Int J Flex Manuf Syst (2006) 18:175–190
DOI 10.1007/s10696-006-9008-y

Reconfigurable modular automation systems for automotive power-train manufacture

R. Harrison · A. W. Colombo · A. A. West · S. M. Lee

Published online: 1 March 2007

Abstract This paper describes research towards the realization of reconfigurable modular automated machines and the associated engineering methods and tools necessary to support their lifecycle needs. UK-based research, in collaboration with the Ford Motor Company and several machine builders, has resulted in the development of full-scale prototype reconfigurable modular automation systems for both engine assembly and machining applications. The implementation of an assembly system is featured in this paper. An engineering environment and associated reconfigurable component-based control system architecture have been created aimed at supporting the lifecycle needs of a new generation of agile automated systems, i.e., providing reconfigurable, easily scalable automated machinery. This approach has the potential to fit within a wider collaborative automation strategy where manufacturing systems are implemented as a conglomerate of distributed, autonomous, and reusable units.

Keywords Automation · Modular · Reconfigurable

R. Harrison (✉) · A. A. West · S. M. Lee
Mechanical and Manufacturing Engineering, Loughborough University, Leicestershire, UK
e-mail: r.harrison@lboro.ac.uk

A. A. West
e-mail: a.a.west@lboro.ac.uk

S. M. Lee
e-mail: s.m.lee@lboro.ac.uk

A. W. Colombo
Schneider Electric GmbH BU Automation/CoPS HUB, Steinheimer Str. 117,
63500 Seligenstadt, Germany
e-mail: armando.colombo@de.schneider-electric.com

A. W. Colombo
e-mail: awcolombo@ieee.org

1 Introduction

Reconfigurability is defined by the NSF Engineering Research Center for Reconfigurable Manufacturing Systems as the ability to adjust the production capacity and functionality of a manufacturing system to new circumstances through rearrangement or change of the system's components. There is a growing need for production machinery to be reconfigured and re-used more efficiently in order to maximise return on investment. The use of fixed configuration, mass production machinery is increasingly seen as being a relatively high-risk option because of the constant threat of obsolescence. Modular production systems although perhaps initially more expensive are more amenable to change and reconfiguration. An important consideration is that such systems must be designed at the onset to be reconfigurable and must be created from basic hardware and software modules that can be arranged quickly and reliably (Koren and Ulsoy 2002). Reconfigurability is sub-divided by Tuokko into dynamic-reconfigurability, inferring real-time volume/variant flexibility, and static-reconfigurability, meaning structural rearrangement and reuse (Tuokko 2004). Both aspects are obviously important in the context of agile manufacturing.

2 Assembly automation

Figure 1 shows a typical layout of an engine assembly line. It consists of a transport system that links together various assembly stations. Raw engine blocks are loaded onto empty pallets on the transport system and then carried into different assembly and test stations distributed along the transport system. The assembly stations operate independently of each other; there is no control coupling between them. Radio frequency identification (RFID) tags are installed in each pallet for storing process information associated with the assembly part on the pallet. Diverters or indexing stations are located at the

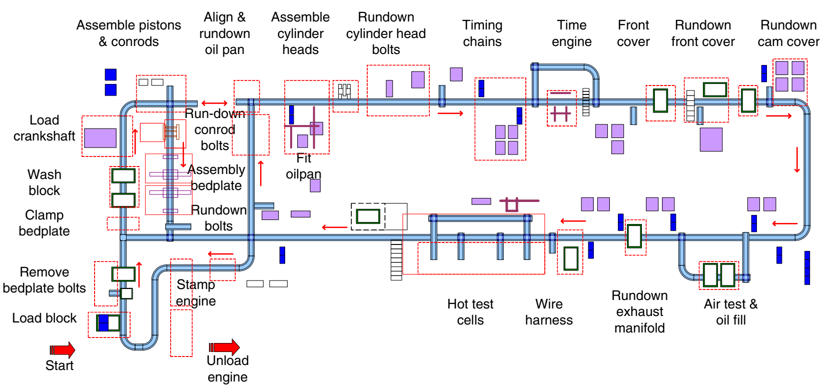

Fig. 1 Example of an engine block assembly line. (Adapted from J.A. Krause Machinenfabrik GmbH)

conveyor intersections to direct pallets to different stations or to change the orientation of the pallet for the subsequent assembly operation. Sensors and mechanical stops are used throughout the transport system to track the pallets and direct them down different conveyors according to the information stored in each respective pallet (Lee et al. 2003).

