



HAL
open science

Contribution to reusability and modularity of Manufacturing Systems Simulation Models: application to distributed control simulation within DFT context

Hind El Haouzi, André Thomas, Jean-François Pétin

► To cite this version:

Hind El Haouzi, André Thomas, Jean-François Pétin. Contribution to reusability and modularity of Manufacturing Systems Simulation Models: application to distributed control simulation within DFT context. *International Journal of Production Economics*, 2008, 112 (1), pp.48-61. 10.1016/j.ijpe.2006.12.067 . hal-00121149

HAL Id: hal-00121149

<https://hal.science/hal-00121149>

Submitted on 22 Dec 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.

**Contribution to reusability and modularity of Manufacturing Systems Simulation Models:
application to distributed control simulation within DFT context**

H. El Haouzi ^{*}, A. Thomas, JF. Pétin

Université Henri Poincaré

Centre de Recherche en Automatique de Nancy (CRAN, UMR 7039)

NANCY university, CNRS

Faculté des Sciences - B.P. 239 54506

Vandoeuvre les Nancy Cedex France

Tel.: +33 (0)3 83 59 5647

Fax: +33 (0)3 83 59 5644

[{hind.elhaouzi} {andre.thomas} {Jean-francois.petin} @cran.uhp-nancy.fr](mailto:{hind.elhaouzi}{andre.thomas}{Jean-francois.petin}@cran.uhp-nancy.fr)

Abstract:

Requirements for manufacturing control evolve from traditional centralized approaches where decision making is hierarchically broadcasted to more complex distributed control architectures involving autonomous entities and processes. Moreover, manufacturing processes are facing to standardisation and globalisation such as promoted by the Demand Flow Technology (DFT) concepts. In order to evaluate these new architectures, discrete-event simulation seems the most appropriate tool. However, complexity of distributed architectures and DFT standardisation requires introducing modularity and reusability in the modelling process. This paper deals with a methodological approach, based on ASDI (analysis-specification-design-implementation), to develop a library of generic simulation components that can be, as automatically as possible, instantiated into a modular simulation model. This approach is illustrated using an industrial case study where simulation aims at evaluating the impact of operator's flexibility induced by DFT context.

Key words: Discrete-event simulation, model reusability, model modularity, distributed control, ASDI.

1.Introduction

Today manufacturing systems need to be adapted to the internal (e.g. machine breakdown) as well as the external disturbances (e.g. changes in demands or product specifications). Consequently research in manufacturing system control has moved away from traditional centralized approaches where decision making was hierarchically broadcasted from the higher decisional levels down to the operational units to more distributed architectures. In this way, heterarchical architectures promote production control by distributing every decision capacities in autonomous entities, without any centralised view of the shop floor status. To ensure the consistency of a decision making, more pragmatic approaches are based on hybrid control which combines the predictability of the centralized control with the agility and robustness against disturbances of the heterarchical control. Holonic Manufacturing Systems (HMS) has been suggested as a concept for these futur manufacturing systems (Koestler, 1967).

In order to evaluate these new manufacturing systems or to choose a management production organization rather than another, Law and Kelton (1991) showed that discrete-event simulation is the more adaptable method (in the following, the term simulation will always be related to discrete-event simulation). While simulation has much strength, it is difficult to identify in a given model the different flows that are processed by the system.

Consequently, decisional and physical systems can't be separated in the model which is a serious limitation for evaluation of several control policies without a complete simulation model redesign. It emphasizes the need for an underlying modelling discipline or structured approach (Douglas and al, 2002) to guarantee modularity and thus facilitate modification on the model. Moreover, nowadays the majority of companies are evolving towards a standardisation of their various physical and decisional processes to ensure coherence and interoperability of their processes. In this way, TRANE Company, which is our industrial case study, chose to implement the Demand Flow Technology (DFT) principles (Costanza, 1996) to standardise all its 29 production sites. DFT methodology is a particular implementation of Just in Time concepts where, all production lines are structured in the same way in every shop floor . Consequently, all shop floor production lines have to be modelled in a similar way. This fact, justifies the need of reusability of process models to be used in simulation. Effectively, it is obvious that the time savings in simulation model design can be obtained if it is possible to reuse some simulation model modules to construct new assembly line models.

In this paper, we propose a structured approach (ASDI-dc) to build reusable and modular simulation models for manufacturing systems with distributed control. This approach is based on the ASDI (Analysis-Specification-Design-

Implementation), Kellert and Force, (1998a), Kellert and Ruch, (1998b) methodology which implements the object-oriented concepts and a systemic modelling framework to the simulation techniques. The main goal of our study will be to give a framework to generate automatically specific models from generic one by using standards objects and automated functions in DFT context . According to the Trane objectives, this framework could be used by a person who is not necessarily expert in software tools. The remainder of this paper is organized as follows. In section 2, we highlight the reusability and modularity challenges in distributed control context. Section 3 presents the proposed methodological approach. Section 4 describes an industrial application related to an assembly line manufacturing. In section 5 we will discuss the credibility of the reuse and the modularity . Conclusions and open issues for future research will be presented in section 6.

2.Reusability and modularity challenges

In order to face requirements for distributed control in simulation models, this section stress reuse and modularity challenges, and presents a comparative study between some simulation modelling methodologies.

2.1.Simulation models modularity

The concept of modularity in software development refers to the encapsulation of key sub-systems, objects and components behind well-defined and relatively stable interfaces. This facilitates the integration of these components with multiple applications, as well as the evolution or possible re-implementation of these components with minimal impact on the applications using them.

