

REAL-TIME SCHEDULING IN A SENSORISED FACTORY USING CLOUD-BASED SIMULATION WITH MOBILE DEVICE ACCESS

S. Snyman^{1*#} & J. Bekker¹

ARTICLE INFO

Article details

Presented at the 28th annual conference of the Southern African Institute for Industrial Engineering (SAIIE), held from 25-27 October 2017 in Vanderbijlpark, South Africa

Available online 13 Dec 2017

Contact details

* 17762456@sun.ac.za

Author affiliations

1 Department of Industrial Engineering, Stellenbosch University, South Africa

The author was enrolled for an MEng degree at the Department of Industrial Engineering, Stellenbosch University

DOI

<http://dx.doi.org/10.7166/28-4-1860>

ABSTRACT

Scheduling is a challenge that persists in the operational phase of the manufacturing life-cycle. The challenge can be attributed to the complex, dynamic, and stochastic nature of a manufacturing system. Computer simulation is often used to assist with scheduling, as it can sufficiently mimic complex, discrete, dynamic, stochastic processes. We propose an architecture of a real-time simulation scheduling system that incorporates the use of a sensorised-network of a job-shop, mobile devices, and cloud computing with simulation and scheduling methods. A simulation model is also created to describe the environment and operations of a job-shop.

OPSOMMING

Skedulering is 'n uitdaging wat in die operasionele fase van die vervaardigingslebensiklus voortduur. Hierdie uitdaging kan toegeskryf word aan die komplekse, dinamiese, en stogastiese aard van 'n vervaardigingstelsel. Simulasie word dikwels gebruik om te help met die skeduleringproses, aangesien dit komplekse, diskrete, dinamiese, stogastiese prosesse voldoende kan naboots. Ons stel 'n argitektuur van 'n reële tyd simulasie skeduleringstelsel voor, wat gebruik maak van 'n sensornetwerk in 'n werkswinkel, mobiele toestelle, sowel as simulasie- en skeduleringmetodes in die elektroniese wolk. 'n Simulasiemodel is ook geskep om die omgewing en aktiwiteite van 'n werkswinkel te beskryf.

1 INTRODUCTION

Manufacturing is described by Groover [1] as the transformation of materials into items of greater value by one or more processing or assembly operations. The processes to accomplish the transformation involve a combination of machinery, tools, power, and manual labour. From Groover's description of manufacturing, the process can be considered a complex engineering endeavour, where the coordination of people, material, equipment, and information to accomplish a manufacturing goal demands considerable time and effort.

According to Mitsuishi, Ueda and Kimura [2], the manufacturing system life-cycle can be divided into design, planning, implementation, operation, and termination phases. The coordination challenges that are faced in the design and implementation phases can be overcome by careful planning; however, such challenges persist into the operational phase. For many manufacturing systems, scheduling is one such challenge that can be attributed to the complex, dynamic, and stochastic environments exhibited by these systems.

Traditional approaches used to address the scheduling problems involve creating and evaluating schedules prior to beginning production. However, according to Suwa and Sandoh [3], uncertainties that are not expected nor taken into account at the planning phase can cause delays of these schedules. Common uncertainties that occur in a manufacturing system include machine operator absence, material shortages, and machine failure. In such scenarios, the manager must react quickly by selecting a new or revised schedule to ensure that production continues while maintaining the

required performance level. On-line scheduling, which is a real-time decision-making process, attempts to address the shortcomings of the traditional approaches by performing scheduling concurrently with the production process. Computer simulation is often used to assist with scheduling, especially of complex, discrete, dynamic, stochastic processes; this is generally known as *simulation-based scheduling*. Kim and Kim [4] state that computer technology has significantly reduced computation time for this evaluation, so that simulation may be used as a tool for real-time evaluation in many circumstances.

