Proceedings of the 11th International Conference on Manufacturing Research (ICMR2013), Cranfield University, UK, 19th – 20th September 2013, pp 57-62

AN IMPROVED CELL CONTROLLER FOR THE AEROSPACE MANUFACTURING

Seemal Asif
Department of Integrated Systems
Cranfield University
Cranfield
Bedford, MK43 0AL, UK
s.asif@cranfield.ac.uk

Philip Webb
Department of Integrated Systems
Cranfield University
Cranfield
Bedford, MK43 0AL, UK
p.f.webb@cranfield.ac.uk

ABSTRACT

The aerospace manufacturing industry is unique in that production typically focuses on high variety and quality but low volume. Existing flexible manufacturing cells are limited to certain types of machines, robots and cells which makes it difficult to introduce any changes. In this paper idea of treating machines, robots, any hardware and software as resource has been introduced. It describes the development of the Flexa Cell Coordinator (FCC), a system that is providing a solution to manage cells and their resources in a new flexible manner. It can control, organise and coordinate between cells and resources and is capable of controlling remote cells because of its distributed nature. It also provides connectivity with company systems e.g., Enterprise Resource Planner (ERP). It is extendable and capable of adding multiple cells inside the system. In FCC resources (e.g., tracker) can also be shared between cells. The paper presents its development and results of initial successful testing.

Keywords: Flexible Manufacturing System (FMS), Distributed Network Cell Controller, Cell Controller, SoftPLC, Aerospace Cell Controller, Web Services

1 INTRODUCTION

There is little use of automation in aerospace manufacturing compared with other industry sectors such as automotive, pharmaceutical or white goods. This is because the product volumes tend be small but the product lifecycles may exceed 60 years. High quality parts must be used and all the associated processes must be traceable and verifiable. The variety of parts is also high and this low volume high variety mix means that a large number of conventional automated systems would be required that will have low utilisation rates and therefore be uneconomical.

The Flexible Manufacturing System (FMS) was developed to overcome some of these issues (Greenwood, 1988). FMS utilises the integration of automated component storage, tool delivery and CNC machines with an overall computer control unit to support and monitor the performance of the system (Shivanand, 2006). These FMS installations have been very successful; however in many cases the flexibility has been very low when trying to produce a wide variety of changing components within existing cells. In the case of aero-structure assembly the problem becomes even more complex. This means that multiple processes are likely to be performed using the same processing equipment but in different locations. To enable this, a new approach of cell control and organisation is required. For example a robot may be used for a drilling task in one part of the factory and then used later for applying sealant in a completely different part of the factory.

2 BACKGROUND

A manufacturing robot cell can consist of 20 or more robots along with other machines. They are then controlled by Programmable Logic Controller (PLC). Flexa is the acronym of Flexible Automation Cell. The term flexibility is defined as the system's ability to adopt the change easily. (Fricke, 2000) Flexa is based on the concept of FMS. There are four classes of manufacturing attributes that need to be considered while designing manufacturing system: (i) cost, (ii) quality, (iii) time and (iv) flexibility. FMS is the approach for flexible and cost-effective means of manufacturing. FMS can consists of two or more computer-managed workstations, material transport system and another computer that controls the transportation operations, tools and other related information. There are a number of components involved in forming FMS like material handling and storage system, programming language and network infrastructure, workstations and human labour. The reason behind using FMS is that businesses want the groups of machines and tools to form a system along with programming and network that can work continuously and with minimal intervention from human labour. (Chryssolouris, 2006)

There are lots of operational tasks of FMS implementation like machine loading, part routing, grouping, tool management, scheduling, etc. Scheduling is an important element (when for about) of FMS operations. Holonic manufacturing scheduling has been used for the scheduling of cells (Gou, et al., 1998) but it doesn't state any communication between the cells and its resources. Moreover although it refers to the automation of cells but any modification in the cells' resources and the processes could take up to a year for planning and implementation.

