Workshop on

Control Systems for Integrated Manufacturing

Sponsored by the Commission of the European Communities, DG XIII/A, in association with Digital Equipment and the Technical University of Eindhoven.
ESPRIT Workshop

PRODUCTION ACTIVITY CONTROL SYSTEMS:
Design, Development & Implementation

Contents

1. ESPRIT PROJECT 477::COSIMA, Control Systems for integrated Manufacturing.


3. A Simulation Tool to Evaluate Factory Level Schedules.


5. COSIMA: ESPRIT Project No. 477, COMAU Implementation Description.

1 Introduction

ESPRIT, the European Strategic Program for Research in Information Technology, was set up by the Commission of European Communities (CEC) in the middle of this decade to promote research and development in the areas of information technology, with the overall aim of improving Europe’s technological base through pre-competitive research. It has research projects in five different areas, including microelectronics, software technology, advanced information processing, office systems and computer integrated manufacturing.

ESPRIT project 477 is in the computer integrated manufacturing area, and is entitled COSIMA (Control Systems for Integrated Manufacturing). It involves three major European manufacturing industries, COMAU of Turin, Italy, who produce robotic systems for flexible manufacturing environments; Digital Equipment Corporation, the largest worldwide manufacturer of networked computer systems; and Renault Automobiles of Cleon, France, who manufacture a range of engines for automobiles and trucks. The COSIMA project team also involves SESAM, a software house from Turin, and the CIM Research Unit from University College Galway.

This project was initiated to research an area in which the quality of commercially available solutions is poor, namely shop floor control, or Production Activity Control (PAC). The work on the project can be divided into three closely related phases:

- Firstly, to develop a generic architecture for Production Activity Control, which is suitable for each of the partners diverse manufacturing environments.
• Secondly, to develop a suite of software tools which can be used to aid the design and development of PAC solutions.

• Finally, to implement PAC within each of the partners manufacturing facilities, in order to validate the architecture and show that it has a role to play within future manufacturing systems.

The following three sections of this document will elaborate on each of these three important phases, commencing with a brief description of the architecture for Production Activity Control.

2 PAC Architecture

Before describing the PAC architecture, it is important to see where PAC lies in the hierarchy of planning and control systems that exist within manufacturing. This hierarchy can been categorised into three distinct levels: (1) the strategic level where decisions specify what to do; (2) the tactical level where decisions specify how and where to do it; and (3) the operational level where decisions are converted into action (see figure 1).

The strategic level has a time frame of between two and five years. It includes business planning and master schedule development. The tactical level corresponds to the planning functions such as master schedule development and materials requirements planning. The operational level includes PAC at a workcell level and a Factory Co-ordination system which co-ordinates individual PAC units. Factory Co-ordination may be seen as a higher level recursion of PAC in that it aims to control different PAC systems within a manufacturing environment. It contains a scheduling function for the plant which drives PAC systems at a workcell level. Besides this scheduling role, Factory Co-ordination also involves the design of workcells, taking Group Technology principles into account. PAC ensures that the production orders for each workcell are met; it also includes analysis after the order is completed to evaluate the effectiveness of the manufacturing activities.

The COSIMA project has modeled PAC systems based on an architecture with five distinct functional blocks (see figure 2). The role of these building blocks is to generate a production plan for a specific workcell and ensure the implementation of the plan by co-ordinating the work through its different operation steps. The Scheduler develops the plan and the Dispatcher, Monitor, Mover and Producer implement the plan based on the state of the system.

• **Scheduler:** The scheduler performs its function of planning production activities
on the shop floor by specifying the timing of operations in order to comply with due dates, priorities, availability of resources etc. Driven from the guidelines of a factory co-ordination system, the scheduler provides a time sequenced operation plan to the dispatcher building block.

- **Dispatcher**: The dispatcher executes in real time the sequenced operation plan provided by the scheduler. It does this by assigning sequenced production orders to producers and sending appropriate commands to movers to ensure the availability of materials in the right place at the right time.

- **Monitor**: The monitor performs the real time feedback function and provides a *window* to shop floor activities. It collects data on equipment utilisation, materials, stock status and quality management and reports them back to the appropriate PAC building blocks, thereby supporting the decision making process of the PAC system.

- **Mover**: The mover, which controls such shop floor transport devices as conveyors, carousels, robots, automated guided vehicles (AGVs) and manual transporters, co-ordinates the transport of material in the manufacturing plant.
• **Producer:** The context of the producer varies depending on the manufacturing environment. For example, in an automated environment, the producer is a piece of software translating high level commands into low level instructions, while in a manual environment the producer is the operator.

Thus, the development of this five-moduled architecture marked the end of the initial phase of the COSIMA project. The next phase involved developing a design tool to validate and experiment with proposed PAC solutions, and this design tool is now described.

### 3 PAC Design Tool

The project partners combined their experience of manufacturing and information technology to design and develop a Production Activity Control design tool. The primary goal of this tool is to test and validate the PAC architecture in a controlled and simulated computing environment. It is based on state-of-the-art software technology such as artificial intelligence, communications systems, relational databases and data-driven sim-
The design tool enables an experienced engineer/systems analyst to configure a PAC system, and then experiment with different production strategies prior to their actual implementation. The unique features of this design tool are:

1. It contains several different strategies for controlling activities on the shop floor. The user, with the aid of a decision-support system, can select from these various strategies, and simulate the resultant possible effects on their manufacturing system. This simulation, because it is data-driven (i.e. driven by manufacturing data in a database), does not require the modeler to have an intimate knowledge of simulation. The results of the simulation can then be analysed with a view to identifying the most appropriate control strategy.

2. It has a unique architecture, one which facilitates a migration path from the simulation of control strategies to the implementation of the same control strategies on the shop floor. Essentially, this migration path allows the main software modules of the Scheduler, Dispatcher and Monitor to “migrate” from the modeled world of simulation, to the reality of shop floor control. This is achieved by decoupling the Emulator, and connecting the Movers and Producers on the shop floor with the Scheduler, Dispatcher and Monitor of the PAC design tool.

3. It also can serve as an important educational tool for manufacturing personnel who wish to learn and experiment with Production Activity Control. Thus, people can “play around” with the design tool, creating their own factories using the database interface, and experiment by using various control strategies to achieve the most appropriate control system for fulfilling customer requirements.

These features are illustrated in figure 3, which shows the different stages of the PAC life cycle. Initially, in the design and development stage, the database is populated with manufacturing information such each product structure, and the operation steps required to produce each process. The rulesbase may be used to select an appropriate control strategy, and the next stage, that of simulation, may be attempted. The final stage, implementation, may finally be undertaken after experimentation at the simulation stage.

Thus, the PAC design tool provides a framework for developing and implementing PAC system within a broad range of manufacturing environments. Two pilot implementations of PAC systems took place during 1989, one at Digital Clonmel and the other at the COMAUU facility in Grugliasco. These implementations are now briefly described.
4 PAC Implementation at Digital

Digital's facility in Clonmel employs approximately 350 people, of which 100 are directly engaged in the manufacturing process. The plant manufactures in the region of 60 different products, and introduces over 15 new products annually. It manufactures network and communications products for Digital's wide range of computer systems. The total weekly volume of products manufactured is in excess of 3000.

The PAC architecture was implemented at a workcell control (figure 4) level in a selected area of the manufacturing process. The implementation consisted of: a manufacturing database with a user interface; the PAC scheduler, dispatcher and monitor modules; the workcell reporter and factory data collection interface, (the digital DECstr system is used at Clonmel); and a communications system, the application network (AN).

- **Database**: A manufacturing database is essential for any shop floor control system, as it holds the manufacturing data needed to plan and control the activities of the shop floor. The Clonmel database was structured to contain data describing resources, raw materials, products, processes and production requirements. A user friendly database interface was also implemented which allowed engineers to view and edit database entries.
- **Scheduler**: The role of the scheduler was to provide the shopfloor with a well coordinated plan for daily production requirements. The Scheduler had a number of options available which could be used to develop schedules, including *production smoothing, bottleneck search, bottleneck analysis* and *scheduling strategies*.

- **Dispatcher**: The dispatcher facilitated the implementation of a recommended schedule by relaying this schedule to the workcell supervisors; this was transmitted as either a hard copy output or by using the distributed message passing system provided by the application network.

- **Monitor**: The role of a monitor is to provide feedback on the current status of work in progress on the shop floor. This data, when compared with the planned work data from the plant level scheduler, gave an accurate picture of the effectiveness of the implemented schedule. The Monitor compared the planned schedule (received from the scheduler) and the actual implementation based on data received from the DECstr data collection system and the workcell reporter. It consisted of a menu driven interface which allowed the user to monitor work in progress on a plant-wide or a cell-wide basis. Information from the monitor was of the form:
  - **Workcell Name**: Machine Insertion
  - **Part Name**: DHQ11
  - **Transfer Batch Number**: ES000121

![Figure 4: PAC in the Test Environment](image-url)
- **Scheduled Start Period**: 10:45 - 12:30
- **Actual Start Time**: 10:56
- **Status**: On Schedule

- **Workcell Reporter**: The workcell reporter received the daily schedule from the dispatcher via the application network and then facilitated the recording of shop floor events. The workcell reporter ran in two modes: a *supervisory* mode and an *operational* mode. The *supervisory* mode showed a list of all the jobs to be completed at a workcell and highlighted *started, completed and late* jobs. The *operational* mode facilitated the recording of start and finish times for each transfer batch to pass through the workcell. These messages were then relayed through the application network to the monitor module to update the work in progress records.

- **Shop Floor Data Collection Interface**: This interface was written to make use of the existing data collection system in Clonmel, (DECstr). It took messages from the DECstr system and relayed them to the *Monitor* via the application network.

- **Application Network**: The application network provides the message passing facility for the live PAC system. The main benefit of the AN is that it ensures *flexibility* and *modularity* in the PAC system. The flexibility means that different building blocks can reside on different nodes in a *local area network*. This feature of the AN is critical to the implementation of PAC, as the Clonmel PAC system used three different nodes to achieve its goals. The benefit of the modularity is that more building blocks (e.g. *workcell reporters*) can be added on to the PAC system without difficulty.

Some of the main observations, which arose as a result of the pilot implementation were:

1. The PAC architecture is valid within the electronics manufacturing environment at Digital Clonmel;

2. The *participative approach*, (i.e. user participation in the development), adopted for the pilot implementation allowed both users and developers to understand the requirements for PAC in the target environment.

3. The modular and flexible nature of the five-moduled PAC architecture facilitates integration with existing manufacturing systems such as requirements planning and data collection;
5 PAC Implementation at COMAU

The COMAU plant at Grugliasco, Turin employs approximately 450 people, of which 225 are directly engaged in the manufacturing process. The plant manufactures parts for machine and assembly lines, as well as robot components and clamping equipment. For the machining and assembly line parts the final product contains around 20% modular parts. The plant operates mostly in an "engineer to order" and "make to order" environment.

The cell chosen for the implementation consisted of four machining centres, each controlled by a SIEMENS Numerical Control. A Vaxstation 2000 was used for on-line interaction with the system. Part programs and schedule information were passed to this VAX using an ethernet connection to the methods department. The pilot implementation took place over a one week period in mid November 1989. The modules developed for the implementation included:

- Scheduler - This scheduler was used not only for the machines in the cell, but was used to generate schedules for the machines prior to and after the cell itself. This was one of the main software modules developed and this is still in use there today.

- Dispatcher - The Dispatcher software routines for the COMAU implementation was really only a set of decision support routines for the supervisor. In reality the supervisor made all the dispatching decisions. The software routines compared the schedule with feedback from the Monitor in order to make suggestions to the supervisor and also issue commands to the Mover building block.

- Monitor - The monitoring functions developed for the application included work in progress, job and machine status statistics. The Monitor provided real-time graphical displays on machine status and efficiency, and constantly compared work in progress with the scheduled operations and in this way provided feedback on tardy jobs to the Dispatcher.

- Mover - There were no automatic moving operations in existence in the COMAU facility. All moving of parts is currently being done by forklift operators. The Mover was realised via the presentation of commands coming from the Dispatcher. These were presented on a terminal display to the user.

- Producer - This software controlled all the process control operations for each of the machines i.e. part program download/upload, tool management etc.

One of the most difficult tasks involved in the development of this implementation was the interface to the existing shop floor control software and hardware systems. As with
the Clonmel implementation, a major constraint was that the work could not disrupt the normal day-to-day activities of the plant. The interface to the requirements planning facility was not so difficult as this was done completely off-line and did not affect day-to-day activities. This provided the input to the scheduling system. Downward integration though was much more cumbersome, due to the difficult interfacing problems with some low-level software.

The main observations from the COMAU implementation were that:

- It provided an important opportunity for disciplined experimentation within the area of PAC, and this experimentation was conducted in a real manufacturing environment;
- The PAC architecture is valid within the FMS environment at the COMAU manufacturing facility.