Stankovic, Spuri, Ramamritham and Buttazzo [5] state that a real-time system typically consists of a controlling system and a controlled system. In an automated factory, the controlled system is the factory floor with its resources and equipment, while the controlling system is the computer and human interfaces that manage and coordinate the activities on the factory floor. The controlling system interacts with its environment, based on the information available about the environment from various sensors attached to it. It is essential that the controlling system's perception of the state of the environment be consistent with the actual state of the environment. Many technological advances – including cloud-based computing, the omnipresence of mobile devices, and the improved capabilities of sensor networks – have offered the opportunity to design a real-time scheduling system as described by Stankovic *et al.* [5], and the opportunity to create software architectures to support these real-time scheduling systems.

This paper forms part of an ongoing research project. The objective is to develop a prototype of a real-time simulation scheduling system that will serve as a decision support tool for real-time rescheduling of machine steps in a job-shop. The system will use a sensed network in a job-shop, mobile devices, and cloud computing with simulation and scheduling methods. The aim of this paper is to explain the design and development of the prototype and to provide architectural design and operational issues. To accomplish this, certain objectives need to be completed. These include: (i) describing basic knowledge of simulation-based real-time scheduling and scheduling of job-shops; and (ii) a discussion of the investigation and construction of a suitable software architecture for the proposed system.

We provide a brief outline of a job-shop and its scheduling problem in Sections 3 and 4, followed by the application of discrete-event simulation as a scheduling method in Section 5. In Section 6, the architectural design of the proposed prototype will be discussed. The current state of the research will be discussed in Section 7, followed by an outline of future work in Section 8.

2 OVERVIEW OF MANUFACTURING SYSTEMS

According to Murthy [6], the organisation, planning, and control of production within a manufacturing system greatly depends on the type of production. Furthermore, aspects of production management in fulfilling specific requirements of a plant, and the management approach to problems of inventory, machine selection, machine setting, tooling, routing, scheduling, loading, follow-up, and general control will differ depending on the type of production system. Three main types of production systems are referred to in the literature ([6]-[8]): job-shops or job production, batch production, and continuous production.

1. Job-shops: In this system, products are manufactured to meet the requirements of a specific order. Manufacturing the product will take place to meet the specifications given by the customer.
2. Batch production: In this system, a number of identical products are manufactured either to meet a specific order or to satisfy demand. When the production of the plant and equipment is terminated, the plant and equipment can be used to produce similar products.
3. Continuous production: This system is the specialised manufacture of identical products on which the machinery and equipment are fully engaged. Continuous production is normally associated with large quantities and a high rate of demand.

Figure 1 illustrates the characteristics of intermittent (job and batch) and continuous (mass and flow) production systems. As can be seen in Figure 1, job production has major differences in product variety, but has very low product volumes; batch production has a large product variety with low product volumes; and continuous production has little product variation but high product volumes.

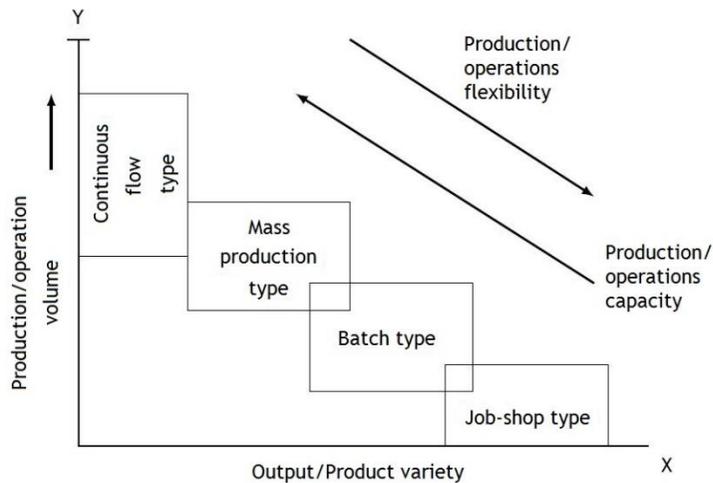


Figure 1: Characteristics of production systems (Murthy [6])

The focus of the next section will shift to the scheduling of jobs within a job-shop production system. For the job scheduling process, machine-level controlling measures such as sequencing, dispatching rules, and performance indicators have to be identified. These will also be discussed in the next section.

