

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



**CC creative commons**  
COMMONS DEED

**Attribution-NonCommercial-NoDerivs 2.5**

**You are free:**

- to copy, distribute, display, and perform the work

**Under the following conditions:**

**BY:** **Attribution.** You must attribute the work in the manner specified by the author or licensor.

**Noncommercial.** You may not use this work for commercial purposes.

**No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

**Your fair use and other rights are in no way affected by the above.**

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:  
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

## Distribution of Machine Information using Blackboard Designed Component for Remote Monitoring of Reconfigurable Manufacturing Systems

V. Barot, R. Harrison and C. S. McLeod

Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University  
Loughborough, United Kingdom

Email: {V.Barot, R.Harrison, C.S.Mcleod}@lboro.ac.uk

**Abstract**— A blackboard-based design for a system component called the “Broadcaster” is described in this paper. It supports remote monitoring of reconfigurable manufacturing systems using a novel system architecture coupled with the Component-Based system paradigm. The design of this component has been evaluated using a case study on a web services-enabled test rig funded by the Ford Motor Company, U.K. The test rig has been implemented using a fully distributed control device called FTB, designed by the Schneider Electric Company. Evaluation of this component has been carried out using three scenario test cases which demonstrate the potentials offered when deploying this solution to a real production environment. The system component not only operates in a heterogeneous reconfigurable manufacturing environment, offering a vendor-independent solution to monitoring machines, but it also supports remote monitoring of the machines throughout their development and management lifecycles.

**Keywords**- *Blackboard design; Broadcaster; Remote monitoring; Reconfigurable manufacturing system; Component-Based.*

### I. INTRODUCTION

In the current economic downturn, manufacturing automation trends for industrial maintenance solutions should adopt profitable practices such as provision of remote monitoring and support services for machines. Implementing such solutions are tremendously beneficial [1] in minimising costs associated with technical support. It has been estimated that remote monitoring functionality cannot only save 20% to 30% of costs associated to the after-sales support [2] but also improve service efficiency, reduce any equipment downtime, and provide better end user satisfaction. Strategically, it establishes strong relationship between business partners (such as control vendors, component suppliers, manufacturing end users and the machine builders) involved within a machine development and management lifecycle [3].

#### A. Aim

In today’s manufacturing automation domain, majority of activities are usually geographically distributed spanning various cities, countries and even continents, for example, 14/15 engine production program of the Ford Motor Company [4]. Constant commercial pressures and outsourcing requires a remote monitoring solution which not only supports these distributed activities throughout machine

lifecycle, but also caters for any future reconfigurations. Authors aim is to highlight design of a system component which distributes machine status data emerging from a manufacturing shop floor, in a timely manner, to a number of remote engineering partner resources such as Human Machine Interface (HMI), Enterprise Resource Planning (ERP) systems, Virtual models, Engineering tools, etc, regardless of their implementation nature and mechanisms. This system component (called the “Broadcaster”) has been designed using a well known Artificial Intelligence methodology called the blackboard-based approach [5].

#### B. Review and Industrial Need

In order to realise the benefits identified earlier in this paper, a number of sophisticated approaches to remote monitoring have been reported in literature recently [6-9]. With machine sensors, actuators, computer networks and system technologies advancing in the current manufacturing era, some of the solutions sound practically implementable. Papadakis [10] have proposed an integrated data management system which monitors renewable energy sources plants. In [6], a framework of building web-based systems supporting remote monitoring functionality in a distributed operational environment has been presented. Some machine tool manufacturers in the United States have also implemented simple but limited remote alarm monitoring systems. In Japan, Toyota have implemented Makino’s technology for remote support which is proprietary and complex to maintain for end users [2].

Majority of the proposed approaches are domain-specific, functionally limited, highly expensive, complex to implement, difficult to change or vendor dependent. In ideal circumstances, engineers would prefer monitoring and performing necessary maintenance to manufacturing products from their physical locations, without incurring costs and time associated with travelling to the actual site. Achieving this sound appealing and realistic with the current technological exposure to Internet; however there are a number of major factors which prevent this idealism to be realised satisfactorily in today’s era of manufacturing automation.