6 Conclusion

Thus, over five years the COSIMA project moved from the design and definition of a PAC architecture to its pilot implementation within an electronics assembly and a flexible manufacturing environment. The factors which significantly aided this progression was the blend of diverse manufacturing knowledge and software technology experience brought together through the ongoing links between each of the project partners. The pilot implementation was the most important step taken in the project, because it showed that research ideas can survive the migration to the real world of manufacturing, and make an important contribution to the pursuit of manufacturing excellence.
Production Management Systems.
A Hybrid PMS Architecture

Paul Higgins†
Gerard Lyons‡
Jimmie Browne†

†CIM Research Unit, University College Galway, Ireland.
‡Digital Equipment International B.V., Clonmel, Ireland.
An Hybrid PMS Architecture

Abstract

In preparing the manufacturing enterprise for CIM, it is essential that manufacturing can describe its planning and control activities in a logical, consistent and enduring systems framework. This paper attempts to define an architecture for production management systems. It begins by giving an overview of the hierarchy of such systems and then proceeds to discuss the architecture under the three management time horizons of strategy, tactics and operations. The main levels in the architecture are business planning, master production scheduling, requirements planning, factory co-ordination and production activity control. The paper also discusses a simulation tool used for the modelling of the lowest level in the architecture, namely production activity control.

1 Introduction

In this paper we attempt to describe an architecture for Production Management Systems (PMS). Up to now PMS have tended to be viewed in terms of Materials Requirements Planning / Manufacturing Resource Planning, Just in Time, or Optimised Production Technology paradigms. The problem is to define a complete PMS architecture that combines and uses aspects of these differing philosophies. The overall objective is to provide a “generic” PMS framework for future manufacturing systems. It is intended that paper will just summarise the main issues in this hierarchy and perhaps highlight certain modelling techniques used in different levels. For more detailed discussions in these areas the reader should consult the references mentioned in the text.

The architecture chosen is based on one put forward by Browne et al. [2]. The premise on which this architecture is based arises from a recognition of the key failings inherent in each of the above mentioned PMS approaches. Each level in this architecture will be detailed and the different links between each one enlarged upon. Figure 1 illustrates an architecture for PMS, which extends from strategic to operational levels [2].

These levels represent different planning horizons. The length of these time horizons depend on which production environment you are operating in. (That is job shop, batch, repetitive or mass production). The strategic planning horizon may cover one to five years, tactical planning, one month to one year and operational planning, real-time to one week.

This architecture reflects a situation where a factory has been decomposed in so far as possible into a series of Group Technology based cells where each cell is responsible for a family of its products, components or processes and is controlled by a production activity control system. Another possibility is that each group is actually geographically dispersed, that is they are a number of different factories. The factory co-ordination module ensures that the individual cells/factories interact to meet an overall production plan.

The different issues involved in each level are as follows:
Figure 1: An Architecture for Production Management Systems

**Strategic Issues**: Strategic production management issues relate to the determination of the products to be manufactured, the matching of products to markets and customers' expectations, and the design of the manufacturing system. These should ensure short production lead times and sufficient flexibility to facilitate the production of the required variety and mix of products for the market.

**Tactical Issues**: Tactical issues in PMSs relate to the generation of detailed plans to meet the demands imposed by the master production schedule. It involves the breakdown of the products in the master production schedule into their assemblies, sub-assemblies and components, and the creation of a time-phased plan of requirements, which is realistic in terms of capacity and material available.

**Operational Issues**: Operational PMS issues essentially involve taking the output from the tactical planning phase, e.g. the planned orders from an MRP system, and managing the manufacturing system in quasi real-time to meet these requirements.

As already stated PMS tend to be viewed in terms of requirements planning (MRP and MRP II), Just in Time (JIT) and Optimised Production Technology (OPT) systems.

- MRP II attempts to address all the different levels in a PMS from business planning right down to real-time shop floor control. However, the architecture
An Hybrid PMS Architecture

proposed here shows two more levels below the requirements planning stage, namely, the factory co-ordination and production activity control systems.

- The JIT approach to manufacturing is also very broad ranging. Some of the key elements in this approach are [2]:
  - Close matching of product design to market demand. Skinner [19] suggests that the idea of a focused factory may be of a help towards a solution in this area.
  - The ideas of product families and flow based manufacturing.
  - Relationships with suppliers to achieve just in time deliveries.

JIT is a very important philosophy from which many lessons can be learned for future PMS. It is envisaged that most future systems will involve some degree of the JIT philosophy in order to achieve manufacturing excellence.

- The OPT philosophy contends that the main goal of a manufacturing enterprise is “to make money”. OPT addresses many aspects of manufacturing, both in production and business. Browne et al. [2] state that OPT will, for the near future, be used primarily as a scheduling tool and later in conjunction with existing MRP II modules as a complete PMS. Therefore, OPT can be seen as only being concerned with the factory co-ordination and production activity control levels in the PMS hierarchy proposed in this paper.

The remainder of this paper involves specifying the different layers of the PMS hierarchy proposed above under the different headings of business planning, master production scheduling requirements planning, factory co-ordination and production activity control. The paper concludes by giving a description of a simulation tool for use in the design of production activity control systems.

2 Business Planning

Production is of vital importance to the competitiveness of manufacturing companies. In recent years world-wide competition has increased and this calls for increased efficiency and strategic thinking within all functions including production. Operations managers need to become better at long range planning. Conventional emphasis has been on productivity, meeting schedules and maintaining efficiency on a month by month basis.

This section discusses the highest level shown in the PMS architecture i.e. business planning. Business planning provides plans necessary to drive the sales, manufacturing and financial groups within an organisation. These plans define markets to be addressed, products to be manufactured, required volumes and resources, and the financial impact of meeting the overall objectives set by strategic planning within the organisation [17]. Only the manufacturing aspects of business planning within the
An Hybrid PMS Architecture

architecture will be developed here. The primary aim here is to discuss manufacturing strategy.

A manufacturing strategy is formed by the decisions taken in, and in connection with, manufacturing which have a strategic influence on the company’s competitive approach. The purpose of a manufacturing strategy is to direct a company’s manufacturing resources in such a way as to support competitive ability.

A key deficiency in the management of batch production of discrete parts is a methodology to allow the articulation and specification of an appropriate manufacturing strategy to support the overall business goals of the enterprise. In recent years, the subject of manufacturing strategy has attracted much attention, yet it is still a relatively unresearched area. All manufacturing strategies to be developed must take into account issues such as:

- Reduced product life cycles.
- Increasingly complex and varied product options.
- Reduced design and delivery lead times.
- Increasing customer expectations in terms of higher quality products and greatly improved delivery performance.
- Increased flexibility to respond to demand changes within shorter delivery lead times.
- A balanced supply/demand pattern to provide a basis for manufacturing planning.

The output from the business planning system is a high quality production plan developed within the constraints of an overall manufacturing strategy. This production plan then becomes the major input to master production scheduling process.

3 Master Production Scheduling

The Master Production Schedule (MPS) is a statement of the anticipated manufacturing schedule for selected items by quantity per planning period [8]. It is a listing of the end items that are to be produced, how many of each item are to be produced, and when they are to be ready for shipment. End items may be products, major assemblies, groups of components, or even individual parts used at the highest level in the product structure. The MPS provides the basis for making customer delivery promises, utilising the capacity of the plant effectively, attaining the strategic objectives of the business as reflected in the production plan, and resolving trade-offs between marketing and manufacturing. Unlike a forecast of demand, the master schedule represents a management commitment authorizing the procurement of raw material and the production of component items.
The master schedule is a disaggregation of the production plan developed at the higher level, and directly drives the requirements planning function at the lower level. The master schedule development and validation function involves demand management, rough-cut resource planning and final assembly scheduling.

The production plan reflects the desired aggregate output from manufacturing business planning necessary to support the company's overall financial objectives. This aggregate plan constrains the master schedule, since the sum of the detailed MPS quantities must always equal the whole dictated by the production plan.

Demand management takes the aggregate sales plan from sales planning and produces a forecast of products which is fed into the MPS. The MPS then interacts on an ongoing basis with this module through its order entry and order promising activities. Rough-cut resource planning involves an analysis of the MPS to determine the existence of potential bottlenecks in capacity on the shop floor or material unavailability, that is, this linkage provides a rough evaluation of potential capacity and material problems for a particular MPS. The Final Assembly Schedule (FAS) is the actual build schedule, whereas the MPS is the anticipated build schedule. It is a statement of the exact set of end products to be built over some time period.

The challenge for a master scheduling system is to produce a master schedule which is realistic with respect to material and capacity and meets customer requirements and financial objectives. There are three main stages in a master production scheduling system: master schedule development, master schedule verification and validation, and master schedule maintenance and change management.

Three key decisions must be made in the development of a Master Schedule, i.e.

1. **What to schedule** – This decision addresses the question of the unit to be used in the MPS, i.e. is the MPS to be based on end items, specific customer orders, or some group of end items and product options. The choice of unit varies greatly depending on whether you are in a make to stock, make to order or assemble to order environment.

2. **When to schedule** – This addresses the question of the planning horizon. The minimum planning horizon acceptable is that of the cumulative lead time for the product. This lead time may include engineering design time, in the case of a make to order company, as well as material procurement time, production lead time and assembly lead time. The planning horizon should extend beyond the cumulative lead time in order to provide visibility into the future.

3. **How much to schedule** – This decides how demand for products is derived from various sources. This may vary depending on the environment. In the manufacture of products to stock, future requirements are usually derived from past demand. In the manufacture to order, the backlog of customer orders may represent total production requirements. In custom assembly, a mixture of forecasting and customer orders generates requirements.

Master schedule verification and validation involves two main activities:
An Hybrid PMS Architecture

- A check and balance to ensure that the master schedule aggregates up to the production plan, (Verification) and
- Rough-cut resource planning to ensure that the available resources, primarily material and capacity, are adequate to meet the requirements represented by a given master schedule (Validation).

Figure 2 shows an overall diagram depicting the interaction between the first and second stages of the master scheduling system. The output of stage 1 is a prospective master schedule. In the validation stage i.e. through rough-cut resource planning, a materials and capacity check is done. The outcome of this stage is either (i) a return to stage 1 to modify the prospective MPS, due to unavailability of resources, or (ii) an authorized MPS which can be released for execution to the requirements planning system. It is at this point that the master schedule maintenance and change management stage of the master scheduling system comes into play.

Figure 2: Master Production Schedule Development

Master schedule maintenance and change management involves the ongoing maintenance of the initial authorized master schedule, once it has commenced execution. Orlicky [18] describes the master schedule under a net change MRP system as resembling a Chinese scroll unwinding with passage of time. The schedule extends
An Hybrid PMS Architecture

indefinitely into the future, and passage of time brings segments of the future within the planning horizon. Maintenance of the MPS involves updating and revising it with the passage of time. The frequency of maintenance is usually geared to the forecasting cycle, which is almost always monthly. Between these "official" issues of an updated version of the schedule, however, there may arise a need for revision at any time, caused by a particular mix of new customer orders and by various unplanned developments in procurement and in manufacturing. The MPS is one of the main inputs for the requirements planning stage in the PMS architecture. Requirements planning will now be described.

4 Requirements Planning

Requirements planning resides in the tactical level of the PMS architecture. It may involve Materials Requirements Planning (MRP) and Manufacturing Resource Planning (MRP II), the latter essentially being an extension of the former.

MRP is a computational technique that converts the MPS for end products into a detailed schedule for the raw materials and components used in the products [9]. It is a priority planning technique that evolved from an approach to inventory management in which the following two principles are combined:

• Calculation of Dependent Demand for component items i.e. the demand is derived from demand for end products or other subassemblies.

• The Time Phasing of requirements i.e. the recognition of the idea of lead times in the placing of orders.

The main objectives of MRP are to determine what and how much to order.

Prior to the widespread use of MRP, the planning of manufacturing inventory and production was generally handled through inventory control approaches. The implicit assumptions of these approaches is that the replenishment of inventory items can be planned independently of each other. Orlicky [18] offers several important insights into MRP which have revolutionised manufacturing inventory management practice. These are summarised by Browne et al. [2] as follows:

• Manufacturing inventory, unlike finished goods or service parts inventory, cannot usefully be treated as independent items. The demand for component items is dependent on the demand for the assemblies for which they are part.

• Once a time-phased schedule of requirements for top level assemblies is put in place (MPS), it follows that the dependent time-phased requirements for all components can be calculated.

• The assumptions underlying inventory control models usually involve a uniform or at least well defined demand pattern. However, the dependency of components
An Hybrid PMS Architecture

demand on the demand for their parent items gives rise to a phenomenon of discontinuous demand at the component level.

- A computer provides the data processing capability to perform the necessary calculations efficiently.

MRP can thus be seen as discrete control on the flow of materials rather than control of the level of stock. MRP orders the components that are required to maintain manufacturing flow.

MRP II is an extension of MRP features to support many other manufacturing functions beyond material planning, inventory control and BOM control. MRP II evolved from MRP by a gradual series of extensions to MRP system functionality. These extensions included master planning, Rough Cut Capacity Planning (RCCP), Capacity Requirements Planning (CRP) and production activity control. Wight [21] states that the characteristics of MRP II are as follows:

- The operating and financial system are one and the same.
- It contains a “what if” capability.
- It involves every facet of the business from planning to execution.

One argument in this paper, however, is that MRP II is not good enough to cover all the PMS activities. There have been many difficulties and failings in MRP II systems in operation today. The system is good for creating lead times for planning purposes, but lower levels than the requirements planning need finite scheduling mechanisms to operate efficiently.

We now turn to factory co-ordination which takes the output from the requirements planning section and provides the input for the lower level production activity control systems.