3 SCHEDULING IN JOB-SHOPS

Wisner [9] presents a number of factors involved in scheduling jobs in a job-shop. These factors are described below.

- If more work is accepted per day than the organisation can complete, then the overall work-in-progress inventories will increase and cause shop congestion, erode the firm's output rate, and lengthen job completion times.
- If completion times or dates are promised to customers, then estimates of lead times for each job must be determined, and jobs must be started far enough in advance to complete the job by the promised date.
- Facilities can finish more jobs per period and satisfy more customers if they work on the shortest jobs first. However, longer jobs will ultimately be completed late or behind schedule.
- More highly valued customers may require earlier completion dates, which will then be given processing priority in the shop. This, in turn, will make it more difficult to estimate accurate completion dates for other jobs.
- Interruptions such as machine breakdowns, employee absence, poor raw-material quality, and processing errors can cause unforeseen delays in processing.

The job scheduling process therefore includes machine-level controlling measures such as sequencing, dispatching rules, and performance indicators. The dispatch rules guide the production sequence of the jobs within the shop, and ensure that operators know what job to process next when the machine becomes available. Dispatch rules, as mentioned by Wisner [9] and Dominic, Kaliyamoorthy and Kumar [10], include:

1. shortest process time,
2. earliest due date,
3. minimum slack time per operation,
4. critical ratio,
5. first-come-first-served, and
6. most-important-job-first.

According to Pezzella, Morganti and Ciaschetti [11], one of the most difficult problems in the area of *job-shop scheduling problems* (JSSP), where a set of jobs must be processed on a set of machines, is deciding how to sequence the operations on the machines. This decision is dictated by the job-

shop performance indicators, which need to be optimised. Table 1 provides common job-shop performance indicators, and a description of each.

Table 1: Common job-shop performance indicators ([9], [10]).

Performance Indicator	Description
Average flow time	The flow time begins when a job arrives at the shop, and ends when it leaves. This flow time is averaged over a number of jobs.
Average queue time	The queue time is the total flow time minus the process time of the job. The queue time is then averaged over a number of jobs.
Average job lateness	Lateness is the difference between the completion date and the due date. The lateness is averaged over a number of jobs.
Average job tardiness	Tardiness is the amount of time beyond the due date that a job finishes. Tardiness is averaged over a number of jobs.
Makespan	Makespan is the total elapsed time to complete a number of jobs.

According to Vredeveld [12], in standard deterministic scheduling all relevant data on the problem is known beforehand; but in a real-world problem, this assumption is not always realistic. Therefore, in many scenarios, a good schedule needs to be found when data is incomplete and decisions with wide-ranging implications need to be made. An approach that can cope with this uncertainty is *stochastic scheduling*.

The attributes of the stochastic scheduling problem, including processing times, due dates, unexpected releases of high priority jobs, and machine up/downtimes, can be viewed as decision variables. Furthermore, probability distributions are used to describe the decision variables, and ultimately help to determine a schedule in a stochastic environment [13], [14]. These decision variables are then used in the JSSP, which will be discussed next.

4 THE JOB-SHOP SCHEDULING PROBLEM

The JSSP is referred to in the literature as an NP-hard problem, where the environment, as defined by Kuhpfahl [15], is as follows:

- There is a set J of n jobs $\{J_1, J_2, \dots, J_n\}$ that need to be processed on a set M of m machines $\{M_1, M_2, \dots, M_m\}$;
- Each job i consists of a finite and predetermined sequence of j operations $O_i = \{o_{i,1}, o_{i,2}, \dots, o_{i,j}\}$, where the operation order is fixed;
- An operation may only be assigned to an available machine forming part of the set M ;
- A machine can process at most one operation at a time, and no pre-emption can take place;
- Each operation $o_{i,j}$ has a fixed processing time $p_{i,j}$; and
- The aim is to find a schedule for processing these n jobs on the m machines.

