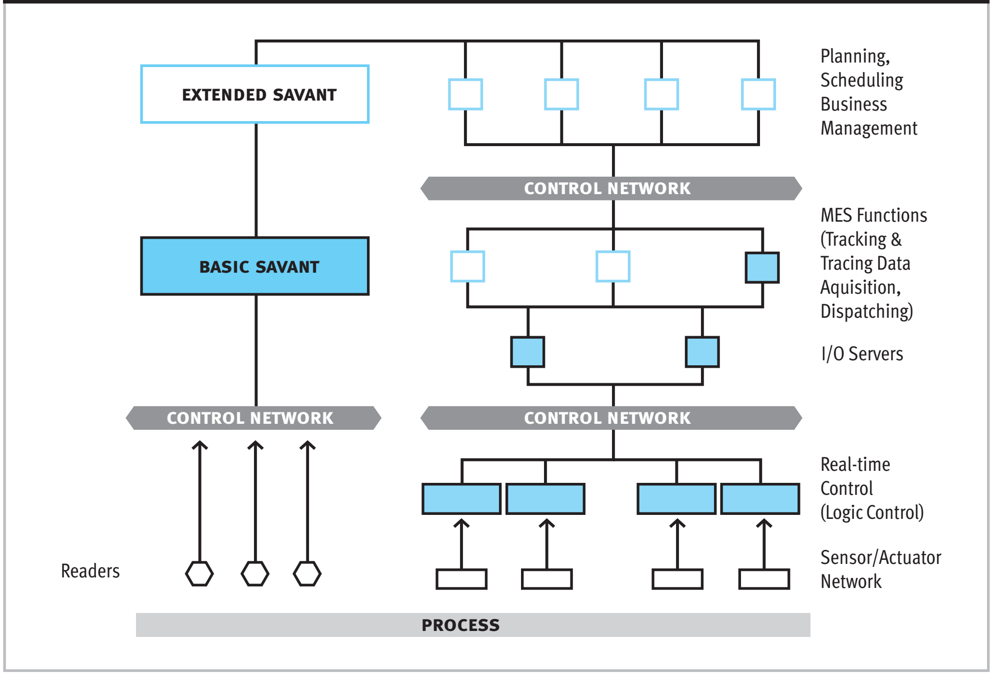
4.2. Vertical Distribution of Savant™ Functionality

The horizontal distribution of Savant™ functionality represents one facet of its distribution. In all four cases shown in the previous sub-section, we have assumed however that the Auto-ID information is simply integrated through appropriate MES clients at the top-level in the control hierarchy. This long information channel could be broken down by disintegrating the combination of basic and extended Savant™ components vertically. Such vertical distribution, together with an appropriate combination of horizontal distribution should then allow for a time and resource optimised information pattern wherein the Auto-ID data is used exactly where it is needed in the control hierarchy and without following a long, top-down routes.

4.2.1. Approach 4: Integration of Auto-ID Information at Business Management Level
As depicted in Figure 12, the first option for vertical distribution represents a similar layout as it was for Approach 1 in the horizontal distribution case, namely that the Auto-ID information is integrated at the business management level only. The information may be presented here in a consolidated form and may only be used to make higher-level management decisions.

