One approach of data-mining for Product Driven Systems

Philippe Thomas, André Thomas

One approach of data-mining for Product Driven Systems

Philippe Thomas. André Thomas.

Centre de Recherche en Automatique de Nancy (CRAN-UMR 7039),
Université de Lorraine, CNRS, Campus Sciences,
B.P. 70239, 54506 Vandœuvre lès Nancy cedex France
(e-mail: philippe.thomas@univ-lorraine.fr)

Abstract: The first objective of this paper is to highlight some new Product Driven Systems (PDS) issues. Effectively, several possibilities have been proposed to give to products or objects capacities to react to environment modifications (especially in manufacturing and logistics context here). In particular, bio-inspired approaches are now promising. All these new perspectives lead putting products in action according to collected information. That's why all technics leading to exploit and organize data are necessary. The main objective of the paper is addressed in a second part, where we highlight why learning machines could be seen as a new way to do that.

Keywords: Product Driven Systems, Viable systems, Learning machines, Neural network, Data mining.

1. INTRODUCTION

After the second industrial revolution the main companies' objective has been productivity. Ford Motor Company has introduced the concept of mass production. Since then, many techniques have been introduced leading to process automation and optimization of planning and production control activities. Among Manufacturing Planning and Control Systems, MRP Systems emerged during the seventies in order to solve problems such as those related to delays of orders, to intermittent stock consumption or to forecasting of raw materials consumption. However, inertia facing unexpected events occurring on the shop-floor has been seen as a residual issue. In order to compensate for this drawback, a new MRP² generation has been proposed with a closed loop approach (Vollman *et al.*, 1997). The main characteristic of these systems is the multi decision level horizons structuration. These horizons may be classified into long, medium, short and very short term. Considering them, four decision levels have been identified: strategic, tactical, operational and execution one. At the beginning of eighties, new management philosophies appeared which implied drastic changes in production management area. The main goals of these changes have been to improve the system reactivity and flexibility, on the one hand, and the service quality, on the other. These challenges are still valid today and have been mainly implemented by Just in Time (JiT) and Theory of Constraints (ToC) philosophies.

The main idea of JiT philosophy is based on the efficient use of productive resources. Various approaches have been proposed such as Lean manufacturing, Demand Flow Technology or Six Sigma. On production workshops, the main used tool to implement this philosophy to control physical/material flows is the kanban system. Lean manufacturing and, in particular, kanban system have implied a great revolution in production system management. The management functions, which is in MRP2 systems centralized and hierarchical, become, in lean philosophy, completely or partialy distributed. In spite of the results obtained with these decentralized approaches, Theory of Constraints (ToC) introduces a new point of view, different and complementary, based on a global optimum attainment which brings back to a centralized approach (Goldratt et Cox, 1992). ToC induces that organization has to be evaluated and controlled by using three indicators: profits generated by sales, operating costs and inventories. In ToC, a good bottleneck management is the key of the success. Finally, the existence of these three concurrent philosophies of production management have led to the proposition of many hybrid systems using techniques coming from MRP², JiT or ToC which were implemented in software such as ERP (Enterprise Resources Planning), APS (Advanced Planning Systems), SCM (Supply Chain Management Systems)...

In the next section, a brief overview of intelligent manufacturing systems is recalled with their advantages and weaknesses. Section 3 presents a viable system model for product driven system. Section 4 focuses on the need of knowledge of these systems and the learning approaches used to design knowledge. An illustration of this approach is presented in section 5 before to conclude.

2. INTELLIGENT MANUFACTURING SYSTEMS

The development of production management systems has led to the "Computer Integrated Manufacturing" concept. The main goal of these systems is to interconnect all the information systems included in the production system. CIM systems have to supervise and control all company operations. At that time, the paradigm generally accepted was that the CIM system would be able to have a great flexibility when changes occur and would give the best solution to the problems encountered in production system. Nevertheless, implementations led to centralized and rigid structures unable to adapt quickly to changes. However, some flexible manufacturing systems were very productive. So, at the beginning of nineties, CIM systems were no longer considered as the solution of all problems of production companies (Babiceanu and Chen, 2006).

Considering the bad results of integrated systems in terms of flexibility and reactivity, collaborations between research centers, universities and companies have been initiated in order to design and develop the production systems for the future. The most important of them was the “Intelligent Manufacturing Systems” Project (IMS) (Yoshikawa, 1995). The basic idea of IMS was the design and implementation of decentralized systems. Its main goal was system flexibility in order to deal quickly with disturbances inherent to the production processes. Before this function was allocated to men who monitored and changed the shop orders. The idea was to automate all or a part of this function by using new communications technologies (Auto-ID, Multi-Agent systems…). These new systems have to be robust, reconfigurable and reusable (Leitao, 2009).

Considering the centralization criterion, the production systems have been divided into four types (Babiceanu and Chen, 2006): centralized systems, hierarchical ones, modified hierarchical ones and heterarchical systems. Around the design of decentralized systems appeared different types of systems and concepts. The main decentralized production systems are bionic, fractal and holonic ones (HMS). This paper focuses on the last one.

HMS consortium has proposed holonic production systems based on the holon concept. A holon is an entity which may be included in other holons (Van Brussel *et al.*, 1998) and are organized in holarchies. Holons have the abilities of autonomy and cooperation. Nevertheless, concepts of agent and holon are often confused. Although the holon may be viewed as an agent, the main difference is that the control part is associated to the physical part in a holon. In an agent (which is an abstract entity), a physical entity may be merged or modeled by an abstract entity. Product Driven Systems is an evolution of holonic system where interoperability and intelligence are improved. In PDS, products become the company resources controllers (Morel, *et al*. 2003). This leads to the intelligent product concept. It has been defined as an entity equipped with physical and informational representations, able to affect decisions which may affect the intelligent product itself (McFarlane *et al.*, 2003). In practice, the Radio Frequency identification (RFID) is a technology able to link information and physical environment. The central idea is to move from a classical hierarchical and aggregated control to a distributed decision making where a part of the decision is made locally, all along the products life cycle. So, the needed information is reduced and locally processed. The PDS have been generally designed as a particular class of holonic systems. The main advantages of IMS approaches are feasibility, robustness, flexibility, reconfigurability and reusability.

