ENGINEERING CHANGE ORDER PROCESSING IN ERP SYSTEMS: AN INTEGRATED REACTIVE MODEL

B. SAENZ DE UGARTE [a], A. HAJJI [a], R. PELLERIN [a], A. ARTIBA [b]

[a] - Ecole Polytechnique de Montréal
Montréal (QC) Canada
benoit.saenz-de-ugarte@polymtl.ca
adnene.hajji.1@ens.etsmtl.ca
robert.pellerin@polymtl.ca

[b] - LISMMA Supméca
Paris, France
hakim.artiba@supmeca.fr

ABSTRACT: *Manufacturing Execution System (MES) provides a common and single system to support most of the manufacturing execution processes from the production order release to the delivery of finished goods. However, MES applications do not address all of the manufacturing execution processes required to replenish the supply chain while dynamically responding to unpredicted change and fall short of adding decision-making tools capabilities. To fill this gap, we propose a generic manufacturing execution platform where real-time optimization models complement the traditional ERP/MES model. The main objective of our architecture is to provide real-time decision support in reaction to disturbing events in manufacturing environments. More precisely, we demonstrate how this approach can support engineering change order (ECO) processing in ERP-controlled environments. Our demonstrative example based on a real scenario in the aerospace industry illustrates how a genetic algorithm and a real-time discrete event simulation model can be integrated within an ERP and MES system platform.*

KEYWORDS: *MES, ERP, simulation modelling, genetic algorithm, adaptive manufacturing, integration.*

1. INTRODUCTION

Enterprise Resource Planning (ERP) systems are designed to integrate all internal enterprise business processes. However, ERP systems still show a number of limitations for production management in dynamic and highly variable environments. First of all, ERP inherits two major shortcomings associated with its central MRP planning function: the unlimited capacity resources assumption and its non-stochastic nature (Hopp and Spearman, 2000). Moreover, ERP systems need additional external systems to monitor and collect real-time data. Consequently, the ERP system's inability to handle uncertainties and unexpected events limits its use for supporting decision making processes in dynamic production environment.

This limitation creates additional challenges in organizations producing highly complex products and where the engineering change management effort is important. Complex products such as automotives, aircraft, and major capital equipment and systems sometimes consist of thousands to hundreds of thousands parts. In addition, there are related to tooling, fixtures, gauges, templates, jigs & dies, test equipment, and software. Each part may undergo ten engineering changes or more over its life which suggests that a company may evaluate and process many thousands of engineering changes for a complex product. Over the product life cycle, the manufacturer must assure that the as-designed configuration at any point in time will satisfy functional requirements and that the product delivered corresponds to the approved as-designed configuration.

In order to support the engineering change management process, ERP systems must operate in coordination with the engineering and product development systems. Formal change management ensure that changes to products are captured, authorized, tracked and communicated throughout the enterprise and the entire life cycle of the product. When a change is being evaluated, the following must be considered:

- **Inventory status**: How many items are in inventory? Must they be scrapped or can they be used on other products or modified? What is the cost to modify or scrap?
- **Production status**: How many items are in work-in-process? Can they be reworked to the new configuration considering their current stage of completion, completed and used up before the change is effective? Must they be scrapped? What is the lead-time and cost for production of the new item? What is the additional lead-time for building tooling, fixtures and test equipment?
- **Procurement status**: Is the old item on order? Can it be cancelled and at what cost? What is the lead-time for procuring the new item? Are new suppliers required?
- **Impact on the customers and field service organizations**: What notification is required? How long will the process take? What documentation needs to be updated? What are the implications on spare parts requirements?

In all cases, real-time information is critical to properly evaluate the impact of any design changes. In that context, it is not surprising to note that production departments have always favoured the development of custom-made production data collection systems using either databases or spreadsheets to monitor and control real-time and variable execution processes (MESA, 1997). Software maintenance and data consolidation is obviously complex in such environment as the number and the structure of these small applications vary over time.
In this paper, we propose a real-time execution system to support engineering change order (ECO) processing in ERP-controlled environments. The main objective of our architecture is to complement the traditional ERP/MES model by providing real-time flexible and well-integrated decision support in reaction to disturbing events in manufacturing environments.

Our paper is structured as follow. We will first present a brief literature review on MES systems, real-time simulation platforms and genetic algorithms used in manufacturing context. The proposed real-time execution architecture will then be presented in Section 3. Section 4 will presents the demonstrative example. A brief discussion of future perspectives will conclude this paper.

2. LITERATURE REVIEW

For more than 25 years, companies have invested in information systems to achieve productivity gains which brought the enterprise information system market to grow steadily. But for most of that period, the information system specialists did not pay attention to the shop floor (Holst, 2001). In that context, it is not surprising to note that production departments have always favoured the development of custom-made software applications to fill specific necessities as an industrial operations support. Fortunately, the difficulty of integrating multiple point systems has brought software providers to package multiple execution management components into single and integrated solutions. These systems, commonly referred as Manufacturing Execution System (MES), provide a common user interface and data management system. MES functionalities are typically broad and can support the production operations management from point of order release into manufacturing to point of product delivery into finished goods. Using current and accurate data, MES can guide, trigger and reports on plant activities as events occur (MESA, 1997). However, MES applications do not address all of the manufacturing execution processes required to replenish the supply chain while dynamically responding to unpredicted change (Homem-de-Mello *et al.*, 1999, Lobecke and Slawinski, 2004). Reactive manufacturing strategies also require defining the process management layer between ERP and process control and translating demand driven supply chain requirements into a set of capabilities, systems, and workflow integration investments. Therefore, adaptive manufacturing organizations need an execution platform that connect manufacturing processes with engineering and supply chain processes, that enables closed-loop mechanisms, and that provides decision support to production personnel so they can deliver on their performance goals.