5 Factory Co–ordination

The function of the factory co-ordination system is to manage the MPS throughout a factory-wide control architecture. The problem is to ensure that the MPS developed at the requirements planning stage is realised across the various work cells at the operational level of PMS.

Factory co-ordination is a set of procedures concerned with planning the flow of products at a plant level and which has close links with the design task. The design task concerns the design of the production environment in terms of product families and product based layout. The complexity of the planning task can be reduced if the production environment is designed efficiently [1].
An Hybrid PMS Architecture

The factory co-ordination system within the PMS hierarchy acts as a bridge between the requirements planning stage carried out by an MRP system and shop floor control carried out by the production activity control system [3].

The two main flaws in an MRP system for managing product flow are as follows [2]:

- MRP does not recognise that lead times are variable. Therefore the estimates used within a MRP system are inaccurate.
- The design of the production environment is ignored by the MRP system.

Production Activity Control (PAC) systems control production within each cell on the shop floor, but have no capability at a plant level.

Factory co-ordination bridges this gap by using principles from JIT and OPT. The main theme of factory co-ordination involves viewing the entire factory as one system (i.e. systems approach) and places great emphasis on planning in order to reduce the complexity of the control task at lower levels. The link with the design module provides the capability of structuring the production environment through group technology principles.

The major benefits of factory co-ordination derive from the regular and stable product flow which will result from the use of such a system. The introduction of new products will be accomplished more efficiently and with less disruption to the process. The planning and control tasks concerning raw materials and work in progress will be relatively less complex. Due to the stability of the production process, further production requirements for satisfying market demand can be determined with greater certainty. As demand fluctuates in batch manufacturing, the environment will have the flexibility to handle the demand fluctuations because of the inherent stability and regularity.

The architecture for factory co-ordination systems is illustrated in figure 3. As can be seen from the diagram the architecture consists of four main items. These are:

- Plant Level Scheduler.
- Plant Level Dispatcher.
- Plant Level Monitor.
- Production Environment Design.

Each of these modules within the factory co-ordination system will now be outlined.

**Plant Level Scheduler** – The main purpose of the plant level scheduler is to produce a plant level plan which can be used to co-ordinate the flow of batches between workcells. Any scheduling within the cell can be accomplished by the PAC system using the guidelines stated in the plant level plan. By using the guidelines stated in the plant
level plan, any schedules which are developed for individual resources within a cell will facilitate the overall flow of batches between cells. The procedure for determining a plant level schedule involves the following steps.

- **Production allocation** – This task involves deciding on a suitable policy for determining how to spread the monthly production requirements for each product throughout the month and to decide on a suitable batch size to be assigned to each product. If there are any late batches which did not enter the process according to schedule, this task will decide how to re-allocate them.

- **Bottleneck Management** – The purpose of this module is to provide some guidelines to a supervisor on minimising the effects of potential bottlenecks on product flow. These bottlenecks may refer to workcells or individual resources. Due to the effect of potential bottlenecks on product flow, they cannot be totally ignored by the plant level schedule, hence the necessity of this management module. These recommendations are separate from the plant level schedule, because the schedule is concerned with product flow between cells, and cannot be changed to facilitate one potential bottleneck within a cell.

This module will act in a decision support mode, generating a range of suitable
An Hybrid PMS Architecture

suggestions, which are based on some basic principles from OPT. The recommendations relating to managing the potential bottlenecks are based on three general areas; setup reduction techniques, alternative routes and specific solutions.

- **Scheduler.** – The schedule generated by this module is intended as a plan, which initially can be used to co-ordinate the product flow between different cells in a plant. The plant level schedule is labelled a plan because if some major problems occur within the process, the schedule may not be feasible. In this instance, there may be a requirement for generating another plant level schedule.

The factory co-ordination system should be able to function in a range of different production environments (e.g. push or pull etc.). Therefore the plant level scheduler must contain a library of rules to create schedules for different types of production environments.

**Plant Level Dispatcher** – The implementation of the schedule is handled by the plant level dispatcher. The plant level dispatcher should communicate with each of the PAC schedulers and dispatchers. Any events which occur in a particular cell can be communicated to the other cells through the plant level dispatcher. The control capability of the plant level dispatcher is not important since it operates at a high level, most of the control capability is handled by the PAC dispatchers. Any real-time scheduling which must be done on a plant level in relation to late batches will be handled by the plant level dispatcher. The plant level dispatcher contains different control strategies (e.g. pull & push etc.).

**Plant Level Monitor** – The function of the plant level monitor is to provide relevant data on performance measurement and a real-time status of the shop floor to the plant level dispatcher and scheduler. This data is accessed from each PAC monitor. The main data requirements will be the amount of finished daily batches, throughput times, and work in progress levels.

### 5.1 Production Environment Design in Factory Co-ordination

Product groupings and the resultant categorisation of resources as either shared or dedicated has a major influence on organising workflow. The main problem to be solved here is the proliferation of new products, which would ensure that the workflow becomes unpredictable, thus making the task of organising the product flow more complex. This module should provide a concise set of procedures to help a user generate a list of suitable product groups and categorise the resources into workcells. This module can be used when a group of new products are coming on-line for production.

There are two types of procedures that can be used within the environment design module. These are variant and generative procedures. The variant approach seeks to integrate the new products within existing groupings according to a rulesbase. However through successive integration of new products into existing groupings the size of individual groupings may be excessive, and the benefits will diminish. Should such a situation arise, the generative approach can be used whereby existing groupings
are disbanded, and new grouping arrangements are formed from the range of new and old products.

In summary, the factory co-ordination system provides a tool which places the emphasis on developing an efficient plan, which will reduce the complexity of the control task. By having close links with the design task, which structures the production environment, the planning task is less difficult. There is a need for closer communication between the product design and PMS which manages the manufacturing system. Therefore, a design tool could be incorporated into a factory co-ordination system.

The plan developed by the factory co-ordination system provides guidelines on starting batches at each workcell, so the detailed scheduling in each cell is developed in accordance with an strategy for the entire factory. By using one factory level plan efficiencies are gained for the overall production system, rather than at one particular set of workcells. Each particular workcell will then be managed by its individual PAC system. In the next sections we describe this lowest level in the PMS architecture.

6 Production Activity Control

The overall factory wide plan provided by the factory co-ordination level is downloaded to individual workcells. These workcells then have their own control system to coordinate the flow of work. In this section we describe this control system and each of the functional elements involved. Also we describe a simulation tool used to test different PAC solutions for various manufacturing environments.

Production Activity Control (PAC) "describes the principles and techniques used by management to plan in the short term, control and evaluate the production activities of the manufacturing organisation" [10].

PAC resides at the lowest level of the architecture for PMS and due to being at the operational level of this hierarchy, PAC operates in a time horizon of between one month and real-time. It is desirable, for greater control, that PAC activities be as close to real-time as possible, and consistent with actual industry requirements.

The PAC hierarchy consists of a number of functions [11] which are illustrated in figure 4. These include:

- The scheduling function,
- The dispatching function,
- The monitoring function,
- The materials transportation function,
- The process control function.

These are the real-time functional control blocks of a manufacturing system operating in a CIM environment. Each of these will now be outlined in turn:
The Scheduling Function: Scheduling is performed as part of the production planning and control function. Schedules serve as a guide for production and for establishing manufacturing resource requirements in terms of manpower, facilities, tooling and machine capacity. From the wide range of tasks controlled through scheduling it is obvious that the quality of schedules produced is a major influence on the effectiveness of a manufacturing enterprise [16].

The PAC scheduling function is a short term scheduler whose goals are that only what is required is produced, when it is needed and in the correct quantity. The task of this function is therefore, to develop a detailed schedule for a time frame of between one day and one week which brings the manufacturing organisation closer to the achievement of production targets set by the long-term master production schedule.

Prior to releasing orders onto the shop floor the PAC scheduler develops a production plan or schedule. This identifies the orders to be worked on, their sequence and their timing. It may specify either dates of completion for a product or at a much more detailed level, the start and completion times for individual operations on products. A wide range of techniques may be used to develop a shop floor schedule including simple Gantt charts, integer programming and simple heuristics [3].

The Dispatching Function: Dispatching is the final determination of job sequencing for a workcenter and it is responsible for co-ordinating the individual work-
An Hybrid PMS Architecture

center schedules, the workcenter itself and material movement control. When an
operation is to be executed, the required material, if available, can be dispatched
to the desired location. This is aided by knowledge of the quantities and loca-
tions of all the work in process (WIP) items. Selection of which item to dispatch
may be based on highest priority, workcenter availability or material handling
availability.

The dispatching of orders requires up to date information on many aspects of the
production process. It represents real-time control within PAC. It determines
the job sequence for each work center and oversees transportation of material.
Working from daily or more frequent schedules, the dispatching function ensures
that these schedules are adhered to as closely as possible. If disturbances occur,
it is the function of the Dispatcher to resequence the jobs within the limits of
the schedule. This involves deciding when and what to do with each job. The
scheduling function is notified of any shortfalls and may incorporate these as a
backlog in the next schedule.

The Monitoring Function: The monitoring function can be seen as comprising of
three different areas [13], namely:

- Data Collection.
- Data Analysis.
- Decision Support.

The data collection system collects all the relevant information from the shop
floor and this is then analysed to produce both real-time and historical reports.
Examples of real-time reporting include current utilisation levels, inventory levels
etc. Historical reporting involves producing graphs and reports on a variety of
items of interest to manufacturing personnel. As well as acting as a reporting
mechanism, the Monitor also has a decision support capability. For example, if
the level of materials on the shop floor was below the desired level at a certain
point in time, then the Monitor would have to signal this to the higher control
functions in the PAC system.

The Materials Transport Function: The materials transport/handling function
controls the shop floor transportation devices. These can consist of carousels,
robots, automated guided vehicles (AGVs), manual transporters, etc.

It is the job of the materials transport function to co-ordinate these devices
and to make sure they are in the right place at the right time according to the
schedule. It also should ensure the no collisions occur between the transportation
devices. This is an important function in the PAC system because the movers
tcontrol not only the movement of parts around the system but also look after
the transport of materials in the manufacturing plant, under the direction of the
Dispatcher.

The Process Control Function: The process control function of PAC controls
specific types of production equipment such as CNC machines and robots through
An Hybrid PMS Architecture

standard protocols. The process control function isolates the physical level of production devices from the strategic control level by translating general instructions from the Dispatcher into specific device instructions. The process controller also communicates performance data to the monitoring system.

For a more detailed discussion on all of the above work on PAC the reader should refer to the papers referenced.

7  PAC Simulation Tool

This section discusses a software tool called the Application Generator (AG) developed as part of the COSIMA project – ESPRIT 477 [15]. The primary objective of the AG is to develop a decision support system which generates a PAC system for a user. The AG is a tool to be used by a manufacturing engineer who has intimate knowledge and experience in the factory for which the new PAC system is being designed. The AG has a large suite of rules which encode knowledge about the building blocks (the software modules) from which it chooses the appropriate components for a PAC system. At the moment, there exists an early prototype version of this AG (See Duggan and Browne [5] and [6]). This uses a simulation generator to test different PAC strategies.

The simulator consists primarily of an application network, which couples a simulation model of a manufacturing system to the various building blocks of PAC. For example, there are numerous different scheduling strategies that could be used in a manufacturing system, but it is hoped that, through the use of the simulator, the user will be given a good idea as to which strategy is the best for his/her system.

The modules developed to date are summarised below:

- A Manufacturing Profile interface which allows the engineer to describe in detail the manufacturing system [12]. Information specified using this interface includes the following:
  1. Resource Data (Workstations, Buffers and Moving Devices).
  2. Stock Data (Complete listing of all the raw materials used in the model).
  3. Product/Process Data (All the parts with their process data).
  4. Planned Orders (Typically for a weeks production).

- The Manufacturing Profile database stores all of this information so that it may be used by the other building blocks of the PAC system. Once this database is populated the engineer may select from several different scheduling building blocks; choosing the one that is most suited for the factory floor [12]. This database has been developed using RdB [20], a relational database product.

- A Scheduler which consists of several scheduling strategies that the engineer can choose from to schedule the flow of work through the shop floor. The Scheduler
An Hybrid PMS Architecture
takes as its input the planned orders for the week and proceeds to develop a
detailed schedule for all of the products through their respective workstations.

- A rulesbase, developed using AI techniques, which analyses a manufacturing
  system and recommends what type of scheduling strategy to use.

- An Interactive Scheduler (IS) which is a sophisticated software planning tool
  that allows the manufacturing engineer/supervisor to manually alter the sched­
  ule developed by the Scheduler. The reasoning behind its development is that
  Scheduler may not provide the most suitable solution and that by using the IS
  the engineer may bring his/her own experience into play, hence improving the
  Schedule. [14]

- The Dispatcher building block containing different dispatching rules that can be
  selected to control the flow of work in the production system. The Dispatcher
  is the real–time controller of the PAC system and it takes the detailed schedule
  from the Scheduler and ensures the passage of the jobs through the production
  system.

- The Shop Floor Emulator simulates all of the events that occur on the shop floor,
  including machine breakdown, workstation usage and work in progress (WIP).
  This Emulator was initially developed using Artificial Intelligence (AI) based
  tools which simulated the events of the shop floor using Petri net structures (See
  Duggan and Browne 1988 [7]).