The computer software applications can be utilized for assistance in FMS. There are several software techniques that are used for FMS and application of service oriented architecture (SOA) in the aerospace industry. This was proposed by the Telematics Department of Hamburg University of Technology (Kazlauskaite, et al., 2005). Their methodology proposed the collaboration between various enterprise systems in aerospace industry. They used web services interfaces for the integration and communication between existing systems i.e., ERP (Enterprise Resource Planner). The study shows how to make the system flexible by using the SOA approach. They proposed a set of processes along with verification for the implementation of SOA in aerospace industry applications. Their approach gave the idea of implementing SOA for an aero-structure assembly FMS. However their concept does not give any detail how it can interface with the cell and its resources.

RAPOLAC (Rapid Production of Large Aerospace Components) is another example of an automated cell (Gault, 2010) referring to automation of a cell and the resources using the feedback data from a laser tracker. It is also talking about the automation of a single cell and not multiple cells. It does not detail the underlying architecture for the automation. The Automated Assembly of Wing Panel for A340-600 has the similar footprints. (Holden, et al., 2007)

The problem with the current automated cells is that they are not fully automated. There is still a need for manual intervention such as loading programs on controllers, loading programs on PLC, changing programs on PLC etc. A little advancement towards minimising the human intervention is the implementation of SOA. The work for the aerospace industry using SOA was analysed (Kazlauskaite, et al., 2005) it is giving idea for coordinating between the aerospace industry applications but that is not giving information about cell itself and how to communicate between resources e.g., robots.

Automotive industry may not need rapid changes in the cells and processes but the aerospace industry is unique in the sense of every part and its manufacturing. The need to reposition the entire robot around the part is understandable in aerospace industry as work can be done on different models in the same cell. There may be need to change the entire cell structure to assemble different parts of the wing. Hence there is a need for a completely automated and flexible system which can cater for the needs of aerospace manufacturing industry.

3 FLEXA CELL COORDINATOR

FCC is a fully automated system which receives programs and transfers it to the required resources in a controlled manner. Resource can be robot, machine, hardware or software. FCC is the part of Flexa Cell in which it is coordinating between cells and its components. It is responsible for execution of received programs in a conflict free way on the required resources. It is also important to run and manage multiple cell coordinators at the same point in time. For this purpose the use of a software Programmable Logic Controller (SoftPLC) was introduced which means there is no need for hardwired binding of resources with the cell.

The FCC receives the program outside of the system using web services in a specific format called recipe. It un-marshals the recipes and schedules it according to the availability of resources. It then activates or creates the Cell Sub Coordinator (CsC) which is having SoftPLC and the controller of the programs and resources. The controller called Programme Manager (PM) downloads the program onto the resources and gives control to SoftPLC program to control the resources and program execution. It is also responsible of getting data back from resources (if required) and sends it back to main program of FCC called Application Manager (AM). And from here the data can send back to the resource of the recipe. The FCC is having two way communications as it accepts the data in the form of recipe and sends information back to the sender e.g., ERP system.

4 FCC ARCHITECTURE

The methodology for control and organization developed makes a number of assumptions about the process and the production resources being used. These are as follows:

1. Production is assumed to be chaotic due to the number of processes and the likelihood of concessions needing to cleared.
2. The resources (robots, machine tools etc.) can be used in different sequences for different operations either by physically relocating them or changing the root of a part through the resources.

The issues noted above mean that the use of a conventional control methodology using a Programmable Logic Controller (PLC) and a number of machines (resources) physically coupled together in a cell(s). The approach of running preloaded programmes on resources is impractical as in the cases above the individual cells would need to be physically reconfigured for each operation. A new way of approaching this problem is to use a central cell controller. That is capable of producing any number of virtual controllers which can take control of local groups of machines to form virtual cells which then behave like conventional physical cells. These CsC are software applications which are tied to their resources using a common interface. The overall cell controller is responsible for decoding recipes, allocating and scheduling resources and launching and destroying instances of the CsC when required. It also contains a database which is used to store status information to allow recovery of system status in the case of an equipment or process failure. In summary:

1. The use of a SoftPLC means that there is no longer any dedicated physical hardware associated with the cell.
2. All the production systems for example machine tools, robots or measuring systems are classed as networked resources which have a common interface which allows them to be interrogated and identified automatically.
3. When a particular production sequence is identified a ‘recipe’ is generated from which the required resources can be identified, allocated and programming information loaded.
4. The resources are co-ordinated using SoftPLC which is connected to the resources over a network. The overall control is provided by a cell co-ordinator which allocates a virtual CsC for each cell. Once the task is finished then the virtual CsC is closed and all the resources are freed up and made available for other tasks that may be waiting for resources.

A diagram showing the structure of the cell coordinator is shown in figure 1. The individual elements of the architecture have the following functions:

1. Application Manager (AM): It is responsible for the control of the FCC. It has responsibility to activate the FCC and communicate between the FCC’s components. It also has responsibility to communicate with the rest of the world via the web services layer.
2. Scheduler: This is needed to schedule the task among resources. The Scheduler will be able to identify the available resources and will allocate the task to them according to the recipe (sent from FDB). The Scheduler will also be able to resolve and avoid conflicts (Deadlocks, etc).
3. Status Database and Monitor: The Cell coordinator status database is used to record the status of the resources available and the availability of resources (if they are available for handing over to a task). The status database monitor has active two way connection with the status database monitor which monitors all the activities of the resources and records the status of all the current recipes.
4. Recipe Queue (RQ): It accepts the request from the AM to execute a particular recipe, if there is one present. The recipes will be sent out for execution by the scheduler on first come first serve basis.
5. CsC: It is comprised of a PM and the SoftPLC. The PM receives the recipe from Scheduler and delivers programs to the resources(s). The Soft PLC also takes its program from PM and

controls the resource(s) accordingly. There can be multiple CsCs all of which will be controlled by the AM. One CsC will work with one set of resources and each other one will use a different set of resources.

6. Recipe is the XML file which contains the configuration of recipe files i.e., resource programs. There are a number of ways to set up the recipe. It can point programs from other recipes as well. A single recipe can run in one CsC and collaborate with other CsC.

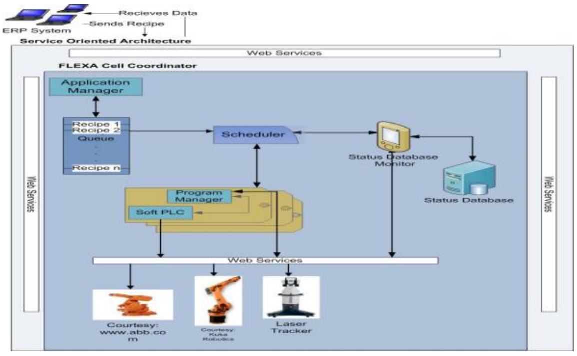

Figure 1: FCC Architecture.

5 FCC FEATURES

Main interface displays the summary of the FCC. It presents numbers about the recipes (their details), CsC, and resource status. Figure 2 shows the main interface for the operator of FCC.

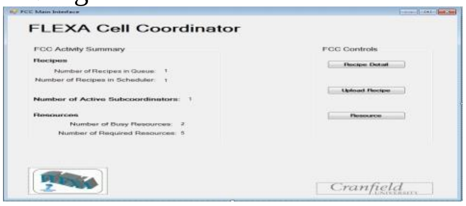

Figure 2: FCC Main Interface

FCC features are highlighted in figure 3.