Up to now, the scientific literature has mostly address these issues by proposing different integration models between ERP and MES systems (Barry *et al.*, 1998, SEMATECH Inc., 1998, Hori *et al.*, 1999, PABADIS Group, 2001, Huang, 2002, Simao *et al.*, 2006). While these models may resolve some interoperability problems, they fall short of adding decision-making tools capabilities to both systems or to gain insight into manufacturing systems.

In some cases, traditional simulation models may be used off-line to either understand the system behaviour or to evaluate various strategies for the system operation (Moon and Phatak, 2005). However, simulation models are usually not built for repetitive and real-time usage as the models are not directly coupled with the real information systems (Drake and Smith, 1996). By contrast, on-line decision support systems take a snapshot of the actual factory status and predict future events according to certain modelling assumptions. In a rapidly changing and highly competitive business environment, the decision system therefore needs to have access to critical real-time production information. On-line systems also need to combine different solving tools with multiple data sources to elaborate solutions in reasonable time compatible with the planning horizon.

The advances in computing power and memory over the -last decades have opened up the possibility of online simulation based optimization. This recent research development offers one of the most interesting opportunities in simulation and the potential benefits in this field are significant. A general overview of the different approaches found in the literature including references to the state of the art is provided in (Scott, 2005). A typical area of application for online simulation, also called real-time simulation, is proactive decision support for scheduling problems in manufacturing systems (Siemiatkowski and Przybylski, 2006). Kouiss and Pierreval (Kouiss and Pierreval, 1999) present a combination of a MES and an on-line simulation model for controlling a flexible manufacturing system. Simulation is used to evaluate different scenarios proposed by the decision module which takes into account the actual state of the manufacturing system. The selected solution is proposed to a human operator for validation and execution. Shin et al. (Shin *et al.*, 2004) also discuss a decision problem faced by assembly line personnel when a breakdown occurs in a multiple production line system. A discrete-event simulation based approach is presented to help line personnel to minimize throughput degradation of the line.

In terms of resolution approach, the gradient based search methods and the heuristic methods are the most encountered on-line simulation based optimization approaches (Carson and Maria, 1997). Gradient based search methods cover finite difference estimation, likelihood ratio estimation, perturbation analysis and frequency domain experiments. These methods aim to estimate the retained performance measure with respect to the decision variables. In the other hand, heuristic methods consist on a random exploration of the admissible solutions in the whole decisions space. The search process ends when the best solution is found. At each point of the search process, the objective function value of the problem is estimated via the simulation model. Thus, no information regarding the analytic form of the objective function is required. This category covers simplex search (Azadivar and Lee, 1998), tabu search, simulated annealing (Ogbu and Smith, 1990, Lee and Iwata, 1991) and genetic algorithms. It is interesting to note that previous researches and survey (Azadivar and Tompkins, 1999, Chaudhry and Luo, 2005, Ruiz *et al.*, 2007) have demonstrated the effectiveness of Genetic algorithms solutions.

In summary, integration between ERP and MES systems is not sufficient to guaranty that organizations will be capable of quickly responding to events as they occur on the shop floor. One of the key differences between off-line simulation model and on-line decision support systems is data. It is why the decision system needs to have access to critical real-time production information. On-line decision support systems also need to combine different solving tools to be flexible. With those objectives in mind, we propose in the following section a generic manufacturing execution platform where a genetic algorithm integrated with a real-time discrete event simulation models complement the traditional ERP/MES model.

3. PROPOSED EXECUTION ARCHITECTURE

It is usually difficult to calculate the impact of a decision on a complex system. In this context, using on-line simulation is attractive but requires the integration of the simulation model and the optimization engine into the enterprise information system landscape and the communication network. It must also combine multiple optimization techniques in order to support different decision-making scenarios.

Our proposed architecture is presented in Figure 1. The main objective of our architecture is to provide a flexible and integrated real-time decision support in reaction to disturbing events in manufacturing environments. Events can occur from the top level – a new order – or the down level – a breakdown on the shop floor. The different software components are spread over several servers linked by local area networks and/or global network. The connection to the real manufacturing processes is supported by OPC (OLE for Process Control) clients connected to different OPC servers collecting data directly from the process. These clients take in charge raw data acquisition, preliminary storing, and processing. Depending on storing/processing rules, an OPC client can store processed data in an SQL database, exchange the data via XML clients, and generate messages to other components if significant events occur. SQL/XML clients or servers assure data exchange with external software, such as ERP, Web servers, Supervisors etc. They can also generate messages to other system components on the basis of their configuration.