According to Sadeh [16], job-shop scheduling must be regarded as a constraint satisfaction problem (CSP) or a constraint optimisation problem (COP). The constraints applicable to the JSSP are the *precedence constraints* specified by the process routings, and the *capacity constraints* that prevent resources from being allocated to more operations than they can process at one time.

Error! Reference source not found. illustrates a JSSP where there are four jobs on five machines. Each node in the figure represents an operation, and is labelled with the name of the operation ($o_{i,j}$), where i is the job index and j is the operation index. The k -th resource/machine required to process the operation is indicated by M_k . The processing time p_{ijk} of each operation is shown as a number in each node. The arrows in the figure represent the precedence constraints, and the broken lines represent the capacity constraints. In this example, it is assumed that each machine can only process one operation at a time; hence the use of a capacity constraint. Therefore, if more than one operation is competing for a machine, all but one must wait, as they cannot be processed at the same time.

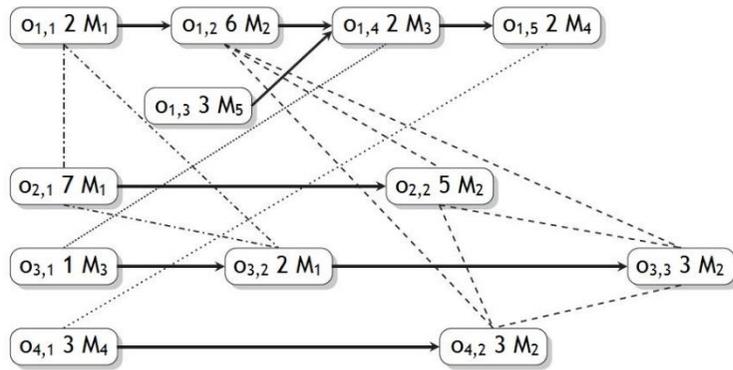


Figure 2: A sample job-shop problem (Sadeh [16])

An example of the precedence constraints in **Error! Reference source not found.** is the precedence between $o_{3,1}$, $o_{3,2}$, and $o_{3,3}$. The precedence between these operations ensures that $o_{3,1}$ is processed before $o_{3,2}$ and $o_{3,3}$. Furthermore, operations $o_{1,2}$, $o_{2,2}$, $o_{3,3}$, and $o_{4,2}$ all need to be processed on machine M_2 ; hence the capacity constraint.

In this study, the use of simulation as a scheduling method is investigated, and will be discussed next.

5 SCHEDULING WITH DISCRETE-EVENT SIMULATION

Discrete-event simulation is described by Schriber, Brunner and Smith [17] as the imitation of a real-world process or system where the state of the model changes only at a discrete, but possibly random, set of simulated time points, called *event times*. The rescheduling of jobs in the operational job-shop, where system state changes are caused by unexpected machine breakdowns, operator absence, or the arrival of new orders, can be defined as an *online, dynamic, and stochastic* process. Discrete-event simulation is therefore a suitable tool to study such problems.

Herrmann [18] describes the use of simulation as part of the production control approach, where three uses of simulation are identified: simulation-based schedule generation, refinement, and optimisation. Simulation-based schedule generation and refinement use simulation directly to determine initial schedules (generation) and to improve existing schedules (refinement). In addition, simulation-based optimisation applies simulation repeatedly to estimate a certain objective function value and improve it. The optimisation of the objective function refers to the optimisation of the performance indicators of the job-shop, as described in Section 3.

Frantzén, Ng and Moore [19] present an application of simulation-based optimisation in the scheduling and real-time rescheduling of a complex machining line. The developed scheduler generates robust and adaptive schedules, while it also has the capability of rescheduling in almost real time to cope with the system uncertainty. The scheduler is integrated with the shop floor database system, and uses real-time data from the production line. Initial results from this industrial application indicated that the scheduling system helped to improve the line throughput while simultaneously supporting real-time decision-making.