- The Monitor building block gathers all of the information in real–time from the
  shop floor emulator and then displays this for the user in terms of workstation
  availability, work in progress and material status. It also issues reorder com­
  mands if material stocks are too low. When the simulation is finished, the user
  is provided with the opportunity to analyse the results using the Monitoring
  Reporting Module [13].

- Finally, each the above applications interact with each other via the Application
  Network. The Application Network is a distributed software bus that allows all of
  the building blocks to communicate with each other in a controlled environment
  [4].

Figure 5 shows the structure of the simulator software.

This section has described a simulator which has shown that the modular PAC
structure allows development and testing of different modules of code for PAC systems.
It is an excellent example of a true PAC system in operation and many ideas from
this simulator are being developed and used in real industrial applications as ongoing
work in the COSIMA project.
8 Conclusion

In this paper we have attempted to give an overall view or framework for PMS. Each level was expanded and connections between and reasons for their existence described. It is intended as a summary paper from which further enhancement and ideas regarding the framework can be accommodated. The important point is that this framework encompasses much of what we have learned from other PMS “ideologies” such as JIT, MRP II and OPT, into one coherent framework for PMS. The paper finished by giving an overview of a simulation tool for use in the design and development of PAC systems.

References


An Hybrid PMS Architecture


A Simulation Tool to Evaluate Factory Level Schedules

J. Duggan, R. Bowden, J. Browne,
CIM Research Unit
University College Galway,
Rep. of Ireland,
353-91-24411
Abstract

Due to the complex and variable nature of modern batch manufacturing, there is a need for knowledge based simulation tools to aid engineers in the task of planning production. To be of use to the engineer, these tools need to be easy to use and provide the capability of using manufacturing data through a structured database. The simulation tool described in this paper is based on an AI approach to simulation and it is data driven from a relational database. It can be used to support the planning functions of an electronics plant to provide information on the impact of scheduled work loads as well as testing the effects of new products on a manufacturing environment.

1 Introduction

Modern factory layout is organised by a series of product related workcells, each of these workcells being defined based on group technology principles. The advantages of grouping products into cells are many, and they include shortening of lead time through reduced setup times, simplifying the process and increased worker satisfaction.

Clearly, each of these work cells are not independant, and as a result there is a need for a cell co-ordination system which designs the cell layout and organises the flow of work between cells. This system, termed factory co-ordination has as part of it a short term planning tool which takes as its input the planned orders of a requirements system and develops schedule guidelines for each of the workcells. The implementation of these guidelines within each workcell is the role of the production activity control system.

This paper describes a simulation tool developed to support the validation of these plant schedules so that their potential impact may be assessed before the plan goes live on the shop floor. The simulation tool simulates the activities of a factory co-ordination system controlling the flow of work between each of the workcells. It also simulates the PAC system within each workcell based on the guidelines specified by the plant scheduler and in doing so gives important statistics regarding total throughput times, resource utilisations and work in progress levels.

The structure of the paper is as follows:

- The first section of the paper describes the environment within which the
simulation tool can be used to support manufacturing planning. This environment is based on production activity control systems interacting with a factory co-ordination system.

- The final section gives an account of the simulation tool with particular reference to:

  1. How the architecture of the simulation tool reflects the control architecture of PAC and factory co-ordination.
  2. The simulation tool is driven from a relational database which holds the manufacturing data describing the factory. The advantage of this is that the simulation tool need not be changed when new products and processes are modeled.
  3. The simulation is based on an emulator, a rulesbase and a data collection module. The emulator was developed using an AI approach to modeling based on Petri nets while the rulesbase was developed using a rules based AI language. The simulation tool has been tested on a modern electronics plant.

2 Environment of the Simulation Tool

From figure 1 it is clear that below the Requirements Planning level (which are typically implemented using MRP type systems) the emphasis switches from planning to control - i.e. control in the sense of ensuring that the detailed requirements for individual components or assemblies are met. The use of the simulation tool is aimed at the operational level shown in the diagram, that is, the area of factory co-ordination and production activity control.

PAC ensures that the production orders for each work cell are met and it also includes analysis after the order is completed to evaluate the effectiveness of the manufacturing activities. Shop floor control has always been critical to the success of a manufacturing operation, and it is viewed by many as they key to successful CIM implementation [4,7]. PAC consists of three main building blocks [2] which control the flow of work through the shop floor.

- The scheduler performs its function of scheduling production activities on the shop floor by specifying the timing of operations in order to comply with due dates, priorities, availability of resources etc. Driven from the guidelines of a factory co-ordination system, the scheduler provides a time sequenced operation plan to the dispatcher building block.
- The *dispatcher* executes in real time the sequenced operation plan provided by the scheduler. It does this by assigning sequenced production orders to operators, transport systems and production equipment on the shop floor.

- The *monitor* performs the real time feedback function. It collects data on equipment utilisation, materials, stock status and quality management and reports them back to the appropriate building blocks within a PAC system or to the PAC user interface. The monitor thus supports the decision making process of the PAC system.

Factory co-ordination may be seen as a *higher level recursion of PAC* in that it aims to control different PAC systems within a manufacturing environment. It contains a scheduling function for the plant which drives PAC systems at a workcell level. Besides this scheduling role, factory co-ordination also involves the design of workcells taking group technology principles into account.

The simulation tool is then used to support the planning function of factory co-ordination through simulating a suggested schedule and its potential impact on the shop floor. The advantage of using the tool is that it can establish a level of confidence for a schedule and also the impact of new products on the existing process can be assessed.
3 Description of the Simulation Tool

The simulation tool is a data driven simulator which is based on a modular approach to the modeling of production systems. It consists of a dispatcher, an emulator and a monitor and uses Petri net structures initially developed using AI tools ([5,6]). The main advantage of the simulator is that the user does not need to have any simulation expertise, as the simulation runs off a relational database which describes the manufacturing system.

![Figure 2: Structure of the Simulator]

The main function of the simulation tool is to verify the plant level schedule by simulating its operation on each of the workcells. Therefore the simulation tool takes as its input the factory schedule generated by the factory co-ordination system and simulates the production activity control task within each of the work cells. This section of the paper describes each of the modules shown in figure 2, starting with the database and the plant scheduling system.
3.1 Manufacturing Database

The database is the data foundation upon which the scheduling and simulation tools can perform, and it is structured so that it can represent the most important features of a manufacturing plant from an operational perspective. The database has been designed to perform as a *generic* database, with the overall goal of allowing different manufacturing environments (e.g., car assembly, Flexible Manufacturing Systems (FMS) and electronics assembly) to be defined. It is composed of four sections:

1. Resource data: This includes the number of Workstations, Buffers and Moving Devices which make up the physical resources of the manufacturing plant. Each of the individual attributes of the resources (i.e., station capacity, breakdown rates etc) are contained in resource data. The data included in this section is used by the *emulator* to build a physical model of the manufacturing systems.

2. Stock data: This contains detailed list of all the raw materials used by the manufacturing facility. It includes information on the *location of the materials*, the *reorder point* and also *vendor lead times*.

3. Product/Process data (All the parts with their process data). The product data holds general information on the product while the process data specifies the operational sequence for each product through the manufacturing system. The data describing each operation includes the *class of operation*, the *setup time*, the *processing time* and the location of possible *alternative operations*.

4. Planned orders: This are a list of the parts to be produced over a definite time frame. Typical planned order data includes the *job number*, *part name*, *required quantity*, and *due date*.

This data then provides all of the necessary information for the software modules to work. For example the plant wide scheduler takes the *planned orders* for a certain time period and calculates start and finish times based on the *product and process data*. The emulator uses the breakdown distributions described in *resource data* to generate machine breakdowns during the simulation run. The monitor can use the database to store the information from different simulation runs for the purposes of comparison.

The database is implementated using Rdb [12], and a full interface with editing, viewing and data input procedures has been developed using a the fourth generation product RALLY.
3.2 Plant Scheduling System

The plant scheduler assumes responsibility for planning product flow over a short time horizon (daily or weekly) because an MRP system cannot produce an accurate plan since it uses estimated lead times. This system uses actual processing and setup times based on data gathered from the shop floor [1], and this data is stored in the manufacturing database described in the previous section.

The task of organising product flow in the plant is approached in two stages; firstly, designing a structured production environment (i.e. design task) and, secondly, producing a plant level schedule (i.e. operations task). On its own, scheduling only achieves so much in a certain production environment, but together with group technology principles, scheduling achieves greater efficiencies. This paper concentrates on the operations task of the factory co-ordination system, and this function is now described in more detail.

The plant level schedule provides a plan for co-ordinating flow of transfer batches between workcells by providing start and finish times for each batch at each workcell. The flow of products is such that no workcell is totally independent of any other cell, so there is a need for some co-ordination. The schedule provides an initial plan for each day/week, the method of implementation (i.e. kanban or push system) of this plan is not within the scope of the plant scheduling systems. The schedule may be developed using a number of strategies, namely,

- **backward scheduling technique**: this technique works from the required due date at the final work cell of each job and calculates the start times of the job at each of the preceeding work cells,

- **priority scheduler**: priorities are assigned to each of the jobs and the work plan is generated taking these priorities into consideration,

- **shortest processing time**: the total processing time of each job is calculated, and the work plan list is made based on the shortest processing time of the jobs.

The output from each of the above scheduling strategies is the same, namely, a list of start and finish times for each of the job orders at each of the work cells. These schedule guidelines are then taken by the simulation tool, and simulated using the three different modules of the simulator. This simulation process is described in the following three sections.
3.3 Shop Floor Emulator

The Shop floor emulator simulates all of the events that occur on the shop floor, including machine breakdown, workstation usage and the creation of work in progress (WIP). This emulator was initially developed using Artificial Intelligence (AI) based tools (OPS5) which simulated the events of the shop floor using Petri nets structures ([5],[9] and [10]). Petri nets are a tool used in the design and analysis of systems. Petri nets enables the modeling of a systems and analysis of the Petri net net can then reveal important information about the structure and dynamic behaviour of the modeled system.

OPS5 is a rule based language and is a member of a class of programming languages known as production systems. A production system is a program composed entirely of conditional statements called productions. These productions operate on expressions stored in a global data base called working memory. The productions are similar to the IF-THEN statements of conventional programming languages. The firing strategy of OPS5 and the execution procedure of Petri nets are very similar, and this was the major reason for choosing OPS5 as a simulation language. It was found that OPS5 is a good tool for developing initial prototypes of the shop floor emulator, but these prototypes were then re-developed using PASCAL because of the processing requirements of large simulation models.

The emulator, which runs of the plant database, consists of several modules (see figure 3), each of which will now be described.

- Petri Net Generator
  This takes the information from the plant database on each workstation and buffer in the system and generates a Petri net model which then simulates the activities of the entity. These models generated are called Petri net structures.

- Petri Net Structures
  As mentioned above, a different data structure is generated for each resource in the database. These data structures can then simulate the necessary event which has to happen on the shop floor. The goal of the emulator is to have the capability to simulate the different types of manufacturing operations as defined by [11]:

  - Combinative, where separate pieces are selected and assembled into a functional end product.
  - Disjunctive, where material is disassembled into useful components;
- Sequential, where one piece of material is progressively modified by operations;
- Inspection, where material is inspected and a certain percentage rejected and re-routed.

The Petri net shown in figure 4 is how a simple operation (sequential) is modeled, and there are similar Petri net models constructed for the other classes of operations.

The place in the net, labelled dispatcher command received receives a token from the dispatcher signalling the start of a job. This is a simplified diagram of the emulator as we use timed Petri nets ([13], [14]) with attribute carrying tokens which allow a detailed simulation to take place.

The following events are simulated by the emulator:
- A job arrives in the system
- A buffer is loaded with a job
- A job is unloaded from a buffer
Figure 4: Petri Net Model of a Manufacturing Operation

- Setup has started on a workstation
- The workstation has started processing the job
- The job has finished on a workstation
- The workstation has broken down
- A broken down workstation has been repaired
- The capacity of a workstation has been reached
- A job has started a move between two locations
- A job has finished a move between two locations

• Event Calendar and Simulated Clock
  The event calendar and simulated clock is the means by which the appropriate events are fired at the correct clock time. The Dispatching Rulesbase or emulator can enter an event into this calendar and the calendar is then searched for an event to happen at the current clock time. When an event is fired, it is removed from the calendar.