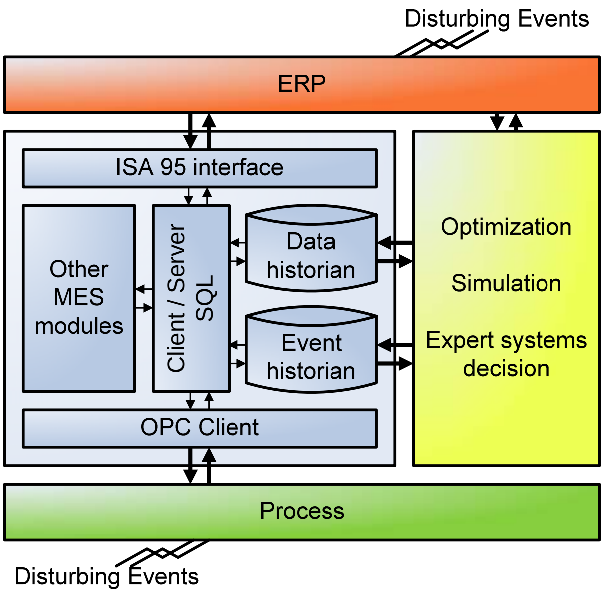

Figure 1. Real time execution architecture

The decision-making nucleus of our system is a hybrid of a discrete-event simulator and optimization tool. The simulator runs models of different parts from the controlled system and responds to incoming messages by initiating decision-making logic depending on the disturbing event and the current system state. At the beginning of the decision making process, the decision-making nucleus is initialized with data from the ERP (i.e. production order, production sequence, etc.) and from the MES (i.e. order status, machine status, etc.). Simulation is used like a meta-heuristic support to evaluate performance indicators under a set of constraints or production rules. In its turn, it sends messages representing decisions made for other actors of the architecture including OPC clients to act on the process controlled.

4. CASE STUDY

In the following example, we illustrate how genetic algorithm and a real-time discrete event simulation model can be integrated within the ERP and MES applications. Readers should note that this demonstrative example is based on a real scenario from the aerospace industry.

4.1. ECO production process

When a design problem is identified by a customer or by an air safety bureau (like the Air Safety Investigation Office in Canada or the National Transportation Safety Board in USA), the equipment manufacturer needs to take corrective actions across multiple departments (engineering, supply, manufacturing, services…) and systems (PLM, ERP, SCM, ...). An engineering change request can impact the product design, the process design, the procurement process, the manufacturing operations, and the in-service operations. In the demonstrative example, we address the case where the engineering change request triggers an immediate part replacement and ignore the product and process design revisions. As the part to be replaced can be temporary changed by a similar part, the reader will understand that the scenario is not concerned with safety or regulatory issues.

Once the engineering change request triggers an immediate part replacement action, the manufacturer first needs to check if the item is already in inventory or in work-in-process. If it is not the case, this item has to be produced as soon as possible. In our example, tooling availability is the main constraint and the planner objective is to schedule and release a new production order. The scheduling objective is to minimize the replacement part lead time while minimizing the total make span.

4.2. Systems architecture

The reactive process, based on the proposed real-time execution architecture, is presented in Figure 2. In this scenario, the engineering support team first creates an engineering change request (ECR) in the ERP system. After the completion of a preliminary impact analysis, the ECR is converted into a formal engineering change order (ECO). The reactive scheduling process is then initiated when the ECO status is updated to a special status asking for a replacement part. The Event Manager automatically sends an alert to the planner who will use the MES to check the inventory and the work-in-process. If needed, a production order will be created in the ERP system. The production planner then determines the tooling and material needed for the newly created production order and then reschedules all orders through the on-line decision support system. In this case, the allotted time period for decision making is sufficient to run an efficient schedule algorithm.