Nowadays, the concept of modularity has been introduced as a logical choice in the complex systems organisation and it can be applied:

- to gather elements (model elements, meta-model elements, as meta-meta-model elements),
- to classify group modelling elements in different subject areas,
- to allow the representation of the extension of such groups (ability to describe a group as an extension of other group),
- to allow nesting of such groups (in order to clarify organization),
- to allow the representation of the dependencies between such groups (ability to describe the fact that elements defined in a group).

In the area of system modelling, modularity is used to improve system understanding, to reduce the model complexity, and to facilitate the reuse of standardised components. For example, systemic view subdivides the manufacturing system in three main subsystems: the physical factory subsystem, the informational subsystem (called logical subsystem in some simulation methodologies) and the decisional subsystem (Le Moigne, 1992).

In distributed control context, we believe that one appropriate rationale to production system simulation should give a rise to modular models according to functional and structural criteria:

- The functional criterion aims to separate physical, information and control flows. This can be very useful in order to evaluate different control policies without changing the physical system model (figure 1a)



Fig. 1a: An example of functional criterion — Fig. 1b: An example of structural criterion

- The structural criterion aims to identify autonomous processes independently from their functional role. For example, in the case of distributed control, some entities or processes belong to physical subsystem and decisional one, rising the interaction and exchanges between the two subsystems. Thus, we must operate a particular merging that rises the autonomous processes having ability of local decision making and physical acting. An example of those processes can be a worker who has decisional capabilities to make a local load levelling decision as well as physical capabilities to operate physical tasks. Another example can be the intelligent product having information, decisional and communication capabilities, thanks to Radio Frequency Identification (RFID) technology (Lee et al, 2004), to make itself active in the scheduling and execution of its manufacturing operations.(figure 1b)

Unfortunately, the actual simulation tools do not provide natural constructs for those kinds of modularity. Indeed, traditional discrete-event simulation models contain both the description of stochastic behaviour of the physical processes and the decisional aspects linked to product creation and parameters tuning. Bridging the gap from the current practices in simulation toward modular simulation is the main goal of our approach.

2.2. Simulation Model Reuse

Reusability is one of the benefits of writing modular code, but it is also a separate goal in itself. For software objects, components or systems, to be reusable, their interfaces and implementations need to be designed so that functionality that is in an useful generic is separated from functionality that is specific to a particular application or domain. This allows the general functionality to be reused directly in other applications or domains, without the application-specific details getting in the way. This principle cuts across the interface and implementation of software elements for an object, component or service to be reusable in multiple situations, their interfaces need to be crafted using these principles, and their implementations need to be similarly organized, so that broadly useful methods and operations can be accessed separate from application-specific methods or specializations of methods. The interest of reusability in simulation community issues is not new, lots of papers focusing on it appear in major conferences and journals (Sargent, 1986) Reese and Wyatt, (1987) (Wyatt, 1990). Recently, S. Robinson et al (2004) 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. The figure 2 illustrates different levels of reuse according to two axes: complexity and frequency.

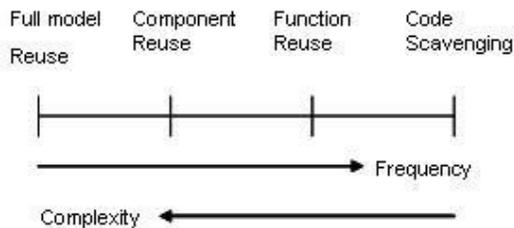


Fig. 2: A spectrum of reuse

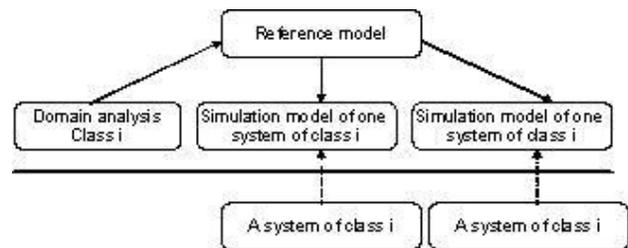


Fig. 3: An example of reusability

Reusability in the area of simulation models means capitalisation of generic knowledge about simulation models of systems that have common properties, instead of studying every system and developing specific components. This leads to generalise particular models through reference models (see figure 3) which provide generic and/or standardized representation for a given class of application. From this global vision we can extract a specific vision dedicated to simulation models for one system of this class.

Allowing reusability in simulation tools by providing a reference model for simulation of production system organised according to the DFT standard is one of our study's goals.

2.3.Literature review

In literature review, there are many persistent object-oriented methodologies for manufacturing modelling and simulation. In conceptual level, the reuse and modularity problems are partially solved by existing modelling methodologies as IDEF1X (Us Air Force,1993) and GRAI (Doumeingts, 1984), but they are focusing on knowledge formalization and they enable the modelling of industrial systems with a point of view that is not easily translatable to simulation models Galland and Grimaud (2000). In the opposite, a preoccupation with federated modelling and High-Level Architecture (HLA) (US Department of defence, 1996) has created a myopia concerning reusability but essentially at the implementation level.

Our aim is to give a methodological approach which involves conceptual and implementation levels. CM (Conical Methodology) (Nance, 1994), ASDI and recently ASDI-mi and MAMA-S (Multi-Agent Methodological Approach for Simulation) have explored this way. Initially Gourgand (1984) highlighted the benefit for developing a knowledge model which organizes knowledge about a class of systems (called the domain) or problems, and an “action” model that instantiates 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. CM and ASDI are based on this rationale to propose modelling frameworks providing the developers with guidelines that facilitate elaboration, verification, and validation of complex simulation models. ASDI life cycle is today widely accepted in the scientific community. ASDI proposes to use the object paradigm from analysis phases towards implementation and deploys structured and consistent systemic approach throughout modelling.

The methodology ASDI-mi (m for multiple, i for incremental), proposed by Sarramia (2002), adds to the initial ASDI methodology two main points: a multi-modelling approach for the domain and an incremental approach for modelling a system of the domain. Chabrol et al. (2006) propose a multi-agents modelling based on the Decisional sub-system (DSS) class diagram to model a particular domain related to Urban Traffic System.