To illustrate the idea, consider figure 5 which shows how the event calendar and simulated clock work together. The first diagram shows the calendar at time 20 having two events, namely:

- Start Job 1 on workstation DIP at time 20
Start Job 2 on workstation VCD at time 70

The event calendar selects the first entry, passes the information to the Petri net structures and the event is simulated. The Petri net structure adds a new event to the calendar

Finish Job 1 on workstation DIP at time 60

and the fired event is removed leaving the event calendar as shown in the second diagram of figure 5. Because there is no event to fire at time 20, the clock is updated to the time of the next scheduled event, which is time 60. The selection step is repeated again with the result that the following event is fired at time 60:

Table 1: Before starting Job 1, Clock Time = 20

<table>
<thead>
<tr>
<th>Event</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Job 1 DIP</td>
<td>20</td>
</tr>
<tr>
<td>Start Job 2 VCD</td>
<td>70</td>
</tr>
<tr>
<td>Finish Simulation</td>
<td>1000</td>
</tr>
</tbody>
</table>

Action: Job 1 started, removed from calendar

Table 2: Started Job 1, Clock Time = 20

<table>
<thead>
<tr>
<th>Event</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finish Job 1 DIP</td>
<td>60</td>
</tr>
<tr>
<td>Start Job 2 VCD</td>
<td>70</td>
</tr>
<tr>
<td>Finish Simulation</td>
<td>1000</td>
</tr>
</tbody>
</table>

Action: Clock updated to 60, calendar unchanged

Table 3: Before finishing job 1, Clock Time = 60

<table>
<thead>
<tr>
<th>Event</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finish Job 1 DIP</td>
<td>60</td>
</tr>
<tr>
<td>Start Job 2 VCD</td>
<td>70</td>
</tr>
<tr>
<td>Finish Simulation</td>
<td>1000</td>
</tr>
</tbody>
</table>

Action: Job 1 finished, removed from calendar

Figure 5: Example of Event Calendar

Finish Job 1 on workstation DIP at time 60

All of these events are recorded and manipulated by the monitor module so that appropriate statistics can be calculated.
3.4 Dispatcher

The dispatcher building block containing different dispatching rules that can be selected to control the flow of work within each of the work cells of the plant. Presently, the dispatcher can cater for the following rules:

- *Shortest Processing Time, Earliest Due Date, Least Remaining Time, Priority Rule*

The dispatcher has two roles:

1. It ensures that the jobs start at each work cell as specified by the guidelines of the plant level scheduler.

2. It controls the flow of work within each of the work cells based on a strategy selected by the user. These strategies include the ones described above.

In controlling the flow of work through the manufacturing system, the dispatcher reacts to information from the shop floor ensuring that the schedule is implemented within the specified guidelines. The dispatcher makes decisions similar to those a supervisor makes on the shop floor, and it is composed of three parts as shown in figure 6.

- **Static Data Module**
  This holds all of the relevant static manufacturing data to be used during the simulation run. It essentially maps the database onto *dynamic record structures* written in PASCAL and these records hold the following information:
  
  - Information on all product names
  - Process data for all of the products
  - All of the resources in the model, workstations buffers etc.
  - The planned loading of jobs to pass through the system (i.e. the plant schedule guidelines).

- **Dynamic Data Module**
  This contains information that is continually updated by the events of the shop floor emulator, which essentially means that the current status of the shop floor is held in these records.
• Dispatching Rulesbase
This is the intelligence of the simulation system, which was initially developed using AI techniques (OPS5) and then implemented using data manipulation routines written in PASCAL. It basically has a different rule for every possible event that can happen in the system. This rule contains a series of conditions, and depending on the state of the conditions a certain course of action is taken. To explain this, consider the following message coming from the emulator to the rulesbase:

- Batch Completed on DIP2 Job 111 at time 100 Part DZQ11

The following sequence of steps are carried out:

- Get the next operation of the job 111
- Find the location of the next operation
- Ensure that there is capacity at the next locations buffer
- If there is capacity, send a command to the emulator to move the job to the next buffer
- Check the input buffer for the next job and check for the selection criteria (schedule guidelines, FIFO, SPT etc)
- Select the job according to the criteria
- Issue a command to the emulator to start setup on the next job
- wait for the next message

The next message is the next event held in the event calendar (as described in the previous section) and the procedure of the emulator to dispatcher information flow continues until the end of the simulation is reached.

3.5 Monitor

The role of the monitor collection module (figure 7) is to make sense of all of the events which happen during the simulation run. It monitors the simulation and records the important data with respect to

![Figure 7: Monitor Module](image-url)
• Queue sizes: the average size of a queue to a workstation is an indicator of the effectiveness of the scheduling strategy.

• Workstation data: this calculates the values for setup, processing, idle and down time of each of the workstations.

• Schedule deviations: this compares what should have happened (the schedule guidelines) with what actually happened in the simulation (the simulation output) and provide an indicator as to the effectiveness of the schedule.

The simulation tool is currently being used to support the task of plant level scheduling at Digital Equipment Corporation, Clonmel, Ireland. The database holds information on thirty different products and over one hundred work stations and the simulation takes the planned orders from the live MRP system to simulate the production on a specified time horizon (daily, weekly or monthly).

4 Conclusion

This paper has described the environment of manufacturing control, and how production activity control and factory coordination can bridge the gap between the activities on the shop floor and requirements planning systems. The need for tools to support the major task factory coordination, that is, the plant level scheduling function, has been highlighted.

The simulation tool developed to support plant wide scheduling allows schedules to be validated before being released live on the shop floor. This tool is data driven, and works off a plant database, which in turn has access to the MRP requirements of the factories MRP system.

The tool is modular, and two of these modules were developed using AI. The benefits of the tool lie in its flexibility and modularity, as new products can be accommodated through the database without effecting the simulation tool, and new scheduling strategies added to the dispatcher and plant scheduler without having to change the database, monitor or shop floor emulator.

The next major step in the development of the simulation tool is to design and implement an AI based system to analyse the effectiveness of different scheduling strategies for different manufacturing environments.
5 References


Digital Equipment Corporation, Clonmel, Ireland.

ESPRIT Project 477 Pilot Implementation

1 Summary

This document gives is a summary of the DEC Clonmel pilot implementation of ESPRIT Project 477, COSIMA, Control Systems for Integrated Manufacturing. Figure 1 shows the different stages of the project and how the project brings together manufacturing knowledge and software technology.

The goal of the project was to develop an architecture for Production Activity Control (PAC) and validate that architecture through a software model of PAC. This software model was developed and tested and its main feature is to allow different control strategies to be tested in a controlled environment before their implementation on the shop floor. The final stage of the project is a pilot implementation of PAC in a live environment, and today's demonstration shows the Clonmel pilot implementation. The demonstration is in two parts:

- Off-line Implementation: this shows the use of PAC as an off line planning tool and the following software is demonstrated:
  - The Manufacturing Database which holds the relevant manufacturing data for the PAC modules to function, and the Database Interface which is a user friendly application which allows the data to be updated and changed.
  - The Plant Level Scheduler and Plant Simulator, both of which are used to develop and test schedules before their implementation on the shop floor.
• On-line Implementation: the modules of PAC which are used in real time by the production planning personnel will then be shown. The COSIMA PAC modules have been designed to integrate with existing systems on the shop floor, and are:

- On-Line Scheduler, this has the same features as the Plant Level Scheduler and it is linked to the Dispatcher through the Application Network to facilitate real time schedule requests from the shop floor.

- Dispatcher: this allows the supervisor to request a schedule from the Plant Level Scheduler and then send this schedule to the relevant workcells.

- Monitor: the Monitor compares the schedule plan to what actually is happening on the shop floor, and through a user interface allows the supervisor to track specific parts at each workcell.

- Workcell Reporter: this facilitates the viewing of workcell schedules at the workcell and also allows the operators to enter the current status of a job.
- **Application Network**: this allows each of the building blocks to communicate with each other in a distributed environment.

The remaining sections of the document will describe in more detail the features of the pilot implementation, and will also show screen output from each of the software modules.

## 2 Off-Line Implementation

The aim of the off-line implementation is to allow different scheduling strategies to be developed and tested in a controlled environment before releasing these schedules on the shop floor. The schedules are based on manufacturing data held in the *manufacturing database* (see figure 2.) and this database is populated through a user interface which will be described in the first software demonstration.

![Diagram of the Manufacturing Database and User Interface]

**Figure 2**: The Manufacturing Database and User Interface

The Plant Level Scheduler has a number of options available which can be used to develop schedules. These include *Production Smoothing, Bottleneck Search,*
**Bottleneck Analysis and Scheduling Strategies (SPT, EDD, Backward Schedule).** The schedule developed can then be simulated by the Plant Simulator and the results analysed. The Plant Simulator is based on the PAC architecture and is composed of a Dispatcher, Shop Floor Emulator and Monitor which ensure that the schedule is implemented through each of the production steps. The interaction of the database, plant scheduler and plant simulator is shown in figure 3.

---

3 **On-Line Implementation**

The on-line implementation facilitates the execution of the schedule on the shop floor. Firstly, schedule is generated by the Scheduler and passed to the relevant workcell supervisors through the Dispatcher; progress of the schedule is then monitored by the Monitor building block, based on information from the Workcell Reporter and DECstr Interface (see figure 3). The passing of information is achieved through the Application Network. Each of the software modules for the pilot implementation will now be outlined.

1. **Scheduler**

   On receiving a *schedule request* message from the Dispatcher (specifying the scheduling strategy), the Scheduler will use the data held in the manufacturing
database to generate an appropriate schedule. When this schedule has been generated, it is downloaded to the Dispatcher. The Dispatcher resides on the live production systems cluster in Clonmel and the Scheduler is located on a separate development cluster; the Application Network provides the facility for the schedule to be passed between the two clusters. Screen information on the scheduler is shown in figure 5.

2. Dispatcher
The Dispatcher allows the supervisor to request a new schedule from the Scheduler and to pass this schedule to the workcell supervisors and the Workcell Reporter. The Dispatcher also keeps account of the number of building blocks on the PAC system at any one time (see figure 5 for screen display).

3. Monitor
The role of the Monitor is to compare what actually happened with what should have happened. Therefore it receives a copy of the schedule from the Dispatcher and the latest shop floor events from the Workcell Reporter and DECstr interface. This data is then manipulated and presented to the supervisor in an intelligent way so that the progress of each job can be viewed at any point in time. The menu options for viewing work in progress through the Monitor is shown in figure 7.
4. Workcell Reporter

The function of the Workcell Reporter is to display the schedule for a specific workcell and to facilitate the notification of important system events which have a direct influence on the implementation of the schedule. For this reason, it can work in any of two distinct modes (shown in figure 5). The first mode is a supervisory mode which allows the workcell schedule and its current status to be viewed, while the second mode is an operational one which enables important system events to be recorded (i.e. the start and finish times of jobs at each workcell).

5. DECstr Interface

This interface was written to make use of the existing data collection system.
in Clonmel. It takes messages from the DECstr system and relays them to the Monitor via the Application Network.

6. Application Network

The Application Network provides the message passing facility for the live PAC system. The main benefit of the AN is that it ensures *flexibility* and *modularity* in the PAC system. The flexibility means that different building blocks can reside on different nodes in a *local area network* (see figure 9). This feature of the AN is critical to the implementation of PAC, as the Clonmel PAC system uses three different nodes to achieve its goals. The benefit of the modularity is that more building blocks (i.e. *Workcell Reporters*) can be added on to the PAC system without difficulty. This is an important feature of any future work on PAC in DEC Clonmel.
4 Conclusions

Today's presentation marks the final development stage of the COSIMA project which involved taking the concepts of Production Activity Control and implementing these on the shop floor. The migration path from PAC Simulation to PAC Pilot Implementation has been completed; the next step is to assess the success of the pilot implementation. The theme of this assessment should focus on the potential for implementing PAC in the Clonmel environment based on the lessons learned from the pilot implementation.
PAC Implementation in the Grugliasco FMS

Paul Higgins, Cathal Copas and Rossella Trentin

CIM Research Unit, UCG and SESAM

January 1989
CONTENTS

1 INTRODUCTION ............................................. 2
2 OVERVIEW OF GRUGLIASTO FMS ............................. 3
3 A PAC ARCHITECTURE FOR THE GRUGLIASTO FMS ............ 7
  3.1 Scheduler ............................................. 11
  3.2 Dispatcher ........................................... 12
  3.3 Producer Or Process Control Function .................... 13
  3.4 Monitor ............................................... 14
  3.5 Mover Or Materials Transport Function .................. 15
  3.6 Conclusion ............................................ 16
4 SPECIFICATION FOR PAC IMPLEMENTATION AT GRUGLIASTO .... 17
  4.1 The Manufacturing Profile .............................. 19
  4.2 Application Network ................................... 20
  4.3 Scheduler ............................................. 20
  4.4 Monitor ............................................... 24
  4.4.1 HERMES Standard Monitor ............................ 24
  4.4.2 Production Activity Monitor ........................ 26
  4.5 Dispatcher ............................................ 32
  4.6 Producer .............................................. 37
  4.7 Mover ................................................. 38
5 CONCLUSION ................................................ 39
1 INTRODUCTION

This report details the work underway in implementing Production Activity Control (PAC) for a Flexible Manufacturing System (FMS) in COMAU's Grugliasco plant. The report commences by describing briefly the FMS, including the machines used and the parts manufactured. It continues by giving a functional specification for the proposed PAC system for the plant. This leads into the detailed specifications of each of the building blocks for the control system and the report concludes by discussing the future implementation work needed.

Introducing the PAC architecture to the Grugliasco FMS means developing a modular control structure for the system. The modules in the control system for Grugliasco will consist of the standard PAC building blocks, including a Scheduler, Dispatcher, Monitor, Producer(s), Mover and the Application Network.
The Grugliasco FMS consists of four independent CNC machines made by the COMAU plant in Modena. Two of the machines are similar in that they are each capable of working only one part at a time. The parts arrive at an input bay, are moved to the working position and, when finished their operation, are moved to the output bay using a shuttle. Although these two machines are similar only one can use a multiple head spindle. The other two machines are equivalent in that they are capable of rotating their spindles through 110 degrees (That is 55 degrees either side of a 45 degrees axis). An independent tool magazine is available for each machine. With the use of fixtures the parts are clamped onto pallets, which in turn are mounted on heavy steel bases. The pallets are moved and loaded manually.

