



HAL
open science

A Methodological Approach to Build Simulation Models of Manufacturing Systems with Distributed Control

Hind El Haouzi, André Thomas

► **To cite this version:**

Hind El Haouzi, André Thomas. A Methodological Approach to Build Simulation Models of Manufacturing Systems with Distributed Control. International Conference on Industrial Engineering and Systems Management, IESM05, May 2005, Marrakech, Morocco. pp.10. hal-00082413

HAL Id: hal-00082413

<https://hal.science/hal-00082413>

Submitted on 27 Jun 2006

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.

A Methodological Approach to Build Simulation Models of Manufacturing Systems with Distributed Control

Hind El Haouzi, André Thomas

Centre de Recherche en Automatique de Nancy. CNRS UMR 7039
CRAN-GGP Faculté des Sciences – B.P. 239 54506 Vandoeuvre les
Nancy Cedex France {Hind.elhaouzi ; Andre.Thomas}@cran.uhp-nancy.fr

Abstract

Simulation is a powerful tool to analyse manufacturing systems for purposes of design and on-going operation. Nowadays, simulation modelling and analysis is confronted to increasing evolution of companies towards distributed process control. Thus, simulation modelling must be supported by an underlying modelling discipline, or structured approach to modelling factory operations. Our study focuses on simulation of industrial systems with distributed control. we propose a structured approach to build the simulation models. This approach is based on methodology ASDI (analysis-specification-design-implementation) and it is independent of any platform or software tool. We will illustrate the use of this approach in an assembly line manufacturing application.

Keyword

Methodological approach, distributed vs centralised control, discrete-event simulation, reference model.

1. Introduction

The will of an enterprise to answer increasingly varied demands on time and with low cost, has highlighted the weaknesses of a purely hierarchical structure of decision in production management. In such organisational structures, decisions generally become more detailed and are taken in shorter time periods as one comes closer to the physical system. This approach corresponds to centralised control. Today manufacturing systems are evolving toward distributed control, resulting in the birth of several concepts as Holonic systems [1] technology. In this context, system entities have enough autonomy to make decisions. In order to evaluate the decision impact or to choose a management production organisation rather than another, it is natural to use discrete-event simulation. Law and Kelton [2] summarise some reasons for the use of simulation in manufacturing systems. While simulation has many strengths, it has limitations that must be addressed to improve its effectiveness: it must be supported by an underlying modelling discipline or structured approach [3].

In this paper, we propose a structured approach to build simulation models for manufacturing systems with distributed control. This approach is based on the ASDI (Analysis-Specification-Design-Implementation)[4][5] methodology for several reasons that will be explained thereafter. The remainder of this paper is organised as follows. In section 2, we underlay two major problems which have spurred interest in our approach. In

section 3 we will explain the proposal. Section 4 illustrates the use of the proposal approach in an assembly line manufacturing application. In section 5, we conclude with synthesis of the study works achieved and future research directions.

2. Problems

In modelling and simulation of industrial systems, several problems are encountered like formalisation, understanding, centralisation and maintenance. In this paper, we develop only two problems: building a reusable model and model modularity.

2.2.1 Simulation Model Reuse

A major problem encountered during the analysis and modelling phase is to build reusable models or reusable basic components. Figure 1 illustrates different levels of reuse according to the axes: complexity and frequency. As an example of reuse, to build simulation models of systems that have common properties, instead of studying every system and develop a specific components, the study or analysis of a system class leads to develop in a first time a reference model (see figure 2). Reference models can be seen as a standardised representation for a problem domain such as manufacturing. From this global vision we can extract a specific vision dedicated to simulation models for one system of this class.