While ASDI and CM methodologies resolve problems of modularity and formalization, they disregard the distribution aspect. The MAMA-S methodology gives a framework for building a consistent simulation platform

from many independent simulation models, but does not represent distributed control within the model. At last, the HLA methodology can solve the problem of distribution at the implementation level.

Common features of these approaches are that they don't really match with modularity and reusability according to the requirements stated in sections 2.1 and 2.2. Indeed, the proposed approach ASCI-mi for modelling DSS through Multi-agents modelling concepts is mainly used for large systems composed of interacting entities, but it focuses only on structuring the decisional sub-system without taking into account the modelling of physical sub-system, and furthermore of autonomous processes merging physical and decisional capabilities.

Faced to this limit, our approach aims to propose a simulation model structure well adapted for distributed control by extending ASDI concepts to define an appropriate methodology called ASDI-dc (ASDI-distributed control). Especially, concerning the modularity, we propose to use two criteria : not only the functional but also the structural criterion to differentiate several types of processes and concerning the reusability we implement special constructs allowing to build specific distributed decision models.

3. Proposed methodology

3.1. Simulation model structure

From the above research, we distil two decomposition principles for structuring simulation models:

- The separation of physical, information and control elements.
- The distinction between processes that are purely decisional or physical and the mixed processes that belong to physical and decisional systems. We named *autonomous process* the module which represents the association of a physical unit and a decision-making centre.

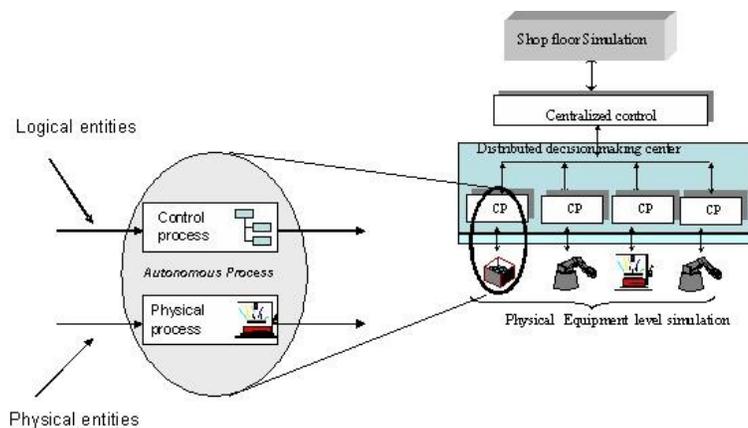


Fig. 4: A autonomous process

In the shop floor simulation, different flows of discrete entities circulate in a model made up of various processes, which will have an influence on the speed of flow, its nature or its advance within the model. These flows can represent logical or physical entities. A physical entity is a physical object, for example a product, which can be modified by physical processes such as a machine or workstation. In this case physical process is characterized by operating time, availability of resources, etc. while the product entity is characterised by arrival and departure dates, quality status, etc. (see figure 4). A logical entity is modified by control processes to make decision for creating and routing physical entities and/or launching, sizing, booking shop orders on physical processes. In discrete event simulation such a control process can be implemented by branch constructs. At last, we call “autonomous process” a process that combines physical and control processes and consequently are able to modify the physical flows and to make local control decision (see figure 4).

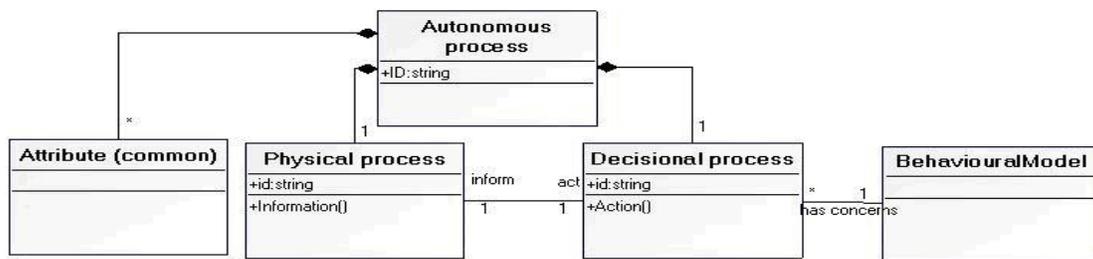


Fig. 5: The autonomous process meta-model

In figure.5, we propose a meta-model of those entities and processes. An autonomous process is composed by a physical part (like resources or stations) and a decisional process; the decisional part controls the physical one according to the behaviour model. It has two different types of attributes: external and internal. The functions and the behaviour model of the system processes needs to be described.

The advantages adherence to these principles has been detailed in section 2.1. The separation of physical, informational and control elements is assumed to facilitate a higher degree of model reusability and modularity. For example, separation of physical and control processes enables to independently define the manufacturing processes and their control strategy. As a result, simulation models allow for flexible response to alternative control structures and rules without requiring a modification on the physical system models. Moreover, *autonomous process* concept enables a local modification of the control without modifying the global control strategy. For example, considering human operator as an *autonomous process* aims to simulate different task allocations taking into account operator’s

flexibility.

3.2. ASDI-dc methodology

The proposed methodology is an extension of ASDI, called ASDI-dc (ASDI-distributed control), to support the simulation models structure as defined below. Moreover, we add before a third principle: “the simulation framework will be used by manager who is not necessarily expert on simulation tools” that’s required to use automation functions to instantiate the simulation model.