Each machine is controlled by a SIEMENS NC that is supported by the COMAU DDL hardware system. A VAX station 2000 is used in a special COMAU HW/SW system called HERMES 2. Part programs are developed on a microVAX II, which is connected using Ethernet to a CAD system and HERMES 2. The following lists the hardware being used [ref. 1].

- HERMES 2 - This consists of
1. VAX station 2000 and colour VDU 19".

2. 6 Mbytes of central memory.


4. Mass and backup storage of 159 and 95 Mbytes.

5. Line printer.

6. Air conditioned cabinet.

- Data Collection Unit (URD) - This is a PC based unit used to count the number of scrapped and good parts on the machines.

- DNC Data Link (DDL) - This is composed of a Motorola 68000 based central unit. It is connected to the machine via digital I/Os, paper tape readers and DNC line ports and to EERMES via a RS422 serial line and to the operator by a keyboard. The functions of the DDL are as follows:

  1. Increase the number of part programs that can be stored locally.

  2. Provide local tool management functions.

  3. Provide data for monitoring functions.
The system is capable of processing a wide variety of parts, ranging from robot components to engine blocks. For this reason the size of the orders can vary from one to several hundred. Another complexity in the system is the fact that the part routings are variable, depending on tool and fixture availability. It is reasonable then to say that this system reflects the architecture of a job shop.

Figure 1 shows a high level data flow diagram of the information flow through the activities for production at the Grugliasco plant. The Methods Department receives a 'weekly production request' document from the production planning department. This weekly production request is modified depending on the result of checks for the material availability and also checks for the presence of the part programs. This plan is then downloaded to the Scheduler, which uses information on fixtures and tools and also the process data to produce a feasible detailed production sequence for the shift. This sequence is then given to the supervisor. DDLs (DNC Data Links) are used for part program and tool management. All the part programs necessary for the weekly production plan must also be downloaded to the DDLs. The DDLs also need the tool parameters every time a tool is mounted on the respective tool magazines. The URD, which is the data collection unit, collects information on the parts scrapped and good on each machine.
FIGURE 1: GRUGLIASCO FMS: DATA FLOW DIAGRAM
The part programs which control each machining operation are developed in the Methods Department. It is also in this department that the routings for the parts are decided. All the product/process and resource information is stored in files on a microVAX in this Methods Department.

The supervisor directs the flow of work throughout the system. In effect, the Supervisor is the 'real-time Scheduler' and must ensure that the system runs smoothly, according to the schedule.

This section has given a general overview of the functionality of the FMS. The next sections will further enlarge upon this and describe each function in terms of the PAC architecture.

3 A PAC ARCHITECTURE FOR THE CRUGLiasCO FMS

PAC 'describes the principles and techniques used by management to plan in the short term, control and evaluate the production activities of the manufacturing organisation' [ref. 2].

PAC is at the lowest level of the hierarchy of the Production and Inventory Management (PIM) subsystem within a manufacturing system. Figure 2 illustrates an architecture for Production Management Systems which extends from the strategic to operational issues. [ref. 3]
Figure 2: An Architecture for Production Management Systems
Begin at the operational level of this hierarchy, PAC operates in a time horizon of between one month and real-time. It is desirable, for greater control, that PAC activities be as close to real-time as possible, consistent with actual industry requirements.

The PAC architecture consists of a number of different building blocks which are illustrated in figure 3. These include [ref. 4]:

- The Scheduler,
- The Dispatcher,
- The Monitor,
- The Producer,
- The Mover.

These are the real-time functional control blocks of any manufacturing system. Each of these blocks will now be discussed in relation to the Grugliasco FMS.
Figure 3: PAC Hierarchy and Data Inputs
3.1 Scheduler

Scheduling is performed as part of the production planning and control function. Schedules serve as a guide for production and for establishing manufacturing resource requirements in terms of manpower, fixtures, tooling and machine capacity. From the wide range of tasks controlled through scheduling it is obvious that the quality of schedules produced is a major influence on the effectiveness of a manufacturing enterprise [ref. 5].

For the Grugliasco system the function of the Scheduler is to produce a sequence of operations for each machine in the system based on the weekly production plan from the Methods Department. Scheduling is performed using the following information:

- The weekly production plan,
- The status of the machines, including:
  1. Machine up/down,
  2. Work-in-progress, i.e. machines still working on the previous weeks production,
- Manually imposed priorities.
Along with the above information, the following criteria have to be adhered to while generating a schedule:

- An analysis of the availability of each machine,
- An analysis of the availability of the fixtures,
- Optimisation of tool changes,
- Optimisation the number of fixture set-ups.

The generated schedule is tested to check its feasibility and is given to the operator who can modify the schedule to incorporate any changes he feels are necessary. The schedule is then transferred to the microVAX in the Methods Department where a final feasibility check is performed. It should be noted that at the moment it is assumed that the tool magazines have infinite capacity.

3.2 Dispatcher

The Dispatcher is in reality the real-time Scheduler of the system. The Dispatcher building block in the PAC architecture takes the generated schedule and implements that plan in as far as it is possible to do so. To implement the schedule the Dispatcher issues instructions which, if carried out correctly, bring the system closer to the goal of producing the scheduled amount of parts. Typical instructions could include the movement of a part from one machine to another.
instruction) or the ordered start of an operation on a part (a Producer instruction).

As well as carrying out the scheduled plan of operations the Dispatcher should have the functionality to be able to take account of unexpected/unplanned events in the system such as a machine breakdown. With the occurrence of an unscheduled event the Dispatcher should either order a reschedule or implement the schedule in so far as it is possible, taking into account this unexpected event.

The Dispatcher for the Grugliasco system uses the scheduled production plan as a guideline to how the system should be controlled. The Dispatcher also receives up to date information about the shop floor from the Producer block concerning machine status and part availability. Based on both the scheduled production plan and the data received from the Producer the Dispatcher issues instructions concerning events that should occur in the system.

3.3 Producer Or Process Control Function

The Process Control function in the PAC hierarchy is performed by the Producer building block. The job of the Producer is to control specific types of production equipment such as CNC machines and robots through standard protocols. The Producer block therefore isolates the physical level production devices from the strategic control level by
translating general instructions from the Dispatcher into specific device instructions. The Producer also communicates data to the other control block in the system.

In the FMS the Producer Block will poll the DDLs and the URD and send relevant information to the other building blocks. The Producer will also be in charge of part program and tools management for each machine.

3.4 Monitor

The Monitor building block of the PAC architecture can be seen as comprising of two different areas, namely:

- Data Collection,
- Data Analysis,

The data collection system collects all the relevant information from the shop floor and this is then analysed to produce both real-time and historical reports. Examples of real-time reporting include current utilisation levels, inventory levels, etc. Historical reporting involves producing graphs and reports on a variety of items of interest to manufacturing personnel. In this way the Monitor also acts as a decision support mechanism for the other PAC functions [ref. 6].

14
In the FMS the on-line data capture is carried out by the DDLs and the URD. These are polled by the Producer block, which supplies the Monitor with the relevant information. The Monitor uses this information to perform a number of tasks. These tasks include:

- Keeping the building blocks informed as to what is happening on the shop floor, e.g. keep the Dispatcher informed as to the status of the machines and the WIP.

- Producing both real-time and historical reports of the events.

- Comparing what is happening on the shop floor with what was actually scheduled. In this way the Monitor can also act as a decision support system for the Dispatcher.

3.5 Mover Or Materials Transport Function

The Mover building block in the PAC architecture controls the movement of parts throughout the system. This involves moving materials to the correct buffers to ensure that scheduled events can take place, and also moving parts from one machine to another when an operation has been completed. The Mover receives instructions from the Dispatcher telling it what part/material to move, where it is to be picked up and what the destination of the moving exercise is.
In the Grugliasco plant the materials transport system consists of a forklift and a driver. Because there is no automatic materials handling system in operation the Mover block will be relatively simple for this system.

3.6 Conclusion

This section has given a general description of the elements of the Grugliasco FMS with reference to the different PAC building blocks. Now this report shall discuss the specifications for the building blocks for the system.
In these sections the use of the PAC architecture is highlighted and each different building block is discussed in some detail. Figure 4 shows the PAC architecture used in the FMS. The main elements in the diagram can be listed as follows:

- Manufacturing Profile
- Application Network
- Producers
- HERMES Standard Monitor
- Monitor
- Dispatcher
- Scheduler
- Mover
Figure 4: PAC ARCHITECTURE
4.1 The Manufacturing Profile

The Manufacturing Profile is the storage area, accessible to all the building blocks, which contains the information that describes the manufacturing system. For the Grugliasco system the Manufacturing Profile is contained on a microVAX in the Methods Department.

The Methods Department receives a weekly issuance of order requests. These 'requests' are stored on a microVAX in the department. This microVAX is then used to verify the existence of the part programs necessary to produce the ordered parts. If certain part programs are not available then they are coded in the Methods Department and stored on the microVAX. Additional information stored on the microVAX includes:

- Information about the machines which will produce the ordered parts,
- The list of fixtures,
- The list of tools,
- The cycle time of the parts,
- The time to mount the tools,
- The time to dismount the tools,
- The time to mount the fixtures,
- The time to dismount the fixtures.

This information is made available to other processes running in the FMS through the use of the Digital product DFS (Distributed File System).

4.2 Application Network

Each of the applications interact with each other via the Application Network (AN). The AN is a distributed software bus that allows all the building blocks to communicate with each other in a controlled environment.

The AN together with the Manufacturing Data Dictionary provides a comfortable application environment for the building blocks of a PAC system and to enable communication between the building blocks via the so defined interfaces. The AN communication facility includes intelligent data passing and data access mechanisms.

4.3 Scheduler

This section defines the general focus of the Grupiiasco FMS Scheduler. It is to be a fully functional FMS scheduling system which will:
- generate a "weekly-term" detailed sequence of jobs at each of the four machines which designates start and complete times for each order.

- generate a "short-term" detailed sequence of jobs to recover the slack between the scheduled weekly production plan and the real production activity.

The Scheduler will support the overall objectives of reducing setup operations, work in process, throughput time and support labor, giving instructions for the fixtures and the tools preparation.

Fixture management is an important aspect in the scheduling of this system. The Scheduler has to allocate the fixtures necessary for the production of an order early enough to allow them to be mounted on a base.

In the future it may be required to evaluate the minimum setup time when scheduling the changing from one order to another. A matrix representation of setup time could be used.

Each machine has the capacity for two fixtures. The parts are moved by means of a crane and loaded onto the fixtures in the input bay and unloaded, after processing, in the output bay at the machines. There are twenty fixture bases available in the system (two per machine for the current order, plus two per machine for the incoming order, plus four others). The fixtures are categorized in fixtures kits that have to be prepared in
time for the scheduled production. Specific data such as "time to mount fixture" and "time to dismount fixtures" for each part family code is available on the microVAX in the Methods Department.

The modularity of the fixtures and their limited quantity are two important aspects in developing a Scheduler that is able to manage the allocation of fixtures between the orders. A list of the tools to be mounted on the machine to produce the scheduled parts is necessary for the operators. This could be obtained externally at the Scheduler using a cross reference between the orders scheduled, the part family and the tools. A setup matrix should be used to minimize the setup time. A limited number of cross references on the setup matrix could be obtained classifying the setup time into three classes: total change, 50% change and no change.

Alternatively a cross check of the tool list for each part family should give precise information about the mount/dismount time. For example, if the difference between the reference tool list of two parts in sequence consists of four tools, and the time to mount a tool takes ten minutes, then the total setup time in the system between these two parts will be forty minutes.

At the moment, it is supposed that at each setup all the tools must be replaced. Thus, the setup time is given by the number of tools for the current setup multiplied by a tool
mounting' constant time plus the number of tools of the previous lot multiplied by a 'tool dismounting' constant time.

Note that the difference in units of time between the part that requires more tools and the part with less tools is small, perhaps ten to twenty minutes. A precise setup time for each part would be not so relevant taking into account the manual aspect of the operations. Therefore, it is not required to schedule the tool setting operations even if the presetting time could require several hours of work.

On analyzing the functionality of the plant, sequencing rules such as following will probably be used:

- If the alternative machines are not used, they are available for another order.

- All orders for parts with the same material should be placed in a successive sequence to allow homogeneous chip collection. If, however, the time needed to change the chip containers is much less that the setup time then this criterion is not so important.

- The orders with minimum matrix setup time must be preferred in the sequencing. At the moment, this has not to be considered because the setup time between orders is not significantly different.
- The operator must be allowed to give a priority to orders. This priority overrides any other sequencing rule.

In general, the schedule will then use an overall sequencing rule based on the following:

"HIGHEST PRIORITY FIRST AND THEN OPTIMIZE THE CHANGING OF MATERIAL". In the future "MINIMIZE SETUP TIME" could also be used.

The weekly window of time will be from the Friday morning of the current week to the Friday night of the following week, so that the work in process at the end of the current week will be considered. There should also be sufficient time to prepare fixtures and tools for the oncoming week.