The first major factor is the constant reconfiguration pressures enforced by ongoing competition and frequent changes in consumer demands and expectations [11]. As globalisation encourages constant innovations to the manufacturing processes, any remote monitoring tool should dynamically reconfigure itself with any changes, avoiding

the need of having further investments such as reprogramming efforts, version updates, driver installations, and additional permissions to bypass the security infrastructure of a manufacturing enterprise. These activities consume a considerable amount of resources. For example, currently, reconfiguring a machine line by adding new components requires reprogramming of an entire HMI system to operate correctly in a changed environment.

Adoption of proprietary standards and technologies by majority of the manufacturing partners is also an issue to consider. This not only results into complexity in identifying and troubleshooting shop floor machine problems effectively, but it also makes users unable to learn from any past mistakes. Furthermore, inconsistencies among hardware and software implementations usually arise when deploying solutions from various vendors. For example, linking an HMI system from a vendor “A” with a control device (such as PLC, etc) from a vendor “B” is a very complex issue in today’s manufacturing domain. Owing to these problems, manufacturers tend to be bound to a specific vendor for a complete solution.

Next generation of remote monitoring solution should not only target these problems but also support machine monitoring throughout its development and management lifecycle [12]. The development lifecycle of a machine corresponds to planning, designing and launching processes, whereas, the management lifecycle involves implementation, maintenance and future reconfigurations carried out to the machines. It is essential to remotely monitor any operational, mechanical or electronic issues associated with the machine design prior to its implementation to save resources such as time, energy and costs. The “Broadcaster” component supports remote monitoring throughout the machine development and management lifecycle and promotes vendor independence in a reconfigurable environment using a Component-Based paradigm coupled with a simple system architecture (described later).

The paper is organised as follows; next section describes a reconfigurable manufacturing systems terminology and the underlying paradigm supporting this functionality. Section three of this article details the design of the “Broadcaster” component and the processes by which it supports remote monitoring in a reconfigurable manufacturing systems domain. Section four discusses an implementation case study of this system component on a web service-controlled Festo test rig provided by the Ford Motor Company, UK. Furthermore, authors also evaluate a reconfigurable test rig environment using three scenario test cases. Based on the design and implementation success, authors have concluded this paper in section five, highlighting the potentials offered.

## II. RECONFIGURABLE MANUFACTURING SYSTEMS

Reconfigurability is the key to handling constant modifications expected in the field of manufacturing automation to achieve cost-effectiveness, and increase life and the utility of a manufacturing product [13]. Reconfigurable systems are usually designed in such a way that a rapid change to their structure can be carried out to

adjust their production capacity, functionality and integration technologies, to newer circumstances owing to the environmental and market pressures. Reconfigurability may either correspond to hardware changes to the control, machines, entire production systems, sensors, actuators, or software modifications to the tool solutions, control logic, virtual models, etc.

With respect to reconfigurability support to the field of manufacturing automation, research at Loughborough University has identified and implemented a Component-Based paradigm to engineering machine control using modular mechatronic device called a “Component”. This paradigm aims to replace the existing widely used PLC based or a PC based system architecture for machine control [14]. The fundamental concept of this approach is to develop manufacturing systems from a set of components residing in pre-compiled libraries as shown in the Fig. 1. Newer systems can be composed from components that have already been developed, evaluated, verified and implemented in the past. Any reconfiguration requires only some newer components to be evaluated, verified and implemented, while using majority (almost 70% [15]) of existing components from the libraries. New components are validated using a “Validation” module of an “Engineering Toolset” which prevents re-validating the entire configuration; instead, it validates newly added configuration definitions. This approach not only reduces the costs and efforts for composing newer systems but also saves a lot of time to market a product based on customer demands and expectations.

Fig. 1 shows a simplified representation of this paradigm. A detailed description and implementation of this paradigm is beyond the scope of this paper, thus readers are encouraged to refer to their implementations in the articles by Harrison et al [15] and Lee et al [16]. In this paradigm, a “System” corresponds to any complete machine implementation. Any system contains one or more “Component(s)” which corresponds to a physical node on a control network, encapsulating the necessary physical and logical properties of a system.