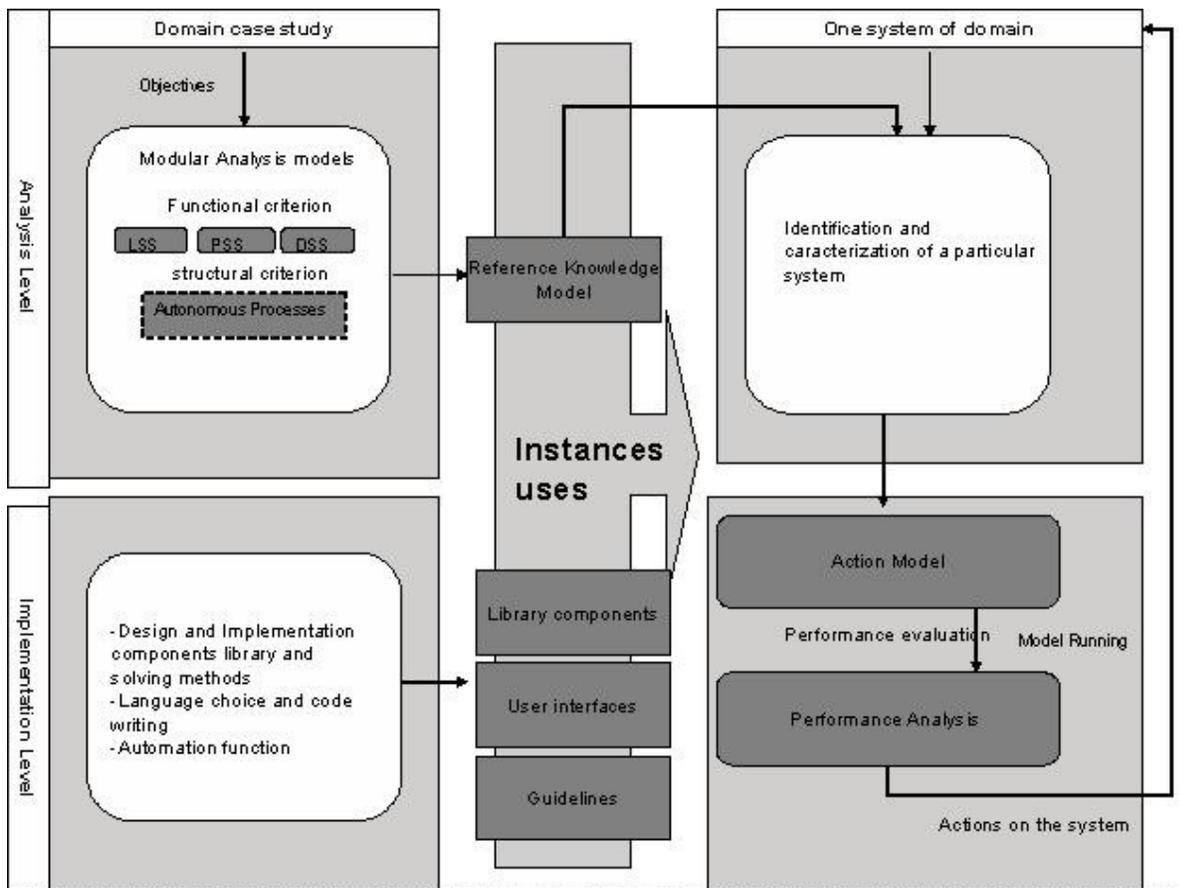


Fig. 6: ASDI-dc modelling process

In the ASDI-dc modelling process, we propose to distinguish two abstraction levels: the analysis and implementation levels (figure 6). Each of those levels is composed by two main parts: the first concerns the systems domain, the second is focused on one instance of this domain. Following we describe the main parts of the figure 6.

3.2.1 Domain study

The aim of **Analysis step** is to analyse domain and to develop a reference (or generic) model of knowledge. This

model is built by identifying the common points of all the subsystems following functional and structural criteria for modularity.

- The logical subsystem is composed of the whole information about transactions flows. As an example we find the information about the raw materials, components, bills of material, and the associated set of manufacturing routing.
- The physical subsystem (PSS) is composed by the means of production and handling, their geographical distribution and their logical and physical interconnections.
- The decisional subsystem (DSS) called also the management system specifying the management rules and structured in hierarchical decision centre (strategic, tactical, operational or reactive).It acts at the same time on the physical subsystem (rules of breakdowns management, rules of resources allocation,...) and on the logical subsystem (rules of passage to an alternate routing,...).
- Communication model between the three subsystems is necessary. It permits to describe interactions between objects of subsystems.
- To ensure structural criterion of modularity, autonomous processes must be defined through its behavioural model.

In **Implementation level**, in first way, we define rules to build action models or basic components; it is a very important step for reusability. We will explain this fact in the case study. In the second way; we build the software components library that will be used automatically to generate models for one system of the studied class.

Concerning the autonomous processes, we propose to associate the physical part and the decisional one. Different problems, like modification of the control strategy, will be simplified.

3.2.2 Instances of domain model: System study

In **analysis level**, the experts use the reference knowledge model to analyze 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.

The last phase, **Implementation level**, is about action model implementation using the software library components and automation function. This action model will be used to evaluate system performances.

4. Case study

4.1. Presentation

Trane Company provides indoor comfort systems and comprehensive facility solutions for residential, commercial and industrial building needs. As we present in the introduction of this paper, the 29 production sites are designed according to DFT concepts. The objective is to optimize production and also standardize processes. The production context is as follows: the shop floor is organized in pull production. To ensure a better balancing of an assembly line every worker must be able to work at three workstations: his workstation (i.e. initial affectation), the upstream workstation, and the downstream workstation. The objective is to model an assembly line and adapt it at all company assembly lines. The control system is globally centralized; i.e. every week the company compiles the Master Production Schedule which proposes finished products manufacturing orders; but some decisions are made locally by workers that represent the distributed aspect of decision. In order to resolve 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 work station. This approach corresponds to the centralized one. The second way is to consider human operator as an “autonomous process”, therefore, we take into account distributed aspect. That corresponds to the distributed control. In both approaches (centralized and distributed control), all necessary data for simulation (manufacturing time, products information, initial affectation of workers etc) are saved in a database and transferred to the simulation model. We have modelled the system using the two approaches. This enables us to compare distributed and centralized control results. To build simulation models for DFT manufacturing system we use our proposed methodology ASDI-dc. The following section describes the major steps of ASDI-dc methodology.