Each new assembly system delivered to the end-user, e.g., Ford, will be composed of a unique combination of these transport, assembly and test related modules. Whilst some modules will inevitably be unique to a new application, the vast majority (typically >70%) will be based on the reuse of previous mechanical modules. The traditional sequential engineering approach to the implementation of such systems involves the configuration of the major mechanical components, addition of the drive systems, electrics, control systems hardware (typically with centralized control and fieldbus-based distributed I/O) and software and finally commissioning activities. So whilst the basic machine elements are largely modular, at least from a mechanical perspective, the final machine, after the addition of drive systems and control systems has become essentially monolithic and difficult to change. This established approach caters well for the paradigm of mass production where a long product life is expected and change occurs relatively infrequently. However, to modify such systems is difficult, risky, and expensive since there is little or no provision for reconfiguration or reuse. See Fig. 2a.

3 Adopting a modular approach

3.1 Introduction

An effective solution to the lifecycle engineering of modular automated machines needs to provide:

- a set of mechatronic modules at an appropriate level of granularity for the intended application domain to enable efficient machine build and re-use of designs,

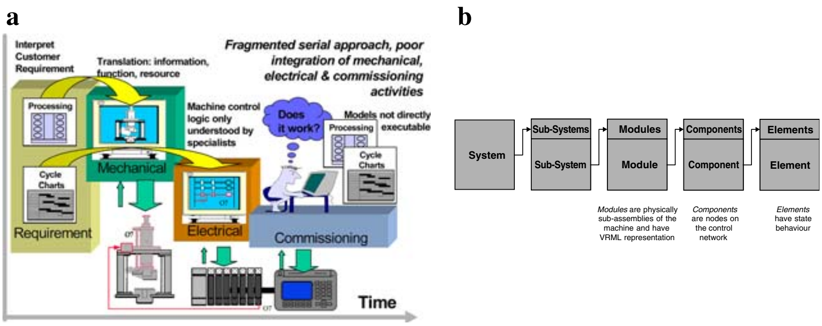

Fig. 2 (**a**) Current machine build process, (**b**) Modular CB architecture

- a system architecture which reflects the specific needs of the application domain, and
- an engineering environment and common engineering model that can effectively support the supply chain partners throughout the machine's lifecycle.

The new approach described in this paper is founded on the creation of a suitable Component-Based (CB) architecture for machine control that allows a control system to be decomposed into a set of distributed components. This CB approach allows the decomposition of control functionality to be matched to the required physical modularity of a machine.

The CB paradigm does not require PLC-based or PC-based central control because the control software is prewritten in a distributed form and embedded into each component. This approach enables the "best practice" for control, diagnostics, error checking, and lifecycle data acquisition to be embedded into each device during component manufacture. Application logic (which is specifically matched to the required state behaviour of a machine and its particular modular composition) is defined via configuration data rather than by writing application specific code in, for example, ladder logic, sequence charts, or structured text (Harrison and West 2000).

A highly simplified representation of the structure of the CB architecture is shown schematically in Fig. 2b. Any complete machine is defined as a system. Each system is composed of one or more control sub-systems that are each in turn made up of modules containing one or more control components. These components are physically seen as nodes on the control network within each mechatronic module. Modules are physically combined together as sub-assemblies of the complete machine. Each component contains one or more elements. Each machine element has its own unique state behaviour. For details of the system implementation please see Harrison et al. (2004).

3.2 Modular decomposition

It is obviously important to choose an appropriate level of granularity within a component-based system architecture that aims to support reuse and reconfiguration. Our research has shown that it is important to carefully define how much functionality each element in the system should provide. A pragmatic approach is to create a system of the coarsest granularity that still offers the ability to provide all the necessary system variants, i.e., to minimise the number of modules required within a given system whilst still being able to build any desired machine configuration. Determining the optimum level of modularity for any system requires the consideration and a trade-off of many factors. As described by Gain (2004), a design method supports effective modularity if it evidences (Gain 2004):

- Decomposability—a systematic mechanism for decomposing the problem
- Composability—able to reuse modules in a new system

- Understandability—the module can be understood as a standalone unit
- Continuity—minimizes change-induced side effects
- Protection—minimizes error-induced side effects

Correct modularity makes systems easier to build, reconfigure, and repair. It also makes systems intellectually more manageable, i.e., reduces the skill level needed to support a given system throughout its lifecycle. Changeability is an important metric for modularity, i.e., the modular decomposition of a system needs to be evaluated in terms of what changes it can accommodate. Good machine-modularity will be characterised by minimal interaction between modules (coupling) and maximal interaction within modules (cohesion); indeed there are many parallel with component-based software engineering (Crnkovic 2001). Granularity is an important issue, since having too many modules has the potential to make integration over complicated. Figure 3 presents a highly simplistic view of some modularity trade-offs. In practice a much more in-depth analysis is required focusing on a study of the functional modularity of the system, e.g., what functionality is to be reused and in what combinations.

3.3 Formalisation of modularity

Based on the experience of the authors working with flexible automation systems (Colombo et al. 2004) and taking into account experimental results in the area of intelligent modular assembly systems obtained at the Technical University Tampere and published in Lastra (2004), the following paragraphs present the core part of a mathematical formalization, applicable to reconfigurable automation systems.