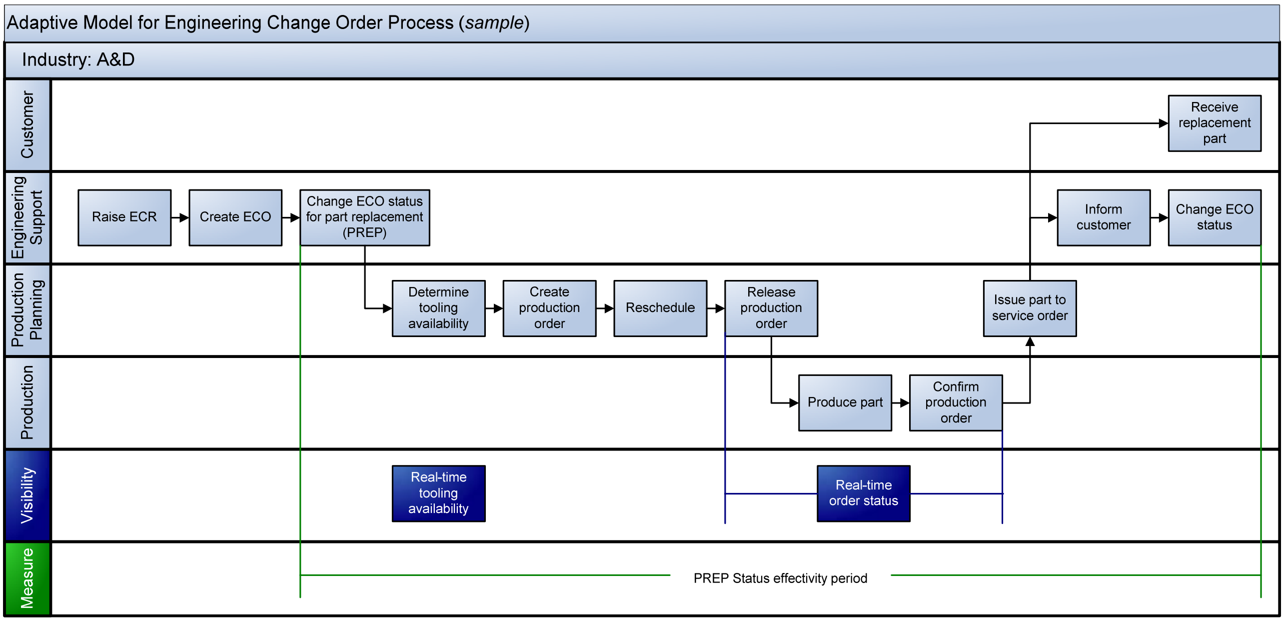

Figure 2. ECO scenario adaptive process

Production orders are released according to this new schedule. When the production is confirmed, an alert is sent to the planner and the replacement part can be shipped to the customer as soon as the good issue transaction is posted.

Figure 3 also presents the decision model process workflow within the proposed execution system architecture. In this example, the execution system is composed of 3 main components: the ERP system (SAP R/3), the MES system (XMII), and the simulation/decision support system (figured by DSS).

4.3. Optimization model

In this scenario, tooling availability is the main constraint. The tooling availability periods are set a long time in advance and can not be renegotiated. In the study case, 30 jobs are in the system with 3 operations by job, each operation used one of the 4 available tools. Operation *i* is always processed on the machine *i*.

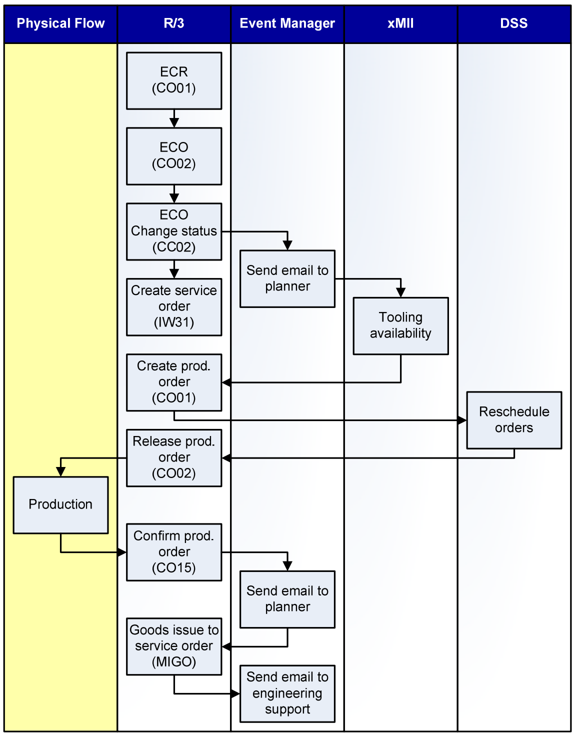

Figure 3. ECO scenario system workflow

Processing times and routings are given in Table 1 and tooling availability periods in Table 2. ECO *E** occurs in the system when the job #10 of the initial sequence S_0 begin to be processed. The decision support system tests 3 scenarios (Figure 4) and the final choice is made by the planner:

- E* is the first of the to do sequence, the 20 other orders sequence is kept like in S0 (this solution is actually always used by the company);
- E* is the first of the to do sequence, the 20 other orders sequence are rescheduled (objective : minimize the total make span);
- The sequence of the 21 orders are rescheduled (objective : minimize the total make span).

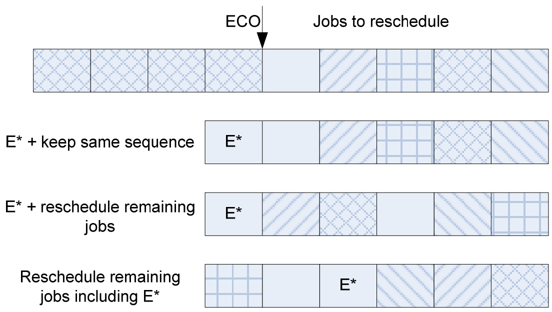

Figure 4. ECO scenario system workflow

Job	$(d_1,t_1)(d_2,t_2)(d_3,t_3)$	Job	$(d_1,t_1)(d_2,t_2)(d_3,t_3)$
1	(54,3)(34,3)(61,3)	16	(11,3)(14,1)(89,3)
2	(2,4)(9,3)(15,4)	17	(33,3)(62,4)(87,4)
3	(89,3)(70,4)(38,1)	18	(38,4)(79,2)(15,3)
4	(19,1)(28,2)(87,1)	19	(28,4)(30,4)(12,1)
5	(95,1)(34,2)(7,1)	20	(30,1)(69,3)(71,2)
6	(29,3)(79,1)(25,3)	21	(98,4)(74,4)(96,1)
7	(59,2)(95,2)(39,4)	22	(53,2)(80,2)(42,3)
8	(66,1)(89,2)(91,2)	23	(51,1)(14,3)(53,4)
9	(53,4)(1,3)(14,3)	24	(93,2)(43,4)(21,1)
10	(66,1)(50,3)(99,2)	25	(58,3)(12,2)(99,3)
11	(46,3)(11,2)(57,4)	26	(19,4)(57,4)(67,2)
12	(68,3)(67,3)(11,1)	27	(54,1)(83,3)(73,3)
13	(35,1)(51,2)(5,1)	28	(93,4)(50,3)(3,1)
14	(46,1)(27,1)(46,3)	29	(49,3)(70,1)(73,1)
15	(28,1)(40,2)(64,1)	30	(57,1)(21,1)(82,4)