6 ARCHITECTURAL DESIGN OF THE PROTOTYPE SIMULATION-SCHEDULER

According to Dori [20], artificial systems require development and support efforts throughout their life-cycle. Furthermore, systematic specifications, analysis, design, and implementation of new systems are becoming more challenging as contradicting requirements of shorter time-to-market, rising quality, and lower cost are on the rise. These trends warrant the creation of a comprehensive methodology that is capable of tracking the mounting challenges posed by the evolution of new systems. A methodology created for this purpose is the *object-process methodology* (OPM).

OPM consists of a concise set of symbols that form a language enabling the expression of the system's building blocks and how they relate to one another. It is, therefore, a symbolic representation of the objects in a system, their structural relations, and the functions they enable. A further

advantage of OPM is that it represents the system simultaneously in a graphic form and in a natural language. By doing so, both the visual and the language interpreters in the human brain – its most powerful processing capabilities – are engaged. OPM also forms part of ISO 19450, which specifies the use of the concepts, semantics, and syntax of OPM as a modelling paradigm and language for producing conceptual models at various levels of detail.

An architecture was developed with the use of the OPM to describe the proposed simulation-scheduler. The architecture is illustrated graphically in **Error! Reference source not found.**, followed by the corresponding semantic description. They are interchangeable, and represent the same information.

The architecture incorporates three uncertainties that initiate the rescheduling process: a user logging a new order, a machine that breaks, and an absent operator. The user handles the process of logging a new order, which changes the state of the information system. A sensor monitoring the state of a machine can pick up when a machine breaks, which then triggers the process of logging the broken machine in the information system. The sensor can also detect the movement of jobs through the shop. When a job arrives at a machine, or the machine finishes the processing of a job, the sensor detects it and registers the movement in the information system. The movement of jobs through the shop is not taken into account when rescheduling, because no pre-emption is allowed. Lastly, when an operator is absent, a process that reports the absence of an operator is initiated and the state of the machine that is used by that specific operator changes to ‘absent operator’. All three uncertainties change the state of the information system from ‘undisturbed’ to ‘disturbed’.

Such a state change results in a notification of a disruption being triggered and sent to a mobile client. The mobile client can be accessed by the manager of the shop-floor, who then handles the notification. If the manager selects the reschedule option within the mobile client, a rescheduling process is initiated. This process consumes information from the integrated information system, and enables a cloud-based simulation model to generate simulation scenarios. A record of the performance of each simulation scenario is kept, from which a new schedule is selected. The new schedule can be either acceptable or non-acceptable. If the new schedule is acceptable, it triggers a process that sends the schedule to the mobile client (which is accessed by the manager) and it then does a system update. The system updating process then changes the state of the information system to ‘undisturbed’, and updates the sequence of the jobs to be processed.