Hypothesis By defining a finite number of basic production operations, it is possible to create more complex production activities. If a production system can be seen as a set of mechatronic components and each device is responsible for a basic operation as its production function goal, then combining the simpler (basic) mechatronic components will generate a complex assembly scenario. If any of these basic operations occur in the configuration of individual complex activities, the simpler mechatronics components can be reused to create different assembly systems targeting complex activities, merely by reconfiguration of the simple components.

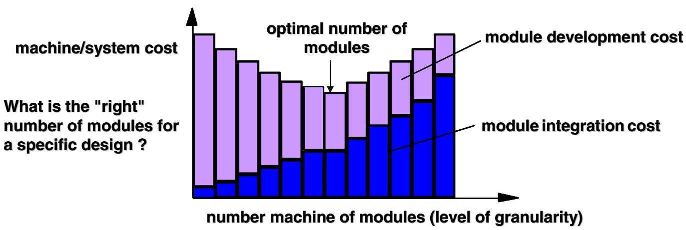

Fig. 3 Optimizing modularity: a simplistic view. (Adapted from J Gains (2004))

The formal demonstration The general background of the following definitions is based on the functional analysis theory (dual space and bag-definition).

Definition 1 A reconfigurable automation system is a 5-tuple

$$\mathrm{RCAS} = <\mathrm{PO}, \mathrm{MD}, C, I^{+}, I^{-}> \quad (1.1)$$

satisfying the following requirements:

- PO = { po_1, po_2,..., po_i,..., po_m} is a finite set "production operations".
- MD = { md_1, md_2,..., md_i,...., md_n} is a finite set of "mechatronics devices/ components".
- C is the colour function defined from PO ∪ MD into Ω, where Ω is a set of finite and not empty sets. C attaches to each production operation a set of possible *operation-colours* C(po) and to each mechatronics device a set of possible *device-colours* C(md). An item of C(po) is called a colour of "po" and C(po) is called the colour set of "po".
- I^{+} (I^{-}) are respectively the input function and the output function defined on PO × MD, such that I^{+}(po,md):C(md) × C(po) → $N \backslash \{0\}$ (i.e., a function from C(mt) to Bag(C(po) = N^{C}(po), ∀ (po,md) ∈, PO × MD). Elements of I^{+} (I^{-}) are denoted, I^{+}(po,(md,c_{md})), where c_{md} belongs to C(md).

Note: The incidence function I of a RCAS is defined by $I = I^{+} - I^{-}$, where

$$I(\mathrm{po}, (\mathrm{md}, c_{\mathrm{md}})) = I^{+}(\mathrm{po}, (\mathrm{md}, c_{\mathrm{md}})) - I^{-}(\mathrm{po}, (\mathrm{md}, c_{\mathrm{md}})) \quad (1.2)$$

The elements of the incidence function can be interpreted as functions

$$I : C(\mathrm{po}) \times C(\mathrm{md}) \rightarrow Z(\text{set of integers})$$

Definition 2 A set Ω_j is called "basic (standard) colour domain" and its elements "colour tones". Ω_j can be extended to the ring $(\Omega_j, \oplus, \otimes)$, where the arithmetic functions $\oplus$ and $\otimes$ are executed module s (s is the cardinality of Ω_j) (Couvreur et al. 1990). For example, Ω_j is the set of transport actors. Then, Ω_j is defined as set $\{\omega_1, \omega_2, ..., \omega_s\}$ with $s \in N$ (number of transport actors), whereby the elements $\omega_j \in \Omega_j$ refers to an actor of the set of a type of transport actors.

Note: For a basic colour domain $\Omega = \{\omega_1, \omega_2, ..., \omega_n\} = \{1, 2, ..., i, ..., n\}$, the operation $\oplus$ is defined as follows:

$$\forall a \in N, \quad \forall i \in \Omega \Rightarrow$$
$$\omega_i \oplus a = \omega_{i+a} \text{ if } \omega_i \leq n_i - a \text{ else } a' = \omega_{i+a} - n_i \text{ if } \omega_i > n_{i-a}$$
$$\omega_i - a = \omega_{i-a} \text{ if } \omega_i > a \text{ else} a' = n_i - (a - \omega_i) \text{ if } \omega_i \leq a$$

Definition 3 A complex colour domain is defined as the Cartesian product of two or more basic colour domains.