Figure 12

APPROACH 4: INTEGRATION OF AUTO-ID INFORMATION AT BUSINESS MANAGEMENT LEVEL

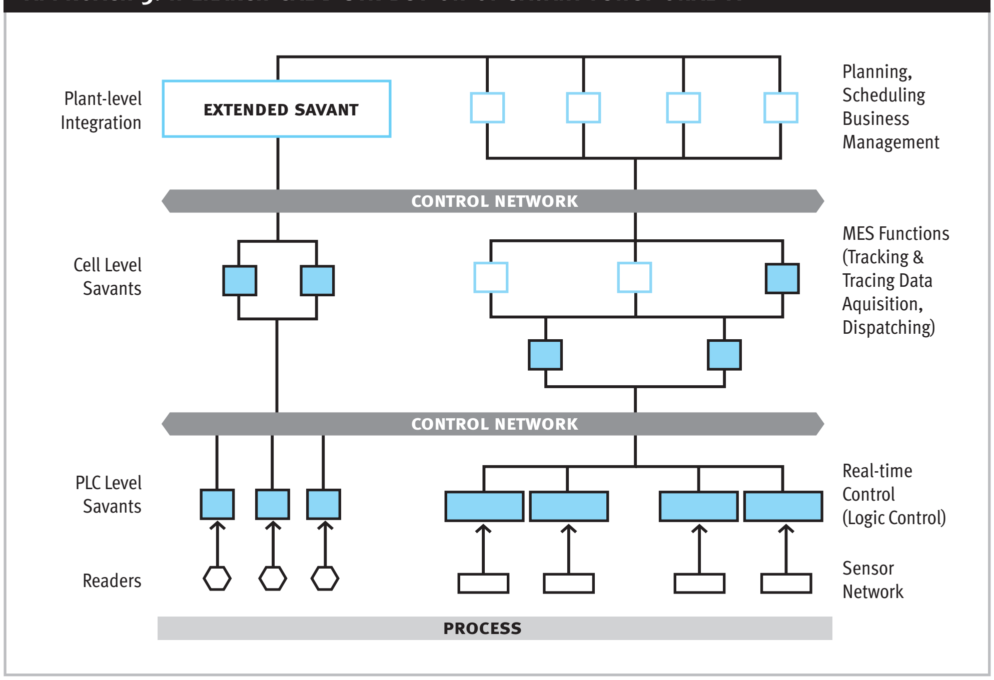


4.2.2. Approach 5: Hierarchical Distribution of Savant™ Functionality

In the next immediate instance depicted in Figure 13, the basic and extended Savant™ modules could be further disintegrated in order to parallel their functionalities with all three levels in the control hierarchy, namely business management, MES and the real-time control operation levels. This concept of a hierarchy of Savant™ modules parallels to the edge and internal Savant™ concept in the Auto-ID literature [18].

