



HAL
open science

Formal DEVS modelling and simulation of a Flow-Shop relocation method without interrupting the production

Patrick Pujo, Massimo Pedetti, Norbert Giambiasi

► To cite this version:

Patrick Pujo, Massimo Pedetti, Norbert Giambiasi. Formal DEVS modelling and simulation of a Flow-Shop relocation method without interrupting the production. *Simulation Modelling Practice and Theory*, 2006, 14, pp.817-842. hal-01212617

HAL Id: hal-01212617

<https://hal.science/hal-01212617>

Submitted on 14 Oct 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Formal DEVS modelling and simulation of a Flow-Shop relocation method without interrupting the production

Patrick Pujo¹, Massimo Pedetti, Norbert Giambiasi

*Research Group in Control and Simulation
Laboratoire des Sciences de l'Information et des Systèmes
UMR CNRS 6168
Avenue Escadrille Normandie Niemen, F-13397 Marseille cedex 20, France*

Abstract

This paper presents an organisational method to keep the production going during the removal of a flow-shop. Assume a flow-shop system is to be moved on a new site and its production has to continue, our method can be applied to follow requests for the removal period. The method works as follows: we segment the removals in groups of machines and move them one after the others. This method can be successfully executed provided that a prime condition is met: envisaging sufficient stocks' plug between each group. The role of the latter stocks is to ensure operations' production continuity between the old and the new site, when the non-operational group is being removed. Removal is then renewed until the whole line is moved and is operational on the new site. To validate this approach, we have used simulation and developed a model of the flow-shop according to coupled DEVS formalism. Our model enables to segment a production line. As a consequence, we can simulate the sequential displacement of machines' groups towards the new site. Among the solutions suggested, those starting with the final group (finished products) and while finishing with the first group are much more effective. In this paper, we present and discuss some simulation results of an industrial case study. The results demonstrate the compared effectiveness of various strategies of removal, and make possible for the industrialist to envisage a good estimated project management.

Keywords: Flow-shop relocation project management, Simulation, DEVS formalism, Facilities Relocation.

1. INTRODUCTION

The relocation of a production site is a difficult industrial project and many firms hesitate to get into this kind of trouble, especially small or medium sized companies that have to go

¹ Corresponding author. Email : patrick.pujo@lsis.org

on operating with the same machines. A simple solution consists in removal during paid holidays: it is then possible to close the firm for the duration of the removal process. When a firm works on orders, without any annual closing, it needs to be continuously present on the market. It also needs a great ability to react making such strategies not conceivable. Although similar situations are commonplace, this problem has not been dealt with yet. We can consider the case of relocating the current facilities of a company to a nearby site. Two companies that were faced with the problem of removing a line of machines from one site to another without interrupting the production twice approached us. The constraints of the removal project were the same for both companies:

- The production could not be stopped completely during the removal of the machines,
- The removal process was of flow-shop type,
- The budget of the removal operation was limited, which forbade any solution of flash removal type (removal the totality of the production system overnight or during a weekend),
- The global production process remained identical, i.e. logical chain of the operations remained the same (the same machines would continue to be used one after the other).

This unusual problem is not dealt with in reference books about production organization [2, 9] or facilities planning [26]. Even the fundamental book on methods of facilities layout [17] does not deal with the organization of removal faced with such constraints. Recent papers have focused on facilities relocation. In fact, these articles deal with location or layout strategies, by considering mathematical approaches [3, 12, 15, 5] or by proposing oriented management methods [24, 19, 20, 16].

However, the removal organization problem is not dealt with in these papers.

In this context, and as a consequence, we have elaborated an organizational method to keep production going during the removal of a flow-shop. We propose here a solution of progressive removal of the resources to the new site by segmenting the whole production line in sub-sets and building up a buffer stock for each elementary removal of a sub-set.

The true problem lies in the choice of the good parameters to balance the relocation organisation. A great number of solutions can be planned to segment the totality of the production line. The use of simulation in the field of manufacturing systems for simplifying the complexity and reducing the problem dimension is recognised by scientists and industrial managers. In our case, simulation aims at determining the best values combination of input parameters, given an output criterion [21].

In this paper, attention is focused on DEVS modelling for the relocation problem. The paper is structured as follows. Section 2 describes the relevance of the DEVS formalism for manufacturing systems' simulation. Section 3 presents theoretical considerations for an elementary removal of a production workstation. Section 4 is devoted to the description of the relocation organisation of the complete production line. We use a case study and different simulations to test various strategies about the removal organization of the complete line. Section 5 describes an industrial case of printed-circuit boards' production line, working on orders. The DEVS models are provided. Section 6 analyses simulation experiments' results that the company can use to choose the best relocation strategy.

2. DEVS APPLICATION FOR MANUFACTURING SYSTEMS SIMULATION

In this section, the manufacturing systems' simulation issue is presented. First, the general characteristics of industrial engineering problems are described. Then, the DEVS formalism concept is introduced and its advantage exposed.

2.1. The issue of manufacturing systems' simulation

Manufacturing systems are regarded as complex systems. As such, various problems arise in the comprehension of their operation, their dimensioning or the evaluation of their performance. These problems are seldom solved by an analytical approach, such as operations research approaches. Indeed, the latter produce models based on relations between various variables in order to seek an optimal configuration of their values. In fact there are two specific factors in production systems that prevent the use of operations research techniques. Firstly, production systems are discrete events systems. Time is one of the principal components in the search for the solution. The temporal constraints are expressed in the form of connected inequations. Their number is equal to the number of events. Secondly, these events are not perfectly known in advance: the events dates rise from various lengths of time duration. It is all about manufacture, transport and/or assembly tasks, which are only known with a quite relative precision. Moreover, unforeseen events can occur, in a random and perturbing way. This contingency of the events implies that a solution obtained by operations research techniques is often only a valid solution with respect to an idealized behavior of the production system. However, discrete events systems can be particularly well studied by simulation. Simulation enables to reproduce production systems' dynamic evolution according to the above-mentioned uncertainties (using stochastic modeling). The dynamic behavior of the system is then described according to its evolution logic: progress of state changes and events generations. This is of course contextual, i.e. the evolution depends above all on the present state. In fact, this evolution logic is examined at each occurrence of event, step by step. This enables to manage the evolution of future event lists. These events are then qualified by the endogenous ones, i.e. they are the consequence of system state changes. Other events can be exogenous; i.e. the external environment of the system generates them. Their occurrence can be described explicitly (such as a list of dates of external events) or implicitly (using any probability distribution). For all events types, it is the behavior of the considered entity in the real world, which enables to formalize the simulation model. An activity is defined, when the time duration has a specified length, i.e. 'know when it begins' (unconditional wait), and a delay, when the time duration has an unspecified or indefinite length, i.e. 'not know until it ends' (conditional wait).

Fishwick [7] lists the advantages of using simulation models and methodology for modelling manufacturing systems as follows:

- Simulation models are more simple to determine than analytic models,
- Realistic models (complex systems representation) can be construct, whereas analytic models require simplifying assumptions,
- Performance measures to estimate the system behavior are easy to obtain, and visual output gives a good assistance to the end-user,
- Independently of the real system, alternative experimentations can be quickly carried out.

These advantages imply that simulation is used in FMS [13], to improve its control [23].

2.2. Advantages of DEVS formalism applications

The Discret EVent system Specification (DEVS) formalism was developed by Zeigler to model and simulate discrete events dynamic systems [27]. This system analysis with unified concepts facilitates the concise description of discrete event systems. DEVS enables the development of robust model representation, based on atomic models concepts and on higher-level models representation closed under coupling. This enables a hierarchical modelling, where atomic models are considered as black boxes.

DEVS is totally independent from the implementation of simulators that use such system's modeling. Various DEVS applications can thus be quoted, in robotics, computer architectures, VLSI design, ecology modelling and analysis and environmental systems, amongst others. In transportation systems, Lee *et al.* [14] performed distributed homogeneous traffic simulation by extending developed DEVS-based I³D² transportation simulation system to an HLA-based distributed environment. Nidumolu *et al.* [18] suggest that DEVS could be applied to simulate and evaluate complex business process models. These two examples have studied connected domains to operations' management domain. In our laboratory, we have used DEVS to support the knowledge of control system behavior of iron-making blast furnace [8].

DEVS begin to be applied to industrial production systems. For example, Choi *et al.* [6] have developed a factory simulator, which can give virtual 3D representation of an automated manufacturing system by employing a job-resource relation network systematically converted to a DEVS model. Equivalent ideas were presented in [1], applied to a FMS plant.

We have shown [25] how simulation can be a useful tool for production systems control aid. The information given by the simulation allows a proactive control to be set up to anticipate blocked situations. This topic was illustrated in the paper by using the coupled DEVS formalism for a Kanban workstation modelling.

The major DEVS concepts can be founded in appendix 1. DEVS defines a method of abstraction of dynamic systems that allows building timed discrete event simulation models (deterministic models) with a good accuracy. DEVS is a symbolic specification of system semantics and there is only one way to execute a DEVS model.

Potential benefits may lie in the ability of DEVS

- a) to implement event filtering, wherein a planned change of state will be realized by a model only when the time interval that separate two key events exceeds a predefined value,
- b) to encapsulate the otherwise mechanical event filtering at the conceptual level.

To our knowledge, only DEVS formalism offers all the concepts necessary to represent, at the conceptual level, phenomena such as filtering and autonomous evolution... phenomena that are described by expert and must be formally represented before implementation occur. The operative semantics of DEVS is clearly defined by the concept of conceptual simulator. Then, a DEVS specification is independent of the simulation technology as well as the underlying host computer. Thus, DEVS models can be translated in all commercial simulation software for manufacturing processes like Arena, AutoMod, eM-Plant, Taylor ED.

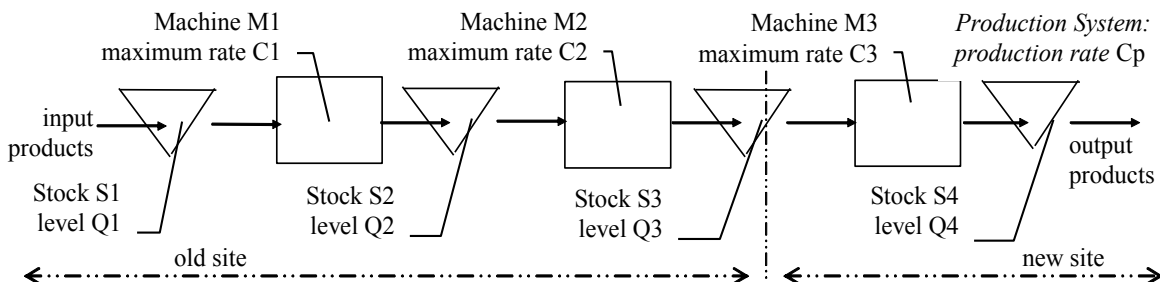
3. ELEMENTARY REMOVAL OF A WORKSTATION: CANONIC STUDY

3.1. Analysis of the modelling problem

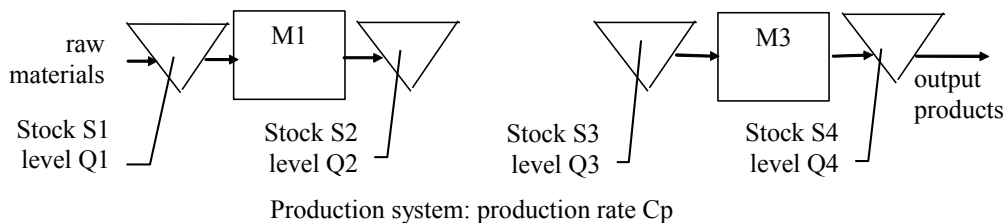
In order to relocate a complete production line, an elementary pattern of how the production system will behave during removal is needed: we describe some of the theoretical facts obtained from the example of a sub-set composed of only one machine. We call M_i the machine number i of a line of n machines. The machine, which is a removal one, is noted k . We also note $C_p \leq \min_{i=1}^n C_i$ the production rate of the line, with C_i the maximal production rate of the machine M_i , S_i the products' stock located upstream of this machine and Q_i the quantity of products in S_i .