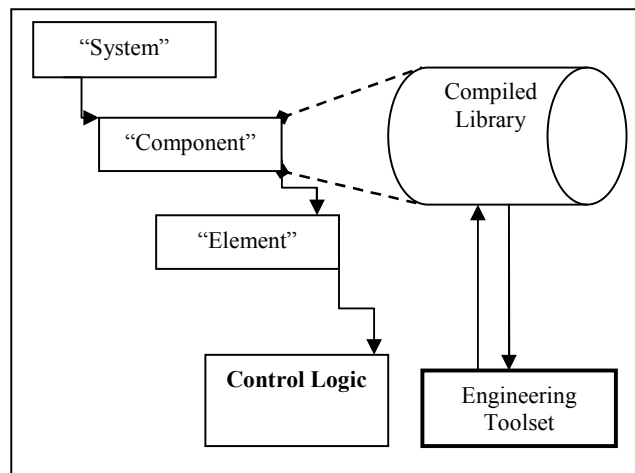


Figure 1. Component-Based system paradigm

Integrating these components by selecting them from a library can eventually compose a desired system. An “Element” can be any input or output device such as a sensor or an actuator respectively, or a more complex drive. Each element can be configured with a unique finite state machine which defines its “control logic” within the system. The lowest granularity level for this paradigm is the component, which can be configured to operate in a new environment using the “Engineering Toolset” designed at the author’s premises. This paradigm promotes control engineering via defining configuration data in a uniform format, which ultimately becomes the application logic of a control system, rather than programming the application specific code using PLC languages like ladder logic, sequence charts, or structured text. Next section describes the design of the “Broadcaster” system component, using the blackboard based design methodology.

### III. “BROADCASTER” DESIGN

#### A. *Blackboard Design for the System Component*

Blackboard design approach is a task-independent A.I. approach based on the traditional blackboard architecture model. Corkill [17] describes three core components of this approach being a blackboard, knowledge source (KS) and a controller. The basic concept has been discussed in literature, where a meeting room metaphor corresponds to the practical application of this design approach solving problems in a natural distributed team environment [5, 17]. A number of different specialists surround a discussion board to cooperatively solve a complex problem which has been posted on to the board. Each specialist only contributes if and when necessary, based on the current state of the discussion board. With an account to this metaphor, the discussion board corresponds to the blackboard component of this methodology. The specialists responsible for applying their expertise in solving problems correspond to the knowledge sources. In order to control the flow of the problem solving process and schedule contributions of each specialist onto the discussion board, a controller component of this methodology is required [5].

Authors have identified target requirements and principles for driving the design of the “Broadcaster” system component in [18]. These requirements motivate adoption of the blackboard based approach to designing this system component. All the requirements and principles have demonstrated a need to adopt a methodology which not only supports soft real-time acceptable communication and implements loose coupling among various sources, but also, supports any indirect dynamic communication requirements among distributed partner resources. Furthermore, this methodology benefits from promoting modularity by regularising interactions between various sources [19]. It exposes rigid interface definitions through which various knowledge sources can be accessed. In the field of software architectures, these systems are normally referred to as repositories [20].

The literature, coupled with author’s experience, has led this component’s blackboard model to be subdivided into

five distinctive panels (as shown in the Fig. 2), each dedicated to solve a different part of an original problem. This division solely aims at implementing efficient data retrieval mechanisms for better performance. Each of this panel has a distinctive hierarchy within which each level is independent of the content processing of the other level, thus any interaction between one level and the next is only via a knowledge source (“ks”). Any interaction between one panel and the other panel is handled by dedicated knowledge sources. These interactions are triggered by event processing techniques within the controller mechanism of the design, thus one can describe the application architecture to be a heterogeneous combination of event invocations and blackboard approach.

In the Fig.2, the “In-load” panel deals with collecting and analysing machine information, in a timely manner, prior to propagating it to the appropriate panel for further processing. Based on the type of event generated on this panel, corresponding knowledge source is triggered and further processing is carried out at the corresponding panel’s level. “In-load” panel is also responsible for importing any global configurations (i.e. Component-Based model’s uniform configurations described earlier) within the system to support reconfigurability. Since any modification to the shop-floor machine is engineered within these global configurations, reconfiguration can dynamically be accommodated within the implemented system component. The “Machine-State” panel processes all machine state information and stores it into a buffer which is dedicated to further propagating the current status of the machine. The “Diagnostic” panel processes machine errors, critical messages and diagnostic information prior to its storage. The “Operation” panel processes any machine action in the required format prior to its storage. The “Out-load” panel deals with amalgamating stored information from all the panels. Furthermore, it efficiently stores all amalgamated information as described in [21]. The storage mechanism adopted for this component is a reconfigurable circular buffer, which enables the “Broadcaster” to process data on real-time. Any stored information is ultimately propagated to unlimited number of remote engineering partner resources (e.g. HMI) in a uniform format, regardless of the mechanisms adopted by these resources or their distribution nature.