1. Supporting multiple cells concurrently
2. Storage of recipes and scheduling
3. Runtime resource management
4. Adding resource into the cells runtime
5. Complete logging and tracking of the system using database management system
6. Capability of restarting the system after recovering itself from error
7. Resource sharing
8. Simple and friendly user interface for operator
9. Adding cell into the FCC system on runtime
10. Distributable control

Figure 3: FCC Salient Features

6 DESIGN OF FCC

The biggest challenge in designing a flexible control system with varying cell resources is to design a loosely-coupled structure without losing efficiency and yet the system should also be easy to reconfigure and extend. Since the cell resources usually come from different manufacturers and use different platforms and communication protocols the traditional distributed computing technologies such as COM (Component Object Model), could lead to very tightly a coupled relationship between cell resources. Any changes to the system such as a newly installed metrology system may need some alterations within the original software which significantly reduces flexibility and re-configurability (Pires & Costa, 2000).

It was therefore proposed that a SOA would be more efficient. An SOA uses a web service as the basic element. This was originally designed to support interoperable machine-to-machine interaction over a network (Aphrodite & Thomi, 2002). A web service is platform-independent as it uses the standardized Simple Object Access Protocol (SOAP) as its communication protocol (Newcomer &

Lomow, 2005). As a result, any device that supports TCP/IP communication can be programmed to provide a web service which greatly reduces the complexity of communicating with different platforms and environments.

7 FCC TESTING

Testing the FCC system proves its strengths and flexibility. It has been tested in various ways such as (i) support of multiple resources, (ii) adding / configuring resource while FCC is running, (iii) support of multiple cells concurrently, etc. SoftPLC was managed to recognize the system and operate all of the resources inside cell. FCC was able to send and receive recipes for all of the cells and was able to create instances of CsCs for all of them. Figure 5 shows how SoftPLC manages multiple cell coordinators concurrently. This is the test which we run in the real cell figure 4 shows the original cell.

Figure 4: The Original Cell

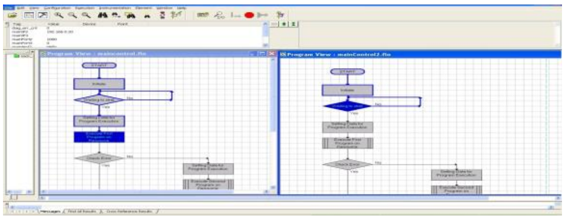

Figure 5: SoftPLC Controlling Resources in Multiple Cells

7.1 Test Case 1

Cell Coordinators 2, Resources 4 (R1: KR200 with clamping end-effector, R2: NM45, R3: LEICA Laser Tracker, R4: H4 with Drilling end-effector)
R1, R2, R4 belongs to CsC 1, R3 belongs to CsC 2
Communication ontology: Cell 1 ←→ Cell 2
Purpose: To test the capability of FCC between multiple resources and cells.
Scenario: The task was to do drilling on a nominal part. R2 was doing the pre and post inspection of the part. R1 was picking the part from loading bay, putting it on the jigs & fixture and then putting it on delivery bay. R4 was responsible for drilling on the part and R3 (virtually in separate cell) was checking the position and feeding its result to the R4 for getting the right position. Figure 6 shows the FCC cell described in this scenario.

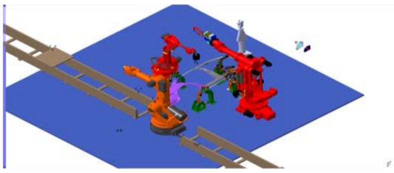

Figure 6: FCC Test Cell

Process: Recipes were sent for these cells. Recipes were received by AM and queued in RQ. After checking the resource availability recipes were picked up by FCC Scheduler (one at a time) un-marshalled and scheduled. Two respective CsCs were created which includes the PM and SoftPLC. Programs were transferred to respective CsCs. PM downloaded the recipe's programs on the resources. Programs were then waiting signal from SoftPLC to get it executed. Program which was dealing with R4 for drilling was scheduled after the completion of pre-inspection by R2. The data was sent back to the PM by the resource's (R2) program which was then transferred to the cell coordinator and then drilling task was carried out by R4. Cell #2 was also activated along with drilling. R3 was constantly checking the position of R4 and giving its feedback to Cell # 1's PM which is passing information to R4 for getting the accurate position. R4 (Cell # 1) and R3 (Cell #2) are communicating constantly via their PMs. R1 lifted the part from loading bay and put it on delivery bay after the completion of drilling process. At the end R2 did the post inspection of the part. Every cell's resources were controlled by their respective SoftPLC and the PMs. After the completion of process data was sent back to AM of FCC. The AM then forwarded the data to the source of recipes i.e., ERP. Complete process was logged at every step.