3.1.1. Analysis of the removal principle

As our removal principle is based on a succession of small removals (one machine after the other or a group of machines after the other), the same production line appears in all the project stages both on the old and the new production sites. The only exceptions are at the very beginning of the removal, when no machine has yet been moved, and at its end, when all the machines are at their proper places in the new site. Let us take a canonical example, where $n=3$ and $k=2$. In all the steady intermediate states, i.e. between two successive removals of two sub-sets, the logistic link between the two sites is ensured through a buffer stock. The buffer stock decouples the operation of the two sites and allows the parts' transport between the sites (i.e. stock S_3). The Work-In-Process (WIP) has to be dispatched to the new site, which generates an additional delay but does not challenge the management of the production system. During the removal of machine M_2 , the production system is modified.



1.a. Model of a production line.



1.b. Model of an interrupted flow production line.

Figure 1. Model of a production line during a removal.

Indeed, M2 is stopped during its removal and so there are two production lines: one downstream on the new site, and one upstream on the old site.

To keep on producing, while M2 is unavailable (i.e. as long as its removal lasts (t_{Mov2})), the level of the stock S3 has to be high enough to allow input in the next machine (M3). As for S2, it will be loaded progressively because there is no more consumption of WIP. To keep this line in operation while avoiding production discontinuities is thus linked to controlling the intermediate stocks S3 and S2. In the sequel, the required decision mechanisms are explained in detail.

3.1.2. Study of the downstream and upstream flow-shops

To ensure finished goods production' continuity, M3 must go on producing during the removal of M2, i.e. S3 must contain a sufficient quantity of products to be able to feed M3 at the rate C_p during t_{Mov2} . This reserve quantity defined as $Q_3 = C_p * t_{Mov2}$ is built up in a

period of time $t = \frac{C_p}{C_{up} - C_p} * t_{Mov2}$, by increasing the upstream rate to the maximum upstream

rate C_{up} . Generally, $C_{up} \leq \min_{i=1}^{k-1} C_i$. This stock builds up while the upstream production goes on as usual. During the removal of M2, the upstream part of the line can go on producing as usual, thereby creating a stock in S2. The volume of this stock, defined as $Q_2 = C_p * t_{Mov2}$,

will be absorbed in a time $t = \frac{C_p}{C_{dw} - C_p} * t_{Mov2}$, where C_{dw} is the downstream maximum rate.

Generally, $C_{dw} \leq \min_{i=k+1}^n C_i$. In a normal running situation, if M2 is already working with a maximal rate, keeping the global rate of the production line is possible by stopping the entire production during this elementary removal. The latter will avoid creating undesirable stocks, which will not be absorbed afterwards.

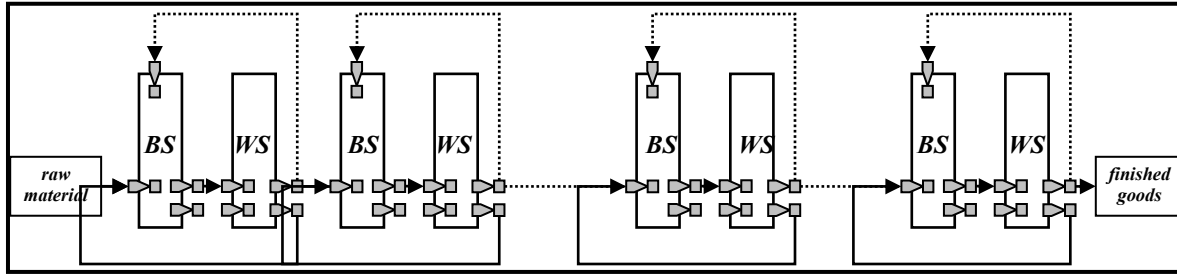
3.2. DEVS model of an elementary removal of a production workstation

In this section, the basic entities of the production line to be modeled are described.

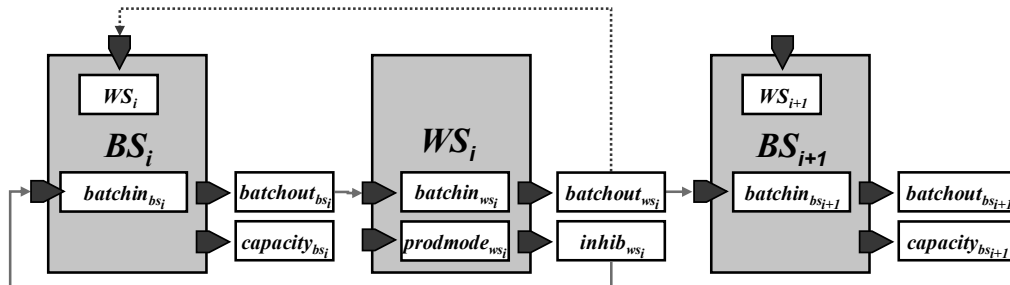
3.2.1. Elementary model

The production line consists of an alternation of stocks, where parts, grouped by batches, are waiting to be transformed. It also consists of machines, where batches are treated part by part before returning to wait. Inputs at the beginning of the production line are raw materials, and outputs are finished goods.

The production line simulation with the DEVS formalism will keep this structure: figure 2a illustrates the model with the alternation of WS (the WorkStation) and BS (the Buffer Stock), where WS is the generic model of a machine and BS is the generic model of a stock, in order to obtain the coupled model by simulation of the complete line, arrows represent the links between these models.



2a. DEVS model of production line.



2b. Coupling between buffer stocks models and workstation model.

Figure 2. DEVS model of production line.

3.2.2. The 'WS' detailed model

A workstation is modelled using a generic model described in coupled DEVS with a set of particular parameters which values specify the operation mode. These models are then associated to the other workstation's models to constitute the complete production line. The association of three functions builds each generic model of workstation: 'upstream interface', 'machine' and 'downstream interface'. We hereby provide an informal description of the generic model behavior.

When the production line is in a nominal state of operation, each workstation works batch by batch. A workstation begins to work when its upstream stock sends in a batch of C parts to be transformed. The reception of this batch by the workstation is made by the function 'upstream interface'. This machine interface enables to manage the parts of the batch one by one. It is thus about an un-stacking operation of the batch to present them to the transformation function 'machine'. After the treatment, each part of WIP is sent to the 'downstream interface', with the aim of reconstructing the batch part by part.

When the batch is completely treated, it is thus reconstituted in the 'downstream interface'. This batch is then sent to the downstream buffer stock.

The workstation, which is now inactive, can then receive a new batch and begin again the previous process.

Note that we have described the workstation's behavior mechanisms by considering the fact that a single batch is treated at the same time: the reception of a new batch is only possible when the treatment of the previous batch is finished by sending this latter in the downstream buffer stock.

Such an operation is modeled using the DEVS model '*WS*' in figure 3. '*WS*' is a coupled DEVS model consisting of three atomic DEVS models, '*upstream interface*', '*machine*' and '*downstream interface*'. Their behavior is respectively described in the next sections.

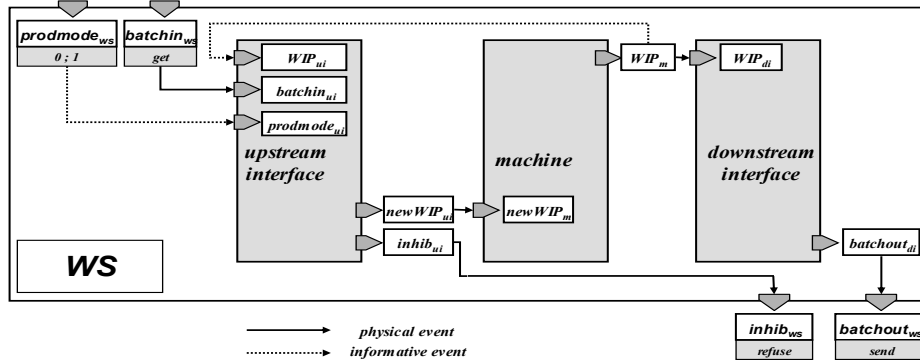


Figure 3. '*WS*': DEVS coupled model of workstation.

3.2.3. '*Upstream interface*' detailed model

a: informal description of the behavior of '*upstream interface*'

The '*upstream interface*' atomic model (figure 4a) can receive or reject batches, requests for parts and information on the operating mode. Two operating modes are available: normal production and stop during removal. The latter allows the machine to stop the production: it blocks the introduction of any new batch in the workstation, but permits to empty the workstation without interrupting the execution of the current batch. Indeed, the signal to stop the production does not produce an immediate effect, but it enables to end the possibly present batch and to have a machine, which remains empty, even if there is, or not batches in wait in the upstream stock.

b: formal description in DEVS of '*upstream interface*' model

'*upstream interface*' = $\langle X_{ui}, Y_{ui}, S, \delta_{ext}, \delta_{int}, \lambda, \tau \rangle$

Input event variables: $X_{ui} = (batchin_{ui}, WIP_{ui}, prodmode_{ui})$, where:

- '*batchin_{ui}*' = {*get*} – this event indicates the arrival of a new batch,
- '*WIP_{ui}*' = {*done*} – this event indicates the request for a new part,
- '*prodmode_{ui}*' = {0, 1} – this port gives the order for the operating mode of the machine: the external event ?*prodmode_{ui}*=1' indicates the 'normal production' operating mode; the external event ?*prodmode_{ui}*=0' indicates the 'stop during removal' operating mode.

State variables: $S = \{(phase, mode, job, sigma)\}$, where:

- *phase* = {*refuse, noprod, waitbatch, receive, queue, wait, lastbatch, lastqueue, wait2*}, is a name representing the situation in the real world,
- mode*={0, 1} is a variable that represents the operating mode, with the following interpretation: *mode*=0 is the 'stop during removal' mode, *mode*=1 is the 'normal production' mode (it is the default value),
- *job* $\in \mathbb{IN}_0$, is the counting variable of the parts (jobs) to be worked presents in the upstream interface,

c: interpretation of DEVS model behavior

In figure 4a, displays on the right side the normal production conditions (normal production operating mode), where $mode='1'$. It displays on the left side the behavior during the removal (stop during removal operating mode), where $mode='0'$. Assume we are in the initial state with $mode='1'$ and no WIP ($job=0$). The initial 'waitbatch' state is passive (i.e. its life time is ∞). A transition occurs at the external event $?batchin_{ui}='get'$, corresponding to the reception of a new batch. The arrival state 'receive' of this transition is transitory (i.e. its life time is 0): its role only consists in updating the number of parts in wait (in fact, the variable job , which was equal to 0, change its value to $job+C$, assuming C is the quantity of parts of the received batch). The following internal transition enables to begin the treatment of the batch: the arrival state 'queue', also transitory, enables to go into a loop of un-stacking until job nullifies. When $job > 0$, i.e. as long as there are parts to be worked (which is necessarily true just after the reception of a new batch), the sending of a part towards the machine is activated (output $!newWIP_{ui}='yes'$) during the transition towards the passive 'wait' state, where the value of job decreases 1 (i.e. it passes from job to $job-1$). From the latter state, one returns to the state 'queue' during the external event $?WIP_{ui}='done'$. This loop continues until the time occurrence when there are not parts anymore in the batch ($job=0$), where we return to the initial state with an internal transition corresponding to the output signal $!newWIP_{ui}='no'$.