The controller mechanism of this blackboard design handles processing of all the machine information which is then published and subscribed through a number of knowledge sources. This controller mechanism synchronises and coordinates information access within the system. The emplacement of the controller is distributed i.e. partly in the prototype’s blackboard model for scheduling, and partly in the knowledge sources for executing. A number of individual knowledge sources (internal as well as external) simultaneously operate on the blackboard. The external sources can remotely be triggered from the partner resources implementation such as HMI. Every knowledge source is specialised to contributing the required functionality through its well defined interface. As every knowledge source has a well defined interface, it promotes modularity in the design.

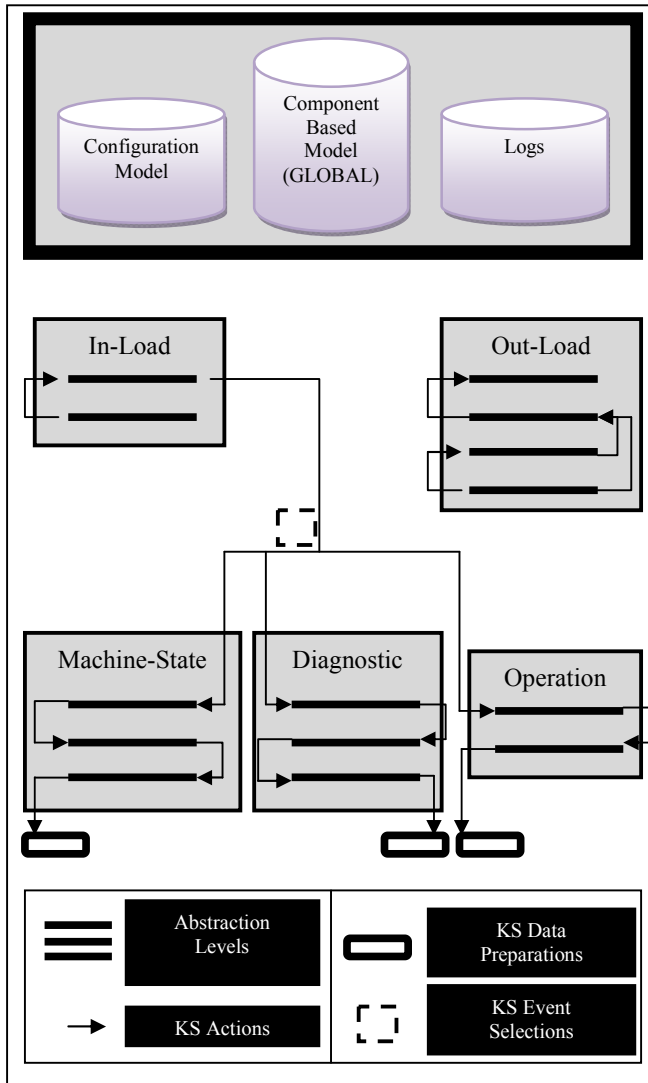


Figure 2. Blackboard based “Broadcaster” component design model

### B. Reconfigurability Support and Vendor Independence

The global configurations (which are defined by the “Engineering Toolset”) are imported by the “In-load” panel at runtime, ensuring that any change to the machine is automatically shared within the system design (as shown in the Fig. 3). Since this research also aims at supporting engineering partner resources such as the HMI, to remotely monitor machine status without any reprogramming efforts and provide a vendor-independent solution, system architecture has been designed by the authors (shown in the Fig. 4). The aim of this is to decompose overall responsibility for supporting remote control and monitoring into three major system components.

