In B. J. Davies (Ed.), *Proceedings of the Twenty-eighth International MATADOR Conference* (pp. 137-146). UMIST, Manchester, UK, 18-19 April 1990: MacMillan Education.

A “MACHINE CONTROL SHELL” FOR “NEXT GENERATION” MANUFACTURING MACHINES

Professor R H WESTON, A H BOOTH, Dr P R MOORE, R HARRISON and Dr K CASE
Department of Manufacturing Engineering,
Loughborough University of Technology

SUMMARY

There is a growing pressure on many manufacturing oganisations to produce products in small volumes. However, to date, most automation projects have centred on high volume production. The major impediment to the application of programmable automation lies in the high cost of engineering solutions. Already a range of control system components are available to produce flexible automation schemes but as yet the selection and use of those components is a highly specialised exercise which is generally not well understood. This paper describes the need for an open control architecture for programmable machines and outlines findings of a proof of concept research project aimed at formalising the design of control systems. The work has resulted in a “motion control shell' which can much reduce the cost and time involved when building machine controllers.

THE NEED FOR PROGRAMMABLE MACHINES

Until fairly recently almost all industrial automation systems could be classified as 'hard' or 'fixed' automation. Conventionally such systems have been mechanical in nature with motion controlled through simple sequences by on-off switches or through fixed position and velocity profiles by cams. Thus, normally such machines could only be used to perform simple tasks and their reprogramming (i.e. to manufacture different products) often implied significant penalty with regard to cost and/or time /l/.

With low cost computer processing power came opportunities for using programmable servo-controlled drives /2/. The need for programmable automation led to a desire to produce robots which could be as flexible as the men they would replace. Thus, during the last decade, the aim of many within the robotics community has been to produce a single reprogrammable machine which can have sufficient flexibility to automate a wide range of manufacturing tasks (including materials handling, processing and assembly) /3/. Not surprisingly such ventures have met with limited success as

(i) a general purpose robot arm will not normally provide a mechanically optimised solution, i.e. offer kinematic and dynamic properties which closely match the requirements of individual applications /4,5/,

(ii) processing power will not be limitless, thereby tending to limit either the machine's ability to respond intelligently to sensory information (i.e. to be flexible) and/or perform tasks quickly and accurately,

(iii) even if a robot is highly flexible it will normally require dedicated tooling and fixturing, thereby much reducing the effective flexibility /5/,

(iv) the cost of engineering a robot solution will usually outweigh the cost of the robot itself, so that a typical robot system will cost as much to design and install as a counterpart hard automated system /3/.

Despite this as a backcloth, the concepts developed within robotics initiatives will grow in importance. As product lifecycles continue to fall rapidly and the cost of skilled manpower increases the use of programmable automation will need to play an increasingly important strategic role in highly competitive manufacturing companies.
How then can emerging robot methods and tools be utilised on a much wider automation front?

The authors suggest that this will best be realised by creating a new breed of modular machines which can be configured quickly and at low cost.

It is widely appreciated that significant advantage can be gained by designing computer software and hardware in a modular fashion. Similarly there are various examples of modular machines where the mechanical elements used demonstrate some degree of modularity, i.e. the mechanical structures involved are built from modules which facilitate motion in one or more degrees of freedom and can be aggregated in a manner dictated by the application involved. To date, however, there has been no generally accepted methodology which addresses the problem of decomposing (or modularising) machines which can comprise both mechanical and control system elements. Such machines would, in general, be built from intelligent modules which can incorporate computer processing power and sensory capabilities to perform their tasks and interact in a consistent manner with other intelligent modules forming the machine. Using contemporary jargon, the modules could collectively be referred to as mechatronic elements.

This paper aims to explain inherent advantages of using modular programmable machines. However, the concepts introduced will be developed primarily with respect to methods of modularising control system functionality. The authors advocate the use of an open control architecture for programmable machines as a means of significantly advancing current automation practice. The concept is one of creating a family of control system modules which themselves conform to an agreed open control architecture and can be combined to enable the control of most types of manufacturing machine. The family of control system modules will collectively provide the functionality available within contemporary robot controllers as well as additional functionality which is commonly required to program, synchronise, sequence and monitor manufacturing machines.