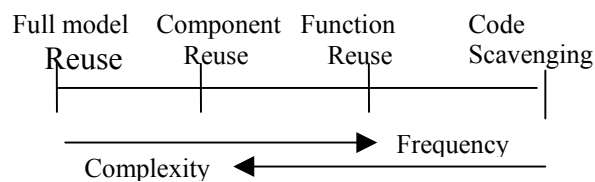


Figure 1 : A spectrum of reuse [8]

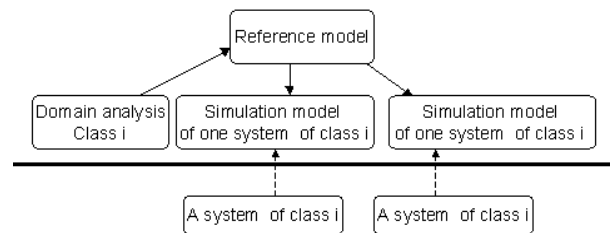


Figure 2 : Example of reusability

This interest is not new, a lot of papers focusing on reusability issues appear in major conferences [6][7][8]. Recently, S. Robinson [9] gave a definition to the term “Simulation model reuse”. He highlights the motivation to develop reusable models, the validity, the cost and credibility of reusability. This problem is partially solved by existing modelling methodologies as CM (Conical Methodology)[10], IDEF[11], GRAI [2] but they focus on knowledge formalisation and they enable the modelling of industrial systems with a point of view that is not easily translatable to simulation models [13]. In the opposite, a preoccupation with federated modelling and High-Level Architecture (HLA)[14] has created a myopia concerning reusability but essentially at the implementation level.

2.2.2 Modularity

Another problem currently encountered by simulation models designers is low modularity. For example, if we wish to simulate a new production management organisation with the same physical factory model. It is often necessary to achieve some significant modifications on the model. Hence, we need to subdivide the manufacturing system along three main subsystems: the physical factory subsystem, the informational subsystem and the decisional subsystem. This decomposition of the manufacturing production system is close to the “Le moigne” one [15] (figure 3). The informational subsystem is essentially a vector of integration or communication between the two other subsystems.

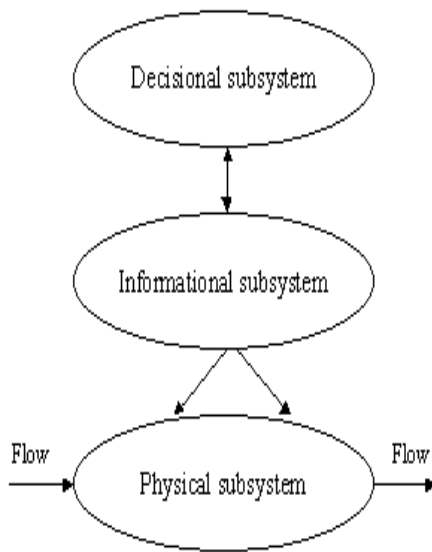


Figure 3: « Le Moigne »