The scope of this paper surrounds description of the “Broadcaster” system component; nevertheless, it is equally important to appreciate the functionality offered by the other two system components namely “Web-HMI” and

“Marshaller” component. The “Marshaller” system component provides remote control functionality by establishing necessary communication protocols between the “Web-HMI” component and the simulated, real or a hybrid machine. Furthermore, this component provides repository services by storing historical machine transactional events for troubleshooting purposes during the machine development and management lifecycle. The “Web-HMI” component supports provision of a vendor-independent web-based HMI solution using the Client/Server architectural model implementation. Furthermore, it enables virtual view of the machines using an integrated VRML visual representation of the shop-floor machines.

Since the Component-Based paradigm serves as an underlying framework for this architecture, visibility and accessibility by all three system components promotes reconfigurability and vendor independence. The configurations are in a uniform format which can be decoded by any vendor’s controller (such as PLC, FTB, LONWORKS, etc) as well as engineering partner resource (such as HMI, ERP systems, etc) regardless of its implementation mechanism or geographical location and distribution. As shown in the Fig. 4, regardless of any industrial controller type (e.g. PLC, FTB, etc), any underlying communication technology (e.g. web services, industrial Ethernet, etc), and the nature of remote clients (e.g. HMI, ERP, etc), “Broadcaster” propagates data from the shop-floor machines to remote destinations without any modifications. For legacy or endemic control systems such as those based on a PLC, a “Middleware” (based on a DAIS or OPC standard) can be implemented which performs necessary communication translations with the “Broadcaster” system component.

### C. Machine Lifecycle Support

“Broadcaster” does not discriminate between machine implementation states namely; real, simulated and hybrid machine state. Real machine state is where an actual machine has been implemented at the shop-floor. Simulated machine state corresponds to a design stage when the machine is only implemented on a visual simulator. The hybrid state enables a machine to be partly real and partly simulated at the same time. For example, at reconfiguration stage, adding a new hardware component requires simultaneous monitoring of its simulation model with the real implemented machine, to identify any potential problems earlier during the machine lifecycle.

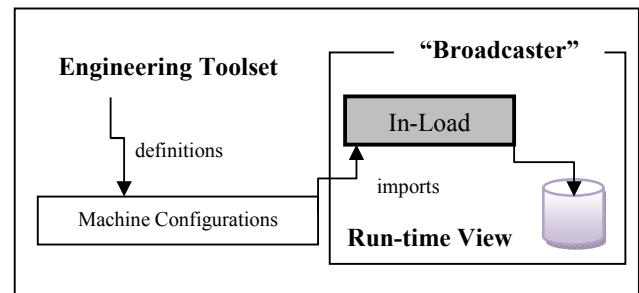


Figure 3. Global configurations at runtime

Since this system component supports all the machine implementation states, it provides support throughout the key development and management stages of the machine life cycle. This feature is beneficial as it saves time, costs and efforts involved in re-designing, implementing, training and travelling to the actual site. Any potential problems during the lifecycle can remotely be monitored from the “Web-HMI” system component using the “Broadcaster”, and rectified using the “Marshaller” system component.