Definition 4 The universal colour domain Ω^* is the Cartesian product of all basic colour domains Ω_j, $j \in [1{:}n]$. That is $\Omega^* = \prod_{j\in[1:n]} \Omega_j = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_n$ then

$$\forall \omega^* \in \Omega^* \Rightarrow \omega^* = \omega_1, \omega_2, \ldots, \omega_n$$

Definition 5 The colour-functions are associated with the elements of the matrix $I^+(I^-)$ and defined $\forall\ \omega^* \in \Omega^*$. They are built from a set of basic—standard—functions or their linear combination. Projection functions: which select a component ω_i (colour) of an item ω^*. Identity functions, which select all the components of an item ω^*. Successor functions, which select some successor of a component of an item. Predecessor functions, which select some predecessor of a component of an item.

Note_1: The elements of the sets PO and MD can represent simple machine components (actuators, sensors, etc.) or alternatively higher level autonomous interacting sections of a production system, e.g., manufacturing/assembly cells.

Note_2: The presented theory is currently being extended to support the formal modelling and validation of the different PO and MD sets addressed above from structural and behavioural viewpoints.

3.3.1 Meaning of the formal elements when they are mapped into a reconfigurable shop floor (Components and functions)

- Based on the product description and process plan, the first step is the definition of the elements of the set PO.
- From the mechatronics perspective, some elements of the set MD can be predefined. At this stage the desired level of system granularity needs to be considered.
- The above theory can easily be applied at the sensor/actuator level as well as at the machine, cell or shop floor level. Independent of the granularity, it is important that each element md should be physically and logically autonomous.
- With simple production operations, then, working with regular sets the mathematical background is reduced to linear algebraic analysis. This means, one mechatronic device is responsible for one functional operation. This is a typical case in simple production systems or stations with a small number of operations.
- The Incidence Function is really a matrix with ‘functions’ as elements and it presents a complex *RELATION* $\Re$ among operations and devices. Each time users build an I-function, they are building a configuration of devices responsible for a given set of manufacturing functions.
- If it is possible to diagonalise this matrix, then it is possible to find the basis of the space ‘Devices’ and the basis of the space ‘Operations’
- Finding the ‘basis’ is the most important task because:

– If the basis is known, then the minimal configuration of devices that is able to offer the set of necessary production functions is known.
– Each complex operation could be performed by some combination of elements of the basis (linear combination law). The result is known as a SPAN of the manufacturing function basis.
– Each complex device is the result of a combination of elements of the basis (linear combination law). The result is formally known as a SPAN of the manufacturing device basis.

As one major result of the above properties, the existence of a homomorphism relation between the 'Hardware Reconfigurability' and the 'Flexible Combination of Manufacturing Functions' can be proved. The mapping Function $\rightarrow$ Device allows the identification of the right combination (configuration) of the devices able to provide the desired operations/functions. By formulating problems in this way it is possible to learn how to formally specify a set of manufacturing functions and mechatronic devices and their interrelationships building complex production automation structures. By means of this functional analysis, a method is emerging to synthesise reconfigurable manufacturing automation structures, consisting of smart devices able to offer the necessary basic manufacturing functions. This formalisation can be seen as a step forward in generalising an engineering method/tool to synthesise reconfigurable modular automation systems, the future application of which can enhance the outcomes addressed in the Section 5.

4 Modular assembly machine implementation

The modular CB approach was implemented on a full-size demonstrator assembly machine in Krause Machinenfabrik GmbH (Krause) in Bremen, Germany. Krause develops, designs and manufactures assembly systems for automotive applications and is a major system builder for Ford Motor Company as well as other automotive manufacturers. Figure 4 shows a picture of

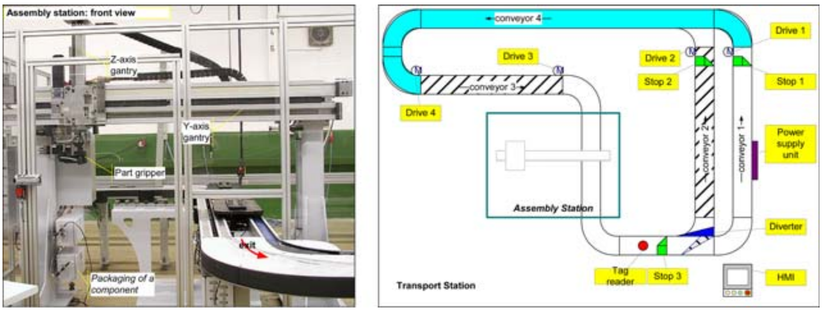

Fig. 4 Implementation of modular assembly automation system

the assembly system and the layout of the transport system. The components that were implemented for the system are labelled in the diagrams.