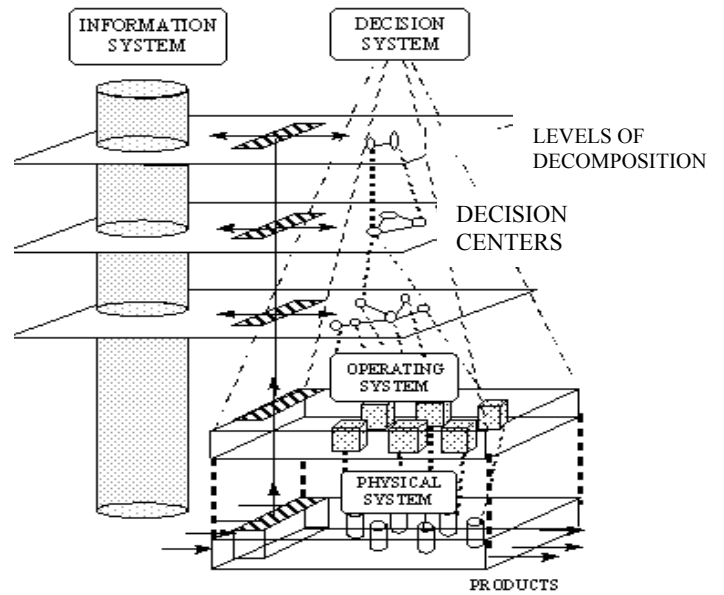


Figure 4: GRAI

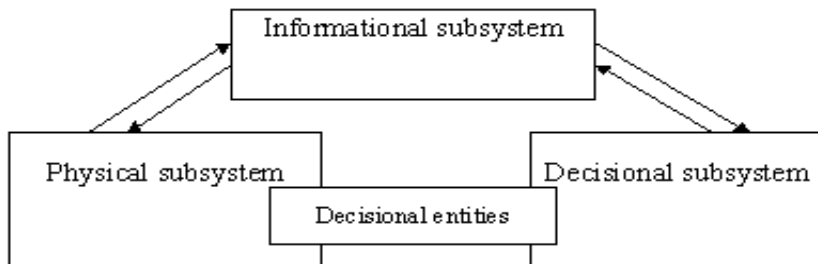


Figure 5: ASDI-dc

The GRAI methodology proposes an other systemic approach based on the system theory and on the hierarchical system theory [16]. Figure 4 illustrates this vision. In the opposite to the “Le Moigne” approach, the informational subsystem is placed in the same level of the decisional subsystem and the physical subsystem. Moreover, both approaches (“Le Moigne” and GRAI) lead to highlight different flows through the system, but don’t take into account the distribution aspect. In systems with distributed control, some physical units have their own decision-making centre. We named decisional entities the result of association between a physical unit and a decision-making centre. In modelling and analysis system decisional entities must be stressed and defined through their behavioural model (figure 5). This vision is more detailed thereafter.

3. Proposal

3.1 Methodological Approach ASDI-dc

While ASDI, CM methodologies resolve problems of modularity and formalisation, they disregard the distribution aspect. The HLA methodology can solve the problem of distribution, but it remains only at the implementation level. The MAMA-S (Multi-Agent

Methodological Approach for Simulation) methodology proposed by Gallan [13] gives a framework for building a consistent simulation platform from many independent simulation models but does not represent distributed control within the model. Faced to this limit, in the ASDI-dc approach (dc mean distributed control) we propose to add “decisional entities”(see figure 5). This approach is based on the ASDI methodology for several reasons.

- The ASDI approach was improved by recent works of simulation modelling as ASDImi (implementation-multiple- incremental).
- Its systemic view subdivides the manufacturing systems in 3 subsystems (physical, informational, decisional) enables to improve design, understanding and modularity of simulation models.
- Its life cycle is close to the CM cycle and is widely accepted in the scientific community [10].
- The ASDI approach uses the modelling process proposed by Gourmand [17] which requests:
 - To develop a knowledge model. The goal of this model is to organise knowledge about a class of systems or problems
 - To develop an “action” model (implementation of the knowledge model).
- This decomposition will ensure the independence between the analysis phase and the choice of the tool or language used in the implementation.

ASCI proposes to use the object paradigm. Indeed the object paradigm is a natural extension of the systemic approach with the use of concepts like encapsulation, heritage, and aggregation or composition [18]. They authorise a potentially high number of users to apprehend and deploy easily, and viably systemic approach. Used throughout modelling, the object-oriented approach ensures a globally coherent modelling process.

3.2 ASCI-dc Life Cycle

In the ASCI-dc modelling process, we propose to distinguish two abstraction levels: the analysis and implementation levels. Each of those levels is composed by two main parts: the first concerns the systems domain, the second is focused on in one instance of this domain.

3.2.1 Analysis Level

Domain : *-Analysis phase:* The aim of analysing domain is to develop a reference (or generic) model of knowledge. We analyse the three subsystems, decisional entities, and the communication model.

- *The logical subsystem* is composed by information about raw materials, components, bills of material , and the associated set of manufacturing routing etc.
- *The physical subsystem* is composed by machines, resources, stock areas, conveyors. A machine includes a non-empty set of stations and/or working posts.
- *The decisional subsystem* has the role to control and to take decisions in the company (strategic, tactical, operational or reactive).
- *The communication model:* It is necessary to develop the model of communication between the three subsystems. It permits to describe interactions between objects of subsystems.
- *Decisional entities:* To answer the problems mentioned in section 2 (reuse and modularity), we must define entities that are able to take distributed decisions. We named these entities “decisional entities” (see figure 5). Each decisional entity is

composed by a physical part (like resources or station) and a decisional part; the decisional part controls the physical one according to the behavioural model. A decisional entity have two different types of attributes: external and internal. In figure 6, we propose a meta-model of those entities.