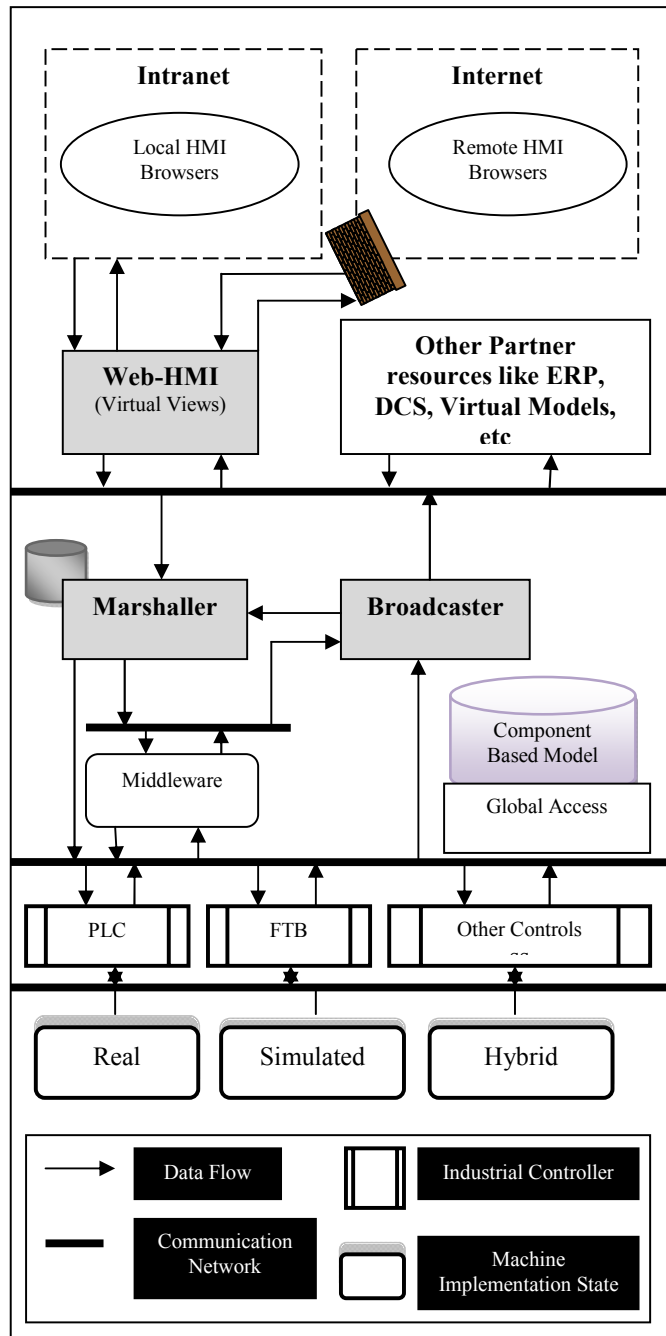


Figure 4. System architecture supporting remote control and monitoring

#### IV. IMPLEMENTATION CASE STUDY

In [18], authors developed special tools which evaluated minimum functional requirements this component has to support, prior to commencing a full implementation study. In this paper, authors have evaluated this component using a case study on a web services-enabled Festo test rig funded by the Ford Motor Company, U.K. This test rig represents a production environment such as an engine assembly line consisting of four operational stations. Each station is composed of one or more components engineered using the Component-Based system paradigm. The communication is purely Ethernet based TCP/IP and the underlying technique adopted to program the control logic is the web services.

Unlike the traditional centralised controller used in production environments, i.e. PLC, this rig has been engineered using a fully distributed control device called FTB (Field Terminal Block) designed by the Schneider Electric Company [22]. FTB provides a truly distributed control architecture enabling devices to communicate with one another in a fully autonomous fashion. The total number of FTB's implemented amount to four, each controlling one station of the test rig. Major aim of this study is to use the “Broadcaster” to provide remote monitoring support by distributing machine information from the test rig to the client resources like remote HMI. Furthermore, the system component should also demonstrate its effectiveness in a reconfigurable manufacturing systems environment.

In order to demonstrate this, three scenario test cases have been developed whose experimental setups have been summarised in the Table I. The corresponding Fig. 5 shows the results obtained in a remote HMI screen for each scenario. Initially in “SN1”, a single station “number 2” is run and the “Broadcaster” automatically configures itself to distribute data to remote HMI for only that station. As seen on the corresponding HMI screen in the Fig. 5, only “Station 2” appears dynamically on the HMI screen. In “SN2”, “Broadcaster” is evaluated on three stations without reprogramming its configuration parameters. The HMI screen automatically monitors three stations (i.e. station 1, station 3 and station 4). When “Station 2” is assembled with other stations in the test rig (in the scenario “SN3”), the “Broadcaster” distributes data from all four stations for remote monitoring purposes as shown in the Fig.5. This demonstrates the strength of the proposed approach as it enables machine data to be distributed to remote client resources, without any further need of reprogramming efforts. The solution works regardless of any machine control implementations (such as PLC, FTB, etc) and supports remote monitoring throughout the major stages of the machine development and management life cycle. Additions, modifications and removals of machine(s) or their elements do not require making any changes to the “Broadcaster” system component.