For the external event $?prodmode_{ui}='0'$, several cases are possible, depending on the internal state of 'upstream interface'. If, for example, it occurs during the initial 'waitbatch' state (the workstation is empty), the model switches directly in the passive 'noprod' state, the value of $mode$ switches from '1' to '0' and job remains equal to '0'. At this time occurrence, it can receive a new batch ($?batchin_{ui}='get'$), but because of the inhibition, it has to refuse it: there is a transition towards the transitory state 'refuse', which induces at once the return of the system towards 'noprod' and the output signal $!inhib_{ui}='refuse'$. It is also possible that it receives the order of the normal production operating mode with the external event $?m=1$: it induces the transition towards the state 'waitbatch', and thus towards the normal conditions of operation.

If, on the contrary, the external event $?prodmode_{ui}='0'$ occurs during the treatment of a batch, it involves the state 'wait'. The passive state 'lastlot' is then reached, where job keeps the same value as in 'wait', but where $mode=0$. The input signal $?WIP_{ui}='done'$, which represents the free machine, induces the transition towards the transitory state 'lastqueue' which, as long as $job > 0$, creates a loop with the passive state 'wait2'. This is similar to the one we have described between both states 'queue' and 'wait'. The loop is ended when $job=0$ by an internal transition. The latter produces as an output the signal $!newWIP_{ui}='no'$ and induces the passive state 'noprod' (already described). Indeed, at this time occurrence, the workstation is empty and it has already received the order to stop the production.

3.2.4. 'machine' detailed model

a: informal description of the behavior of 'machine'

The atomic model '**machine**' can represent unspecified production equipments, able to process a product. For a use in simulation, the most important characteristic is the duration of the transformation task. The other characteristics are thus not taken into account in this model. Moreover, we made the simplifying assumption that the machine was always in a

good operating condition. So it is always available and we do not have to manage the degraded operation modes.

b: formal description in DEVS of 'machine' model

'machine' = $\langle X_m, Y_m, S, \delta_{ext}, \delta_{int}, \lambda, t \rangle$

Input event variables: $X_m = (newWIP_m)$, where:

- 'newWIP_m' = {yes, no} – this port indicates the reception of a new part to be treated (external event ?newWIP_m= 'yes') or the absence of a new part to be worked (external event ?newWIP_m= 'no')

State variables: $S = \{(phase, sigma)\}$, where:

- phase = {waitnewWIP, work}, is a name representing the situation in the real world,
- sigma $\in \mathbb{R}_0^+ \cup \infty$, is the life time of the current state.

Output event variables: $Y_m = (WIP_m)$, where:

- 'WIP_m' = {done} – this event indicates that the machine has finished to work a part.

The functions of the 'machine' DEVS model are defined by the way of the graph given in figure 4b.

c: interpretation of DEVS model behavior

The input, called 'newWIP_m', can have the values 'yes' (i.e. part reception) or 'no' (i.e. absence of part to be worked). The output, called 'WIP_m', has for value 'done' (i.e. the machine ended). The initial state of the model, 'waitnewWIP', is passive. In fact, it is in wait to receive a new WIP. When the event ?newWIP_{ui}= 'yes' arises, a new state is reached, 'work', with a finished life time (equal to the necessary time to make the transformation of the part). At the end of this duration, an internal transition enables the model to return to its initial state and to generate the output !WIP_m= 'done'. As long as the input receives the event ?newWIP_{ui}= 'no', the model does not change its state and waits for a new WIP.

3.2.5. 'downstream interface' detailed model

a: informal description of the behavior of 'downstream interface'

The purpose of the atomic model 'downstream interface' is to reconstitute the batch. Since we work part by part, the model must count the number of parts and test if the batch is complete. A signal is sent when the batch is reconstituted.

b: formal description in DEVS

'downstream interface' = $\langle X_{di}, Y_{di}, S, \delta_{ext}, \delta_{int}, \lambda, t \rangle$

Input event variables: $X_{di} = (WIP_{di})$, where:

- 'WIP_{di}' = {done} – this event indicates the reception of a part that has been treated,

State variables: $S = \{(phase, worked, sigma)\}$, where:

- phase = {waitWIP, queue, receive}, is a name representing the situation in the real world,
- worked $\in \mathbb{N}_0$, is the counting variable of the batch's parts worked received by the model,
- sigma $\in \mathbb{R}_0^+ \cup \infty$, is the life time of the current state.

Output event variables: $Y_{di} = (batchout_{di})$, where:

- $batchout_{di} = \{send\}$ – this event indicates that a batch has been reconstituted and that it goes out from the model.

The functions of the '**downstream interface**' DEVS model are defined in the graph in figure 4c.

c: interpretation of DEVS model behavior

The initial state 'waitWIP' is passive, with $worked=0$. A transition of state is activated by the signal $?WIP_{di}='done'$, representing the reception of the first worked part: this transition brings us in a new transitory state, 'receive', which is characterized for the value of $worked$ incremented by 1. The following internal transition depends on the comparison between the values C and $worked$. If $worked = C$, then the batch was reconstituted and it can thus be sent to the following stock. It generates the output $!batchout_{di}=send$, and one gets via an internal transition to the initial state. On the contrary, if $worked < C$, the batch is not finished. The internal transition reaches a new passive state, called 'queue', which protects the value of $worked$. During a new signal $?WIP_{di}='done'$, one gets into the transitory state 'receive', from which there will be a new comparison between C and $worked$ which will have been incremented by 1 when arriving in this state.

3.2.6. The 'WS' coupled model

We can now easily return to the couplings existing between the three atomics DEVS sub models constituting the DEVS coupled model of workstation '**WS**'.

a: formal description in DEVS

'**WS**' = $\langle X_{ws}, Y_{ws}, M, EIC, EOC, IC \rangle$

Input event variables: $X_{ws} = (batchin_{ws}, prodmode_{ws})$, where:

- $batchin_{ws} = \{get\}$ – this event indicates the reception of a new batch by the workstation,
- $prodmode_{ws} = \{0, 1\}$ – this port gives the order for the operating mode of the workstation: the external event $?prodmode_{ws}='1'$ indicates the normal production operating mode (it is the default value); the external event $?prodmode_{ws}='0'$ indicates the stop during removal operating mode.

Output event variables: $Y_{ws} = (batch_{ws}, inhib_{ws})$, where:

- $batch_{ws} = \{send\}$ – this event indicates that a reconstituted batch is sent out by the workstation,
- $inhib_{ws} = \{refuse\}$ – this event indicates that a new arriving batch is not accepted by the workstation.

Set of the names of components: $M = \{\text{'upstream interface'}, \text{'machine'}, \text{'downstream interface'}\}$

The couplings (internal (IC) or external (EIC for inputs and EOC for outputs)) can be viewed in figure 3. Note that the output $!WIP_m='done'$ emitted by the '**machine**' model can be considered under two aspects. The physical aspect represents the worked in progress part which is sent towards the '**downstream interface**' model for the reconstruction of the batch. The informative aspect represents a signal passed to the '**upstream interface**' model to specify that the machine is free and that it can send a new WIP.

b: interpretation of the behavior of 'WS'

The inputs ('*prod mode*' and '*batchin_{ws}*') and the outputs ('*inhib_{ws}*' and '*batch_{ws}*') of '**WS**' constitute the external coupling. Their operation was already described in the presentation of the atomic model '**upstream interface**', with regards to both inputs and first output, and also in the presentation of the model '**downstream interface**', with regards to the last output.

The internal coupling of the '**WS**' model is constituted by three connections. The first one connects the '*newWIP_{ui}*' port of the '**upstream interface**' atomic model and the '*newWIP_m*' port of the '**machine**' atomic model: way of parts of the batch from the upstream interface towards the machine. The second connects the port '*WIP_m*' of the model '**machine**' atomic model with the port '*WIP_{ui}*' of the '**downstream interface**' atomic model (informative event to request a new part) and the last connects the port '*WIP_m*' of the '**machine**' atomic model with the port '*WIP_{di}*' of the '**downstream interface**' atomic model (physical event to send the worked part).

3.2.7. The 'BS' detailed model

a: informal description of the behavior of 'buffer stock'

Stocks situated on the production line between workstations have an infinite capacity.

The '**buffer stock**' model will describe this characteristic. It can receive, accumulate and send batches. The inputs and the outputs are made batch by batch. The model must manage the number of the present batches in the stock. When it receives a new blanket release and if this number is equal to 0, then it must send information, otherwise the stock transaction consists in one batch delivery.

b: formal description in DEVS

'**BS**' = $\langle X_{bs}, Y_{bs}, S, \delta_{ext}, \delta_{int}, \lambda, \tau \rangle$

Input event variables: $X_{bs} = (WS, batchin_{bs})$, where:

- '*WS*' = {*free*} – this event indicates that the downstream workstation is free,
- '*batchin_{bs}*' = {*new, refused*} – this port represents the reception of a batch: the external event *?batchin_{bs}* = '*new*' represents the reception of a batch from upstream, the external event *?batchin_{bs}* = '*refused*' represents the reception of a batch refused from downstream,

State variables: $S = \{(Phase, batch, sigma)\}$, where:

- *phase* = {*refusedbatch, waitbatch, newbatch, send, empty, onebatch*}, is a name representing the situation in the real world,
- *batch* $\in \mathbb{IN}_0$, is the counting variable of the batches presents in stock,
- *sigma* $\in \mathbb{IR}_0^+ \cup \infty$, is the life time of the current state.

Output event variables: $Y_{bs} = (batchout_{bs}, capacity)$, where:

- *batchout_{bs}* = {*send*} – this event indicates that a batch is sent to the empty downstream workstation,
- *capacity_{bs}* = {*empty*} - this event indicates that there are no more batches to be sent.