Even though only methods of modularising control system functionality will be considered, it will be shown that significant benefit can be gained through enhancing the level of programmability and control functionality at machines. Provided that, where necessary, drive system design modifications can be enabled in a cost effective manner, the approach can be applied to most types of existing production machinery. In this way the best of two worlds becomes possible, i.e. products belonging to a family can be manufactured in a highly efficient manner (i.e. with short cycle times and satisfactory levels of accuracy and quality) yet in small quantities (where adequate levels of programmability can lead to short machine set up times and hence low levels of inventory). The alternative use of a robot arm would in only exceptional circumstances lead to similar duality of programmability and mechanical optimisation /4/.

In the following sections the use of an open approach to machine control is considered and compared with conventional approaches. In that regard, the progress made by the Modular Systems (MS) research group, at Loughborough University of Technology (LUT), in producing a proof of concept specification and prototype implementation of the open control architecture is described.

CONTEMPORARY METHODS CF CONTROLLING MANUFACTURING MACHINES

Despite the relative infancy of the technology, the impact of LSI components has been such that almost all modern machine control systems incorporate some form of digital computer. Provided that systems engineering costs can be spread over a number of units, even the most simple machine sequencer can be produced cost effectively using this technology, once having justified a computer controller the inherent processing power available can also be used to provide improved operator interaction, improved machine diagnostics and opportunities for integrating the machine and its controller into its manufacturing environment (leading to CIM). Thus it is a widely accepted view that the next generation of manufacturing machines will be computer controlled. The producing machine itself will incorporate a number of mechanical mechanisms which operate collectively to transport, process and store workpieces and tools, whereby operations such as materials handling, packaging, metal removal, forming, fabrication, printing, inspection and assembly can be accomplished. The particular attributes of and relationships between the mechanisms will be application specific. Hence, although machines of different types often demonstrate common properties, apparently the functionality required from different machine controllers varies enormously.

Today, two distinctly different approaches are commonly used when creating a computerised machine control system. One approach involves creating a custom designed control system, often built from single board computers and engineered using conventional general purpose software development tools. A vast array of proprietary control systems of this type exist, including robot and machine tool controllers (where frequent reprogramming is required) and computer controllers for various semi-dedicated and dedicated producing machines. These custom controllers often demonstrate high levels of functionality but their realisation ties them to specific mechanical hardware. Thus seldom can configuration costs be spread over a large number of machine units. The cost of producing a custom controller can be extremely high for complex machines, typically tens of man years of highly skilled design and development work being involved.

The second common approach is based on the use of various forms of 'industrial controller' (including programmable logic controllers PLCs) which offer a degree of modularity and configurability. Industrial controllers are available 'off the shelf' and can be customised to control specific machines by selecting hardware modules and creating control software using specialised development tools, e.g.

symbolic 'relay ladder' programming debugging facilities. Unfortunately, however although system engineering costs can be minimised significantly, contemporary industrial controllers can demonstrate major limitations where high-level of functionality is required, or where product or process changes dictate frequent reprogramming. Essentially, industrial controllers have been evolved as a replacement for relay and hard-wired electronic logic controllers, with new functions added over the years in a 'bottom-up' manner. Today, although they are well accepted by industry, their use often results in relatively 'hard-wired' solutions, which has relegated their usage to a subset of machine control problems which usually can be classified as simple sequential or continuous process control tasks. For example, it is not appropriate to attempt to control even a simple end-stop robot arm with a PLC.

The philosophy embodied in 'industrial controllers' is an excellent one, recognising that there are many common problems when controlling manufacturing machines and providing modules and configuration tools for creating specific machine control systems. However, with advances in computer science, since their conception (in the early 1970s), it is time to re-think the problem in a 'top-down' way and retain the best features of current approaches.

Essentially, many of the major problems associated with contemporary machine control methods, come from the fact that they are 'closed' solutions in the sense that they are (i) tied to particular vendors, (ii) utilise a specific hardware base and set of configuration tools, and (iii) are not widely accepted solutions. A step change in approach can only be realised through standardisation in regard both to design methodology and available automation products, i.e. through open solutions which apply generally across various classes of manufacturing machine.

AN OPEN APPROACH TO MACHINE CONTROL

The problems involved in specifying an open control approach suitable for any producing machine are extremely complex.