TABLE I. SCENARIO TEST CASES EVALUATION

Scenario Details				
Implementation Target: Festo Test Rig				
Control Device: FTB				
Ref	# Stations	Configuration Type	Client Resources	System Changes Needed
SN1	1	Independent (Station 2)	Local/Remote HMI	None
SN2	3	Existing Line (Station 1,3,4)	Local/Remote HMI	None
SN3	4	Assembled Line (Station 1,2,3,4)	Local/Remote HMI	None

V. CONCLUSION

A blackboard-based design for the “Broadcaster” system component has been described in this paper. Novelty of this research is the design of a remote monitoring component which provides vendor-independent (open) solution in a reconfigurable manufacturing environment, and supports machines throughout major stages of their lifecycle. To realise this solution, system architecture, consisting of three major system components, is implemented in a Component-Based controlled environment. The system component’s blackboard design supports remote data propagation in a uniform format, enabling remote clients to plug into the system architecture and monitor machine status. The design also enables this system component to collect data efficiently from manufacturing systems, regardless of industrial controller type implementation, and process it, ready for distribution to unlimited number of partner resources at real-time.

This research is a step forward to establishing stronger business relationships among stakeholders involved in the machine lifecycle. It opens up opportunities to remotely support machines in a reconfigurable domain, making information access more transparent and efficient. Application of this research is not just limited to the automation industry, other industries such as food packaging, petrochemical, etc, can benefit with the ever changing requirements of faster ramp-up processes, and remote monitoring the status of their physical processes by analysing and implementing this system architecture, coupled with the benefits offered by the Component-Based paradigm.

ACKNOWLEDGMENT

The authors acknowledge the support of the EPSRC in carrying out this work on the Loughborough University IMCRC Business Driven Automation (BDA) project.