The functions of the '**BS**' DEVS model are defined in the graph in figure 5:

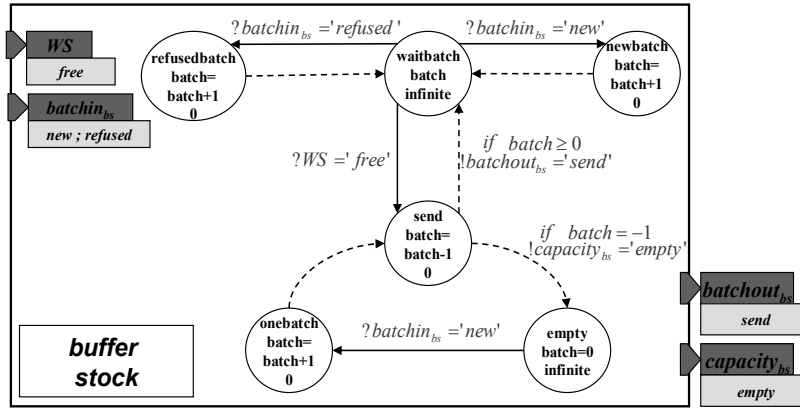


Figure 5. DEVS atomic model of 'buffer stock'.

c: interpretation of DEVS model behavior

To describe the model's behavior, assume its initial state is the passive 'waitbatch' state, characterized by a positive or not value of the variable *batch*. At this time occurrence, several transitions are possible according to the external event that arises. If *?batchin_{bs}='new'* or *?batchin_{bs}='refused'* occur, a new batch is received, with an external transition towards two different transitory states, called respectively 'refusedbatch' and 'newbatch'. The number of present batches in the stock is incremented from *batch* to *batch+1*, then the system returns to the passive 'waitbatch' state. The third external transition from the passive state 'waitbatch' is connected to the external event *?WS='free'*, which induces the transitory 'send' state. During this transition the variable *batch* decreases its value by 1: a new batch was taken out of the stock, to be sent towards the downstream workstation. Two cases appear: Either the stock still contains at least one batch (*batch ≥ 0*), the state 'waitbatch' is reached again by emitting the signal *!batchout_{bs}='send'*; Or the stock is empty, and then *batch = -1*. In these conditions, it is impossible to send the batch. The signal *!capacity='empty'* describes this situation when we pass via an internal transition towards the passive state 'empty'. We can only change state if the external event *?batchin_{bs}='new'* occurs. The reception of a new batch from the upstream workstation produces the transition towards the transitory 'onebatch' state, characterized by the value 1 of *batch*. It returns to the state 'send', where *batch = 0*, then via an internal transition to the state 'waitbatch' with the output *!batchout_{bs}='send'*.

3.2.8. Coupling between WS and BS

In this paragraph, we deal with the coupling between the models 'WS' and 'BS' (refer to the situation described in figure 5). In our generic approach, we just have to describe the couplings between a generic model 'WS_i' and both generic models 'BS_i' and 'BS_{i+1}', which are situated respectively on upstream and downstream.

In figure 2b, arrows represent the couplings between the three models.

Between the 'WS_i' workstation and the 'BS_{i+1}' downstream stock, the coupling connects the output 'batchout_{ws}' of 'WS_i' with the input 'batchin_{bs}' of 'BS_{i+1}'. It is representative of the batches flow along the production line.

There is a stronger link between the ' WS_i ' workstation and the ' BS_i ' upstream stock. First, the output ' $batchout_{bs}$ ' of ' BS_i ' feeds directly the input ' $batchin_{ws}$ ' of ' WS_i ': the batches are brought on the workstation.

Then, coupling is undertaken by linking the output ports of ' WS_i ' (respectively ' $batchout_{ws}$ ' and ' $inhib_{ws}$ ') to the input ports of ' BS_i ' (respectively ' WS ' and ' $batchin_{bs}$ '). The first connection gives to the stock BS_i the information that its downstream workstation is free (a worked batch has been sent to stock BS_{i+1}) and can receive a new batch. The second connection is used to bring back rejected batches to their initial stock, i.e. BS_i .

Each stock ' BS_i ' produces the external event ' $!capacity_{bs_i}='empty'$ ' when it is empty and after receiving the new batch blanket released by the downstream workstation, which is also empty. So, this signal is representative of the empty state of the stock ' BS_i ' and of its downstream workstation ' WS_i '.

All the workstations of the production line answer this coupled model. The upstream stock of the first one is ' BS_1 ' the stock of raw materials. The last stock of the production line (' BS_{n+1} ') contains finished goods. The models used for both stocks are thus simplified versions of ' BS '. All other stocks contain WIP.

4. PROGRESSIVE REMOVAL OF A PRODUCTION LINE: REAL STUDY

As we have seen in the previous paragraph, the scheduled stopping of a workstation in a production line because of its removal can be perfectly anticipated by creating a temporary buffer stock. We can thereby suppress the impact of this stopping on the production rate. We will now generalize the use of this method for an entire production line without any interruption of activity.

In order to exploit all the above elements, we suggest a removal organization based on the two following principles:

- Creation of a downstream stock necessary during the removal of a workstation.
- Absorption of the upstream stock created during the removal of the following workstation.

Indeed, the downstream stock of Work-In-Process will allow the upstream part of the manufacturing line to work normally while the downstream part of the line will continue producing by absorbing the upstream stock of semi-finished products (figure 6b). Each stock BS_i of WIP builds up during the removal of the workstation WS_i , and is absorbed during the removal of WS_{i-1} , while BS_{i-1} builds up.

One gets the impression that the location of the buffer stock is shifted upstream of each workstation being moved. This avoids creating a new stock, since the buffer stock is in the right place for the removal of the next workstation. We therefore suggest starting with the removal of the last workstation.

This will create an upstream stock, which will allow the removal of the next to last workstation and so on to the first workstation. This solution reduces the number of removal steps (see figure 6c). On the contrary, by starting the removal with the first workstation on the line, each time a workstation is moved, its buffer stock is not well located for the removal of the next workstation, and must be re-located in its right place: this requires additional steps (figure 6a). This solution is therefore not recommended. The former solution allows optimizing the total removal time: we create a single buffer stock that will be moved and adjusted according to the removal times of the various workstations.

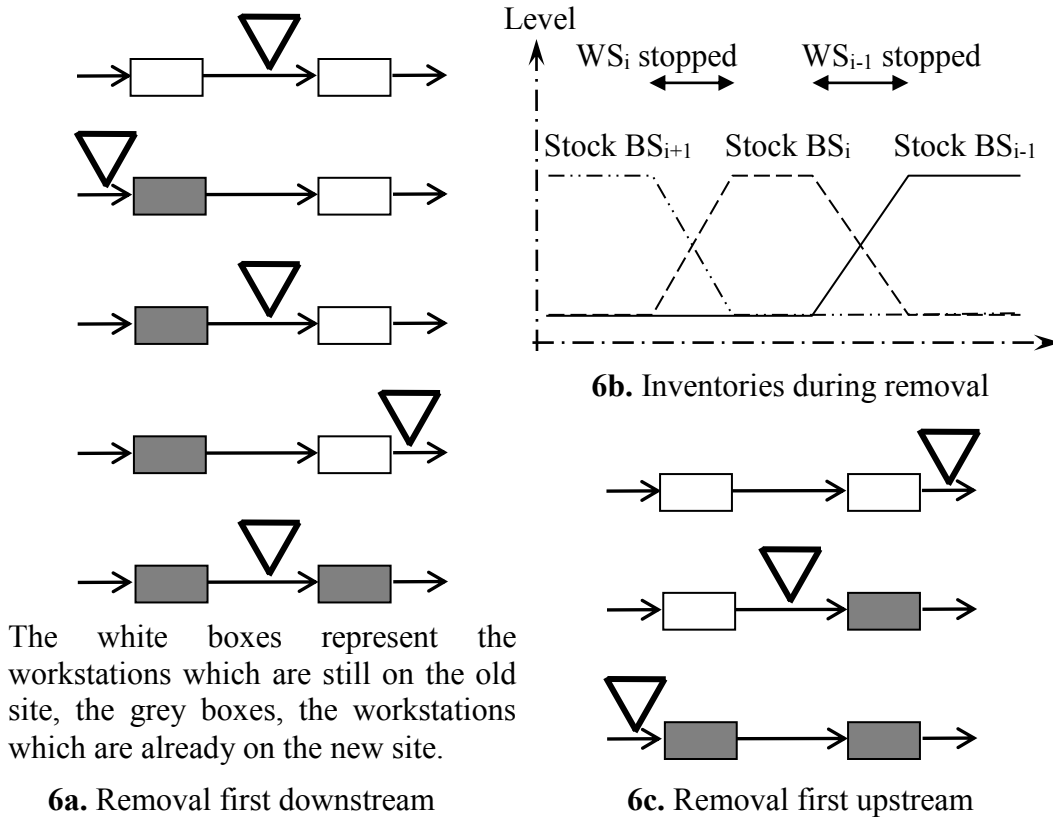


Figure 6. Evolution of stock levels during removal

4.1. Methodology of the relocation of a production line

The principal steps necessary for the relocation of a production line are the following:

Removal of the last workstation: during this time, the upstream stock is created.

do Adaptation of the volume of the buffer stock to the next workstation (the stock must be able to feed the downstream part of the production line, during the removal of this workstation, i.e. that the quantity of products stored will depend on the removal time of the workstation).

Removal of the previous workstation.

until the first workstation of the production line is moved.

Resumption of the consumption of raw materials interrupted during the removal of the first workstation.

Algorithm 1. Removal project management algorithm.

In real production lines, when considering a particular workstation, the upstream production rate differs from the downstream production rate. Therefore, each workstation removal must be specifically studied. Removal a workstation means that it has to be stopped, dismantled, maintained and transported, reassembled, calibrated and controlled.

The time all these activities take is the removal time t_{Mov} . The nominal production rate C_p and the maximum production rate C_{dw} are two other fundamental parameters.

4.2. Adjustment of buffer stocks

The level of the downstream stock is determined according to the removal time of the workstation, which means that the upstream production rate has to be adjusted.

- If the removal time of the workstation is longer than that of the preceding workstation, we will have to increase the stock and thus the upstream production rate.
- If the removal time of the workstation is shorter than that of the preceding workstation, we will have to decrease the stock which means a decrease of the upstream production rate.

Let us study the theoretical case just after WS_2 has been moved. The workstations WS_2 and WS_3 being already on the new production site, the stock BS_2 has not been absorbed and contains a quantity of products, $Q_2 = C_p * t_{Mov2}$.

The stocks BS_3 and BS_1 are back to their normal capacities. To continue the production during the removal of WS_1 , BS_2 will be required to contain a quantity of products $Q'_2 = C_p * t_{Mov1}$. Thus we will have to produce $Q'_2 - Q_2 = C_p * (t_{Mov1} - t_{Mov2})$ more products than in normal production.

The adaptation time of BS_2 is therefore $t = \frac{C_p}{C_1 - C_p} * (t_{Mov1} - t_{Mov2})$ (1)

If $t_{Mov2} < t_{Mov1}$, the level of stock necessary to move WS_1 , i.e. Q'_2 , is higher than the level of stock left by the removal of WS_2 (Q_2). Therefore, more semi-finished products have to be produced. Two methods can be used to increase of the removal time:

- Producing after the removal of WS_2 . The production time is defined by equation (1).
- Producing during the removal of WS_2 , thanks to an increase of the production rate of WS_1 . Let C be the necessary capacity to produce this additional quantity in a time t . We

obtained: $(C - C_p) * t_{Mov2} = C_p * (t_{Mov1} - t_{Mov2})$. Thus $C = C_p * \left(\frac{t_{Mov1} - t_{Mov2}}{t_{Mov2}} + 1 \right)$

If the rate necessary to prepare the stock for the removal of WS_1 during the removal of WS_2 (C) is greater than the maximum production rate of WS_1 (C_1), then we will have to produce the quantity after the removal of WS_2 : $Q'_2 - (C_1 - C_p) * t_{Mov2} = (C_1 - C_p) * t$.

Therefore, we will have to produce at maximum rate for a time $t = \frac{C_p}{C_1 - C_p} * t_{Mov1} - t_{Mov2}$

If $t_{Mov2} > t_{Mov1}$, two solutions are proposed to decrease of the removal time:

- Stopping the production during a time $t = \frac{C_p}{C_1 - C_p} * (t_{Mov2} - t_{Mov1})$
- Decreasing the production rate of WS_1 during the removal of WS_2 . The new rate of WS_1 will be $C = C_p * \left(\frac{t_{Mov2} - t_{Mov1}}{t_{Mov2}} + 1 \right)$.