It is observed during system decomposition that the assembly machine is largely composed of a relatively small number of common control elements that provides standard control functionalities to the system. In fact, Krause's commissioning engineers have expressed that about 80% of the controls and equipment for assembly are standard, and tremendous effort can be saved if the design, development and implementation efforts for such control elements can be encapsulated and reused. Table 1 shows the decomposition of the complete Krause system into modules. The system is decomposed into two subsystems, the transport subsystem and the assembly subsystem. Each

Table 1 Modular decomposition of transport and assembly systems

Sub system	Module/component	Control element
Transport subsystem	Power supply unit	Power supply unit
	Monitor console	Operator console
		Subsystem monitor
	Drive 1	Drive
	Drive 2	Drive
	Drive 3	Drive
	Drive 4	Drive
	Stop 1	Stop actuator
		Pallet sensor
	Stop 2	Stop actuator
		Pallet sensor
	Stop 3	Stop actuator
		Pallet sensor
	Diverter	Diverter
	RF Tag	RF tag ID
		RF tag writer
Assembly subsystem	Power supply unit	Power supply unit
	Monitor console	Operator console
		Subsystem monitor
	Pre-stop	Stop actuator
		Pallet sensor
	Stop	Stop actuator
		Pallet sensor
	Fixing unit	Fixing unit
	Section monitoring	Enter station sensor
		At station sensor
		Leaving station sensor
		Left station sensor
	RF Tag	RF tag ID
		RF tag writer
	Y-axis	Operation
		Position
		Assembly operation control
		Movement status
	Z-axis	Operation control
		Position
	Gripper	Gripper
	Ultrasonic sensor	Ultrasonic sensor

subsystem comprises automation components of which the control behaviours are represented by their respective control elements. Altogether, eleven different types of components were developed—namely, Conveyor Drives, Pallet Sensor, Stop, Diverter, RF (radio frequency) identification unit, Indexing Unit, Gripper, Z-axis vertical gantry, Y-axis horizontal gantry, Power supply unit, and HMI control panel.

5 Engineering environment for modular machines

An integrated engineering environment was developed at Loughborough to support the implementation and lifecycle support/evaluation of the Krause assembly machine. This environment consists of an extendable set of engineering tools that can be used by a globally distributed set of engineering partners at all phases of a machine's lifecycle; see Fig. 5a. These tools include a Process Definition Editor (PDE), which supports the configuration of the machine from a library of mechatronic modules, each module having embedded control and monitoring capabilities. The PDE also supports the

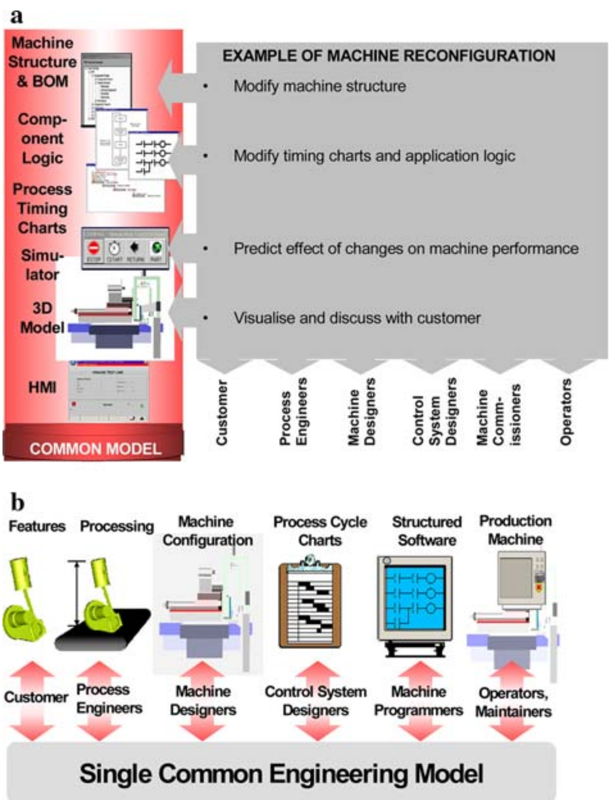

Fig. 5 (**a**) Integrated engineering environment, (**b**) Common engineering model

graphical description of the machine's behaviour. A simulator (or logic engine) enables the machine's control logic to be executed and then viewed in conjunction with a set of visualisation tools, which support remote monitoring of (1) 3D VRML-based representations of the machine's movement, (2) operator interface screens and (3) logic execution via state and cycle timing charts. The same tools can be used through the lifecycle, e.g., for both initial operator training and later remote diagnostic purposes (Thomas et al. 2002).

As illustrated in Fig. 5b, a single common engineering model is used throughout the machine's lifecycle. This model stores information related to customer requirements, process engineering, machine build and so on through the lifecycle of the machine. The model supports all significant data associated with the modular build process. All the data is structured in accordance with the CB architecture. The essential idea is that the information is defined once but used many times by both end-users and suppliers throughout the machine's lifecycle.

6 Discussion

The modular assembly system was subjected to Krause's standard commissioning tests (Anon 2002) conducted by the company's own commissioning engineers. The commissioning checks provided practical verification of the ability of the component-based approach to meet the runtime requirements of an automotive engine assembly system. It has been demonstrated through the commissioning checks that the component-based manufacturing automation system is able to match the operating standards and requirements of current automation systems.