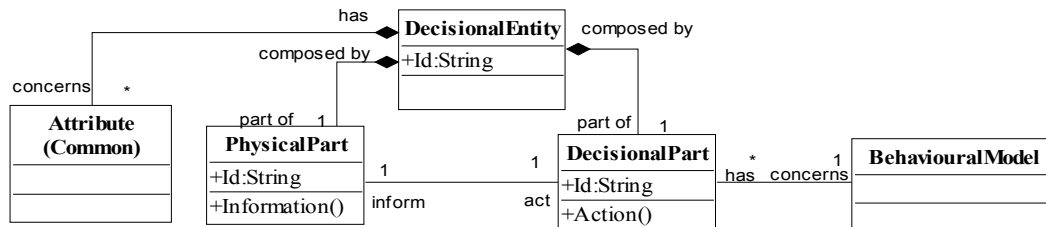


Figure 6: Meta-model of decisional entity

-Specification phase: during the specification phase, the functions and the behavioural model of the system entities are described. This phase was often confounded with analysis one. Indeed, we propose to describe a behavioural model of decisional entities: to ease their implementation by use of mathematical formalisms (matrix of decision, chain of Markov, Automaton....).

System: *Analysis and specification:* in those phases, the experts use the reference knowledge model to analyse and specify their industrial system. This reference model can be adopted at the particular system by specifying the functionality of some system parts, add method or attributes.

3.2.2 Implementation Level

Domain: *-Design phase:* In design phase, we define rules to build action models or basic components; it is very important step for reusability. We will explain this fact in the case study. *-The Implementation phase:* leads the build of the software components library that will be used automatically to generate models for one system of the studied class. Concerning the decisional entity, we propose to separate the physical part and the decisional one. Different problems, like modification of the control strategy, will be simplified.

System: *Implementation phase:* The last phase is about action model implementation using the software library components. This action model will be used to evaluate system performances.

5. Case Study

5.1 Presentation

We were interested in a company which provides indoor comfort systems and comprehensive facility solutions for residential, commercial and industrial building needs. The context production is as follows: The shop floor is organised in pull production. To ensure a better balancing of an assembly line every worker must be able to work at three workstations: its workstation (i.e. initial affectation), the upstream workstation, and the downstream workstation. The objective is to model an assembly line using Arena software [19] and deploy it at all company assembly lines. The control system is globally centralised; i.e. every week a company compiles the Master Production Schedule which proposes manufacturing orders; but some decisions are made locally by workers that represent the distributed aspect of decision. In order to resolve a worker mobility problem we can use two approaches. In the first one, we simplify modelling by using a “capacity

factor”, for example, if an operator spend 70% of his time at work station w_i and 30% on the w_{i+1} or w_{i-1} , at the first work station we will consider that the available capacity will be 70% of theoretical capacity and 30% at the second. This approach corresponds to the centralised one. The second way is to consider operator as decisional entity, therefore, we take into account distributed aspect. That corresponds to the distributed control. In both approaches (centralised and distributed control), all necessary data for simulation (manufacturing time, products information, initial affectation of workers etc) are saved in a data base and transferred to the simulation model. We have Modelled the system using the two approaches. This enables to compare distributed and centralised control results.

5.2 Domain Analysis Phase

To structure our analysis we describe a domain studied in natural language, then we formalise it in UML in order to get a reference model of the domain. (see figure 7 and 8).