4.3. Principle of simulation

The simulation of this flow-shop is possible by developing two generic models via the formalism DEVS: the Group of Sequential Workstations (GSW_i) and the Buffer Stocks (BS_i). Next, by coupling these models, we obtain the complete flow-shop model which will be simulated (refer to figure 7).

The simulation of the relocation requires a Removal Control System (RCS), which inhibits the various groups of machines operation during their removal.

The simulation technique retained is relatively simple. The line in its normal configuration was modeled according to an approach based on activities, each activity corresponding to a machine. All in all, about thirty activities were defined to simulate a production-type representing more than 80% of the cases of manufactured products.

In each case, the simulation starts one week before the beginning of the removal, in order to initialize the flow and to prepare the buffer stock, and ends one week after the end of the removal, once the stabilization of the flow has been achieved. The simulation of a removal stage (thus concerning a group) was obtained by the following mechanism:

```

for  $i=N$  to 1
    Inhibition ( $m_i=0$ ) of new products entrance in the group of
        workstations being moved (the creation of the upstream stock
        begins, while these machines empty gradually),
    When the last Work-In-Process gets out of this group of machines
        (empty output  $=1$ ), the time duration of its removal count down
        is started ( $t_{Mov\ i}$ ),
    At the end of this time, the inhibition of the entrance of products is
        suppressed ( $m_i=1$ ) and the products start to circulate again in the
        machine group that has just been moved,
    Production by waiting for the next  $GSW$  removal.
next  $i$ .
    
```

Algorithm 2. Removal simulation algorithm.

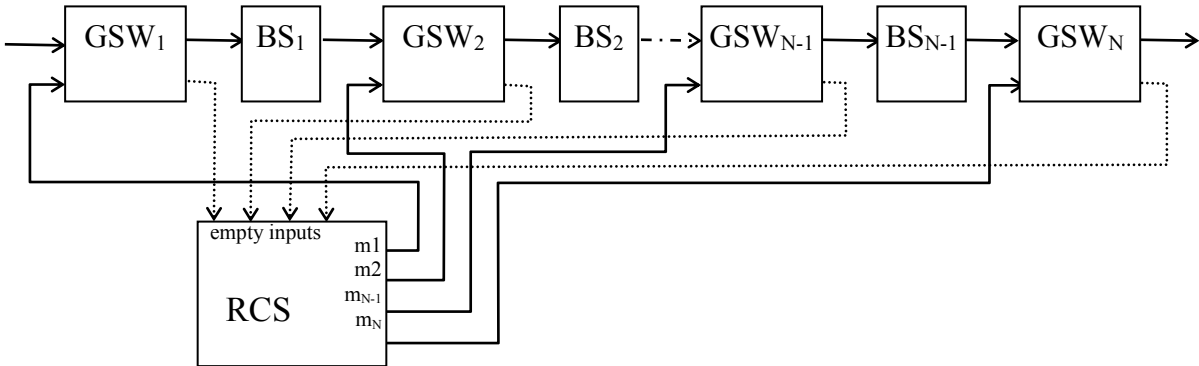


Figure 7. Relocation flow-shop model.

5. SIMULATION OF A REMOVAL: AN INDUSTRIAL CASE

We present a removal project made necessary by the decrepitude of the initial site and a lack of available surface, due to the accumulation of new equipment over the years.

Having a new industrial building with more floor space allows a different organization of the machines, while keeping the same functional logic. This presents various advantages: better quality of life for the production staff, facilitated inventory control, and better brand image towards the clientele.

5.1. Presentation of the industrial study

This study was performed for a small sized company that produces printed circuit boards. Whereas the mass production of such products implies long production delays (several months), this firm produces small and medium sized series within 2 weeks, and even within 5 days in case of urgency. Its market is served by engineer to order manner. Each item to be manufactured is defined by the customer. Thus, each order needs to interpret the CAD requirements of the customer before manufacture can start: masks manufacturing, programming instructions of computer numerically controlled drill press...

Their requirement for responsiveness and continuous presence on the market means that the removal must not interrupt the production, even temporarily.

The production takes place on a production line composed of 24 workstations, automated or not, organized in 6 work-shops. Some workstations have several machines in parallel. Transfers between machines are principally manual, the integrated circuits being temporarily stored vertically on pallets where they are not in contact with one another.

Some of these workshops are characterized by particular environmental conditions. Thus, in the chemistry and engraving workshops, where many foul-smelling effluents are emitted, extractions and air treatments must be performed regularly, while the place where photosensitive products are stored needs a polarized light and certain level of dust filtration. Figure 8 gives an idea of the sequence of production operations which allows the transformation of an epoxy plate covered with brass into two-sided PCBs.

The removal machine after machine is no longer adapted because the machines are too numerous and too heterogeneous. Some are simple to move whereas others need complex procedures. Therefore, we determined different possible machine groupings with specific removal characteristics: homogeneous removal times and small variations of the buffer stock volume.

5.2. Study of machine grouping strategies

5.2.1. Selecting the production line groups

The same production line can be moved machine after machine, or groups of machines after groups of machines. The choice of sequences of machine grouping must comply with technological rules and requirements. In fact, sometimes a product getting out of a certain machine must get into the following machine right away. For example, before the beginning of each of the engraving stages, a photosensitive layer is deposited on the epoxy plate by a first machine, then it is necessary to develop the layer immediately in an

exposure appliance and then another machine suppresses the non developed product: these three machines must not be moved separately, since they are part of a group. This grouping for technological reasons may be needed in other instances, such as, in the food industry when there are shelf time constraints.

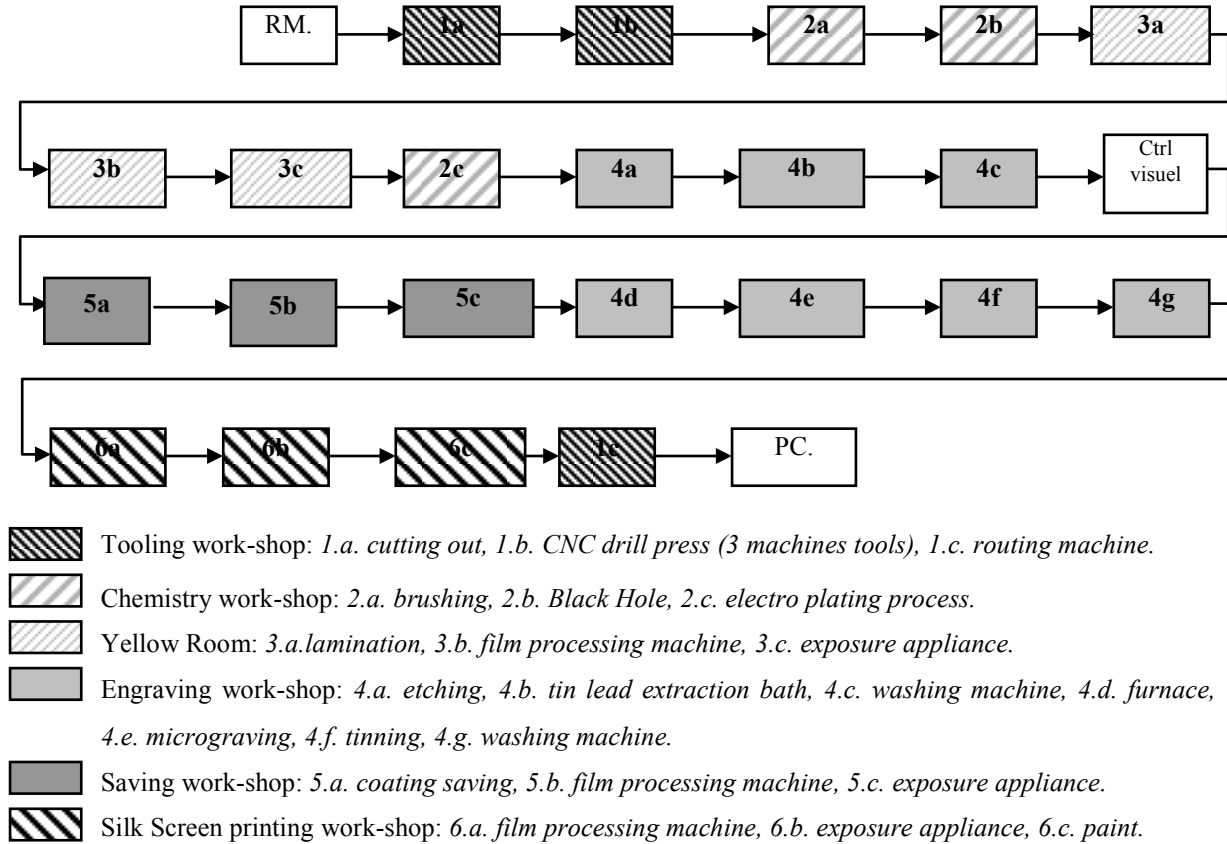


Figure 8. The production line.

Furthermore, for the proposed method to operate properly, it is necessary to have a certain balance between the volumes of the successive stocks to be created. This imposes to create groups of machines with homogeneous removal times.

5.2.2. Application

In our application, only a dozen strategies of segmentation have complied with the technological constraints. The smaller the buffer stock is, the more delicate the removal is. Therefore six segmentation strategies were studied. The segmentation by machine or by workshop proved inadequate. The best three solutions are shown in Table 1:

- Solution A: grouping machines from different workshops.
- Solution B: dividing different workshops so as to optimize the variations of the buffer stock.

- Solution C: using the parallelism of the first drilling workshop and maintaining the production of some drilling machines during the removal of the other, to minimize the average buffer stock.

To test these solutions, an accurate simulation model of the production system and of its flow was necessary. It needed a ground inquiry to collect the exact specifications of the machines concerning their productivity and move ability.

Strategy	Description (Refer to figure 8)	Number of groups	Time taken (weeks)
A	4 groups from different workshops: (6, 1c) ; (4,5) ; (2,3) ; (1(ab))	4	5,5
B	same as A, but sectioning of the group (4,5) gives: (4(abc), 5(abc)) ; (4(defg))	5	6,5
C	same as B + drilling workshop decomposed into 3 parallel tool machines: (1b1 // 1b2 // 1b3)	7	10,5

Table 1: Description of the proposed strategies A, B and C.

5.3. Exploring the simulation mechanisms

We describe the coupled DEVS modeled components announced in section 4.3 (figure 7).

5.3.1. The ‘GSW’ model description

In this section, we give an informal description of the behavior of ‘GSW’.

In a ‘GSW’ coupled model, several ‘WS’ models of workstations are associated in alternation with several ‘BS’ models of stocks. The goal is to obtain a ‘GSW’ model having globally an equivalent operation to that of a workstation ‘WS’ model. From the point of view of their external coupling, these two models have thus to be alike, even if ‘GSW’ model contains several ‘WS’ models. So a modular and hierarchical approach of the production line modelling problem is obtained, which enables several configurations of the relocation. In figure 9, the case of a Group of Sequential Workstations (GSW) including 4 workstations is described. The principle is the same whatever the number n of buffer stocks and the number $n+1$ of workstations contained in the GSW is.