This is true in regard both to technical and 'political' issues. Some would argue that the need to gain widespread acceptance of an open specification against which all could work is in fact mission impossible. However, the authors contend that much of the standardisation effort required will follow on from current standardisation initiatives in the Information Technology (IT) arena. Proof of concept initiatives are required to address at least a subset of the overall machine control problem and hence clearly identify pilot methodologies and areas in which new standards need to be defined or adopted. To develop this argument further let us consider certain of the technical issues involved.

An appreciation of the machine control problem

For a typical manufacturing machine the high-speed concurrent manipulation and synchronisation of workpieces, tools and sensors may be involved. Thus, a number of motion controlled and sensing elements (which will be referred to as machine components) will be required to operate in a co-ordinated manner, the number and type of those components being chosen to accomplish the specific producing function. The individual components will each require their own subset of information processing and control functions. The individual components will be required to interact with other components forming one or more 'closely coupled' component groups, thereby performing one or more logically separate parts of the producing function. The term 'closely coupled' is used here to imply a 'close relationship between'; an example of such a relationship occurs when more than one machine component is involved in locating a workpiece or tool according to some specified profile of positions, velocities and/or forces. The individual components of a 'closely coupled' component group may, in fact, be operating at different locations in a specific machine. Indeed, the individual components may also be distributed at remote locations along a production line but be required to operate in a synchronised manner (a common requirement in packaging and process industries). Thus, the individual components of a 'closely coupled' component group can be considered to be logically related, but may or may not be physically (e.g. mechanically) linked to each other.

If more than one logically separate manufacturing activity is involved (e.g. separate processing, assembly and/or inspection operations might be required) those concurrent activities must also be synchronised and controlled to perform the complete production function. Here, events such as starting, indexing, tool changing and stopping times must be initiated.

In certain application areas machines will require infrequent reprogramming, whereas in others products may be manufactured in small batches leading to the need for operational changes on a regular basis. In any event job changes should be facilitated efficiently and effectively without the need for high levels of operator skill or long periods of machine downtime (during which there may even be the need to produce defective products). For a typical machine reprogramming will involve the defining of new synchronisation conditions/events, new motion sequences, new motion profiles and new I/O conditions. Generally the definition of such parameters relating to machine motions will form only part of the new functional description of the machine. For example, it will also be necessary to define the way in which feedback information is processed and interpreted, exception conditions are handled and the way in which the machine needs to communicate with its environment (e.g. with the operator and other machines). Implicit in the programming requirement of manufacturing machines (as for robots) is the need for computer assistance (i.e. high level programming tools) to minimise the cost of engineering support functions.

At this point it is appropriate to draw out the distinction between machine programming and machine configuration. There is overlap here, but machine programming is used in this paper to imply changing the task performed by an installed machine, whereas machine configuration refers to the building and installation of a machine, which may, of course, subsequently require programming/reprogramming.

A ‘generic’ reference architecture for machine control

The foregoing rather abstract description of the general machine control problem, serves to illustrate some of the difficulties of producing a general or open control architecture for machines. It also provides some clues as to how the mechanical elements of machines might be modularised but here we will not consider mechanical design issues further.

Clearly, not all producing machines will involve the same level of complexity, thus any open control architecture which purports to being ‘generic’ must support the range of complexity level found in both simple and complex producing machines. Inherent in this statement is a definitive need for modularity as a control system capable of controlling a complex group of machines would have significant redundant capability when used to control a simple machine, leading to solutions which would not be cost effective. A natural machine control hierarchy exists as illustrated by Fig. 1 and this provides some basis for decomposing the general control problem. However, on closer examination it becomes evident that there is also often a need for heterarchical interaction between control modules which would operate at each layer in the hierarchy.

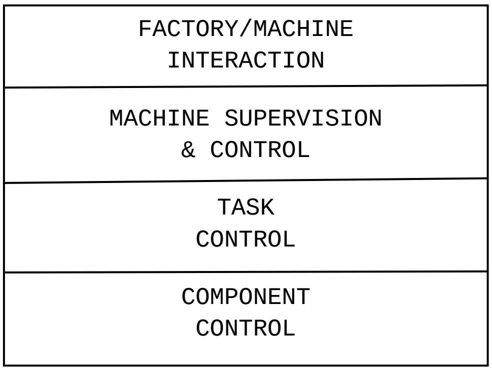

Figure 1. Natural Control Hierarchy of Programmable Machines

With regard to the lower levels of the hierarchy described by Fig. 1, there already exists a number of commercial motion control products which, with modification, could naturally exist as modular building elements of machines. For example, single board computer control systems, which achieve motion control for one or more axis of motion can be purchased from various sources. Similarly, programmable transmission elements (e.g. software gearboxes, cams and linkages are beginning to emerge).