4.2. Domain Analysis Phase

To structure our analysis we describe a domain studied in natural language, and then we formalize it in UML in order to get a reference model of the domain. (See figure 7, 8 and 9).

In the decisional subsystem we represent a relational structure between organizational decision-making centres. We distinguish two types of decision-making centres: centralized system and distributed decision centres. The centralized system control centres 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 (figure 7). The distributed

decision centres can represent the decisional processes of “*autonomous processes*”.

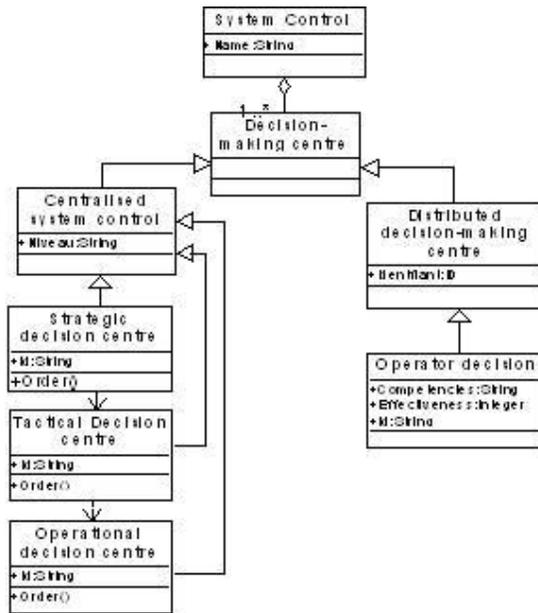


Fig. 7: A decisional subsystem of an assembly line

The physical subsystem concerns an assembly line. As we can see in figure 8 each assembly line is composed by a set of work centres 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 communication between the three systems.

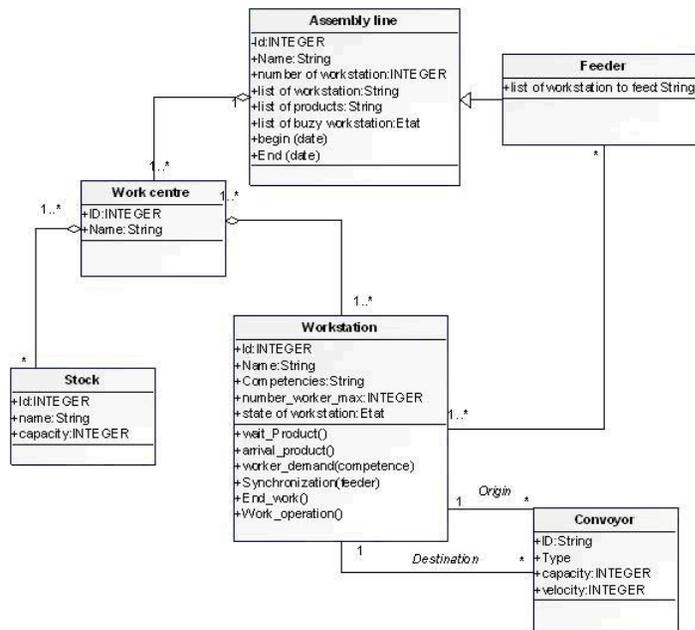


Fig. 8: A physical subsystem of an assembly line

The informational subsystem is based on the product. It describes its characteristics, its range (bill of materials) and in our case we find also the MPS which contains information on the number of products to be manufactured for one period given (in major case it will be one week), the order of product manufacturing, and the release date of each product (figure 9).

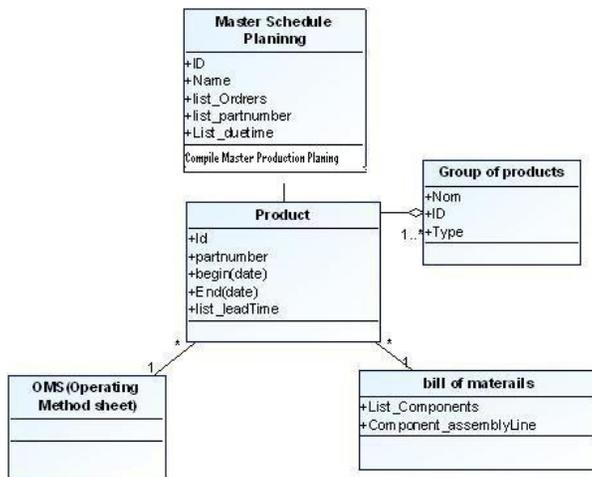


Fig. 9: A logical subsystem of an assembly line

4.3. Specification

In the first step, filtering operations of 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 behaviour models of autonomous processes. The algorithm describing operator decision process is described after:

Operator j takes decisions according to variables of his environment that are: (P_{wi}) and (S_{wi}) with w_i indicating 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 . $D_{wije} \{-1, 0, 1\}$ 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

$C_j \in \{C1, C2 \dots Cn\}$ and effectiveness $E_j \in \{0 \dots 1\}$.

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 be competent for work at workstation i).