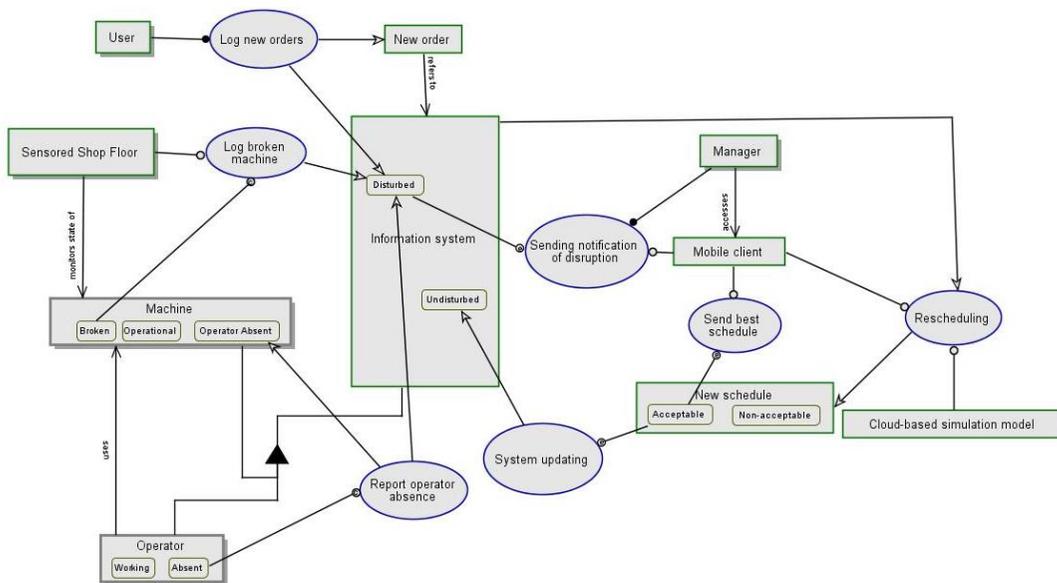


Figure 3: Top-level architecture of proposed scheduler

The semantic representation of the architecture is as follows:

User is physical.
User handles Log new orders.
New order refers to Information system.
Information system can be Undisturbed by default or Disturbed.
Information system consists of many Machines and many Operators.
 Machine is physical.
 Machine can be Operational by default, Broken, or Operator Absent.
 Operator is physical.
 Operator can be Working by default or Absent.
 Operator uses Machine.
Information system triggers Sending notification of disruption when it enters Disturbed.
Sensored Shop Floor is physical.
Sensored Shop Floor monitors state of Machine.
Manager is physical.
Manager accesses Mobile client.
Manager handles Sending notification of disruption.
New schedule can be Acceptable or Non-acceptable.
New schedule triggers Send best schedule and System updating when it enters Acceptable.
Log new orders yields Disturbed Information system and New order.
Log broken machine occurs if Machine is Broken.
Log broken machine requires Sensored Shop Floor.
Log broken machine yields Disturbed Information system.
Report operator absence occurs if Operator is Absent.
Report operator absence yields Disturbed Information system and Operator Absent Machine.
Sending notification of disruption requires Mobile client and Disturbed Information system.
Rescheduling requires Cloud-based simulation model and Mobile client.
Rescheduling consumes Information system.
Rescheduling yields New schedule.
Send best schedule requires Mobile client and Acceptable New schedule.
System updating requires Acceptable New schedule.
System updating yields Undisturbed Information system.

7 CURRENT STATE OF THE PROJECT

The current state of the project includes the development of a simulation model to illustrate the shop floor of a job-shop. The developed shop floor is based on the job-shop simulation model created by Bangsow [21], using Tecnomatix Plant Simulation. The base model used by Bangsow consisted of four turning machines, two milling machines, two gear milling machines, one induction hardening machine, and five grinding machines. The model also made use of four elements: work plan management, order management, resource management, and production control. These elements are defined by Bangsow as follows:

1. The work plan contains the sequence of operations for each job, the corresponding processing times, and the required resource for processing the operation. The sources of the information in practice are normally Microsoft Excel workbooks or database tables.
2. The order management element provides the instruction to process a certain number of jobs up to a certain date.
3. The resource management element contains all the resources or machines that are present in the real-world job-shop. All the information about the machines is stored in this element. The information can then be used to assign jobs to the appropriate machines.
4. The production control element contains the processing times on the machines, information about the loading and unloading of machines, and finding and moving the jobs to the next machine as described in the work plan.

The model of Bangsow was used and studied by the authors to understand better the required elements for modelling the job-shop problem in Tecnomatix Plant Simulation. The model developed by the authors is illustrated in **Error! Reference source not found.**, and contains five manufacturing cells: turning, grinding, milling, gear milling, and induction hardening. The turning, grinding, and

milling cells contain two machines each, while the other cells only contain one machine each. Furthermore, the model contains a resource management element that includes the information of all the machines in the model and the corresponding buffer locations. An example of this is provided in Table 2.