Any open control architecture should seek to encompass available motion control products of this type as well as new machine components as they emerge. In this way, it is possible to avoid duplication of effort (i.e. significantly reduce the research and development work involved) and much improve the opportunities for gaining a widespread industrial and commercial acceptance of the methodology.

This has been the approach adopted by the MS group, where concentration of effort has largely been in studying the remainder of the generic machine control problem assuming the availability of motion controllers, intelligent tools and sensors, and industrial controllers which can be included as required.
Loughborough research has aimed to specify an open framework for the

design of machine control systems (see Fig. 2).

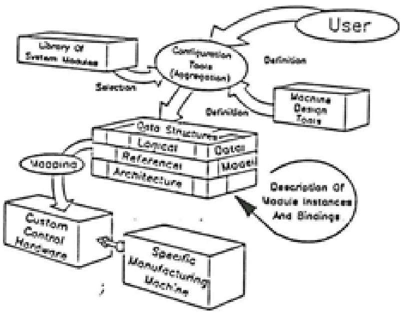

Figure 2. Reference Architecture Concept

To establish this framework concept, a family of software modules have been created which can be assembled together to produce controllers for various types of manufacturing machine. Machine configuration (i. e. the selection and binding together of modules) is enabled through the use of configuration tools, also as depicted conceptually by Fig. 2. Through significantly reducing the cost of building machine controllers the aim is to offer increased functionality at an equivalent cost (see Fig. 3) so that a successor to the established practice of employing industrial controllers can be found.

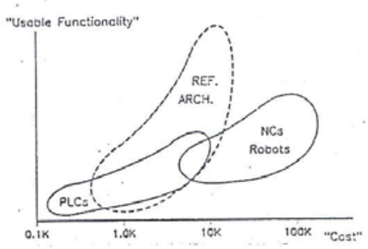

Figure 3. Role of Reference Architecture Approach

In an effort to maximise configurability, the approach has thus been to decompose the functionality required from the general machine control system into the hierarchy of functional layers described by Fig. l. Subsequently a further decomposition of the required functionality at individual lavers has resulted in the specification of the family of modular control system elements shown in Fig. 4.

Various commercially available motion controllers (comprising both computer hardware and software) have been purchased and incorporated within the control architecture and are treated as modular building elements. The approach used here has been to include ‘driver’

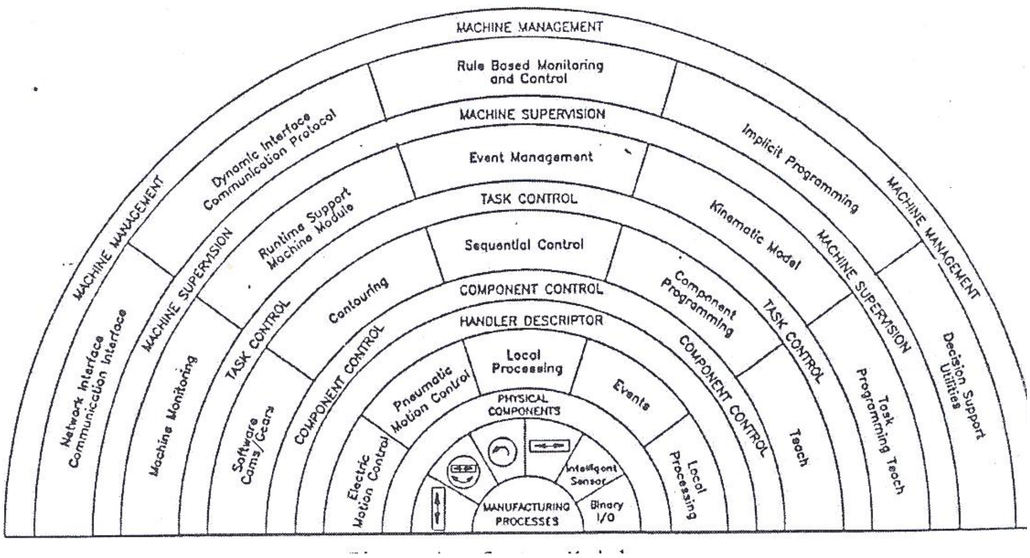

Figure 4. System Modules

modules within the module library which operate to provide a neutral interface to proprietary building elements. For example a choice of electric and pneumatic motion controllers can be used, with appropriate 'driver' modules converting them into a standardised or 'virtual' motion controller. In this way higher level modules can treat these units in a consistent manner as standard control system building blocks.