Table 1. Processing time and routings (d_i: operation *i* processing time, t_i: operation *i* tooling)

Tooling 1		Tooling 2		Tooling 3		Tooling 4	
open	close	open	close	open	close	open	close
80	205	0	155	0	55	0	26
245	448	192	439	111	644	39	238
469	773	547	997	656	980	254	502
825	889	1036	1341	1041	1523	547	626
973	1509	1379	1617	1543	1947	660	710
1583	1863	1651	2000			765	896
1939	2000					951	954
						1038	1264
						1301	1727
						1750	1850
						1867	2000

Table 2. Tooling availability periods
(load / available capacity on 2000 time units = 67%)

4.3.1.Genetic algorithm

The search process is detailed in figure 5. It consists on running the Genetic Algorithm (GA) with respect to its stopping rule and evaluates each desired configuration through the simulation model.

In terms of an optimization problem, the genetic algorithm approach is summarized as follows. At any given point in time, the genetic algorithm generates a population of possible candidate solutions. Initially, the population size is chosen randomly. However this choice typically depends on the characteristics of the problem. Each population component is a string entity of chromosome which represents a possible solution to the problem. The population components are evaluated based on a given fitness function. Highly fit population components are given the chance to reproduce through a crossover process with other highly fit population elements by exchanging pieces of their genetic information. This process produces «offspring» or new solutions to the optimization problem based upon the high-performance characteristics of the parents. Premature loss of important information by randomly altering bits within a chromosome is prevented by a mutation process. This procedure continues until a continues until a satisfactory solution is achieved (Legault, 1994, Chaudhry and Luo, 2005).

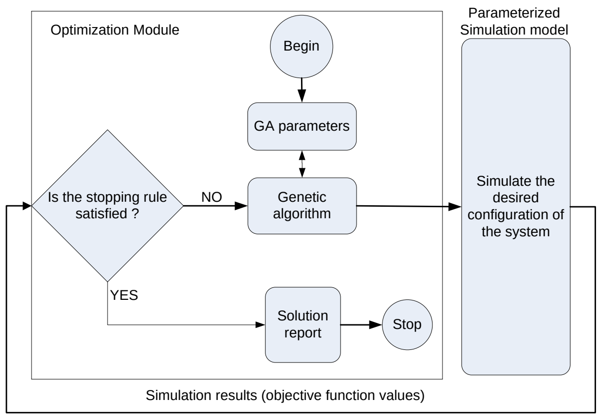

Figure 4. Simulation / GA optimization module

To solve the problem under study, we have developed a GA program based on an existing Toolbox (Chipperfield *et al.*, 1994). The main data structures in the GA toolbox are chromosomes, phenotypes, objective function values and fitness values. The chromosome structure stores an entire population in a single matrix of size $N_{ind} \times L_{ind}$, where, N_{ind} is the number of individuals and L_{ind} is the length of the chromosome structure. Phenotypes are stored in a matrix of dimension $N_{ind} \times N_{\text{var}}$ where, N_{var} is the number of decision variables. A $N_{ind} \times N_{obj}$ matrix stores the objective function values, where N_{obj} is the number of objectives. Finally, the fitness values are stored in a vector of length N_{ind}. In all of these data structures, each row corresponds to a particular individual.

The GA toolbox uses MATLAB matrix functions to build a set of versatile routines for implementing a wide range of genetic algorithm methods. The following steps summarize the employed Genetic Algorithm:

1- Population representation and initialisation: permutation coding representation with « N_{ind} » the number of individuals and $L_{ind} = 30$ the length of the chromosome.
2- Fitness: the linear-**ranking** method of Baker (Baker, 1985).
3- Selection: roulette wheel selection (Goldberg, 1989)(routine rws)
4- Crossover: alternate edges crossover (Poon and Carter, 1995) with crossover probability « Pc », a crossover method for ordered chromosomes. It recombines pairs of individuals with given probability to produce offspring with respect to the existing jobs.
5- Mutation: shift mutation (Chang *et al.*, 2007) with probability P_m. A mutation method for ordered chromosomes.

Let « MaxGen » be the maximum number of generation if the stopping algorithm rule is fixed following this criteria.

4.3.2.Simulation model

The simulation model is build to describe the dynamics of the system governed by the scheduling sequence and the tooling constraints defined previously (section 4.3). These factors are considered as input of such a model and the related total make span is defined as its output. The parameterized simulation model is developed using the Visual SLAM language (Pritsker *et al.*, 1997).

The Visual SLAM portion is composed of various networks describing specific tasks (tooling constraints, sequence, data exchange with Genetic algorithm, etc...). The simulation ends when we reach the end of the defined production sequence. Figure 5 shows a bloc diagram representation of the simulation model.