Through evaluation work it was determined that designing and implementing a conventional control system on a reference assembly machine at Krause would typically require 40 days work (Anon 2002; Ong 2004). The equivalent activities undertaken using Loughborough University's modular component-based approach (developed on the COMPAG and COMPANION research projects (Harrison et al. 2004)), with a component library available to support the reuse of standard machine modules, was evaluated at Krause taking 19 days, a saving of about 50% in overall build time. In terms of software design activities, from initial tests it is predicted that by utilising standard modules for common machine elements, savings of around 80% in time could be achieved along with a 25% reduction in the commissioning time for a typical machine. These saving can be made through heavily exploiting the advantages of a virtual engineering environment coupled with the reuse of standard components (Ong 2004).

The achievement of better performance (relative to conventional automation systems utilising centralised PLC/PC-based control) was not an objective of this research. However, a certain degree of performance optimisation has been observed. This is due to the inherent nature of the distributed control system where automation components can concurrently execute their respective automation functions so long as their respective interlocking conditions are satisfied.

Currently virtually every new automation system project has its own unique control system specification governed chiefly by end-user preferences. This factor is currently a severe inhibitor of machine reconfigurability, and hence reuse, since there is little association at machine-module level between control hardware and software, electrical systems specifications/implementations and the mechanical modules of the machine. It is estimated that 70% of the engineering teams' effort is involved in re-implementing the control and related electrical systems each time a new machine is implemented on a new project (Ong 2004; Lee 2004). This effort could be significantly reduced if the component-based approach were well established, i.e., if proven pre-assembled mechatronic modules were used as the common building blocks to compose manufacturing automation systems by simply reconfiguring and interlocking them together instead of developing new control programs for each application (Ong 2004). The component-based approach described here in this research emphasises 'black box' reuse strategy, i.e., once a given module has been implemented and entered into the system library, the system builder no longer needs to know how the component is implemented in order to reuse the functionality offered by the module (Brown and Wallnau 1998; Brereton and Budgen 2000; Luders 2003).

One critical phase in contemporary component-based software engineering involves the mapping of logical software components to physical resources, i.e., processing and data storage. Non-functional issues have to be taken into consideration, e.g., how long it takes to execute the software components and how much memory is needed to ensure the system can deliver the required performance at runtime (Van Brussel et al. 1998). In the component-based approach proposed in this research, the component has predefined physical resources within the component boundary, and these are not accessible across components. Hence, the runtime performance of a given component is independent of that of the other components. This enables larger systems to be constructed from components without violating the principle of composability, i.e., properties that have been established at the component level will also hold at the system level (Kopetz 1997).

7 Towards collaborative automation

This paper has so far described a modular approach to the implementation of automated machines and to the support of their lifecycle needs. The approach presented is however applicable to, and can be enhanced through further research to support the modularization and reconfigurability of production structures in a broader production automation context, e.g., the Collaborative Manufacturing Automation strategy used by Schneider Electric (Colombo et al. 2004) or the Holonic Manufacturing System proposed by the IMS-HMS consortium (see http://hms.ifw.uni-hannover.de/). In order to help the reader better understand the above addressed concepts, the next section summarizes the first of the approaches, i.e., Collaborative Automation.

The collaborative automation paradigm is a result of the integration of three main emerging technologies/paradigms: holonic control systems utilizing agent-based technology, object/component-oriented approaches to software, and mechatronics. The aim is to utilize these technologies and methods effectively to achieve flexible, network-enabled collaboration between decentralized and distributed intelligent production competencies. Autonomous automation units with embedded local supervisory functionality, installed in each production site, are able to collaborate to achieve production objectives at the shop floor level, and to interact/co-operate in order to meet global (network-wide) supervisory needs (e.g., related to control, monitoring, diagnosis, HMI, and maintenance) (Colombo et al. 2004). A brief overview of the latest results appearing in the literature reveals some solutions in the area of collaborative automation systems (Van Brussel et al. 1998; Leitao et al. 2005) and also (Colombo et al. 2004) and the references therein.