Table 2: Resource management element

Machine name	Buffer
Turning machine	Turning queue
Milling machine	Milling queue
Grinding machine	Grinding queue
Induction hardening machine	Induction hardening queue
Gear milling machine	Gear milling queue

The model also contains the work plan of the jobs. The work plan includes the jobs to be scheduled and the sequence of operations to complete the processing of a job. An example of a typical job and its required sequence of operations is illustrated in Table 3. In this example, the job consists of three operations: turning, hardening, and finally, grinding. The processing time in the developed model takes the set-up, processing, and dismantle time of the operation into account, and is displayed in minutes and seconds. This was decided before modelling started for simplicity. The processing times and the required machines for each operation are also defined, as seen in Table 3. The data in the work plan is then used to generate the dynamic schedules or sequence of jobs to be processed for each machine.

Table 3: Example of the sequence of operations of a typical job

Operation	Machine	Processing time
1	Turning machine	5:00
2	Induction hardening machine	10:00
3	Grinding machine	2:00

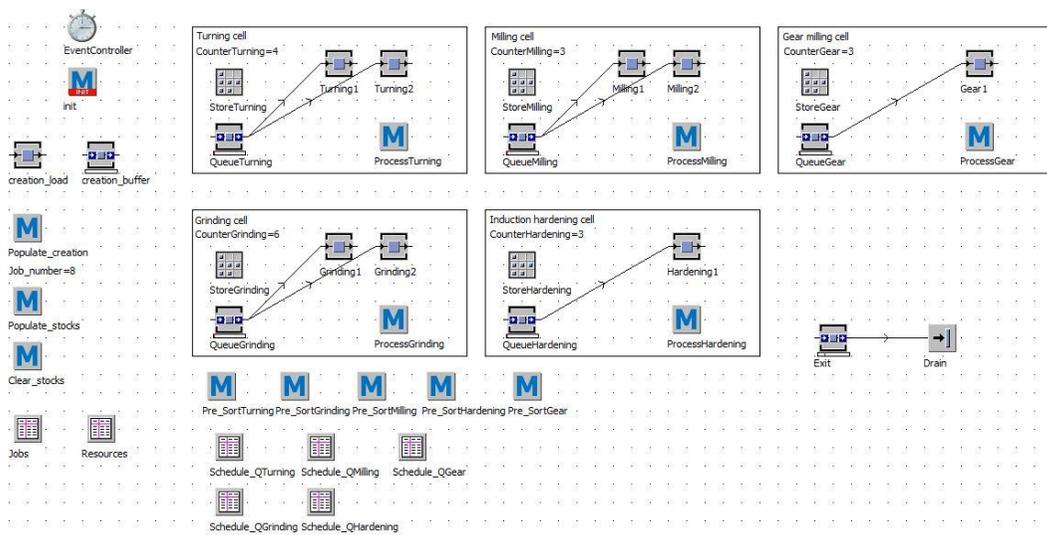


Figure 4: Job-shop model as developed in Tecnomatix Plant Simulation

8 FUTURE WORK

This section serves to discuss the future work required to complete the project successfully. The next step in the development of the software is to sensorise the simulation model. This must be done by developing an information system that hosts the real-time data of the job-shop. This step also includes incorporating sensors that generate the real-time data and store it in the information system. The information system must then be linked to the simulation model to provide the

simulation model with real-time data. Thereafter, the developed software must be put on the cloud to enable the software to be accessed and used at any location with internet connectivity. This will enable the users of the system (e.g., the shop floor manager) to use the software without being at the job-shop.

The final step in the software development requires the development of a mobile client that should be linked to both the information system and the simulation model. The mobile client should be able to receiving notifications from the information system if disruptions occur. This notification must then be handled by the shop manager. It will also initiate the rescheduling process if the manager selects the rescheduling option. The mobile client will inform the simulation model to determine a new schedule. After the development of the proposed software, the system must be tested and validated to determine the usefulness and efficiency of the system.