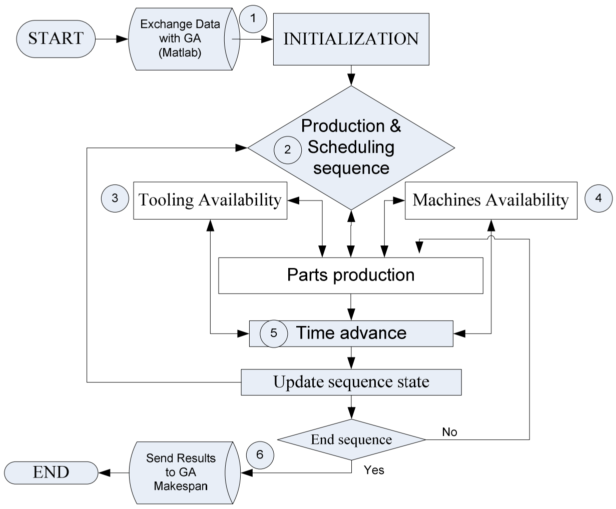

Figure 5. Simulation model bloc diagram

1) The Exchange data block read the parameters of each individual of the population set by the Genetic Algorithm. To run the model the INITIALIZATION block sets these values and other parameters defining the system (e.g., the tooling constraints, the machines availabilities, the processing times and routings,...) as well as the simulation stopping criteria.

2) The production and scheduling sequence mechanism set the processing time and the part type to produce according to the generated sequence. This block is in connection with the «update sequence state» block charged to send a signal when a given part reaches the end of the process. The scheduling sequence is then activated to launch the following part.

3) The tooling availability bloc set the availability periods of tooling 1 to 4 according to the cycle defined in table 2. This bloc is in connection with the « parts production » and the « time advance » blocs. In fact, when a tooling is being used it becomes unavailable for other jobs. Moreover, when the close time defined by table 2 is reached the tooling should be pre-empted.

4) The machines availability block samples the times to failure and times to repair for the machines from their respective probability distributions. This bloc is in connection with the « parts production » and the « time advance » blocs. In fact, when a machine is being repaired it becomes unavailable for other jobs. According to the time advance the available / unavailable states are defined by a two well defined signals.

5) The time advance block follows the system events progress and governs the tooling and machines availabilities blocks.

6) The Send Results block writes the incurred total makespan of each individual of the genetic algorithm population in an external file. This file being available to the genetic algorithm program reiterates the optimization process.

4.4. Validation

It is extremely difficult to compare the performance of different evolutionary algorithms since most researchers use their own instances of test problems, i.e., problems where the processing times and due dates of the jobs are selected randomly out of a uniform distribution. To validate our model, we use a standard benchmark problem taken from Taillard (Taillard, 1993). This flow shop instance has 20 jobs and 5 machines, uses the time seed 379008056. A calculated lower bound is 1290 and the best make span found by Taillard is 1359. With our model we found 1360 after 100 iterations and with Pc equal to 0.6. Being very close to Taillard results we decided to go further with the developed genetic algorithm and the problem under study. It is interesting to note that when dealing with genetic algorithms, the choice of the GA parameters (i.e., Pc, Pm,..) is an important issue to be taken into account since it can affect the optimization process and the final results. In the research literature the choice of these parameters is generally based on experience.

4.5. Results – discussion – extension

The system parameters and ECO data used to run the optimization module and to characterize the best scheduling / rescheduling scenario are given in table 1, 2 and 3, respectively.

Job	$(d_1,t_1)(d_2,t_2)(d_3,t_3)$
ECO	(57,1)(21,1)(82,4)

Table 3. ECO Processing time and routing

The obtained results are given in figure 6. When the same sequence before ECO is kept, the insertion of E* in the first position leads to a total make span of 3621 for the 21 orders. This is the solution actually used by the company. When the decision to reschedule the remaining jobs including ECO is taken, the total make-span is reduced to up 13.6 %. It is interesting to note that under the considered case and with respect to the optimization objective, the difference between the second and the third scenario (see figure 6) is not significant. In the next section, it will be shown that the gap between the two results will be amplified under more constrained configurations.

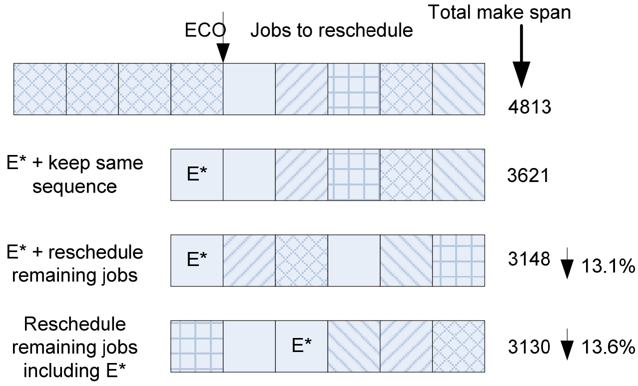

Figure 6. obtained results

To illustrate the effect that some considered system parameters variation have on the solutions, a sensitivity analysis was conducted. Tables 4, 5 and 7 detail he considered parameters variations, and present the obtained total make-span for the sensitivity analysis cases.