The Collaborative Automation approach considers the set of production units/agents/actors as a conglomerate of distributed, autonomous, intelligent, fault-tolerant, and reusable units, which operate as a set of cooperating entities. Each entity is typically constituted from hardware, control software and embedded intelligence, as depicted in the right-hand side of Fig. 6. Due to this internal structure, these production entities (collaborative automation units/physical-agents/actors) are capable of dynamically interacting with each other to achieve both local and global production objectives, from the physical/machine control level on the shop floor to the higher levels of the factory management systems. The terms physical-agent, actor and collaborative automation unit, and their associated concepts, are now quasi-synonymous, although they have somewhat different origins (Harrison and Colombo 2005). The promise of self-optimizing and self-configuring systems in this context, capable of providing rapid and inexpensive customization, has led to increasing interest in agent-based manufacturing systems. This type of collective functionality distributed across many system devices and machine control components has the potential to replace the logical programming of

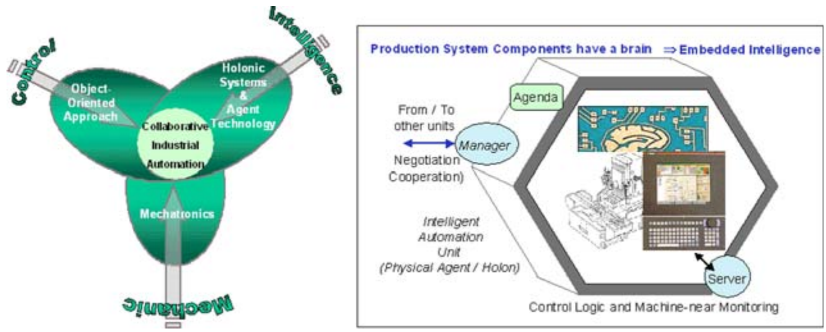

Fig. 6 Collaborative automation paradigm

manufacturing sequences and supervisory functions in traditional production systems. However, a consistent approach to enable the reconfigurability of such systems throughout their lifecycle is needed (Fletcher and Brusey 2003).

8 Conclusions

In this paper, the design and implementation of a modular assembly automation system has been discussed. It has been demonstrated, through the successful implementation and commissioning of the Krause demonstrator system, that a modular component-based approach provides substantially better system reconfigurability and reusability. This can be observed in terms of reduced lifecycle costs, improved quality and reduced risk and effort through reuse of proven components (Ong 2004). The system developed embodies an appropriate architecture and a well integrated set of lifecycle engineering tools that enable modular component-based automation systems to be efficiently built, improved and upgraded. It enables automation systems to be designed from the onset to be reconfigurable.

The integration of reconfigurable automation systems within a wider collaborative automation framework is an important goal for future research (Gorbach and Mick 2002). Higher level automation components have a natural requirement for more agent-based functionality (i.e., more goal directed and less reactive behaviour), which needs to be realised in a reusable and reconfigurable manner.

Acknowledgements The authors gratefully acknowledged the support of EPSRC, the Ford Motor Company Limited, Johann A. Krause Maschinenfabrik GmbH and the other collaborating companies in carrying out this research. The authors would also like to thank the European Commission and the partners of the Innovative Production Machines and Systems (I*PROMS; http://www.iproms.org/) Network of Excellence, and of the Service-Oriented Cross-Layer Infrastructure for Distributed Smart Embedded Components (SOCRADES; http://www.socrades.eu) Integrated Project, for their support. Schneider Electric is a core member of the I*PROMS consortium.

References

Anon (2002) J.A.K, Krause assembly system - commissioning check list, Krause Machinenfabrik GmbH

Brown AW, Wallnau KC (1998) The current state of component-based software engineering. IEEE Software September/October

Brereton P, Budgen D (2000) Component-based systems: a classification of issues. IEEE Comput 33:54–62

Colombo AW, Schoop R, Neubert R (2004) Collaborative (Agent-Based) Factory Automation. In: Zurawski R (ed) The industrial information technology handbook. CRC Press

Couvreur JM, Martínez J (1990). Linear invariants in commutative high-level nets. Lecture Notes in Computer Sciences, vol. 483. Springer Verlag, Berlin Heidelberg, pp 146–165

Crnkovic I (2001) Component-based Software Engineering - New Challenges in Software Development, Software Focus. John Wiley & Sons, December