Figure 13

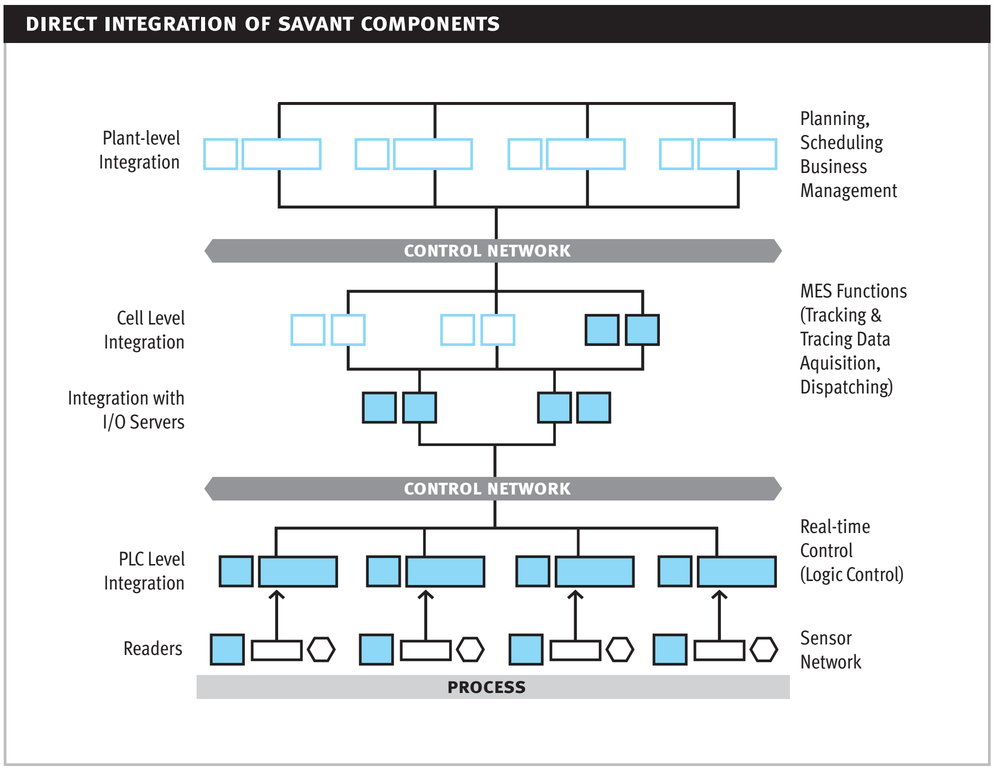
APPROACH 5: HIERARCHICAL DISTRIBUTION OF SAVANT FUNCTIONALITY



4.2.3. Approach 6: Direct Integration of Vertically Distributed Savant™ Components within Control Hierarchy

Finally, the vertically distributed components of Savant™ modules could be integrated entirely within different levels of control hierarchy. Figure 14 represents the resulting configuration. Thus each control system component now possesses its own separate processing module for handling the Auto-ID information. The information thus passes up in the hierarchy exactly in a similar manner and in parallel to the discrete sensing in the conventional control operations.

Figure 14: Direct Integration of Vertically Distributed Savant Components within Control Hierarchy



5. A RECOMMENDED PROCEDURE FOR AUTO-ID INTEGRATED CONTROL DESIGN

In this final section of the paper, we describe a draft set of guideline procedure for integrating Auto-ID into manufacturing control environments. The steps listed here are largely based on the experience gained in the development of Cambridge Auto-ID Centre's demonstrator environment.

1. Perform a Use-case Analysis [7] for identifying relevant business and production processes

A use-case based pre-cursory analysis [7] should be first performed in order identify a set of business and production processes where the use of Auto-ID will have substantial benefits. The analysis should also quantify the benefits by overlooking the impact of new technology on existing procedures, e.g. labour saving, hardware/software costs, increased business profits etc. Such an analysis may also give a general overview of how the Auto-ID enabled system would operate in terms overall information architecture. It is expected that the analysis remain generic at this stage. It may or may not refer to specific components in the actual system to be deployed. It should rather define the goal of the whole project in terms of what is required (must, should have and nice to have) and how to get there.

2. Define control logic for Auto-ID integrated operations
Given the production goals and control requirements, the definition of control logic is the critical next step in identifying the closed-loop information pattern. The definition should take into account the presence of product identity information as well as the redundancy measures.

3. Categorize plant locations from where the product identity information could be acquired
The plant locations could be categorized as must, should and nice to haves. Further, each location may have different frequency (read rate), read range, aggregation requirements. It is also possible to define whether instance specific information is necessary or simply product-specific information could serve the control purpose.

4. Define the overall information channel
Information channel refers to the close-loop flow of product identity information as well as subsequent sequence of control command. These steps, being a part of analysis activity, should identify an abstract architecture of how information will flow in the network.

A useful guideline to minimize the complexity of analysis would be to examine individual sub-components of this channel separately and independently. For instance, developing the upward channel from tags to Savant™ should generally be kept separate from other information flow. The consideration for optimising the information transfer or the use of sophisticated techniques such as agile or intelligent readers and/or downward link from Savant™/Control system for early filtering of tags, etc. should also be avoided at this stage. Such should be done only once the conceptual channel is established in a full-blown form.

5. Define Reader Network Design
Identify the locations for placement of readers. Such identification should follow the decisions made in the previous step, the nature of decisions being made and the operational layout. Another criterion could be to place a reader where a product-identity based switching of product flow is used, e.g. before a gate or switch that routes products to different outgoing paths, or vice versa the incoming paths. In the latter case, one may even need a separate reader for each incoming routes. (This is a topic of ongoing research.)

6. Define Basic and Extended Savant™ Operations
Define how the Savant™ functionality could be best distributed. As discussed previously, the criteria considered here may include synchronization with legacy systems, fault tolerance, hardware/software availability, cell layout, reader network design, the amount of data processed, etc.

7. Define the Information Channel for Networked PML Integration in Control Decisions
The channel should define the usage of PML tools as well as ONS services. The link may be both outward (e.g. storing new product information) as well as inward (as usual, e.g. for receiving process recipe information).

8. Analyse/Optimise Information Channel
From the conceptual design of the information channel, develop different use-cases defining the ways in which basic Auto-ID components could be placed within the channel. For each use-case, analyse specific information metrics, e.g. fault tolerance, information delay etc.