7.2 Test Case 2

Cell Coordinator 3 (includes 2 cells from test case 1), Cell #3 Digital I/O (acting as four different resources for testing purposes) R5, R6, R7, R8
Communication ontology: Cell 1 ←→ Cell 2, Cell 3
Purpose: To test the capability of FCC across multiple cells on multiple locations.
Scenario: The test case 1 along with an additional cell which was physically distributed and placed on different location. R5, R6, R7, R8 were the lights attached to Digital I/O which flashed one by one.
Process: The same process was followed by CsC 1 & 2 which is detailed in the process of Test Case 1. The CsC 3 gives respective signal to SoftPLC to control R4, R5, R6 and R7. All of these resources light up when activated by corresponding physical switch. CsC 3 worked independently as it did not need to communicate with any other cell. The rest process is same as described in test case 1.

Table 1: Test Results

	Resources	Cells	Operation Time *	Errors	Total Time **
Test Case 1	4	2	12 µ sec	0	20 sec
Test Case 2	8	3	168 µ sec	0	75 sec

* Operation means the execution of recipe on the resource under the control of Soft PLC

** Total time means time taken for all of the processes of FCC i.e., receiving of recipe, un-marshalling, scheduling, executing it on resources and sending result back.

The time of the completed operation was calculated carefully by using internal time calculation routine which is double checked by using system stop watch. The test case results show that system was able to execute the recipes successfully on the FCC without any errors.

8 CONCLUSION

As proved in the section above FCC can control multiple cells and collaborate in between them. Multiple cells can work independently as well and the can also be controlled by FCC as proved in Test Case 2. There are multiple scenarios which are tested with FCC but because of the space issue all of them cannot be illustrated here. In short FCC is fully capable of managing, coordinating and controlling cells. It becomes more cost effective with resource sharing and scheduling system.

ACKNOWLEDGMENTS

This work was funded as part of FLEXA within the European Seventh Framework Programme (FP7).

REFERENCES

Aphrodite, T. & Thomi, P., 2002. An overview of standards and related technology in Web Services. 12(3), pp. 135-162.

Chryssolouris, G., 2006. *Manufacturing Systems: Theory and Practice.* 2nd ed. New York: Springer Science + Business Media, Inc.

Fricke, E. S. A. P. W. S. &. N. H., 2000. *Systems Approach to Product Innovation and Development in Hyper-Competitive Environments.* Colorado, INCOSE.

Gault, R., 2010. *RAPOLAC readies weld-based production for the real world - SMD process offers significant savings in cost and time,* s.l.: s.n.

Gou, L., Luh, P. B. & Kyoya, Y., 1998. Holonic manufacturing scheduling: architecture, cooperation mechanism, and implementation. pp. 213-231.

Greenwood, N. R., 1988. *Implementing flexible manufacturing systems.* s.l.:John Wiley & Sons.

Holden, R., Haworth, P., Kendrick, I. & Smith, A., 2007. Automated Riveting Cell for A320 Wing Panels with Improved Throughput and Reliability (SA2).

Kazlauskaite, J., Minhas, A. & Vogt, F. H., 2005. *Applying Service Oriented Architecture in the Aerospace Industry.* Klagenfurt/Austria, International Mass Customization Meeting.

Newcomer, E. & Lomow, G., 2005. *Understanding SOA with Web services.* s.l.:Addison-Wesley.

Shivanand, H. K., 2006. *Flexible Manufacturing System.* s.l.:New Age International Pvt Ltd Publishers.