4.4 Monitor

The Monitor building block for the FMS consists of both the HERMES Standard Monitor functions and a Production Activity Monitor. The Production Activity Monitor for the FMS will act as a supplementary monitoring function to reside alongside the existing HERMES monitoring functions.

4.4.1 HERMES Standard Monitor -
The HERMES Standard Monitor provides both diagnostic and monitoring information. As regards diagnostic information HERMES provides a schematic layout of the FMS showing the machines with different colours relating to the different machine states. Log messages provides explanations of what is happening.

HERMES also provides a machine data display, which displays current productions features such as the following:

- Machining
  1. executing part program code,
  2. elapsed time from the start of the part program execution.

- Tool status
  1. physical tool code,
  2. cutter code,
  3. tool status,
  4. tool nominal life,
  5. tool remaining life,

25
6. tool accumulated life,

7. warning threshold,

8. pre-warning threshold.

- Executing part program data

1. part program code,

2. part program description,

3. part program length.

At the end of each shift a summary of machine conditions and part production rates is presented, as well as listings and displays on the FMS efficiency and productivity.

4.4.2 Production Activity Monitor -

The different elements of the Production Activity Monitor are:

- Collection of the messages.

- Analysing the messages.

- Reporting.
It receives messages from both the Producer AN interface and Dispatcher processes. It then analyses these messages and updates relevant statistics on the FMS in memory. Reporting routines then provide this information in readable form to the user.

Figure 5 shows the data flow diagram for the Production Activity Monitor.
Figure 5: Production Activity MONITOR
The Production Activity Monitor receives the following information:

- A Scheduled Production Plan. This is output from the Scheduler. Using the data stored in this file the Monitor compares the schedule to what is actually happening on the shop floor. It displays whether or not the orders are meeting the schedule and gives values regarding the slack of each order.

- The number of parts worked on the machines. This comes from the Producer AN Interface. Using this piece of information the Monitor updates variables concerning the WIP in the system. This is used in conjunction with the scheduled production plan.

- The status of the machines. Also from the Producer AN interface. This provides the Monitor with the ability to manage processing and waiting timer variables concerning each machine, thus giving reports on machine utilisation and other time statistics.

- A message from the Dispatcher referring to the start of a job on a machine. When the Monitor receives this message, it can then compare it to the scheduled finish time in order to give an account of the slack.
The following real-time reporting functions will be provided by the Monitor:

- Production Statistics Report. The information for this report is calculated from the machine status messages sent by the Producer AN interface. This option gives information on the following:

1. Current status of each machine (Busy, idle etc.).
2. Current Utilisation.
3. The time that each machine has been busy.
4. The time that each machine has been down.
5. The time that each machine has been idle.

- WIP Information -- The information for this report is calculated from the messages sent by the Producer AN interface and data from the scheduled production plan. This option gives information on the following:

1. Part program name i.e. the part family id and the list of associated jobsteps.
2. The number of parts to be done.
3. The number of parts done.
4. The time spent in process for each jobstep.

5. The time spent in queueing for each jobstep.

6. The slack value for each jobstep.

- Log Report -- This option provides the user with a means of viewing the previous messages sent on the AN. These messages are displayed on a screen when requested by the user.

The next sections describe the logic involved within each module in the Production Activity Monitor process.

- 1. Receive AN Messages.

   In this section the messages are collected from the AN and recorded in a log file. The messages are then manipulated and sent to the analyse and update procedures of the code.

- 2. Analyse and Update.

   When this section receives a message it initially checks what type of message it is (i.e. either from the Dispatcher or Producer process). The receipt of a message allows the Monitor process to update all variables detailing the system status. Comparisons are made with the scheduled production plan on receipt of the messages and variables updated accordingly. Also all time variables are managed by
these procedures.

- 3. Report

This section allows a user to interact with the Production Activity Monitor via some screen procedures. The user can choose between three different options and dependent on which one is chosen the screen is updated after each new message has arrived and been analysed in the system.

4.5 Dispatcher

The function of the Dispatcher is to act as the real-time Scheduler of the system. This involves taking the detailed schedule and interpreting it. Then based on that interpretation the Dispatcher will issue instructions to the other control blocks of the system which will lead to the implementation of the schedule.

In the Grugliasco system a human operator will preform the actual dispatching operations. However, this operator will be supported by a software Dispatcher block which will have limited decision support capabilities and will also be linked to the AN. This Dispatcher will have three functions to perform for the system, these include:
- Receiving AN messages. The Dispatcher will receive information from the DDL's and the URD via the Producer software. This information will be sent along the AN using the Producer AN interface.

- Decision Support functions. When a message is received the Dispatcher analyses the information contained in it. If the status of any of the system's entities has changed then the Dispatcher decides if an event should be fired as a result of that change.

- Sending AN messages. If the Dispatcher fires an event that requires materials, components or products to be moved, then a 'Move' instruction is issued by the Dispatcher to the Mover, via the AN. This instruction contains information regarding the items to be moved, where they have to be moved from and their final destination. All instructions issued by the Dispatcher are sent, via the AN, to the Monitor.

When a schedule is generated it is stored in a Scheduled Production Plan file. This file contains a list of all the events that are scheduled to take place in the system. The Dispatcher has access to this file and uses the data contained in it when deciding what events should be fired.

When deciding what events it should fire the Dispatcher takes into account machine status and part availability. If a machine is 'Down', then the only event that can occur for that
machine, is a machine 'Up' event. If a machine is 'Busy' then no other part can be processed on that machine until the machine becomes 'Idle'. Similarly, if a part is being processed on a machine, it cannot be moved to, or processed on another machine until it has finished its current operation. The Dispatcher therefore requires a constant flow of information from the shop floor. This information if provided by the Producer, via the AN.

The Dispatcher can fire two kinds of events, a 'Dispatch' event and a 'Move' event. If it fires a 'Dispatch' event then the information regarding this event is displayed on the operator's terminal screen. If this were a fully automatic system, then the instruction would be sent direct, via the AN, to the Producer which would then translate the instruction into device specific commands for the machines. However, in this system it is the operator who performs the dispatching operations. A typical 'Dispatch' command would contain the name of the machine the event was to occur on, the name of the part that was due to be processed and the number that part was of the total batch.

If the event is a 'Move' instruction, then the necessary information is sent to the Mover process, via the AN. Both the 'Dispatch' and the 'Move' instruction are sent, also via the AN, to the Monitor process.
Figure 6 shows the data flow diagram for the Dispatcher building block.
FROM PRODUCER AN INTERFACE
PART WORKED AN
MACHINE_STATUS_AN
SYSTEM_STATUS_AN

RECEIVE AN MESSAGES

DISPATCH

SEND AN MESSAGES

GIVE INSTRUCTIONS

TO MONITOR
SCHEDULED PRODUCTION PLAN
JOB_RELEASED_AN
TO MOVER
MOVERMISSION_AN

DISPATCHER_REPORT

Figure 6: DISPATCHER
Because of the complexity of scheduling an FMS system the Dispatcher will not change the sequence of jobs generated by the Scheduler. In the case of an unexpected/unplanned event, e.g. machine breakdown, the Dispatcher will continue to issue instructions until no more events can occur. It will then wait for the machine to become available again before recommencing its dispatching functions.

4.6 Producer

In the standard COMAU FMSs the Producers send data to the Monitor. This data consists of information on machine status and parts worked. The communication is facilitated through the use of mailboxes. This system will replace that communication by using the AN to send information.

A Producer AN interface process is linked to with the Producer software. This interface is used to send the messages from the Producers to the Monitor building block (HERMES Standard Monitor and Production Activity Monitor) and the Dispatcher.

Other important functions of the Producer are part program and tool management. Part program management involves the uploading and downloading of part programs to the DDLs. Tool management involves both static and dynamic data concerning the tools on the methods department microVAX and DDL respectively.
In summary the Producer collects all the data from the DDLs and URD, manages all the part program transfers and tools and sends messages to the HERMES Standard Monitor, Production Activity Monitor and Dispatcher using the AN.

4.7 Mover

The Mover for the Grugliasco system consists of terminal messages provided by the Dispatcher instructing the forklift operators to move particular items from one area of the system to another.

The layout for the Mover screen contains the following sections:

- Current command to the mover e.g. "Move order 45 from CNC1 to CNC2"
- Current date and time
- Previous messages to the Mover.

The Mover process collects messages sent on the AN by the Dispatcher and prints these on the screen for the operator. For the moment no more functionality is needed here as the moving operations are performed manually and controlled directly by the supervisor.
The working of the Mover modules is summarised as follows:

- 1. Receive AN Messages.

   In this section the messages are collected from the AN and recorded in a log file. Then the message is then sent to the issue instructions procedures.

- 2. Issue Instructions.

   This section just takes the message received on the AN and prints it on the screen in a readable format.

   If at a later date an automatic materials handling system is incorporated into the system then instead of printing the instructions from the Dispatcher on a terminal, those instructions would be translated into commands compatible with the mover system.

5 CONCLUSION

A PAC control system is being developed for a four machine FMS at the COMAU Grugliasco plant. It will consist of the five standard PAC building blocks:

- Scheduler,

- Dispatcher,
At the moment prototypes of these blocks are being created, and when fully developed will be integrated to form a fully functional PAC modular control system.

This report is designed to give an overview of the work being done to develop the control system for Grugliasco. It initially looks at the FMS hardware, explaining how it works, and then discusses briefly the parts and components that will be processed by the system. The following section maps the PAC control hierarchy onto the Grugliasco FMS by imposing each building block on the different areas of control in the system. The final section discusses the specifications for the building blocks by examining the purpose and functionality of each block.
References


Scheduling In The Renault Testbed, 
A Repetitive Manufacturing Environment

Declan Kennedy M.Eng.Sc., 
Digital, Clonmel.
1 Introduction

This paper details the work done in University College Galway, toward the development of scheduling strategies for Production Activity Control within the Renault test site at Cleon as part of ESPRIT project 477, COSIMA. The test site represents a repetitive manufacturing environment, while still retaining batch production in some shops.

2 PAC and Repetitive Manufacturing

This section will briefly outline the difference between the various types of manufacturing and the implications for production co-ordination in repetitive manufacturing environments. The type of PAC strategies needed in repetitive environments will be dealt with before examining the test site requirements in the succeeding section.

2.1 Production Mode

Production Mode is the fundamental arrangement and method for manufacture. Fixed Location and Job Shop, through to Mass Production are used to represent the production mode or manufacturing type continuum. Wrennall and Lee[?] have characterised a Production Mode by several key features as shown in Figure ?? . As the type of manufacturing moves along the production mode continuum production becomes more repetitive and ultimately mass production is achieved. In parallel to this production mode hierarchy, Wrennall and Lee[?] propose a production co-ordination hierarchy, as shown in Figure ?? . This diagram indicates that production co-ordination becomes simpler as manufacturing moves up the production mode hierarchy where simpler and hence, more effective production control systems may be employed. This phenomena is due to the decreasing variation in the manufacturing system as we move toward mass production, with increased repetition and increased focus on one product or product family. Hence, production control and co-ordination will be simpler, as "control can be obtained only if the variety of the controller is at least as great as the variety of the situation to be controlled", this is Beer's[?, p.4] redefinition of Ashby's Law of Requisite Variety[?, p.207].
## Production Modes

### Characteristic Features

<table>
<thead>
<tr>
<th>Features</th>
<th>Continuous Line</th>
<th>Toyota Cellular Functional</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrangement</td>
<td>Sequential</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Focus</td>
<td>Product</td>
<td></td>
<td>Process</td>
</tr>
<tr>
<td>Product Flow</td>
<td>Simple &amp; Direct</td>
<td></td>
<td>Complex &amp; Mixed</td>
</tr>
<tr>
<td>Resource Flow</td>
<td>Fixed</td>
<td></td>
<td>Very Flexible</td>
</tr>
<tr>
<td>Sequence</td>
<td>Fixed</td>
<td></td>
<td>Very Flexible</td>
</tr>
<tr>
<td>Cycle Time</td>
<td>Very Low</td>
<td></td>
<td>Very High</td>
</tr>
<tr>
<td>Repetition</td>
<td>Very High</td>
<td></td>
<td>Very Low</td>
</tr>
<tr>
<td>Balance</td>
<td>Very High</td>
<td></td>
<td>Very Low</td>
</tr>
<tr>
<td>Synch.</td>
<td>Very High</td>
<td></td>
<td>Very Low</td>
</tr>
</tbody>
</table>

Figure 1: Production Modes and Characteristics Features
A Hierarchy for Production Coordination

Most Desirable

Flow Line

Broadcast

Kanban

MRP

ReOrder Point (ROP)

Least Desirable


Figure 2: Production Co-ordination Hierarchy
3 Repetitive Manufacturing

Repetitive manufacturing is making the same thing over and over - repeatedly if not frequently. Then potential exists to produce by a rhythm, or repetitive pattern. It also includes making items by a repetitive process, even if the products themselves are not in a standard sequence, or even if no two products are exactly alike. Browne, Harhen and Shivnan, have a similar view of repetitive manufacturing with one important difference, this is that the products are produced in volume. I will use this definition when discussing repetitive manufacturing. Hence, repetitive production refers to the production of discrete parts by a process which will lie on the continuum from Job to Mass production between Batch and Mass production. From Wrennall and Lee's Production Mode hierarchy we see repetitive manufacturing could take place in the Line and Toyota modes and possibly in the Cellular mode. From what we discussed earlier it is clear that repetitive manufacturing systems should have a relatively simple PAC system, i.e., repetitive systems are easier to plan and control in the short term.