9. Analyse different routes in which Information Channel could be optimised
Use the conceptual analysis and design as a basis to delegate or transfer data processing or controlling responsibilities to other parts in the channel. These might be – for example:

- Seeking to distribute of Extended Savant™ module to provide information decentralization
- Transferring filtering functionality directly to the location where excess, unwanted information is likely to be generated
- Consider the option of Tags with memory where tags can carry specific information about the product tracking/tracing within production operations.
- Deciding between Push vs. Pull mode of information channelling – that is, readers or Savants™ could take decisions about where the information is needed and would send such directly to specific location.

6. CONCLUSION

In this paper we discussed certain issues concerned with the integration of Auto-ID infrastructure within manufacturing control middleware environment. Since Auto-ID information is expected to generate a vast amount of real-time data, it will be crucial that efficient means are provided in the integrated architecture for dealing with such information in an economic fashion. We analysed the requirements for efficient integration from a software/hardware development perspective as well as a data management perspective. We also showed a set of potential routes in which this integration can be performed. The horizontal and vertical distribution of Savant™ functionality has been shown to be the key to these routes.

Future work will address issues related to development of Extended Savant™ modules to be directly integrated within PLCs or MES operations as in Figure 14. A collaborative project with two different commercial vendors is under a proposal stage and is expected to lead towards an initial feasibility study for the first major undertaking of Auto-ID infrastructure within industrial control operations.

7. REFERENCES

1. **Anonymous, "MES explained: A high level vision".**
   White Paper 6, MESA International, http://www.mesa.org, 1997.

2. **Y. Chang, D. C. McFarlane, R. Koh, C. Floerkmeier & L. Putta, "Methodologies for integrating Auto-ID data with existing manufacturing business information systems".**
   White Paper CAM-AUTOID-WH009, Auto-ID Center, http://www.autoidcenter.org, Nov. 2002.

3. **A. Garcia, D.C. McFarlane, M. Fletcher & A. Thorne, "Auto-ID in materials handling".**
   White Paper CAM-AUTOID-WH013, Auto-ID Center, http://www.autoidcenter.org, Nov. 2002.

4. **J.E. Kendall & K.E. Kendall, "Information delivery systems: An exploration of web pull and push technologies".**
   Communications of AIS, 1:Article 14, Apr 1999.

5. **D.C. McFarlane, "Auto-ID based control: An overview".**
   White Paper CAMAUTOID-WH004, Auto-ID Center, http://www.autoidcenter.org, Feb. 2002.

6. **D.C. McFarlane, J. Carr, M. Harrison & A. McDonald, "Auto-ID's three R's: Rules and recipes for product requirements".**
   White Paper CAM-AUTOID-WH008, Auto-ID Center, http://www.autoidcenter.org, Nov. 2002.

7. **H.I. Moran, D.C. McFarlane & T.P. Milne, "The development of use cases as an alternative approach to identify the impact of auto-id implements on business information systems".**
   White Paper CAM-AUTOID-WH016, Auto-ID Center, Cambridge University, Cambridge, UK, 2003.

8. **M.T. Ozsu & P. Valduries, "Principles of Distributed Database Systems".**
   Prentice Hall, New Jersey, USA, 2nd edition, 1999.

9. **K. Ramamritham, "Real-time databases".**
   International Journal of Distributed and Parallel Databases, 1(2):199–226, 1993.

10. **A. Stankovic, S.H. Son & J. Hansson, "Misconceptions about real-time databases".**
    Cybersquare, pages 29–36, June 1999.

11. **A.J. Thorne, D.C. McFarlane, S.E. Hodges, S. Smith, M. Harrison, J. Brusey & Garcia, "The auto-id automation laboratory: Building tomorrow's systems today".**
    White Paper CAM-AUTOID-WH018, Auto-ID Center, Cambridge University, Cambridge, UK, June 2003.

12. **E. van Daalen, R.J.G.M. Lenders & L.F.A. van Zunders, "Manufacturing execution systems (MES) 2003: An overview of the vendors, developments and trends in the dutch manufacturing execution system (MES) software market".**
    Technical consultancy report, Cap Gemini Ernst & Young, www.cgey.com, 2003.