When using the approach derived by the MS group to configure a specific machine control system, the particular requirements of that system are determined to enable a selection of control system modules from the library to be made. Instances of the modules selected are then bound together by data structures, which themselves fit into a hierarchical data model of the machine and its controller. This data driven methodology offers not only considerable flexibility, but also visibility and extendability, which can enable the machine to be more fully integrated within its manufacturing environment.

The concepts embodied in the Loughborough reference architecture have been derived in an iterative manner, with various versions of the computer hardware and software produced being tested and subsequently enhanced by utilising representative machine hardware. Up to 14 servo-driven motions have been simultaneously controlled, along with various binary motions for which actuation is to mechanical stops. Many of those drive elements are modular in construction and can be arranged to form various multi-axis groups, which in various evaluation scenarios represent a number of concurrently operating, closely-coupled activities which require synchronism, sequencing and motion control. Thus the machine hardware is itself highly configurable and allows a wide range of work-handling and assembly tasks to be emulated. Custom tooling and various sensing elements have been incorporated within this test facility to illustrate the use of the 'configurable machine control concept' when handling, transporting, or partially assembling products.

OPEN VERSUS CLOSED

Generalities

The axiom on which the open architecture concept is based is that specific manufacturing machines can be considered to comprise a subset of the general manufacturing machine. Hence the functionality of a specific machine can be decomposed into modular elements which form part of a larger family of modules for the general machine. In this paper we have concentrated on control system modules, but the approach can equally apply to mechanical building elements.

Assuming that the modules can be bound together (the purpose of the data model), several major advantages accrue which essentially are the result of a 'standardisation' process, i.e. a consistent approach to a wide range of problems. Tried, tested and well documented library modules can be used many times and, over a period of time, additional modules can be included within the library. As the automation marketplace is vast, the widespread and repeated use of component elements can facilitate low unit cost and high support and reliability levels - common attributes of well-established software products.

The inherent property of extendability gives rise, in this case, to two clearly identifiable advantages. First, very high functionality levels can be achieved (with new application areas served by generating appropriate library modules) and second, support and diagnostic modules can be incorporated which are designed to serve the specific requirements of the system builders, manufacturing engineers, shop floor workers and maintenance personnel. Essentially, extension is only limited by cost and implementation considerations, but, provided that a common base of requirement can be identified, the advantage of spreading development costs can be retained.

It is also important to stress that the hierarchical nature of the Loughborough reference architecture provides important advantageous properties when compared with a 'flat' architecture. There is a natural hierarchy in machine control problems and a growing need to extend that hierarchy both upwards, to encompass factory control and information systems, and downwards, to include low-level machine components (such as mechanical modules as well as actuate or sense 'by wire' components). This hierarchical segmentation of the problems allows specific expertise and methods at each level in the hierarchy to be brought to bear on the problem.

The use of a hierarchical data model also allows access to data structures at each level, i.e. it provides 'visibility'. This visibility results in an ability to configure, programme and monitor the control system with varying degrees of cost and sophistication. For example, where appropriate, task programming can be achieved via the use of on-line explicit language and/or teach methods, whereas other situations may best be served by task oriented methods based on graphical and implicit language programming tools.