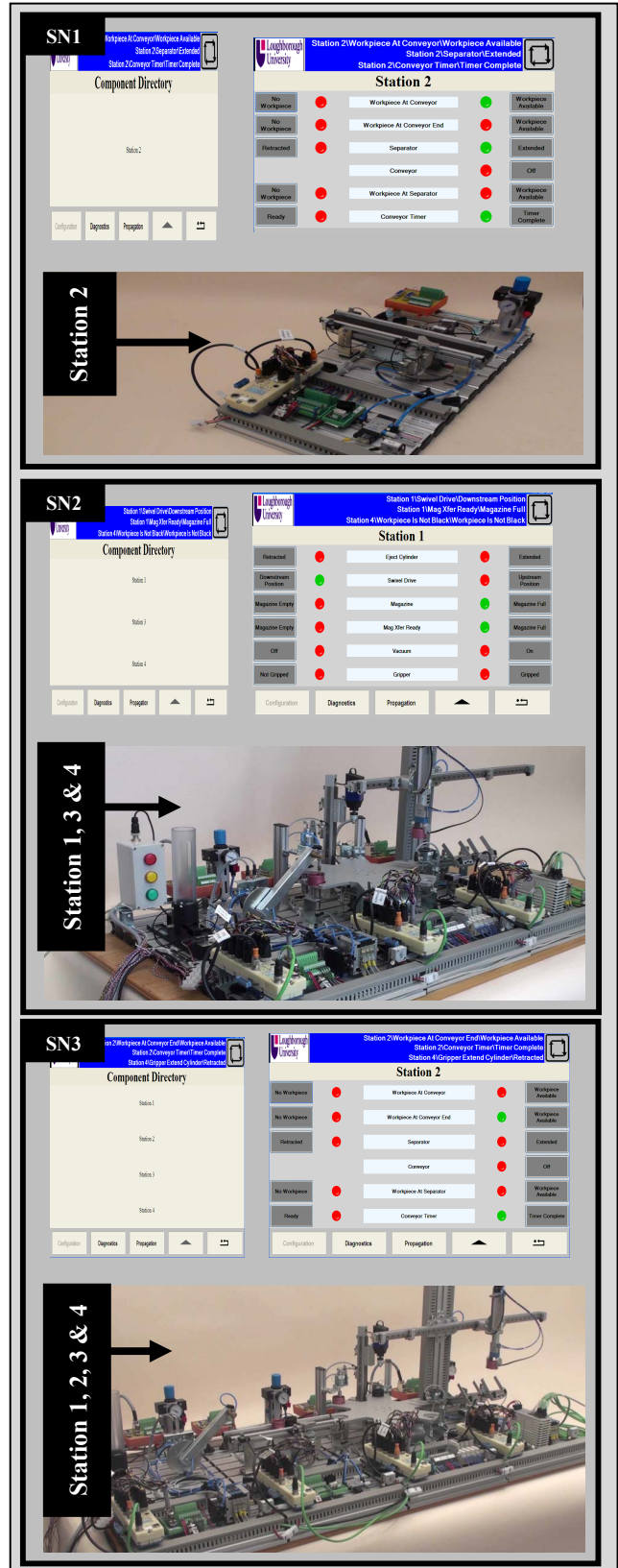


Figure 5. Scenario cases evaluation results

## REFERENCES

- [1] Hibbert, L. and T. Here, Everywhere. *Professional Engineering*, 2000. 13: p. 16-40.
- [2] Biehl, M., E. Prater, and J.R. McIntyre, Remote repair, diagnostics, and maintenance. 2004.
- [3] Wang, C., L. Xu, and W. Peng, Conceptual design of remote monitoring and fault diagnosis systems. *Information Systems*, 2007. 32(7): p. 996-1004.
- [4] Harrison, R., et al., Distributed engineering of manufacturing machines. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 2001. 215(2): p. 217-231.
- [5] Corkill, D.D. Collaborating software: Blackboard and multi-agent systems & the future. 2003.
- [6] Wang, L., et al., Integrating Java 3D model and sensor data for remote monitoring and control. *Robotics and Computer Integrated Manufacturing*, 2003. 19(1-2): p. 13-19.
- [7] Kimura, T. and Y. Kanda, Development of a remote monitoring system for a manufacturing support system for small and medium-sized enterprises. *Computers in Industry*, 2005. 56(1): p. 3-12.
- [8] Aziz, A., et al. Remote monitoring using sensor in greenhouse agriculture. 2008.
- [9] Hou, T.H., W.L. Liu, and L. Lin, Intelligent remote monitoring and diagnosis of manufacturing processes using an integrated approach of neural networks and rough sets. *Journal of Intelligent Manufacturing*, 2003. 14(2): p. 239-253.
- [10] Papadakis, K., E. Koutroulis, and K. Kalaitzakis, A server database system for remote monitoring and operational evaluation of renewable energy sources plants. *Renewable energy*, 2005. 30(11): p. 1649-1669.
- [11] Ong, M.H. and S.M. Lee, Evaluating the use of multimedia tool in remote maintenance of production machinery in the automotive sector. *Robotics, Automation and Mechatronics*, 2004 IEEE Conference on, 2004. 2.
- [12] Takata, S., et al., Maintenance: changing role in life cycle management. *CIRP Annals-Manufacturing Technology*, 2004. 53(2): p. 643-655.
- [13] ElMaraghy, H.A., Flexible and reconfigurable manufacturing systems paradigms. *International journal of flexible manufacturing systems*, 2005. 17(4): p. 261-276.
- [14] Ong, M.H., Evaluating the Impact of Adopting a Component-Based Approach within the Automotive Domain. PhD dissertation, MSI Research Institute, Loughborough University (internal), 2004.
- [15] Harrison, R., et al., Reconfigurable modular automation systems for automotive power-train manufacture. *International Journal of Flexible Manufacturing Systems*, 2006. 18(3): p. 175-190.
- [16] Lee, S.M., R. Harrison, and A.A. West, A component-based control system for agile manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 2005. 219(1): p. 123-135.
- [17] Corkill, D.D., Blackboard systems. *AI expert*, 1991. 6(9): p. 40-47.
- [18] Barot, V., et al., "Broadcaster": An Architectural Description of a Prototype Supporting Real-time Remote Data Propagation in Distributed Manufacturing, in 7th IEEE Conference on Industrial Informatics 2009: Cardiff, United Kingdom.
- [19] Lau, T.L., H.Y.K. Lau, and A. Ko, A Distributed Blackboard-based Control System for Modular Self-Reconfigurable Robots. The University of Hong Kong. Department of Industrial and Manufacturing Systems Engineering, 2003.
- [20] Shaw, M. and D. Garlan, *Software architecture: perspectives on an emerging discipline*. 1996: Prentice-Hall, Inc. Upper Saddle River, NJ, USA.
- [21] Barot, V., et al., Efficient real-time remote data propagation mechanism for a Component-Based approach to distributed manufacturing, in International Conference on Manufacturing Systems Engineering. 2009: Rome, Italy.
- [22] Colombo, A.W., SOCRADES. <http://www.socrates.eu/Documents/objects/file1224780946.72>, Accessed on: September 2009.