In this model, the problem consists in making sure that the whole of the group was really emptied before beginning the removal. For that, we must build a model which objective is to control the logical conditions indicating the empty states of each model component and to aggregate them. It is the goal of the sub-model ‘E’.

5.3.2. The ‘E’ detailed model

a. informal description of the behavior of ‘E’

The model 'E' is used to generate a logical signal indicating that a 'GSW' is totally empty of WIP: this implies it can be moved. The operations of model 'E' are activated when the model receives the order for the 'stop during removal' operating mode.

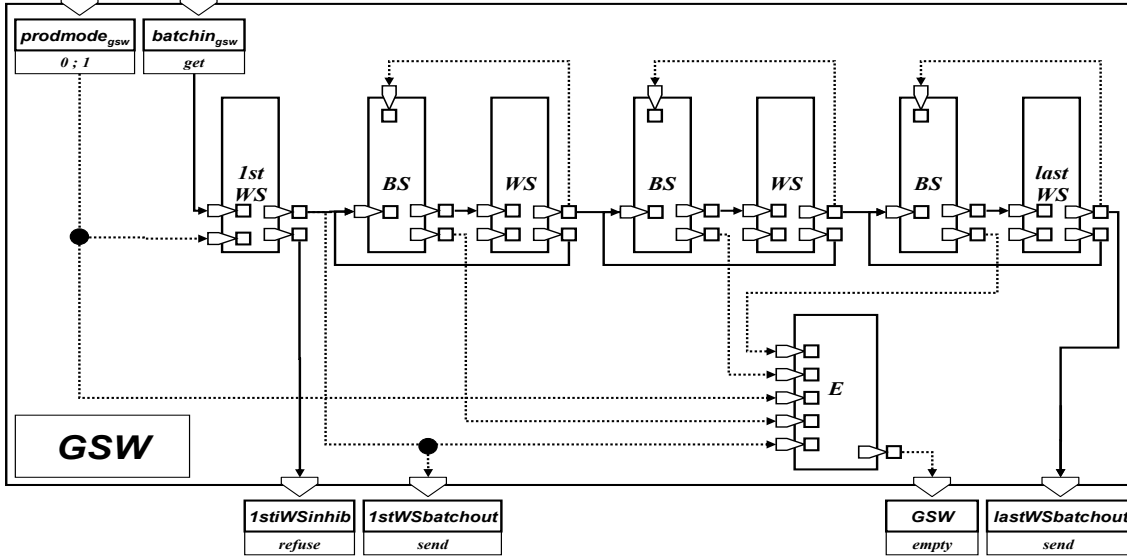


Figure 9. DEVS model of a group of sequential workstations (**GSW**) for relocation control.

b: formal description in DEVS

'E' = $\langle X_E, Y_E, S, \delta_{ext}, \delta_{int}, \lambda, \tau \rangle$

Input event variables: $X_E = (prodmode_E, capacity_1, \dots, capacity_i, \dots, capacity_n, capacity_{n+1})$, where:

- ' $prodmode_E$ ' = $\{0, 1\}$ – this port gives the order for the operating mode of the machine: the external event $?prodmode_E=1$ indicates the normal production operating mode, the external event $?prodmode_E=0$ indicates the stop during removal operating mode.
- ' $capacity_i$ ' = $\{empty\}$ for $i=1, \dots, N, N+1$ – this event indicates that the i-th element is empty (for $i=1, \dots, N$ it is a buffer stock, for $i=N+1$ is the first workstation),

State variables: $S = \{(phase, \vec{X}, sigma)\}$, where:

- $Phase = \{init, inhib, empty, verify, wait\}$, is a name representing the situation in the real world,
- \vec{X} is a vector of dimension $N+1$: each coordinate x_i ($i=1, \dots, N+1$) of \vec{X} represent the empty state of the i-th element: the value '0' is the default value and it refers to the normal production operating mode, and the value '1' refers to the stop during moving operating mode and it signifies that the element is empty from parts,
- $sigma \in \mathbb{R}_0^+ \cup \infty$, is the life time of the current state.

Output event variables: $Y_E = (GSW_E)$, where:

- $GSW_E = \{empty\}$ – this event indicates that there are no more parts in the group of sequential workstations.

The functions of the 'E' DEVS model are defined in the graph in figure 10:

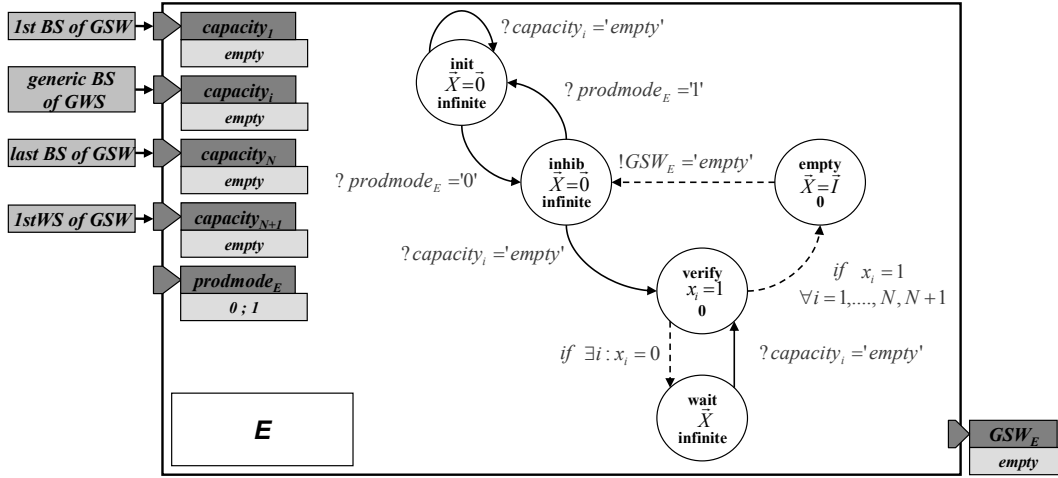


Figure 10. E model.

c: interpretation of DEVS model behavior

At the beginning, in the passive state 'init', characterized by a normal production, each coordinate x of \bar{X} is equal to 0 ($\bar{X} = \bar{0}$). Each event $?capacity_i = 'empty'$ is ignored and the first external transition is caused by the event $?prodmode_E = '0'$. The model reaches the passive state 'inhib'. Each time an external event $?capacity_i = 'empty'$ occurs, the i -th coordinate of \bar{X} changes its value from 0 to 1 and the model reaches the transition with the state 'verify'. The internal transition which derives from this state depends on the following condition: when $x_i = 1$, for each $i = 1, \dots, N+1$, the transition is activated and a new transitory state 'empty' is reached. The output signal $!GSW_E = 'empty'$ is produced when the state 'inhib' is reached again. As long as there is still at least one i such as $x_i = 0$, the transition 'verify' reaches a passive state 'wait' from which 'verify' can be reached again if a new external event $?capacity_i = 'empty'$ occurs. When the model returns in 'inhib', it waits for the signal $?prodmode_E = '1'$ to return to the initial state 'init'.

5.3.3 The 'GSW' detailed coupled model

a: formal description in DEVS

'GSW' = $\langle X_{gsw}, Y_{gsw}, M, EIC, EOC, IC \rangle$

Input event variables: $X_{gsw} = (batchin_{gsw}, prodmode_{gsw})$, where:

- $batchin_{gsw} = \{get\}$ – this event indicates the reception of a new batch by the group of sequential workstation,
- $prodmode_{gsw} = \{0, 1\}$ – this port gives the order for the operating mode of the group: the external event $?prodmode_{gsw} = '1'$ indicates the normal production operating mode (it is the default value), the external event $?prodmode_{gsw} = '0'$ indicates the stop during removal operating mode,

Output event variables: $Y_{gsw} = (1stWSinhib, 1stWSbatchout, GSW, lastWSbatchout)$, where:

- $IstWSinhib = \{refuse\}$ – this event indicates that a new arrived batch is not accepted by the group,
- $IstWSbatchout = \{send\}$ – this event indicates that the first WS has sent out a reconstituted batch,
- $GSW = \{empty\}$ - this event indicates that there are no more parts inside the group,
- $lastWSbatchout = \{send\}$ – this event indicates that the last WS has sent out a reconstituted batch,

Set of the components names : $M = \{ 'IstWS', \{ 'WS_i' | i \in \{1, \dots, n+1\} \}, \{ 'BS_i' | i \in \{1, \dots, n\} \}, 'E' \}$

Refer to figure 9 for the couplings (internal (IC) or external (EIC for inputs and EOC for outputs)).

b: interpretation of DEVS model behavior

One can note the similarities at the level of the external coupling. Both ' $prodmode_{gsw}$ ' and ' $batchin_{gsw}$ ' inputs, for which the operation have already been described, are directly connected with the coupled model ' $IstWS$ '. ' $prodmode_{gsw}$ ' is also connected with the input of the ' E ' model. For four outputs ' $IstWSinhib$ ', ' $IstWSbatchout$ ', ' GSW ' and ' $lastWSbatchout$ ', there are some modifications. The first two outputs have a similar operation to that of the outputs ' $inhib_{ws}$ ' and ' $batchout_{ws}$ ' of ' WS '. They are connected to the outputs of the first one of n ' WS ' models present in ' GSW ', by keeping a similar role: the event ' $?IstWSinhib='refuse'$ ' is used to refuse new batches when $prodmode_{gsw}=0$, and the event ' $?IstWSbatchout='send'$ ' gives the order to the upstream stock to send a new batch to ' GSW ', and then to ' $IstWS$ '. The output ' GSW ' is connected to the output of the model ' E ': it is the information transmission according to which there is no more WIP in the GSW. This information indicates that the removal of the GSW can start. The last output ' $lastWSbatchout$ ' is connected to the output ' $batchout_{ws}$ ' of the last ' WS ': it is about the transfer of batches towards the downstream stock after the last workstation of the GSW. The similarities between ' WS ' and ' GSW ' at the external coupling level are the removal control input ' $prodmode_{gsw}$ ', the input and the output for the WIP flow and the inhibition output.

The internal coupling of the coupled model follows the behavior described in section 3.2 (refer to figure 2b). The outputs ' $capacity_{bs}$ ' of the various models ' BS ' are connected in ' GSW ' with the inputs of the model ' E '.

5.3.4 The ' RCS ' detailed model

a: informal description of the behavior of ' RCS '

This model is used to manage each stage of the production site removal: it gets the information about characteristic times of the relocation (time duration, initial time) and then uses them to manage the relocation.