Due to the number of the involved parameters we decided to limit our analysis to the tooling constraints and the ECO parameters. These parameters could be considered as illustrative in our context. Moreover, our objective is to insure the robustness of the approach and the proposed solutions. We claim that, at this point, this objective is reached and it will be reinforced with this sensitivity analysis.

Tooling 1		Tooling 2		Tooling 3		Tooling 4	
open	close	open	close	open	close	open	close
80	152	0	114	13	158	0	96
192	309	177	287	174	355	158	162
330	504	335	435	400	458	211	217
556	593	518	804	492	528	313	501
677	983	813	987	583	678	512	672
1057	1218	1079	1124	733	735	728	886
1294	1630	1180	1367	819	984	983	1086
1657	1772	1450	1519	1021	1332	1121	1172
1817	2000	1589	1840	1355	1428	1255	1496
		1907	2000	1445	1808	1560	1756
				1828	2000	1856	1860
						1927	2000

Table 4. Tooling availability periods for the 2^{nd} run (load / available capacity on 2000 time units = 77%)

Run	Job	$(d_1,t_1)(d_2,t_2)(d_3,t_3)$
2nd	ECO	(47,1)(53,3)(44,4)
3rd	ECO	(47,3)(53,4)(44,1)

Table 5. ECO Processing time and routing

	Total make span	Variation
30 jobs sequence	5647	
E* + keep same sequence	4660	
E* + reschedule remaining jobs	3984	↓ 14,51%
Reschedule remaining jobs including E*	3913	↓ 16,05%

Table 6. Obtained results for the 2nd run

Tooling 1		Tooling 2		Tooling 3		Tooling 4	
open	close	open	close	open	close	open	close
155	227	67	211	0	222	165	294
304	421	405	663	371	445	366	610
462	636	733	909	570	872	655	713
737	774	977	1115	992	1011	746	1030
936	1242	1176	1405	1037	1182	1069	1218
1386	1547	1593	1628	1214	1395	1322	1418
1694	2000	1740	1993	1485	1543	1540	1544
				1610	1646	1640	1646
				1754	1849	1834	2000
				1958	2000		

Table 7. Tooling availability periods for the 3rd run
(load / available capacity on 2000 time units = 96%)

	Total make span	Variation
30 jobs sequence	7044	
E* + keep same sequence	6485	
E* + reschedule remaining jobs	4730	↓ 27,06%
Reschedule remaining jobs including E*	4537	↓ 30,04%

Table 8. Obtained results for the 3d run

The obtained results are shown in table 6 and 8 for the second and third run respectively. Following our expectations, the results show that facing a more constrained system (i.e., tooling availability and ECO routing) the total make-span is increasing. Moreover, the gap between the actual practice (i.e., insert ECO at the beginning and keep the same schedule) and the two other solutions is increasing. These results point toward a considerable benefit for the company. In this context, to minimize the total make-span, it is always more interesting to reschedule the remaining jobs including E*. However, based on the due date of the ECO the planner should make a multi objective choice (i.e., minimize total make-span and deliver ECO as soon as possible).

5. CONCLUSION

This paper presents our contribution to the development of an open architecture framework supporting real-time decision-making. The proposed real-time control system is integrating optimisation and simulation methods with ERP and MES systems. The functionalities and the interactions between the different components of the real-time execution system were presented. A demonstrative example also illustrated the operating mode of the control system functional architecture. This real life based application highlighted the integration aspects of the different control system modules and their usefulness to solve real-time manufacturing decision-making problems. Other scenarios will be studied in the future to demonstrate the generic aspect of our approach in solving real-time decision-making problems of adaptive manufacturing systems. Future works will also involve the integration of other optimisation techniques.

It is interesting to note that we have only considered a production shop floor without external constraint. In fact, the original case consists on a production shop floor which supplies an assembly line. This configuration set additional constraints on the production shop floor schedule. In this context, the production shop floor produces items on stock and on demand. When an ECO occurs, the planner has to re-schedule the shop floor with a minimum impact on the assembly line schedule while always minimizing the total make-span. This case study will be investigated in further research.

REFERENCES

Azadivar, F. & Lee, Y.-H., 1998. Optimization of discrete variable stochastic systems by computer simulation. *Mathematics and Computers in Simulation*, 30(4), 331-345.

Azadivar, F. & Tompkins, G., 1999. Simulation optimization with qualitative variables and structural model changes: A genetic algorithm approach. *European Journal of Operational Research*, 113(1), 169-182.

Baker, J.E., 1985. Adaptive selection methods for genetic algorithms. *Proceedings of the 1st International Conference on Genetic Algorithms*, 101-111.

Barry, J., Aparicio, M., Durniak, T., Herman, P., Karuturi, J., Woods, C., Gilman, C., Lam, H. & Ramnath, R., 1998. NIIIP-SMART: an investigation of distributed object approaches to support MES development and deployment in a virtual enterprise. *Proceedings of Second International Enterprise Distributed Object Computing Workshop, 1998. EDOC '98*, 366-377.

Carson, Y. & Maria, A., 1997. Simulation optimization: Methods and applications. *Proceedings of the 1997 Winter Simulation Conference*, Atlanta, GA: IEEE Piscataway NJ USA, 118-126.

Chang, P.-C., Hsieh, J.-C. & Wang, C.-Y., 2007. Adaptive multi-objective genetic algorithms for scheduling of drilling operation in printed circuit of drilling operation in printed circuit board industry. *Applied Soft Computing Journal,* 7(3)**,** 800-806.