In the decisional subsystem we represent a relational structure between organisational decision-making centres. We distinguish two types of decision-making centres: centralised system control centres and distributed decision centres. The centralised system control centre can make decision in the short (operational), medium (tactical) or long (strategic) horizon. the Relation between these centres can be hierarchical or at the same level. The distributed decision centres can represent the decisional parts of decisional entities

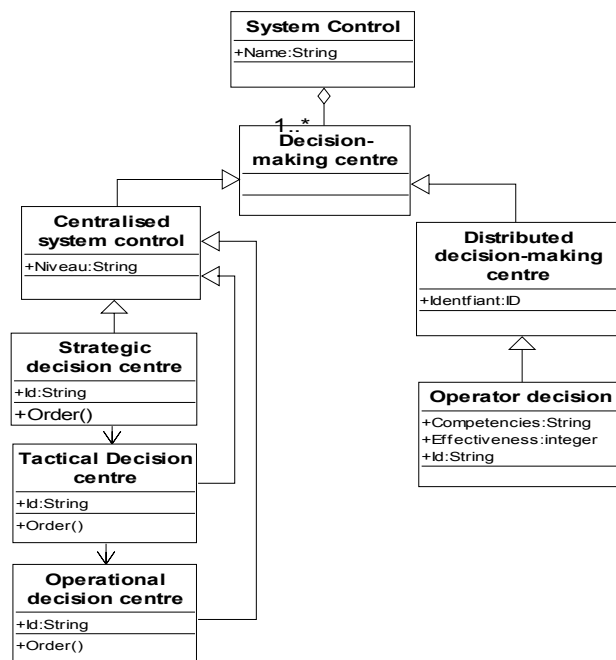


Figure 7 – Decisional subsystem of an assembly line

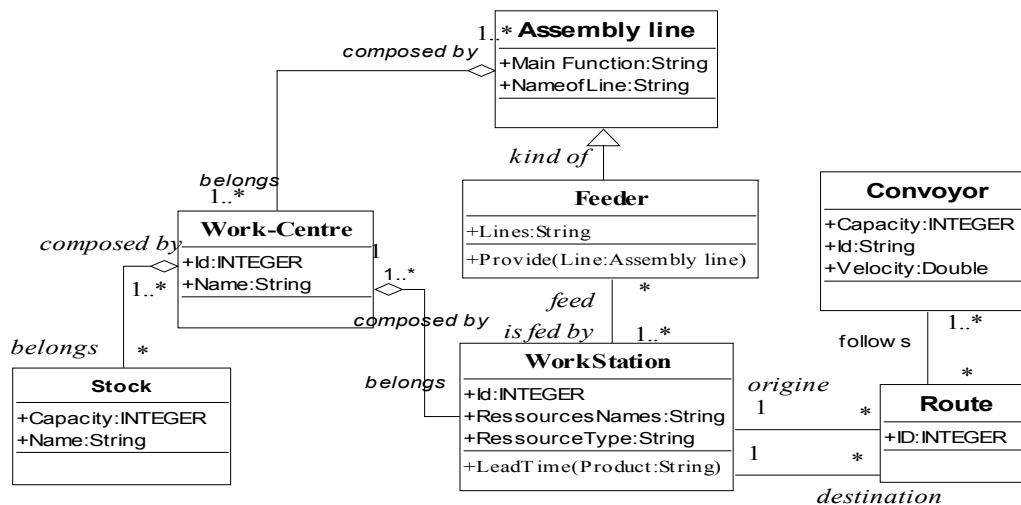


Figure 8 : Physical subsystem of an assembly line

The physical subsystem concerns an assembly line. As we can see in figure 7, each line is composed by a set of work centre each work centre is composed by one or several work stations. A work station can be connected to a feeder etc. The communication model can be developed by the use of UML Sequences. It describes interactions and the communication between the three systems. (In this paper we don't present the informational subsystem).

5.3 Specification

In the first step an operation of filtering the domain reference knowledge model has been achieved to extract the particular vision dedicated to the simulation. Then, we added attributes and methods to the existing objects for the simulation models implementation. As we said before, in this phase, functions and behavioural model of system objects will be described. In this paper, we show only behavioural models of decisional entities.

The algorithm describing operator decision process :

- Operator j takes decisions according to variables of his environment that are: (P_{wi}) and (S_{wi}) with w_i indicate initial work station of operator.