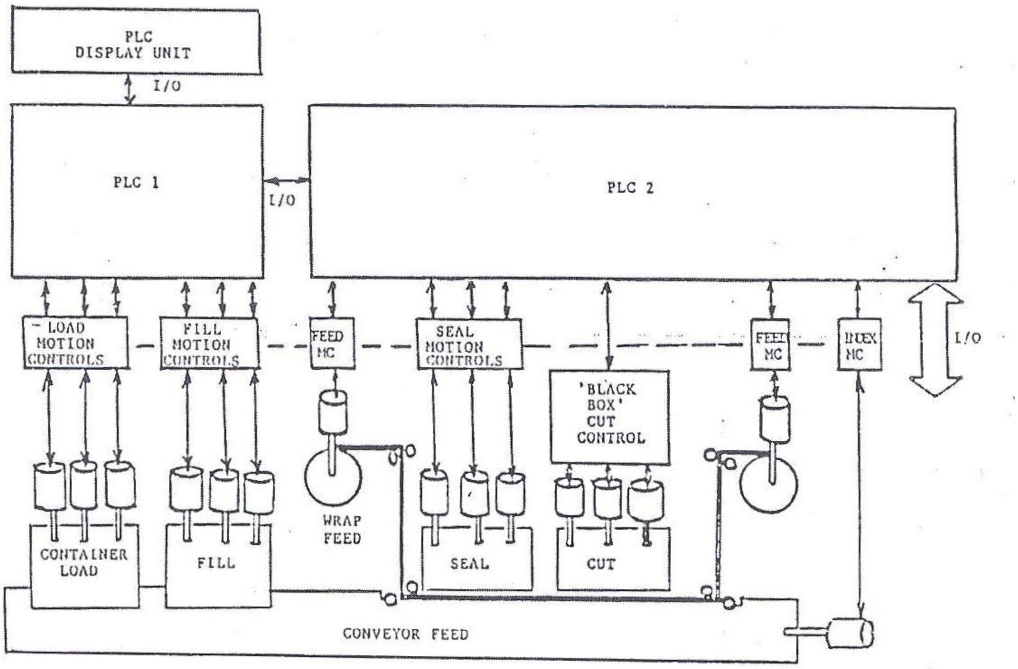

Figure 5A. Conventional Control Approach

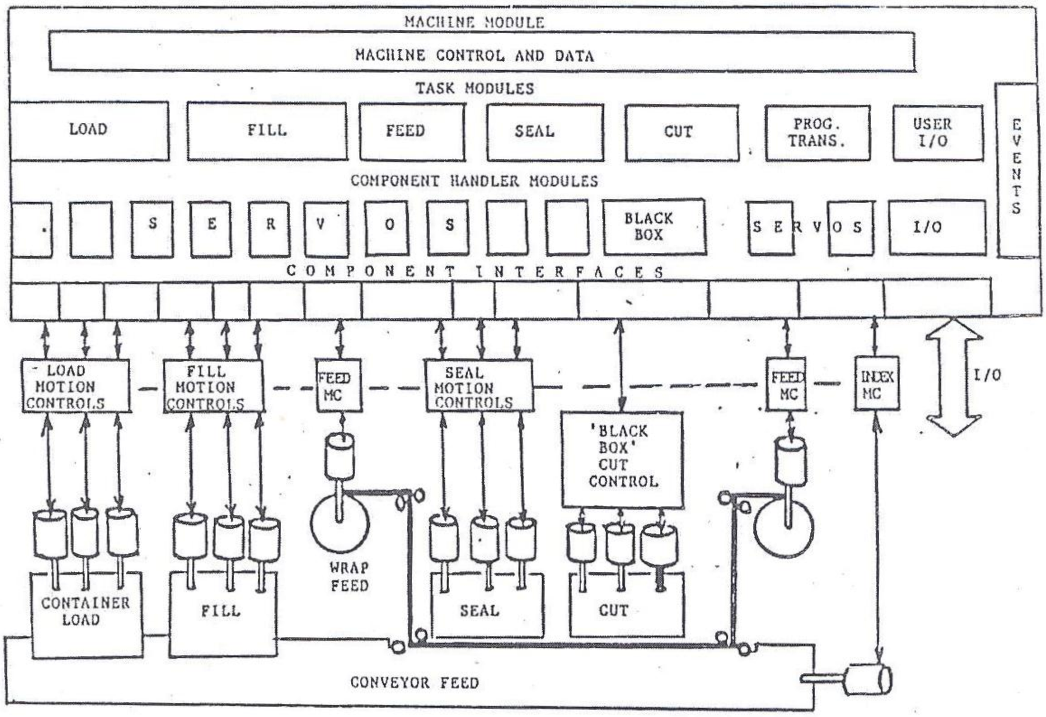

Figure 5B. Open Control Approach

A specific case studv

It is difficult to describe the potential impact that an open approach to control might have as the generic nature of this design methodology promotes the use of rather vague statements. With that in mind, let us consider a rather specific machine control problem and then contrast the open and closed approaches. Figure 5 depicts a fictitious packaging machine where various types of container are to be filled and the speed of the machine is to be programmable. Fig. 5A shows a typical conventional method by which control is achieved, whereas Fig. 5B illustrates the use of an open architectural approach.