13. **A.A. Zaharudin, C.Y. Wong, V. Agarwal, D.C. McFarlane, R. Koh & Y.Y Kang, "The intelligent product driven supply chain".**
    White Paper CAM-AUTOID-WH005, Auto-ID Center, http://www.autoidcenter.org, Feb. 2002.

14. **J. A. Tompkins, J.A. White, Y.A.Bozer, E.H. Frazelle, J.M.A. Tanchoco & Jaime Trevino, "Facilities Planning".**
John Wiley & Sons, New York, USA, 1996.

15. **Anonymous, "Toyota extends Body Identification Systems as Derbyshire factory gears up for new model".**
http://www.rockwellautomation.co.uk/case_studies/casestudy_auto_toyota.htm.

16. **D.C. McFarlane, "The Impact of Product Identity on Industrial Control Part 1: "See More, Do More…".**
White Paper CAM-AUTOID-WH012, Auto-ID Center, http://www.autoidcenter.org, Feb, 2003.

17. **M. Reynolds, J. Richards, S. Pathare, H. Tsai et.al, "Multi-band, Low-Cost EPC Tag Reader".**
White paper MIT-AUTOID-WH012, Auto-ID Center, http://www.autoidcenter.org, Sep, 2002.

18. **Oat Systems & MIT Auto ID Center, "The Savant – Vo.1 alpha".**
White paper MIT-AUTOID-TM003, Auto-ID Center, http://www.autoidcenter.org, Feb, 2002.

A. TAXONOMIC CLASSIFICATION OF DIFFERENT PROCESSING MODULES WITHIN SAVANT™

The table 3 below provides a taxonomic classification of different data processing modules (including filters and queries) that could normally be used in the manufacturing control decision-making. A technology vendor could develop such a classification to provide purpose-built filters or queries within basic or extended Savant™ module. Using such a facility, systems integrators or end-users could then straightforward develop their applications by simple mix-and-match of appropriate components. Note that many of these filters or queries may need sophisticated filtering or database querying algorithms particularly when the information being accessed is available in a disparate, incomplete and/or distributed form. It is quite likely that an end -user or systems integrator will not have skill nor time for how to implement them efficiently on their own.

Table 1: Taxonomic Classification of Processing Modules within Basic or Extended Savant

ATTRIBUTE	ASSOCIATED EPC™ FILTER OR QUERY OPERATION
Product-type	– All products of X type, colour, variety
Location	– All products at the X location or all products that passed through X location
Aggregation/Containment	– Batch, lot, unit-load – all products in X lot/sublot/unit-load
Timestamp/History	– Timestamps of products' travel on shop-floor
State History	– Products produced after X grade product or a product with Y EPC™ pattern
Customer Specific	– Products produced for X customer
Raw-material Specific	– Products produced from X raw-material from Y vendor
Genealogy Specific	– What type of materials/machining instructions are used for X product
Recipe Specific	– All products produced using X type of machine instructions
EPC™ Pattern or Interval Specific	– Products from X to Y range of EPCs™
Machine Related Tracking/Tracing	– All products processed on X machine
Operation Specific	– Assembly, disassembly, machining, packaging, palletising instructions
Manual/Operator Specific	– Product EPCs™ associated with particular operator or technician
Production Targets/ Specification Related	– Products to be finished within a specified due-date, time and tolerance
Statutory Requirements, ISO-benchmarking Specific	– All EPCs™ meeting such requirements
Performance, Resource Usage, Process Efficiency	– How many items of X type were processed on this machine – Total spread of item costs – Spread of item processing time
Nose Removal & Smoothing	– Mysterious tag reads
Events, Alarms, Triggers	– When was last item of X type/X fault detected or passed on this machine
Fault/Failure Specific	– EPCs™ passed during X machine was faulty
Configuration Data	– EPCs™ read on a particular reader, combination of readers, savant network
Data Originator or Supplier Specific	– EPCs™ supplied by particular I/O server or external party
Industrial Application Specific	– FMCG, Pharma, Clothing, Electronics, Automobile