- Let P_{wi} be the availability of the product : $P_{wi} \in \{0,1\}$ with $P_{wi}=1$ product in queue i ; $i=0$ there is no product in queue
- Let S_{wi} be the signal of work $S_{wi} \in \{0,1\}$ with $S_{wi}=1$ work station state (w_i) is free, $S_{wi}=0$ work station (w_i) is busy.
- Let D_{wij} be a decision of operator j in work station w_i . $D_{wij} \in \{-1, 0, 1\}$
- with $D_{wij} = -1$ upstream displacement. $D_{wij} = 0$ no displacement. $D_{wij} = 1$ downstream displacement.
- Let O_j be: operator (j) has as attributes W_i initial workstation competencies $C_j = \{C_1, C_2, ..C_n\}$ effectiveness $E_j \in \{0..n\}$.
- Let $Affect(O_j, w_i)$ be: operator j work in workstation i
- Let $Apt(O_j, w_i)$ be: a Aptitude of operator j to work at workstation i ($Apt(O_j, w_i)=0$ operator j don't has competence for work at workstation i).

- The decision matrix for an operator i in workstion w_i is the following:

$Dw(i,j)$	Case 1	Case2	Case3	Case4
S_{wi}	0	0	1	1
P_{wi}	0	1	0	1
D_{wij}	1	1	-1	0

Cases 1 and 2 can be regrouped into one case. If Operator works station is busy the operator moves downstream. Indeed in pull production shop floor the operator pull flow in priority.

- We simplify this decision matrix by a simple function:

$$Dw(i,j) : P_{wi}, S_{wi} \in \{0,1\} \longrightarrow D_{wij} \in \{-1, 0, 1\}$$

$$D_{wij} = S_{wi} \times (P_{wi} - 2) + 1$$

Decision algorithm:

- (1) Intialisation
 - Affect (O_j, w_i)
 - $C_j = C_j = \{C_1, C_2, ..C_n\}$
 - $E_j = X$ // (X will be in $\{0..n\}$)
- (2) Waiting Event
 - If Event then $D = Dw(i,j)$ // execute decision matrix
- (3) Research


```

While (Stop = False)           //while Stop condition is false
  If Apt(Oj,wi+D) = 0           //we test the aptitude of operator j to work in station wi+D
  Then wi=wi+D                 //wi=wi+1 or wi=wi-1 according to D
  Else Stop=True               //when Apt(Oj, wi+D)=1 then stop condition will be true
  Affect(Oj,wi)                //we affect the operator j at work station wi
End While

```

5.4 Domain Design : Rules of Building the Action Model

In the domain design phase, we define rules for building the “action model” basic component

The following table show the more important objects:

Domain Objects	Arena Objects
Work Centre	Process
WorkStation	Ressource
Stock	Queue
Path	Route
Decisional Entity	VBA /ressource
Convoyer	Convoyer
Assembly line	Arena Model

Other rules concern the name of Arena objects, for example : a work station resource is named Mij when i is a work centre and j is a work station in work centre i. (see page 6)

Tij : theoretical operational time in work station j work centre i

Qij: queue in work station j and work centre i

5.5 Domain Implementation :

We design a basic components library: Object “work centre” that is materialised by a sub-model or process in Arena. Each process has one or several objects “workstations”. This architecture enables to keep a hierarchical vision of the assembly line (figure 10). For the distributed control : Decisional entity is the worker, the decisional part is represented by a VBA block; When a product arrives at the VBA block, the control of the entity is passed to the VBA Sub procedure. A corresponding Sub procedure is created in the Visual Basic Editor and it describes the decisional entity behaviour. The physical part is materialised by a resource in Arena (figure 11).