When the hardware costs of the two machine control systems are compared it is unlikely that there will be very significant differences. However, major differences can be expected between the open and closed methods in regard to development costs and time when a new machine control system is specified, implemented and tested. Potentially very significant savings should result from the open approach, primarily as the need for be-spoke (or customised) software writing will be much reduced and the availability of special purpose tools will simplify the software creation. For example, configuration tools provided by the MS groups proof of concept implementation, can much simplify the machine description phase by providing prompts and templates to the machine builder. The tools also allow segmentation or decomposition of the control problem into manageable sub-problems which can be de-bugged separately and run concurrently. Provided that appropriate control system modules already exist, development costs and time can be reduced by a factor of ten or more. In situations where certain control system modules do not already exist, such as in Fig. 5 where the proprietary 'black box' controller of the cut mechanisms has to be fitted into the open architecture, then new software (a driver modules in this case) would have to be written, proven and installed. Nonetheless, the amount of be-spoke software writing would still be much reduced and the driver module could form part of the library of control system modules for future use. As previously indicated, opportunities for much improved data visibility exist and, if possible short cuts should not be encouraged when extending the library of modules (e.g. as in the case of the cut mechanism driver module alluded to above) so that enhancements to control system functionality can be progressively included, e.g. better management and quality reporting might be included at a later date. This perhaps leads on naturally to the observation that the open approach is the only sensible way forward where incremental enhancements need to be made in control functionality leading ultimately to fully automated factories. When using conventional closed automation technology an exponential increase in development costs can result as complexity levels are increased. The natural extendability of the open modular approach means that it does not suffer from such inherent properties.

CONCLUSIONS

The use of modular design strategies will become increasingly important in the creation of new generations of manufacturing machines. This paper has illustrated potential advantages of modularising and standardising control system functionality. A proof of concept open architecture is briefly described as is an example application area.

Currently the Loughborough University open control approach is being released as a commercial product in the form of a Motion Control Shell (see Fig. 6). This shell is written in 'C' and runs under the OS-9 operating system on Motorola family computer hardware. A typical machine control system would thus comprise one or more motherboard single board computer and various motion controllers as required, where interprocessor communication is enabled over VME, G64, RS232 or RS422 data links.

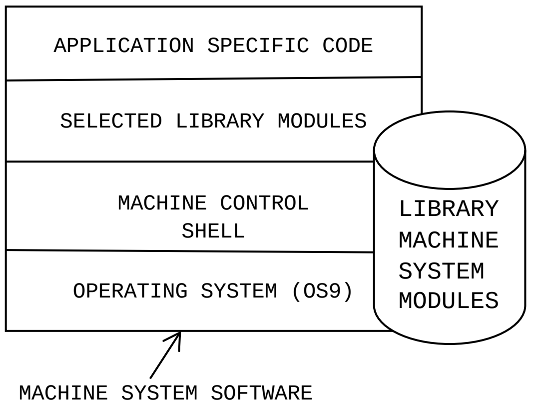

Figure 6. Much Reduced Bespoke Software Using the Machine Control Shell

It is hoped that the motion control shell will become widely used by European OEMs, where their diverse activities may lead to an increased range of control system modules and configuration tools, which in turn should help to establish the approach.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the ACME Directorate of the UK Science and Engineering Research Council, whose funding and technical inputs have been much valued.

REFERENCES

1. G. BOOTHROYD, C. POLI and L.E. MUICH (1982), Automatic Assembly, Marcel Dekker.

2. A.H. REDFORD and E. LO (1986), Robots in Assembly, Open University Press.

3. T. OWEN (1987), Robots Out of Wonderland: How to Use Robots in the Age of CIM, Cranfield Press.

4. R.H. WESTON, R HARRISON, A.H. BOOTH and P.R. Moore (1989), Universal Machine Control Primitives for Modular Distributed Manipulator Systems. International Journal of Production Research, Vol.27, No.3, pp.395-410.

5. H. ASADA and K YOUCEF-TOUMI (1987), Direct-Drive Robots - Theory and Practice, Cambridge, Mass., MIT Press.