Chaudhry, S.S. & Luo, W., 2005. Application of genetic algorithms in production and operations management: A review. *International Journal of Production Research,* 43(19)**,** 4083-4101.

Chipperfield, A.J., Fleming, P.J. & Fonseca, C.M., 1994. Genetic algorithm tools for control systems engineering. *Proceedings of Adaptive Computing in Engineering Design and Control*, Plymouth Engineering Design Center (UK), 128-133.

Drake, G.R. & Smith, J.S., 1996. Simulation system for real-time planning, scheduling, and control. *1996 Winter Simulation Conference.* Coronado, CA, USA: IEEE, Piscataway, NJ, USA, 1083-1090.

Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning* Boston, MA, USA: Addison-Wesley.

Holst, L., 2001. Integrating Discrete-Event Simulation into the Manufacturing System Development Process. Lund University, Department of Mechanical Engineering, Division of Robotics.

Homem-De-Mello, T., Shapiro, A. & Spearman, M.L., 1999. Finding optimal material release times using simulation-based optimization. *Management Science,* 45(1)**,** 86-101.

Hopp, W. & Spearman, M., 2000. *Factory Physics* New York , NY: McGraw-Hill.

Hori, M., Kawamura, T. & Okano, A., 1999. OpenMES: scalable manufacturing execution framework based on distributed object computing. *1999 IEEE International Conference on Systems, Man, and Cybernetics 'Human Communication and Cybernetics'*, Tokyo, Jpn: Institute of Electrical and Electronics Engineers Inc. USA, 398-403.

Huang, C.-Y., 2002. Distributed manufacturing execution systems: A workflow perspective. *Journal of Intelligent Manufacturing,* 13(6)**,** 485-497.

Kouiss, K. & Pierreval, H., 1999. Implementing an on-line simulation in a flexible manufacturing system. *Simulation in Industry'99. 11th European Simulation Symposium 1999. ESS'99.* Erlangen, Germany: SCS, 484-8.

Lee, Y.-H. & Iwata, K., 1991. Part ordering through simulation-optimization in an FMS. *International Journal of Production Research,* 29(7)**,** 1309-1323.

Legault, G., 1994. Un algorithme génétique pour la conception topologique de réseaux téléinformatiques à commutation de paquets. Master thesis. Université du Québec à Montréal.

Lobecke, R. & Slawinski, T., 2004. Integrated manufacturing execution systems - history and current state: from system integration to overall optimisation of business processes. *Automatisierungstechnische Praxis,* 46(1)**,** 21-5.

Mesa, 1997. *White Paper #06: MES Explained: A High Level Vision for Executives* [online]. MESA International. Available from: http://www.mesa.org/knowledge-base/details.php?id=53 [Accessed 5 october 2007].

Moon, Y.B. & Phatak, D., 2005. Enhancing ERP system's functionality with discrete event simulation. *Industrial Management and Data Systems,* 105(9)**,** 1206-1224.

Ogbu, F.A. & Smith, D.K., 1990. Application of the simulated annealing algorithm to the solution of the n/m/C_{max} flowshop problem. *Computers & Operations Research,* 17(3)**,** 243-253.

Pabadis Group, 2001. *Revolutionizing Plant Automation : the PABADIS Approach* [online]. Deliverable 6.3. Available from: http://www.pabadis.org/downloads/pabadis_del_6_3.pdf [Accessed 12 November 2006].

Poon, P.W. & Carter, J.N., 1995. Genetic algorithm crossover operators for ordering applications. *Computers & Operations Research,* 22(1)**,** 135-147.

Pritsker, A.A.B., O'reilly, J.J. & Laval, D.K., 1997. *Simulation with visual SLAM and AweSim* New York (NY): John Wiley & Sons, Inc.

Ruiz, R., Carlos Garcia-Diaz, J. & Maroto, C., 2007. Considering scheduling and preventive maintenance in the flowshop sequencing problem. *Computers and Operations Research,* 34(11)**,** 3314-3330.

Scott, D., 2005. Comparative advantage through manufacturing execution, and applications. *In*: R.G. Ingalls, M.D. Rossetti, J.S. Smith & B.A. Peters, (eds.). *Proceedings of the 2005 Winter Simulation Conference*, 83-95.

Sematech Inc., 1998. *Computer Integrated Manufacturing Framework Specification Version 2.0* Austin, TX.

Shin, F., Ram, B., Gupta, A., Yu, X. & Menassa, R., 2004. A decision tool for assembly line breakdown action. *2004 Winter Simulation Conference.* Washington, DC, United States: Institute of Electrical and Electronics Engineers Inc., New York, NY 10016-5997, United States, 1122-1127.

Siemiatkowski, M. & Przybylski, W., 2006. Simulation studies of process flow with in-line part inspection in machining cells. *Journal of Materials Processing Technology,* 171(1)**,** 27-34.

Simao, J.M., Stadzisz, P.C. & Morel, G., 2006. Manufacturing execution systems for customized production. *Journal of Materials Processing Technology,* 179(1-3)**,** 268-275.

Taillard, E., 1993. Benchmarks for basic scheduling problems. *European Journal of Operational Research,* 64(2)**,** 278-285.