The decision matrix for an operator i in workstation w_i is the following (see table 1):

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:

$Dw(i,j): Pwi, Swi \in \{0,1\} \quad Dwij \in \{-1, 0,1\}$

$Dwij = Swij \times (Pwij - 2) + 1$

Decision algorithm:

(1)Initialization

Affect (O_j, w_i)

$C_j =$ Competence $\in \{C1, C2 \dots Cn\}$

$E_j = X$ // (X will be in $\{0 \dots 1\}$)

(1)Waiting Event

If Event then $D = Dw(i, j)$

“Execute decision matrix

End If

(1)Research

While (Stop = False)

“While Stop condition is false

If Apt ($O_j, w_i + D$) = 0

“We test the aptitude of operator j to work in station $w_i + D$

Then $w_i = w_i + D$

“ $w_i = w_i + 1$ or $w_i = w_i - 1$ according to D

Else Stop=True

End If

```

End While
While Apt ( $O_j, w_i + D$ ) = 1 then
    Stop = True
    Affect ( $O_j, w_i$ )
    "we affect the operator  $j$  at work station  $w_i$ "
End While

```

4.4. Domain Design

In the domain design phase, we define rules for building the "action model" basic components. The table 2 shows the more important objects.

Other rules concern the name of Arena objects, for example: a work station resource is named

M_{ij} when i is a work centre and j is a work station in work centre i .

T_{ij} : theoretical operational time in work station j work centre i

Q_{ij} : queue in work station i and work centre j

In this step we must also choose the tools used for the action model implementation. The choice is dependant of subsystem nature.

- To model the physical subsystem, we were interested in the simulation software using an oriented-object approach for several reasons. First this approach leads to keep coherence between all the ASDI-dc cycle steps: for example the knowledge model in UML resulting of analysis phase will be easily exploited to build the action model. In addition, this approach makes it possible to identify each object of the actual system with an object of model (Anglania et al., 2002). Several software exist (Arena, AutoMod, Modelica, and Witness). The choice is related to Arena, very used in the industrial and academic fields A.M.A Al-Ahmari and K. Ridgway. (1999), H. De Swaan Arons and Csaba Attila Boer. (2000) (Perera and al. 2000) (Kovács et al., 1999) (Guilherme, 2004) as well as in our company. The principle of this software is to generate one or more flows of discrete entities, which can represent flows of information or physical flows. Arena has two interesting properties: a VBA interface to ensure communication with the other software, a "professional Arena" support which makes it possible to create its own library components and thus makes it possible to encapsulate the data of certain parts of the Arena model.

- For informational subsystem, we used Excel to store necessary information of Arena model (Nb of operator, competence on the stations, products to be manufactured.).
- Decisional subsystem has as rule the control of production and, in our case, the control of the Arena model, is developed in VBA.

4.5. Domain Implementation:

We design a basic components library: Object “work centre” that is materialized 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.

For the distributed control, the autonomous process is the worker; the decisional process is represented mainly by a VBA block. When a product arrives at workstation w_i a signal is sent to show availability of product, then a logical entity made up to the corresponding VBA block, the control of this entity is passed to the VBA Sub procedure created in the Visual Basic Editor. This procedure describes the autonomous process behaviour. The physical process is materialized by a resource in Arena (figure 10). The operator interface leads to enter the operator attributes and VBA Code if operator behaviour has been modified (figure 11).

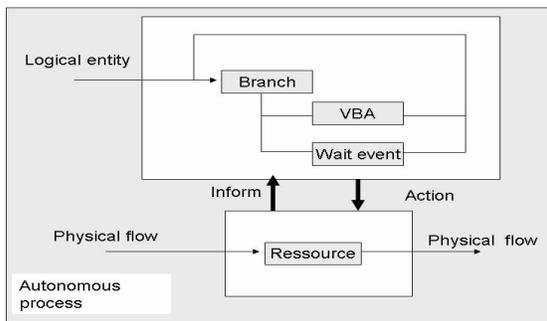


Fig. 10: a autonomous process in Arena environment



Fig 11: The operator interface

4.6. Implementation: Action Model

In the following step, we use the library components to create the action model through Arena software facilities.

5. Discussion

5.1. Credibility of reusability

The reusability is ensured in several ASDI-dc process steps: in domain implementation level, by giving a library of constructs called “Template” in Arena environment, and in system level, the reusability is based on the automation

function to build simulation models thanks to VBA facilities.