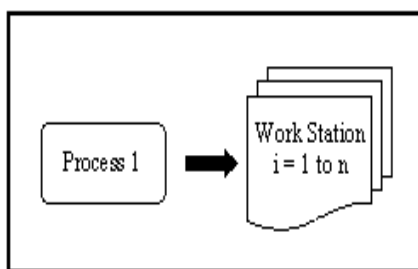


Figure 10: Work centre in Arena

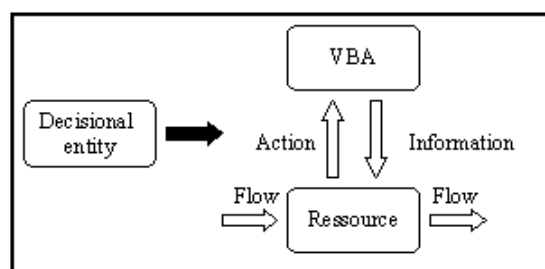


Figure 11: Decisional entity

5.6 Implementation : Action Model

In the following section, we use the components library to create the action model through Arena software facilities. *-Physical subsystem* is built in Arena. *-Decisional subsystem* (or control) is developed in Visual basic for application. *-Informational subsystem*: All products information are saved in Microsoft Excel. *-Communication models*: For more modularity we develop an independent communication module in Visual Basic (i.e. Arena / Excel). *-Decisional Entities*: We used a basic component developed in the domain implementation phase.

5.7 Tests and Validation

In any process modelling, the tests and validation phases are very important, it enables to measure the confidence interval between the simulated results and the actual data. The figure 15 and 16 present two tests done on sample of 15 products.

Figure 15 : Centralised case

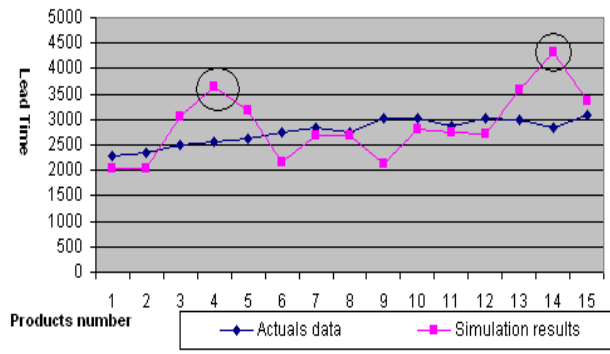
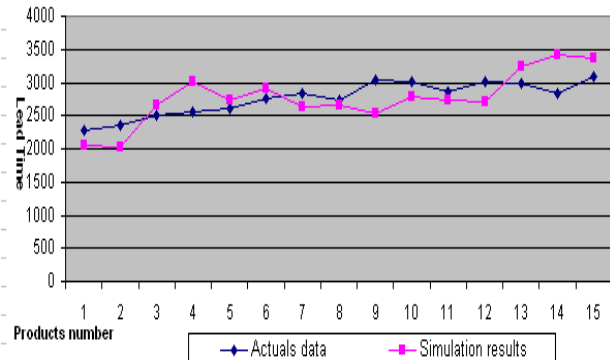


Figure 16 : Distributed case



Picks on the figure 15 show that if a problem occurs at a work station w_i , it has influence on the time passing at stations w_{i-1} and w_{i+1} . This problem is due to workers mobility. Consequently, we use the distributed approach to refine the simulation model. We obtained with the same sample the following results (figure 16). The distributed case results (figure 16) show that it will be possible to improve the shop floor simulation model. Effectively, the variability of the lead time was reduced.

6. Conclusion and Perspectives

The study of different simulation methodologies enabled us to highlight limitations that must be addressed to improve their effectiveness in a distributed control context. The ASDI methodology enables to answers formalisation, modularity and “reuse” problems. But it doesn't take into account the distribution aspect. Our contribution concerns all ASDI life cycle phases in the definition of the decisional entities and their conceptual model (i.e. meta-model). In design and implementation, we used Arena and Visual Basic software for application to develop a generic component for control. But, we can use the same concept with other software. The question of model validity looms large and cannot be ignored. It seems widely accepted in the simulation community that models or modelling approaches cannot be fully validated. It makes sense to have some form of quality assurance so as to ensure that a model is fit for its intended purpose. In the future works we will be able to give a more formal setting for this approach.