Each strategy of production line segmentation specifies a number M of GSW (Group of Sequential Workstation) to be moved. In the definition of each strategy, a specification of the technical operations of each removal stage (dismantling, transport, assembly and setup) enables to calculate the duration of removal and its initial time. Thus, we define for each GSW_k ($k=1, \dots, M$) its data triplet $Y_k = (k, t_k, t_{mov_k})$: k is the number of the GSW, t_k is the

GSW removal start date and t_{mov_k} is its duration. We can obtain *in fine* the \vec{Y} vector of dimension M , composed by each Y_k , which defines all characteristic times of a relocation strategy. The removal procedure described below allows obtaining a vector in the form $\vec{Y} = ((M, t_M, t_{mov_M}), \dots, (k, t_k, t_{mov_k}), \dots, (1, t_1, t_{mov_1}))$: GSW_M is the first group to be moved and GSW_1 is the last one

These GSW data will be exploited by the '**RCS**' (Removal Control System) model.

b: formal description in DEVS

'**RCS**' = $\langle X_{RCS}, Y_{RCS}, S, \delta_{ext}, \delta_{int}, \lambda, t \rangle$

Input event variables: $X_{RCS} = (GSW_1, \dots, GSW_k, \dots, GSW_M, relocation)$, where:

- ' GSW_k ' = $\{free\}$ for $k=1, \dots, M$ – this event indicates that the k -th group of sequential workstation is empty (there is no batch),
- ' $relocation$ ' = $\{\vec{Y}\}$ – this event indicates the specifications for the relocation, i.e. the sequence of the various removals, the time instant at which they have to start and their time duration,

State variables: $S = \{(phase, \vec{Y}, t, sigma)\}$, where:

- $phase = \{wait, stateM, stateM^*, stateM^{**}, \dots, statek, statek^*, statek^{**}, \dots, state1, state1^*, state1^{**}, end\}$, is a name representing the situation in the real world,
- \vec{Y} is a vector of dimension M , where each coordinate Y_k ($k=1, \dots, M$) represents a triplet (k, t_k, t_{mov_k}) with the temporal information characteristic of each removal,
- $t \in \mathbb{R}_0^+ \cup \infty$, is the time variable which concerns the time duration of the simulation,
- $sigma \in \mathbb{R}_0^+ \cup \infty$, is the life time of the current state.

Output event variables: $Y_{RCS} = (pmode_1, \dots, pmode_k, \dots, pmode_M)$, where:

- $pmode_k = \{0, 1\}$ for $k=1, \dots, M$ – this port gives the order for the operating mode of the k -th group: the event $!pmode_k=1$ indicates the order for the normal production operating mode (it is the default value), the event $?pmode_k=0$ indicates the order for the stop during removal operating mode.

The functions of the '**RCS**' DEVS model are defined in the graph in figure 11.

The initial state of the model is a passive state ' $wait$ ' where the vector $\vec{Y} = \vec{0}$ and $t=0$. The event $?relocation=\vec{Y}$ activates the simulation ($t=t_0$). A new state ' $stateM$ ' is reached:

- Updated of the vector \vec{Y} . The event $?relocation=\vec{Y}$ initializes this vector,
- Initialization of the start date (t_M) of the GSW_M removal (this state will wait the good date, and for that, it needs to calculate its life time $\sigma = t_M - t_0$).

In the date t_M , the internal transition causes the inhibition of GSW_M , through the event $!pmode_M=0$. The passive state ' $stateM^*$ ' is reached, where the variable t is updated $t=t_M$. The following transition is reached at t'_M , when the model ' GSW_M ' emitted the signal that it is empty $?GSW_M=free$: the state ' $stateM^{**}$ ' is reached, where the t_{movM} life time corresponds to the removal duration. At $t'_M + t_{movM}$, the production of '**GSW**' can start again: broadcasting of the signal $!pmode_M=1$.

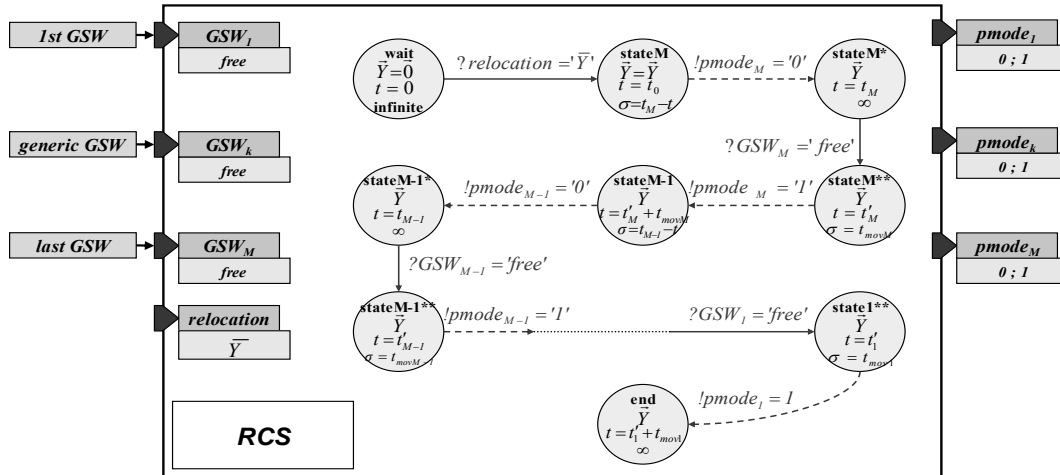


Figure 11. DEVS model of 'RCS'.

This above operation can thus be repeated until the last removal stage is reached. The following sequence describes this operation:

```

i=M
do   Wait until date  $t=t_i$  (statei),
      Blocking the flow WIP at the input of 'GSWi',
      Wait until it will be empty:  $t=t'_i$  (statei*),
      Wait during the 'GSWi' removal duration:  $t=t'_i + t_{movi}$  (statei**),
      Return of the 'GSWi' normal production operating mode,
      i=i-1
until i=0

```

Algorithm 3. Removal scheduling algorithm.

Then, the model reaches the passive state 'end', which involves the end of the relocation and the return of the production line to normal operation conditions, on the new site.

'RCS' allows thus to have a chronological sequencer of the removal stages, so settling the temporal succession of the inhibition events $0 < t_0 < t_M < t_{M-1} < \dots < t_I$. The model behavior enables to also assert that, for each $k=1, \dots, M-1$, we have $t'_{k+1} + t_{movk+1} < t_k < t'_k < t'_k + t_{movk} < t_{k-1}$.

This is particularly true for the treated industrial case:

- The successive removal must be done the weekend,
- The duration of these removal was included between 1,5 and 2,5 days,
- The duration of emptying of a GSW does not exceed few hours.

5.3.5 Relocation flow-shop with DEVS model

The explanations given in the three above paragraphs can now be considered as a whole: the production line during the removal, with all ad hoc models DEVS, is then represented.

On the basis of the graph in figure 7, the situation illustrated by figure 12 is obtained, where only one stage of the removal is considered, i.e. a 'GSW' model, a 'BS' model and a 'RCS' model.

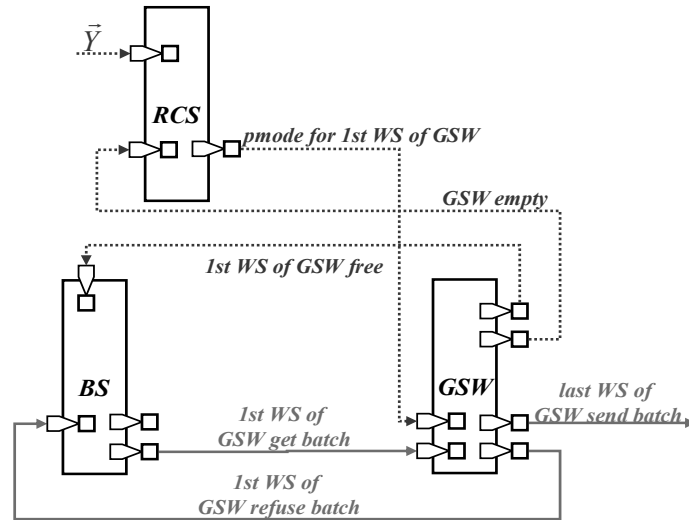


Figure 12. Relocation flow-shop with DEVS model.

The complete production line can thus be obtained by repeating this coupling for all the groups and the buffer stock envisaged by the removal strategy.

6. RESULTS OF A REMOVAL SIMULATION

6.1. Presentation of simulation results

For each regrouping strategy, the complete simulation of the removal allowed us to determine the evolution of the production flow and that of the Work-In-Process stocks, supposing five days of work per week.

In this flow-shop, there are a lot of machines whose elementary removal time t_{Mov} is longer than a weekend's time. Therefore it is impossible to consider a grouped removal distributed over several weekends: it is only possible to organize each removal by including weekends. The duration of each stage can be reduced: $t'_{Mov} = t_{Mov} - 2$ (it would be impossible for a flow-shop that operates 365/366 days a year, but the general principle of the method would remain valid, requiring only more important stocks).

For readability reasons, weekend days were suppressed from the following figure. We will only present here the results of solution C, whose diagrams are given in Figure 13.

In this solution, the simulated average weekly production reaches 336 m², about 67 m² per day, which corresponds to the actual average value of the usual volume of production of the firm. This validates our model.

In Figure 13, the first peak on the finished goods curve corresponds to the creation of the buffer stock.

This stock is created by working overtime during one weekend. We observe the level of this stock the week preceding the beginning of the removal at the engraving workshop. Then, we observe the shift of the buffer stock, which proceeds along the production line from the downstream toward the upstream (routing machine, engraving, then electro plating; the other Work-In-Process stocks have not been represented).

The second peak of production of finished goods, in week 6, corresponds to a correction of a cyclic accumulation at the entrance of the silk screen-printing workshop, which has periods of programmed cyclic maintenance. Besides, we note that values of Work-In-Process stocks are extremely steady throughout the removal.

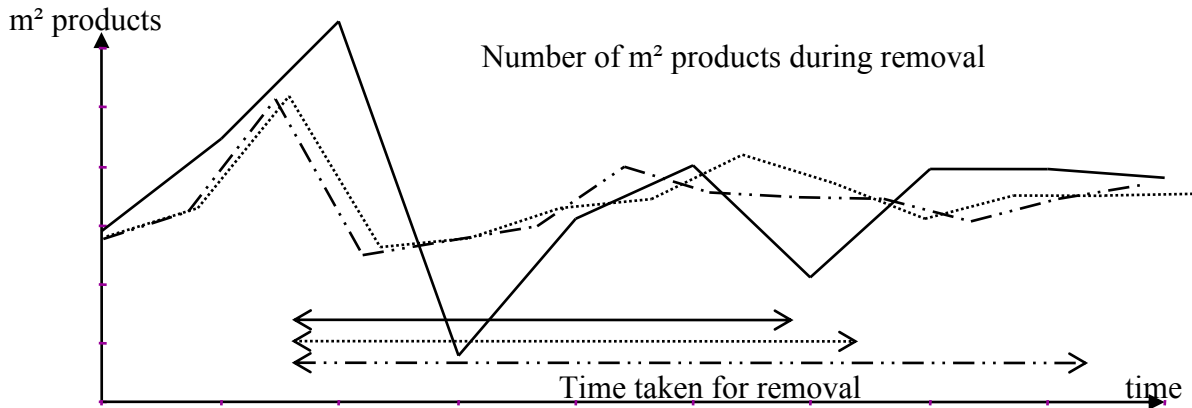


Figure 13. Simulation results for strategies A (/), B (.) and C (- .).

6.2. Comparative analysis according to the strategies of machine grouping

We achieved a graphic representation of the results obtained with each strategy, showing for example two performance criteria: the removal time and the Work-In-Process buffer stock level. These two criteria do not have the same significance. The importance of the average buffer stock could be interpreted from an economical point of view but it represents mostly the criterion of customer service: the smaller the buffer stock is, the earlier the delivery dates.

However, other criteria were used, like the regularity of stocks, which shows that the control of an operation in strategy C is easier because less sensitive to small perturbations. These criteria have been examined cautiously. For example, the increase of the removal time, which maintains the firm in an unstable situation, is *a priori* a bad thing.

Yet, it can be observed that a more grouped removal is advantageous, because of the smaller number of machines to move in each step. It is in our case very important, because the firm wants to use its own staff and use the less possible external workforce.