9 CONCLUSION

This paper describes part of an ongoing research project with the objective of developing a prototype of a cloud-based real-time simulation scheduling system. The architecture for the proposed system, which incorporates a sensorised job-shop floor, an information system, a mobile client, and cloud computing, was presented. A simulation model was created to model the environment and activities in a job-shop. The next step is to continue the development of the proposed software, and put the scheduling system on the cloud. The idea of simulation-based scheduling is not a novel one, but we are striving to develop a cloud-based simulation scheduling system that can be accessed by a mobile client. In doing so, we believe we shall make a small contribution to the overall effort of industrial engineers to take up stewardship and thus advance the economy.

REFERENCES

- [1] Groover, M.P. 2013. *Principles of modern manufacturing*, 5th edition, US: John Wiley and Sons.
- [2] Mitsuishi, M., Ueda, K. & Kimura, F. 2008. *Manufacturing systems and technologies for the new frontier: The 41st CIRP Conference on Manufacturing Systems May 26-28, 2008, Tokyo, Japan*. London: Springer.
- [3] Suwa, H. & Sandoh, H. 2012. *Online scheduling in manufacturing: A cumulative delay approach*. London: Springer.
- [4] Kim, M.H. & Kim, Y.D. 1994. Simulation-based real-time scheduling in a flexible manufacturing system. *Journal of Manufacturing Systems*, 13(2), pp. 85-93.
- [5] Stankovic, J.A., Spuri, M., Ramamritham, K. & Buttazzo, G. 2012. *Deadline scheduling for real-time systems: EDF and related algorithms*. US: Springer.
- [6] Murthy, P.R. 2005. *Production and operations management*. New Delhi: New Age International Publishers.
- [7] Miltenburg, J. 2005. *Manufacturing strategy: How to formulate and implement a winning plan, 2nd edition*. New York: Taylor & Francis.
- [8] Papadopoulos, C.T., O'Kelly, M.E.J., Vidalis, M.J. & Spinellis, D. 2009. *Analysis and design of discrete part production lines*. New York: Springer.
- [9] Wisner, J.D. 2016. *Operations management: A supply chain process approach*. California: SAGE Publications.
- [10] Dominic, P.D.D., Kaliyamoorthy, S. & Kumar, M.S. 2004. Efficient dispatching rules for dynamic job shop scheduling. *International Journal of Advanced Manufacturing Technology*, 24(1), pp. 70-75.
- [11] Pezzella, F., Morganti, G. & Ciaschetti, G. 2008. A genetic algorithm for the flexible job-shop scheduling problem, *Computers and Operations Research*, 35(10), pp. 3202-3212.
- [12] Vredevel, T. 2012. Stochastic online scheduling, *Computer Science – Research and Development*, 27(3), pp. 181-187.
- [13] Cai, X., Wu, X. & Zhou, X. 2014. *Optimal stochastic scheduling*. New York: Springer.
- [14] Pinedo, M.L. 2012. *Scheduling*, 5th edition. New York: Springer.
- [15] Kuhpfahl, J. 2015. *Job shop scheduling with consideration of due dates: Potentials of local search based solution techniques*. Germany: Springer.
- [16] Sadeh, N. 1991. *Look-ahead techniques for micro-opportunistic job shop scheduling*, Pennsylvania: Carnegie Mellon University.
- [17] Schriber, T.J., Brunner, D.T. & Smith, J.S. 2015. How discrete-event simulation software works and why it matters, *Proceedings of the 2015 Winter Simulation Conference*, pp. 1-15.
- [18] Herrmann, J.W. 2006. *Handbook of production scheduling*. New York: Springer.
- [19] Frantzen, M., Ng, A.H.C. & Moore, P. 2011. A simulation-based scheduling system for real-time optimization and decision making support. *Robotics and Computer-Integrated Manufacturing*, 27(4), pp. 696-705.
- [20] Dori, D. 2002. *Object-process methodology: A holistic systems paradigm*. New York: Springer.
- [21] Bangsow, S. 2015. *Tecnomatix Plant Simulation: Modeling and programming by means of examples*. Switzerland: Springer.