Hence a move toward continuous production is desirable. But in terms of discrete parts manufacturing systems, a move to mass production or the Line mode is not desirable. This is due to the characteristics of the new evolving environment in which manufacturing has to operate, namely increased product diversity and significantly reduce product life cycles. Thus, repetitive manufacturing in modes which can accommodate these characteristics is the most desirable. Such modes are the Toyota or Cellular modes. The Toyota mode relies on the Just-In-Time (JIT) philosophy and in particular the realisation of JIT through the use of the shop floor control system Kanban.

Kanban provides a model for the real time PAC in repetitive manufacturing environments. A model for the short term planning (scheduling) of PAC systems for these environments is provided by the production smoothing and planning techniques which are essential for the successful use of Kanban. Kanban is used to co-ordinate production at various levels, at shop floor level, to co-ordinate the activities of the various shops and cells and to link the manufacturing system with its suppliers. Hence, Kanban provides a model for a higher level recursion of PAC, co-ordinating activities between shops, workstations and assembly lines.
4 A PAC Scheduler for Repetitive Manufacturing

There are three basic purposes of scheduling as outlined by Hall[?]:

- Matching the production process to the demands of the market.
- Co-ordinating operations both those internal to the firm and those taking place at the suppliers.
- Stimulating improvements in operations.

An important factor is that scheduling must deal with operations in their current state, both in physical development and in organisational development. The method of scheduling should reflect the current status of manufacturing development. A schedule must be within an organisation's capability to execute it or else it is meaningless. Hence, the use of certain production management strategies will support certain daily scheduling activities. Therefore scheduling in repetitive manufacturing systems will have to reflect the organisation of the processes and the strategies used in controlling production.

The objective of production planning in repetitive manufacturing environments is to develop a level schedule. A schedule is level when the material and labour requirements are evenly distributed. If final assembly is dedicated to producing just one model with a fixed BOM (bill of materials) all day every day, the schedule will be level provided the rate of production does not vary. This is the simplest case for production planning. When a separate final assembly line area is established for each final product, planning would certainly be simplified. However, this cannot usually be done because the volumes demanded for each model cannot economically justify it, and it is necessary to assemble more than one model in each assembly area. In order to keep the flow of materials required at final assembly as level as possible, the assembly lot sizes should be as small as possible. Therefore a mixed sequence of all models to be assembled should be prepared. This method of production is a mixed-model production process, and differs from a multi-model process. In a multi-model production process there are long runs of the individual products, whereas in a mixed-model process products within the same product family are interspersed one after another, e.g. A-B-C-D-A-B-C-D.
4.1 The PAC Scheduling Block

The scheduling building block for repetitive manufacturing environments will carry out short term planning, scheduling and sequencing. It will not carry out real time scheduling, this is the role of the dispatcher. The emphasis is placed on scheduling the end-items, (final assembly products). The end-item assemblies, subassemblies and parts are scheduled in the light of the final assembly or end-item schedule.

4.2 Production Smoothing

Hence, production should be adaptable to demand changes, resulting in the elimination of excess finished product inventories. The means for adapting production to variable demand is called production smoothing, Monden[?, p.55]. Through production smoothing, a production line is no longer committed to the manufacture of a single type of product in large lot sizes; instead a single line must produce many varieties each day in response to varying customer demand. Hence, production is kept up-to-date and inventory is cut.

There are two phases of production smoothing, as shown in Figure ??:

1. Adaptation to monthly demand changes during a year (monthly adaptation).
2. Adaptation to daily demand changes during a month (daily adaptation).

Monthly adaptation is achieved by monthly production planning: the preparation of a master production schedule showing the averaged daily production level of each process in the plant. The daily adaptation is carried out by the PAC system. It involves the formulation of a schedule, based on firm orders, and the use of a pull system for production control to achieve the dispatching of the right products, in the right quantities at the right time.

5 The Renault Test Site

The Renault test site is an engine assembly plant is located in the Renault plant in Cleon, France. The test site is a section of the plant, Department 43, which specialises in the production of one engine family, Engine F, and consists of a number of workshops as shown in figure ??.

The workshops in the test site include:
Production promptly adaptable to demand changes

"Just-in-Time" Production
(produce the salable products in salable quantities)

Monthly adaptation:
adaptation to the monthly
demand change during a year

Daily adaptation:
adaptation to the daily demand
changes during a month

Monthly production planning

Instructing the daily
production level of each unit
to each process

Daily production dispatching

Dispatching the actual
production quantity of each
product daily

Master production schedule:

Determination of the daily
average quantity for each
kind of product, based on
the monthly predetermined
production quantities

Pulling system by kanban

Determination of the sequence
schedule for the mixed model
assembly line to enable smoothed
withdrawals from subassemblies

Three-month forecast
and monthly forecast

Ten-days order and daily order
from dealers

Flexible machinery

Reduction of the
production lead time

Figure 3: Framework for Production Smoothing, Monden.
Figure 4: The Renault Test Site

- Camshaft machining,
- Cylinder Head machining,
- Engine Block machining,
- Crankshaft machining,
- Piston (Connecting) Rod machining,
- Assembly Workshop area, which is divided into:
  - 3 Preparation areas:
    * Cylinder Head preparation,
* Engine Block preparation,
* Assembly Kit preparation or Basket preparation.

One Final Assembly area, which is made up of:
* Engine Dress-up,
* Test benches,
* Final Dress-up,
* Shipment.

Other work areas include:
- F3 Cylinder head preparation,
- Manifold Preparation.

The bill of material (BOM) for Engine F has four levels, i.e. there are three sub-assemblies before the final part is finished. This idea is demonstrated in figure ??.

To build an engine the cylinder head and the engine block have to be assembled together. This means firstly machining the piston rods, crankshafts, the cylinder block, the cylinder head and the camshaft. The cylinder head and the engine block are then assembled together at Assembly Point 1. The result of this assembly is then passed to Final Assembly where a kit, made up of raw materials from external vendors, is used to finish the engine. This kit is made up of parts, such as, the carburettor, distributor, etc. The finished engine is then moved to a buffer until it is shipped off into another department.

5.1 Analysis of Scheduling and Dispatching in the Test Site

The fundamental objective of production planning and control is to match the production process with the demand for the finished products. In the test site the scheduling and dispatching functions provide co-ordination between the assembly, preparation and machining processes. The machining shops would have their own PAC systems but the assembly and preparation processes may be viewed as individual workstations since they consist of assembly lines which complete a number of operations on the one item and where normally the items enter and leave the line in the same order. Hence, the PAC system which provides production planning and control in the Cleon plant is similar to the higher level recursion of PAC.

The test site is able to adapt to changes in demand for its products but this ability is limited by the nature of the raw material supply to the machining shops. Since
Bill of Materials structure for the Car Assembly Industry

Figure 5: The Bill of Materials for Engine F
vendor relationships have not reached a stage where the suppliers can guarantee to change their planned production to match the changing demand from the Cleon plant, a less favourable solution such as large stocks of raw materials has to be employed.

5.2 Push and Pull Production Control

Scheduling and dispatching in department 43 operates in both a “pull” and a “push” manner with the co-ordinating and control functions focused on the procurement operator, and the “launching” of engine batches on to ENGBLP. The machine shops all produce in a “push” fashion although they may have batches expedited or pulled through the shop if required. The machining shops push items into their buffers and these items are subsequently pulled from these buffers by the relevant preparation areas.

The procurement operator performs the scheduling function and some of the dispatching functions. Based on machined parts availability and planned machined parts production he creates a schedule of engines to be produced. He transmits this schedule to all the preparation shops except KITPRP. The engine blocks are then actually started on ENGBLP and these then go through to ASSPT2. The production at ENGBLP, ASSPT1 through to ASSPT2 is the “driving” process, see Figure ??1. Hence, all cylinder heads and manifolds are pulled from their respective buffers to match the engines which are ready for assembly at ASSPT1. Likewise, the production at KITPRP is driven in real time by the engines ready for assembly at ASSPT1. These kits are then pulled from their buffer as engines are brought into ASSPT2.

All withdrawals from buffers are decided upon and made in real time and are not scheduled, but all production is schedule. The machine shops are scheduled weekly and possible daily. The preparation shops ENGBLP, CYLHED, MANPRP and F3PREP are provided with daily schedules which may be altered in real time or rescheduled, (rescheduling would occur at the start of every shift). The production in ASSPT1, KITPRP and ASSPT2 is determined in real time based upon the production of preceding shops not succeeding shop.

Figure?? diagram uses the acronyms used to describe the machine shops, preparation shops and assembly lines. A list of these acronyms and the shops they represent is given below in Table ??.

11
The scheduling solution for the test site is designed to provide co-ordination between the machining shops and the preparation areas and in turn to co-ordinated these preparation areas with the assembly lines. There is no requirement for detailed planning of the machine shops in this solution as each would its own internal activities. The preparation lines and assembly lines can be viewed as individual cells or workstations, thus indicating that a PAC solution for the test site will be similar to a higher level recursion PAC system. Within this the scheduling block will be dependent on the state of the manufacturing organisation and development. This includes the dispatching and monitoring practices employed.

The scheduler is designed to provide a final assembly schedule from which schedules for all the preceding workstations and shops may be derived. A capacity check is provided, and an interface to alter the planned production orders to facilitate the formation of a manageable sequence of products.
Figure 7: The structure of the Kanban Scheduler

- Planned Orders

Each of these forms a data structure using the LISP macro function `defstruct`, e.g. see Figure ??.

6.2.1 Data Retrieval and Storage

The database is made up of data files. These are then loaded and instances of the data structures discussed above are formed. The files are easily retrieved and altered through the interface. The files used are; part data files, operation data files, materials data files, workstation data files, buffer data files and planned orders files. The interface provides a means of entering new data describing new manufacturing environments, editing or appending to existing databases or files. The interface for data entry provides error checking.

6.2.2 Operation of the Scheduler

Having chosen the database the procedure used to form the sequence schedule is straightforward. The user is first given the five options;

\[1\] For further information see reference[?]
(Defstruct W/Station-data
   (Name "" :type String :read-only T)
   (Capacity 0.0 :type float)
   (Production-rate 0.0 :type float)
   (Pou nil :type symbol :read-only T)
   (Buffer nil :type symbol :read-only T)
   (Process-times nil :type list :read-only T)
   (Setup-times nil :type list :read-only T))

Figure 8: The LISP macro defstruct

- To view data.
- To edit the planned orders.
- To check the capacity.
- To create sequence schedule.
- To view sequence schedule.

The user may view and check the data before running the scheduler. The planned orders received may be altered based on the user's knowledge of the existing environment and the results of the capacity check which analyses the time constraints the planned demand imposes on the production process. Having decided on the set of products to be produced and the quantities of these, the composition of the sequence is fixed. The next phase is to actually sequence the products which make up the sequence. A heuristic algorithm is used to do this. The objective of the algorithm is to level the load on the workstations and to keep a constant speed in consuming each part on the line. Schedule data and resultant sequence are shown in Figure ??.

The sequenced schedule may be evaluated to show how evenly the parts are consumed throughout the sequence cycle. This is done graphically as shown in Figures ?? and ?? . In Figure ?? the part selected was PSTRD3-1, the dash one indicates that the part chosen is PSTRD3 after its first operation. Only one part has more than two operations carried out on it before being assembled with other
Figure 9: The Resultant Sequence

Figure 10: Evaluation of the Sequencing Algorithm with respect to PSTRD3
Figure 11: Evaluation of the Sequencing Algorithm with respect to KIT1 parts, this is PRCYH6 which has two operations as it needs special preparation at F3PREP. The vertical axis of the graph represents the number of the part, in this case the part is PSTRD3, which have been consumed in one sequence cycle, while the horizontal axis represents the number of products which have been produced in that sequence.

Hence, if you refer back to Figure ??, zero on the horizontal axis of the graph in Figure ?? implies that none of the products in the sequence have been completed and therefore none of the part PSTRD3 has been consumed. Whereas 22 on the horizontal axis implies that all products from the first, DRENG1, to the last, which is again DRENG1, have been produced with twenty eight of the part PSTRD3 being consumed. The diagonal line represents the mean consumption of the part PSTRD3. Since the jagged line, which represents the actual consumption of PSTRD3, is close to the diagonal line at all points, it indicates smooth consumption of the part PSTRD3. The same evaluation methodology applies to all thirty–nine different parts needed to make up the six different engines in the final assembly sequence.

The user may view the sequenced schedule for all the workstations which are used in the process. The sequenced schedules for all the workstations which feed into the final assembly line are derived from the final assembly sequenced schedule.
7 Conclusion

This paper has detailed the characteristic features of repetitive production and outlined a scheduling strategy which for the scheduling block of a PAC system which was based on the JIT shop floor control philosophy. The PAC system, including the scheduling block, would be orientated toward achieving co-ordination between the machine shops and the preparation and assembly lines similar to the higher level recursion of PAC, factory co-ordination.

The scheduler developed for Renault, realises the implications of the manufacturing environment's characteristic features in the design of scheduling systems and provides a schedule which is executable in a repetitive environment.

References