Such criteria must be elaborated case by case for each company, according to its own specificities and priorities, and cannot be generalized.

Other data were obtained through simulations, such as the increase of the path length of the products flow, the cost of transport of Work-In-Process and the modification of the machine rates.

Strategy	Total stocks created (m ²)	Flexibility indicator	Regularity of Work-In-Process	Regularity of deliveries	Processing time		
					Less than 48 h	2-7 days	7 days more
A	1128	*	**	*	20%	58%	22%
B	987	**	****	**	62%	30%	8%
C	611	****	****	***	68%	26%	6%

: poor; **: average; *: good; ****: excellent*

Table 2. Description of the results of strategies A, B and C.

6.3. Presentation of the results

We can show the performance of the suggested heuristic through the results presented in the table 2. It shows one of the criteria, the increase of the processing time, according to the different strategies (i.e. time passed in the production line). This criterion is very important in view of the responsiveness of the firm and we can observe that for other strategies, it can be extremely damaging.

The final choice of the right strategy depends on a global evaluation, taking into account all the criteria. All these criteria are obtained through simulations and are used as indicators. This evaluation linked to an economical study allows evaluating the efficiency of a given removal strategy, which allows flexibility in the project management and at the same time a removal without any consequences on the customers.

7. CONCLUSION

In this paper, we have approached a method dealing with unusual problem thanks to discrete events simulation. We dealt with the relocation of a production site from an old site to a new one. Relocation implies that the pre-existing production line, consisting of several machines, has to be totally dismantled from the old site and rebuilt later in the new one.

Thus, we propose a method for relocation when the production cannot be completely stopped during the removal of the machines. We have wanted that this method is described in a complete and formal way, in order to be easily reproducible. For that, we have chosen the DEVS formalism. The problem is tackled by partial removals at a time of either a single machine or a group of sequential machines. The choice between the two options is not trivial and must only be done after a careful analysis of the production line.

We first analysed the relocation problem. Then we have suggested a relocation methodology that takes into account different constraints. Our methodology is based on

balancing buffer stocks. After a study of the useful stocks' movements, we have given all the necessary DEVS models for the simulation. Finally, we have wanted to show how we can arrive to practises implementation on a real industrial case from a theoretical study on a canonical situation.

So, our approach was validated using an industrial case study based on DEVS formalism to model the real production system. Three types of models were developed for this purpose. The first type corresponds to production equipments models. The second model type enables controlling logical conditions. The last one enables to follow a projected planning and to organize the relocation scheduling simulation. Using the three model types, we have simulated several strategies of the production line segmentation. The simulation allowed determining the best configuration of the flow-shop relocation parameters.

Appendix A

DEVS (Discret Event system Specification) Formalism: Atomic DEVS model [27, 28]

Formally, an atomic model M (cf. figure A1) is specified by:

$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$: where :

X : $\{(p, v) | p \in \text{input ports}, v \in Xp\}$: the set of input ports through which external events are received

Y : $\{(p, v) | p \in \text{output ports}, v \in Yp\}$ the set of output ports through which external events are sent

S : the set of the states .

δ_{int} : internal transition function : $S \rightarrow S$. The model being in a state s at t_i , it will go in s' , $s' = \delta_{int}(s)$, if no external event occurs before $t_i + ta(s)$.

δ_{ext} : external transition function: $S \rightarrow S$. When an external event occurs, the model being in a state s during a time e , it will goes in s' . $s' = \delta_{ext}(s, e, x)$. At each change state e is reset to 0.

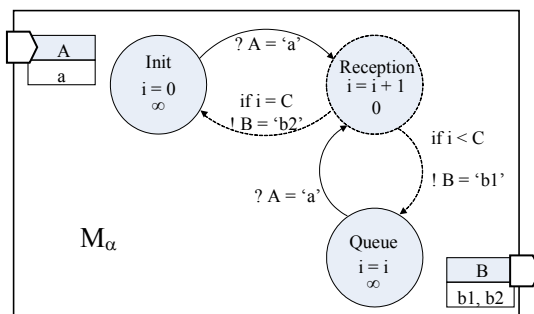
λ : output function : $S \rightarrow Y$. This function is executed before the internal transition function.

ta : the duration of a state if no external event occurs : $S \rightarrow \mathbb{R}_0^+ \cup \infty$.

Q : set of states of a model : $Q = \{(s_i, e) | s_i \in S, 0 < e < ta(s_i)\}$.

e : is the elapsed time in the state s .

The Information and Systems Science Laboratory (LSIS) has developed a graphic representation which allow to simplify the representation of the internal and external transition functions as well as the output function.



Legend: A state is represented by a circle, continuous for a stable state, dotted for a temporary state. This circle contains the name of the state, the operations on variables and the state lifespan (ta). (A) is an input port, which can receive the value (a). (B) is an exit port, which can take the values (b1) & (b2). A dotted line shows an internal transition. A continuous line shows an external transition. $? A='a'$ represents an external event at input port A with value 'a'; $!B='b1'$ represents an internal event at output port B with value 'b1'

Figure A1. Atomic DEVS Model

Appendix B

DEVS (Discret Event system Specification) Formalism: Coupled DEVS model [28, 29]
DEVS Formalism was extended in order to be able to associate in a hierarchical and modular way coupled and/or atomic models.

A coupled DEVS model (cf. figure A2) is defined by the following structure:

$MC\chi = \langle X, Y, D, \{M_d \mid d \in D\}, EIC, EOC, IC \rangle$. Where X : input events set ; Y : output events set ; D : is a set of components names ; M_d : DEVS model for each $d \in D$; EIC : $EIC \subseteq MC\chi \times M$ is the set of the external input couplings ; EOC : $EOC \subseteq M \times MC\chi$ is the set of the external output couplings ; IC : $IC \subseteq M \times M$ is the set of the internal couplings.

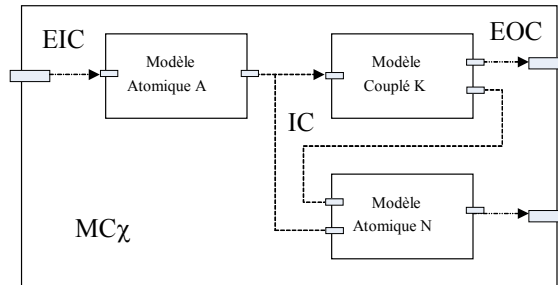


Figure A2. Coupled DEVS model

Reference List

- [1] A. Anglani, A. Grieco, E. Guerriero, F. Nucci, Q. Semeraro and T. Tolio, Object oriented simulation Models based on the DEVS formalism, in: 14th European Simulation Multiconference, May 23-26, 2000, Ghent, Belgium.
- [2] N.J. Aquilano, R.B. Chase and M.M. Davis, The fundamentals of operations management (Irwin, 2nd Ed., 1995).
- [3] R. Batta and W.V. Huang, On the synthetis of adversing and relocation decisions for facility, Computers and Industrial Engineering, 16:1 (1989) 179-187.
- [4] G. Bel and J.P. Kieffer, Pilotage assisté par la simulation discrète, in: P. Pujo and J.P. Kieffer, ed., Méthodes du pilotage des systèmes de production (Edition Hermès, Traité IC2 Productique, Paris, 2002) 99-127
- [5] J. Brimberg and G.O. Wesolowsky, Facility location with closest rectangular distances, Naval Research Logistics 47:1 (2000) 77-84.
- [6] B.K. Choi, B.C Park and J.H. Park, A formal model conversion approach to developing a DEVS-based factory simulator, Trans. of the Society for Modeling and Simulation International 79:8 (2003) 440-461
- [7] P.A. Fishwick, Computer simulation: growth through extension, Transactions of the Society for Computer Simulation International 14 (1997) 13-23.
- [8] C. Frydman, M. Le Goc, N. Giambiasi and L. Torres, Knowledge-Based diagnosis in SACHEM using DEVS models, Transaction of SCS 18:3 (2001) 148-159.
- [9] J. Heizer and B. Render, Operations management (Prentice Hall, 1999).
- [10] N. Giambiasi, B. Escude and S. Ghosh, G-DEVS: a generalized discrete event specification for accurate modelling of dynamic systems, Trans. of S.C.S.I., 17(3) :120-134, Aug. 2000.
- [11] N. Giambiasi, Abstraction à évènements discrets des systèmes dynamiques, European Journal of Automation, 32(3) :275-311, 1998.

- [12] W.V. Huang, R. Batta, and A.J.G. Babu, Relocation promotion problem with Euclidean distance, *European Journal of Operational Research* 46:1 (1990) 61-72.
- [13] J. Kosturiak and M. Gregor, FMS simulation: some experience and recommendations, *Simulation Practice and Theory* 6 (1998) 423-442.
- [14] J.K. Lee, M.W. Lee and S.D. Chi, DEVS/HLA-based modeling and simulation for intelligent transportation Systems, *Trans. of the Society for Modeling and Simulation International* 79:8 (2003) 423-439
- [15] B.M.T. Lin and S.S. Tseng, Generating the best K sequences in relocation problems, *European Journal of Operational Research* 69:1 (1993) 131-137.
- [16] G. Moon and G.P. Kim, Effects of relocation to AS/RS storage location policy with production quantity variation. *Computers and Industrial Engineering* 40 (2001) 1-13.
- [17] R. Muther, *Systematic Layout Planning* (VNR Company, 1969).
- [18] S.R. Nidumolu, N.M. Menon and B.P. Zeigler, Object-oriented business process modelling and simulation: a discrete event system specification framework, *Simulation Practice and Theory* 6 (1998) 533-571.
- [19] L.K. Nozick and M.A. Turnquist, Integrating inventory impacts into a fixed-charge model for locating distribution centers, *Transportation Research Part E* 34 (1998) 173-186.
- [20] L.K. Nozick, The fixed charge facility location problem with coverage restrictions, *Transportation Research Part E* 37 (2001) 281-296.
- [21] H. Pierreval and J.L. Paris, From 'simulation optimization' to 'simulation configuration' of systems, *Simulation Modelling Practice and Theory* 11 (2003) 5-19.
- [22] H. Praehofer, System theoretic formalisms for combined discret-continuous system simulation, *Int.Journal of General Systems*, 19 :219-240, 1991.
- [23] P. Pujo and J.P. Kieffer, Concepts fondamentaux du pilotage des systèmes de production, in: P. Pujo and J.P. Kieffer, ed., *Fondements du pilotage des systèmes de production* (Edition Hermès, *Traité IC2 Productique*, Paris, 2002) 25-50.
- [24] M. Rheault, J.R. Drolet and G. Abdunour, Dynamic cellular manufacturing system. *Computers and Industrial Engineering*, 31:1/2 (1996) 143-146.
- [25] V. Ricciardi, P. Pujo and C. Frydman, DEVS modelling for the proactive control by simulation of kanban production lines, in: *Proc. MAS2003* (Berggeggi, Italie, 2003).
- [26] J. Tompkins and J. White, *Facilities planning* (Wiley, 1984).
- [27] B. Zeigler, *Theory of Modelling and Simulation* (Ed. John Wiley, New York, 1976).
- [28] B. Zeigler, *Multifaceted Modelling and Discrete Event Simulation* (Academic Press, London, 1984).
- [29] B. Zeigler, H. Praehofer, T.G. Kim. *Theory of Modeling and Simulation*. 2nd Edition, Academic Press, New York, NY (2000)