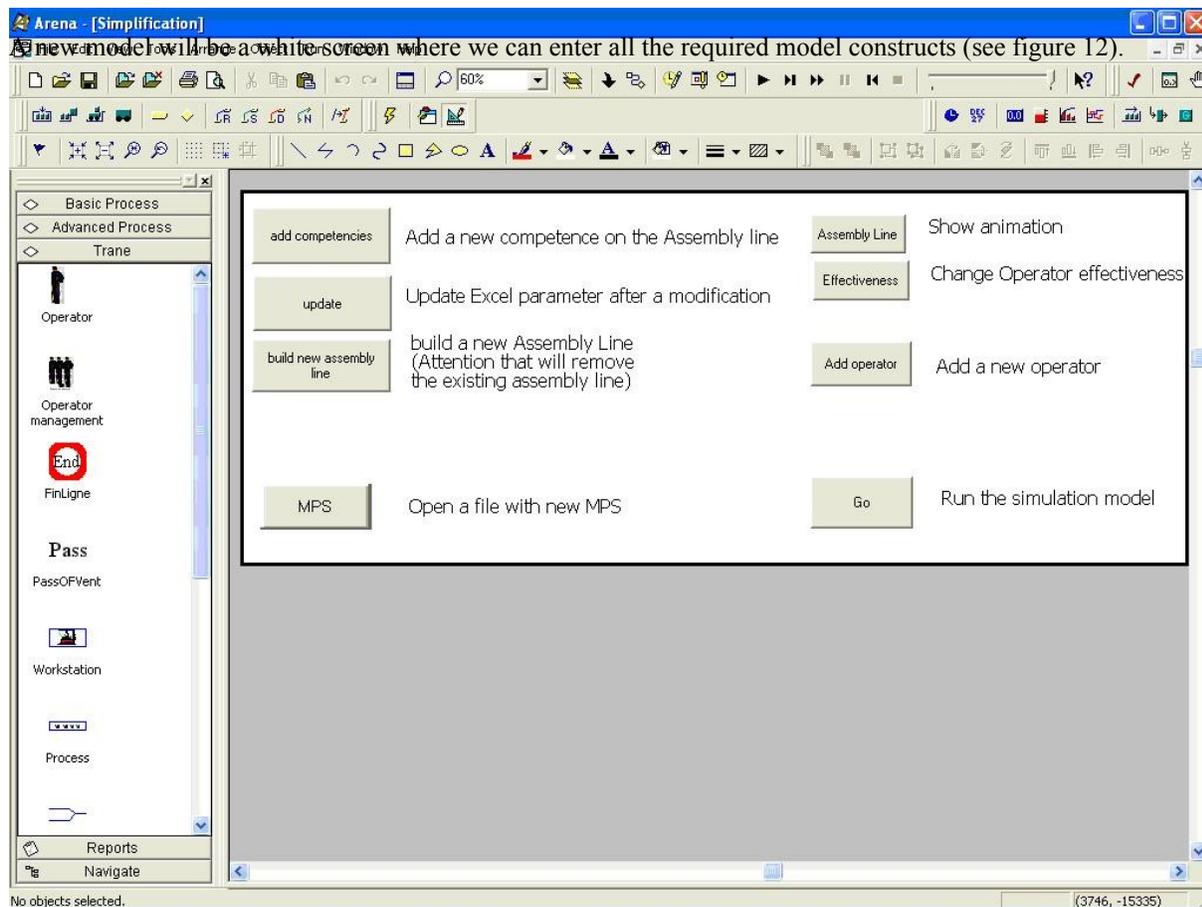


Fig 12: The Arena environment

At the left side we have the main model constructs called “Template” we can use different template to build a model. In fact,

Arena provides several templates. The widely used are the "Basic Process" template, the "Advanced Transfer" template and the "Advanced Process" template. In this way, we are developing our template “Trane” which contain our own custom modules for specific highly repetitive issues as workstation, human operator, orders management etc. The process of developing a simulation model is by instantiating a module out of the template into the modelling area, via drag and drop or via VB interfaces thanks to automation function (for facilitate the use) on those interfaces we can set the parameters concerning to this module (for example, in module human operator: the name of operator, his competences, his first workstation allocation).

5.2. Credibility of modularity

As an example of modularity benefit: on the same physical subsystem (Arena Model) we can simulate several strategies for example testing different rules from sequencing changing job sequences without changing the routing physical processes. The modification will be acting only on Excel product “spreadsheet” or /and VBA procedure and not on Arena model which present physical sub-system.

On the other hand, the concept of autonomous process leads to change behaviour model without changing a centralised control model. For example we can change mobility rules for operators without changing the numbers of operators and their attributes.

5.3. Tests and Validation

In any process modelling, the tests and validation phases are very important; they enable to measure the confidence interval between the simulated results and the actual data and evaluate the credibility of models. The figure 13 and 14 present lead time results relative to two tests done on sample of 15 products. The first one is interesting to centralized approach and the second to distributed approach. The two picks on the figure 13 (A, B) show that if one problem occurs at a work station w_i , it has an influence on the lead time at stations w_{i-1} and w_{i+1} .

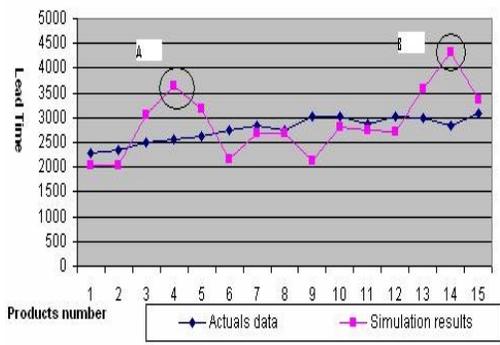


Fig. 13: The centralized case

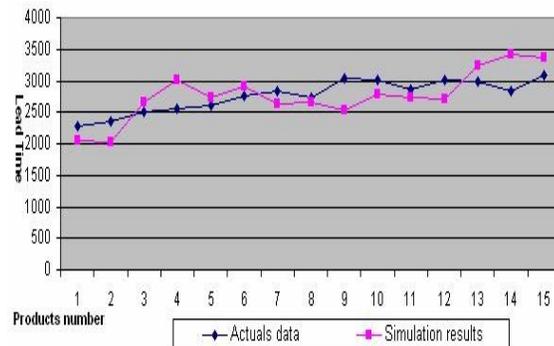


Fig. 14: The distributed case

This problem is due to workers mobility. As we said in preceding section, in centralized approach we model the workers mobility as the change of theoretical capacity on the work station. Indeed, in a stable environment this approximation gives relatively best result but if a problem arrives “on line” we are not able to change a work station capacity dynamically. Consequently, we use the distributed approach to refine the simulation model. We obtained with the same sample the results on figure 14. The distributed case results show that it will be possible to improve the shop floor simulation model and decrease the variability caused by local decision.