Fletcher M, Brusey J (2003) The story of the holonic packing cell. Third International Joint Conference on Autonomous Agents and Multi-Agent Systems, Melbourne, Australia
Gain J (2004) Design (Concepts and Principles), Systems Engineering Course Notes. Department of Computer Science, University of Cape Town
Gorbach G, Mick R (2002) Collaborative Manufacturing Management Strategies, ARC Strategies, ARC Advisory Group, White Paper, Dedham, MA USA
Harrison R, Colombo AW (2005) Collaborative automation from rigid coupling towards dynamic reconfigurable production systems. 16th IFAC World Congress, Prague, Czech Republic
Harrison R, West AA (2000) Component-based paradigm for the design and implementation of control systems in electronics manufacturing machinery. J Elect Manuf 10(1):1–17
Harrison R, Lee SM, West AA (2004) Lifecycle Engineering of Modular Automated Machines, presented at 2nd IEEE International Conference on Industrial Informatics (INDIN'04), Berlin, Germany
Kopetz H (1997) Component-based design of large distributed real-time systems. Control Eng Pract 6:53–60
Koren Y, Ulsoy AG (2002) Vision, principles and impact of reconfigurable manufacturing systems. Powertrain International, pp 14–21
Lastra JL, (2004) Reference mechatronic architecture for actor-based assembly systems, PhD-Thesis, Tampere University of Technology, Finland
Lee SM (2004) A Component-based distributed control system for manufacturing automation, PhD Dissertation, in MSI Research Institute. Wolfson School of Mechanical and Manufacturing Engineering: Loughborough University
Lee SM, Harrison R, West AA (2003) A component-based control system for agile manufacturing, Proc IMechE J Eng Manufact - Part B 219:123–135
Leitao P, Colombo AW, Restivo F (2005) ADACOR, A collaborative production automation and control architecture. IEEE JIS 20:158–66
Luders F (2003) Use of component-based software architectures in industrial control systems, PhD dissertation, Department of Computer Science and Engineering. Malardalen University
Ong MH (2004) Evaluating the Impact of Adopting the Component-based System within the Automotive Domain, PhD dissertation., in MSI Research Institute. Wolfson School of Mechanical and Manufacturing Engineering: Loughborough University
Thomas DW, Harrison R, West AA, McLeod CS (2002) A process definition environment for component based manufacturing machine control systems, Proc. of the SAE World Cong. and Expo., Paper 2002-01-0468, March 2002 ISBN 0-7680-0962-6
Tuokko R (2004) High-precision and micro assembly & micro factories. MASINA Mikromekaniikka-Seminaari, Helsinki Finland
Van Brussel H, Wyns J, Valckenaers P, Bongaerts L, Peeters P (1998) Reference architecture for holonic manufacturing systems: PROSA. Comput Ind 37:255–274

Robert Harrison is a Senior Lecturer in the Department of Mechanical and Manufacturing Engineering, Loughborough University, UK. He began his career with British Aerospace working on real-time control in airframe systems. Dr. Harrison has a degree in mechanical engineering and a Ph.D. in automation. An author of more than 70 refereed publications, he has over 20 years of experience in machine design and automation systems. Dr. Harrison has been a Research Fellow at the National University of Singapore and also at the Ford Motor Company. In 2004, he was the recipient of a Royal Academy of Engineering Global Research Award. Dr. Harrison currently leads a global research initiative focussing on digital manufacturing automation in the automotive sector.

Armando Walter Colombo (Dr., 46) received the Bachelor Degree in Electronics Engineering from the National Technical University Mendoza, Argentina, in 1990; the Master Degree in Control Engineering from the National University San Juan, Argentina, in 1994; and the Doctor-Engineer Degree from the University of Erlangen-Nuremberg (Institute for Manufacturing Automation and Production Systems / Prof. Klaus Feldmann), Germany, in 1998. From 1999 to 2000 he was Adjunct Professor in the Group of Robotic Systems and CIM, Faculty of Technical Sciences, New University of Lisbon, Portugal. In 2001 he joined the Anticipation and Architectures Group, R&D Department, Schneider Electric GmbH, Germany. In 2003 he becomes a senior engineer. Dr. Colombo has extensive experience in managing multi-cultural research teams in multi-regional projects and from July 2004 he serves as Manager of Anticipation/Advanced Projects in the BU Automation CoPS HUB Department, Schneider Electric, France, Germany and the EU. His research interests are in the fields of service-oriented architectures, collaborative agent-based automation, engineering of flexible and reconfigurable production automation and control systems. Dr. Colombo has more than 120 publications (per-review) in journals, books, and chapters of books and conference proceedings. He is a senior member of the IEEE and member of the Gesellschaft für Informatik e.V.. Dr. Colombo is Associated Editor of the IEEE Trans. on Industrial Informatics, Associated Editor of the IFAC Associated Journal ATP-International, member of the IEEE IES Administrative Committee (AdCom) and chair of the IEEE IES Committee on Industrial Agents. He is listed in Who's Who in the World /Engineering 99-00/01 and in Outstanding People of the XX Century (Bibliographic Center Cambridge, UK).

Dr. Andrew West received a first class degree in Physics from Leeds University in 1983 and a Ph.D. in Astrophysics in 1987. He has over 17 years experience of research (at Cambridge and Loughborough Universities) and industrial consultancy in many industrial sectors (e.g. automotive, rail, electronics manufacture, packaging, pharmaceutical, sports and healthcare). His main research focus over the past five years has been on the lifecycle engineering of intelligent, distributed, component-based control and monitoring systems

Szer Ming Lee has a Ph.D. in manufacturing automation from Loughborough University. He also has Master of Engineering, and Bachelor of Manufacturing and Production Engineering degrees from the National University of Singapore. Dr. Lee has extensive technical expertise in distributed control systems including the implementation of distributed control-network architectures and the integration of complex automation systems. He has extensive knowledge of real-time embedded systems software development on a range of platforms and applications engineering experience in the automotive and rail sectors