7. Bibliography

- [1] Koestler A.(1967) *The Ghost in the Machine*. Arkana, London
- [2] A. M. Law and W. D. Kelton, *Simulation Modeling & Analysis*, McGraw-Hill, 2e edition, 1991.
- [3] Douglas A. Bodner and Leon F. McGinnis Keck “A Structured Approach to Simulation Modeling of Manufacturing Systems. Engineering Proceedings of the 2002 Industrial Engineering Research Conference. pp
- [4] P. Kellert et C. Force (1998a), “Méthodologie de Modélisation Orientée Objets de Systèmes de Production – Application à une Chaîne d’Etuvaage de Bobines d’Allumage” *Journal Européen des Systèmes Automatisés*, 32(1), pp. 107-131.

- [5] P. Kellert et S. Ruch (1998b), “ Méthodologie de Modélisation Orientée Objets de Systèmes de Production – Un Processus de Construction/Validation du Modèle Générique Orienté Objets d’un Système de Production ”. Journal Européen des Systèmes Automatisés, 32(1), pp. 51-105.
- [6] R.G Sargent, Issues in simulation model integration, reusability, and adaptability, in: J.Wilson, J.Henriksen, S.Roberts (Eds.), Proceedings of the Winter Simulation Conference, Institute of Electrical Electronic Engineers, Piscataway, NJ, 1986, pp.512-516
- [7] R.Reese, D.L. Wyatt, Software reuse and simulation, in:A. Thesen, H.Grant, W.D. Kelton (Eds), Proceedings of the Winter Simulation Conference, Institute of Electrical and Electronic Engineers, Piscataway, NJ, 1987, pp.185-192.
- [8] D.L Wyatt, A framework for reusability using graph-based models, in:O. Balci, R.P. Sadowki, R.E. Nance (Eds), Proceeding of the 1990 Winter Simulation Conference, Institute of Electrical and Electronic Engineers, Piscataway, NJ, 1990, pp. 472-476
- [9] Stewart Robinson, Richard E. Nance, Ray J.Paul, Michael Pidd, Simon J.E Taylor, “Simulation model reuse: definitions, benefits and obstacles (Simulation Modelling Practice and Theory 12 (2004) pps 479-494.
- [10] R. Nance (1994)”The Conical Methodology and the Evolution of Simulation Model Development”. Annals of Operations Research, 53, pp. 1-45.
- [11] US Air Force (1993a), “ Integrated Computer Aided Manufactured Definition Language(IDEF methods) ”. Department of Commerce, National Institute of Standards and Technology, Computer Systems Laboratory. Rapport technique, Gaithersburg, USA.
- [12] G. DOUMEINGTS (1984), Méthode GRAI : méthode de conception des systèmes productiques. Thèse de doctorat, Laboratoire d’Automatique et de Productique, Université Bordeaux I.
- [13] S. Galland et F. Grimaud (2000), “Methodological approach for distributed simulation: Life cycle of MAMA-S”. In ASIM-workshop 20/21.3 2000 - Multi agent systems and Individual-based simulation.Germany, pp. 83.
- [14] US Department of Defence (1996), “High Level Architecture Federation Development and Execution Process (FEDEP) Model, version 1.0 ” Defense Modeling and simulation Office. (A Technical Report)
- [15] J.-L. Le Moigne 1992, “ La modélisation des systèmes complexes ”. Editions Dunod.
- [16] M. Mezarovic, D. Macko et T. Takahara 1970, “Theory of hierarchical, multilevel systems”. Academic Press.
- [17] M. Gougand 1984, Outils Logiciels Pour L’évaluation Des Performances Des Systèmes Informatiques :Thèse de doctorat, Université Blaise Pascal, Clermont-Ferrand, France.
- [18] G. Booch, I. Jacobson, J. Rumbaugh et al. 1997, Unified Modelling Language Specifications - version 1.1 “. UML consortium - Object Management Group. Rapport technique.
- [19] W. Kelton, R.P. Sadowski and D.A. Sadowski 1998 “ Simulation with Arena.”McGraw-Hill.