6. Conclusion and Perspectives

The study of different simulation methodologies enabled us to answer limitations that must be addressed to improve their effectiveness in a distributed control context. The ASDI methodology enables to answer formalization, 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 *autonomous processes* and their conceptual model. In design and implementation, we used Arena and Visual Basic software for application to develop a generic component for control. The question of model validity can not be ignored. It seems widely accepted in the simulation community that models or modelling approaches can not be fully validated. It makes sense to have some form of model quality insurance to ensure that the model is fit for its intended purpose. In the future research we will interest to the following two objectives. The first one is to give a more formal setting for this approach and propose a simulation platform to build simulation model in DFT context. The second one deals within the scope of our laboratory research team, this scope concerns product driven manufacturing systems. In this project we consider that the product can storage its own information and can be an actor of its own transformations. This fact deals with intelligent products resulting from HMS architecture. In the DFT context, the concept of Product-Driven control is very developed. For example, the signal of work is given by the product state. However, physical flow and information flow synchronisation are not assured. To resolve this problem, new technologies of identification as RFID have appeared. The goal of our next work will be to use simulation as a tool to integrate this technology in Trane Company legacy systems. In our current case, autonomous process concerned labour on the shop floor. Nevertheless, it would be interesting to analyse how to adapt the autonomous process concept to intelligent product concept.

7. References

- Koestler A., 1967. “The Ghost in the Machine”. Arkana, London.
- Law A. M., and Kelton W. D., 1991. “Simulation Modeling & Analysis”, McGraw-Hill, 2e edition,.
- Douglas A., Bodner and Leon F. McGinnis Keck 2002. “A Structured Approach to Simulation Modeling of Manufacturing Systems. Engineering Proceedings of the Industrial Engineering Research Conference.
- Costanza J., Just-In-Time manufacturing excellence John Costanza Institute of Technology Inc.; 3rd edition September, 1996.
- Kellert P., Force C., 1998a. “Méthodologie de Modélisation Orientée Objets de Systèmes de Production -

Application à une Chaîne d'Etuvage de Bobines d'Allumage "Journal Européen des Systèmes Automatisés, 32(1), pp. 107-131.

Kellert P., Ruch S., 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.

Le Moigne J.-L. , 1992. " La modélisation des systèmes complexes ". Editions Dunod. A. Anglania, A. Grieco, M.

Lee Y.M., Cheng F., Leung Y.T., 2004. "Exploring the impact of RFID on Supply Chain dynamics", Proceedings of the 2004 Winter Simulation Conference R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters.

Sargent R.G., 1986. "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, pp.512-516.

Reese R., Wyatt D.L., 1987. "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, pp.185-192.

Wyatt D.L., 1990. "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, pp. 472-476

Robinson S., Richard E Nance R.E., Paul R.J., Pidd M., Simon J.E Taylor, 2004. "Simulation model reuse: definitions, benefits and obstacles": in Simulation Modelling Practice and Theory 12 pps 479-494.

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.

Doumeingts G., 1984. " Méthode GRAI : méthode de conception des systèmes productiques". Thèse de doctorat, Laboratoire d'Automatique et de Productique, Université Bordeaux I.

Galland S., and Grimaud F., 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.

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)

Gourgand M, 1984. “ Outils Logiciels Pour L'évaluation Des Performances Des Systèmes Informatiques”. Thèse de doctorat, Université Blaise Pascal, Clermont-Ferrand, France.

Nance R., 1994. “The Conical Methodology and the Evolution of Simulation Model Development”. *Annals of Operations Research*, 53, pp. 1-45.

Sarramia D., 2002. ASCImi : “Une méthodologie de modélisation et de simulation multiple et incrémentielle”. Application aux Systèmes de Trafic Urbain. Doctorat en informatique, Université de Clermont-Ferrand II

Chabrol M, Sarramia D, Tchernev N. 2006. “Urbain traffic systems modelling methodology. *International journal of Production economics*”, 99, pp. 156-176,

Anglania A., Grieco A, Pacella M., Toliob .T, 2002. “Object-oriented modeling and simulation of flexible manufacturing systems: a rule-based procedure”. *Simulation Modelling Practice and Theory* 10 209–234

Al-Ahmari A.M.A., Ridgway K. , 1999. “An integrated modeling method to support manufacturing system analysis and design”. *Computers in Industry* 38, pp. 225-238.

De Swaan Arons H., Csaba Attila Boer, 2001. “Storage and retrieval of discrete-event simulation model. *Simulation Practice and Theory*”, 8(8), pp. 555-576.

Perera T., Liyanage K., 2000. “Methodology for rapid identification and collection of input data in the simulation manufacturing systems”. *Simulation Practice and Theory* 7, pp. 645-656.

Kovács G.L., Kopácsi S., Nacsa J., Haidegger G., Groumpos P, 1999. “Application of software reuse and object-oriented methodologies for modelling and control of manufacturing systems”. *Computers in Industry* 39

Ernani Vieira G., 2004. “Ideas for modelling and simulation of supply chains with Arena”. *Proceedings of the 2004 Winter Simulation Conference* R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

Figure Captions

Fig. 1a: An example of functional criterion

Fig. 1b: An example of structural criterion

Fig. 2: A spectrum of reuse

Fig. 3: An example of reusability

Fig. 4: A autonomous process

Fig. 5: The autonomous process meta-model

Fig. 6: ASCI-dc modelling process

Fig. 7: A decisional subsystem of an assembly line

Fig. 8: A physical subsystem of an assembly line

Fig. 9: A logical subsystem of an assembly line

Fig. 10: a autonomous process in Arena environment

Fig 11: The operator interface

Fig 12: The Arena environment

Fig. 13: The centralized case

Fig. 14: The distributed case

Dw(i,j)	C1	C2	C3	C4
Swi	0	0	1	1
Pwi	0	1	0	1
Dwij	1	1	-1	0

Table 1: The decision matrix for an operator i in workstation w_i

Domain Objects	Arena Objects
Work Centre	Process
WorkStation	Resource
Stock	Queue
Path	Route
Autonomous process	VBA /resource
Convoyer	Convoyer
Assembly line	Arena Model

Table 2: the most important domain objects