Up to now, many methodologies have been proposed in order to model distributed approaches (PROSA, ADACOR, METAMORPH…). Despite this, no standardized criterion exists allowing model design. The modeling step may be performed by focusing on functional, physical or abstract aspects (Creput 2008). And yet, the tools choice, the criterions choices and the models choices remain linked to the abilities and preferences of designer. This lack of uniformity makes the evaluation and the comparison of different applications in the literature, difficult.

One of the most critical points for the heterarchical approaches (decentralized) comparatively to traditional ones (centralized) is the decision optimization. Heterarchical systems are not able to formally guarantee their performances in terms of quantifiable variables, and more particularly, of costs. Heterarchical systems are interested in classical criterions of cost, time or efficiency, but also to goals relating to flexibility, reconfigurability, reactivity, interoperability… These goals are not easily quantifiable, and so, the comparison and evaluation of benefits of such systems are difficult.

In conclusion, it can be noticed that the two great approaches of production planning and control have strengths and weaknesses. Actually, the conventional approaches (centralized) insure the efficiency of the global system but at the expense of flexibility and reactivity. At the opposite, the IMS approaches (distributed) insure flexibility and reactivity, but are not able to insure performance and consistency between decisions taken to different levels. So the search of global consistency of a system working, through the comportment of sub systems and their own goal is essential (Thomas, 2004).

Verstraete *et al*. (2008) have proposed to associate a HMS to a hierarchical planning system. The function of HMS is to determine an alternative planning when disturbance occurs. Another way would be to design hybrid systems (centralized/distributed). Herrera (Herrera *et al.*, 2011) highlights that to allow acceptable efficiency and consistency between different decision levels and to improve the flexibility and reactivity capabilities of the systems; “Viable Model” could be a good way to structure their architecture. Moreover these systems have to include a data acquisition system in order to collect data from the physical system to be controlled. These data must be filtered, analyzed, possibly aggregated… in order to become exploitable. This paper focuses now on this point.

3. VIABLE SYSTEM MODEL FOR PDS

3.1. Viable System Model (VSM)

The origins of VSM arise from the works of Beer (Beer, 1984) applied to the steel industry in the fifties. This research can be placed in the line of works of Norbert Wiener, Warran McCulloch and Ross Ashby. The main objective of the model was to identify and to explain how systems are viable. Although, VSM is a general model for the study of any viable system, the most concerned application areas has been human activity organizations, i.e., corporations, firms or governments. In this domain, VSM changes the view of the traditional management model based on command and control, in which a control system is designed as a pyramid and such decisions are disaggregated in a top-down manner at different structural levels. The main difference, inspired by the biological organization, consists in mapping this hierarchy into a structural recursion (Herrera *et al.*, 2011). The premise of this change of perspective was inspired from the living beings composition (cells, organs, systems, etc.). Indeed, they have properties of autonomy, self-organization and self-regulation, allowing them to have an independent existence. The differentiation of their functions and the relationships between these elementary components produce more complex systems, without that subsystem essential properties would be lost. However, one of the most important properties of a viable system is their intrinsic recursion. In fact, any viable system contains and is contained by another viable system. Every subsystem maintains its autonomy towards its environment, but it also contributes to generate the viable system in which it is included. In that way, a viable system and its different subsystems have the same structural requirements. A viable system supports its objectives thanks to an overall cohesion and adapts itself by the autonomy of its subsystems. VSM was developed looking for invariances in organic systems. These invariances allow defining a homomorphism of their functions, organization and structure. Beer defines five elementary functions that any viable system must have: implementation, coordination, control, intelligence and policy.

3.2. VSM model of Manufacturing Planning and Control system

The model describes here (Fig. 1) has been proposed by Herrera *et al.* (2011). It is consistent with the five functions of the Manufacturing Planning and Control Systems which can be described as: Strategic Planning, Sales and Operations Planning (S&OP), Master Production Planning (MPS), scheduling and execution. Each of these functions corresponds to a level in the decision making process regarding to different horizons going from a longer to shorter one. In practice, these decisions are taken using a rolling horizon to take into account the frequent changes that occur in the data (demand, capacity, etc.). Thus, the strategic planning is revised once a year, the S&OP is computed monthly, the MPS is get per week, and the schedules are performed daily or more frequently depending on disturbances. Each function deals with a corresponding aggregation level of products respectively families, finished products and items (components). In this context, one of the major issues is to adapt decisions at each level when disturbances (internal or external) happen. The frequently resulting modifications in the decision making process lead to the so-called nervousness system which deteriorates the system performance (productivity and efficiency). One should notice that the shorter the horizon is, the more frequent are the changes. Thus, the performance is more deteriorated at the lower level (scheduling level). More precisely, this model is a generic model based on VSM dealing with production planning considering both MPS level and scheduling (lot-streaming).

In a PDS, the basic unit is the intelligent product, which is capable of i) acquiring and archiving data, ii) communicating with its environment and iii) interacting with and on it. So, intelligent products have autonomy, auto-organization and auto-regulation properties necessary to become the basic subsystem of a VSM model which is able to model all levels of a MRP2 system.

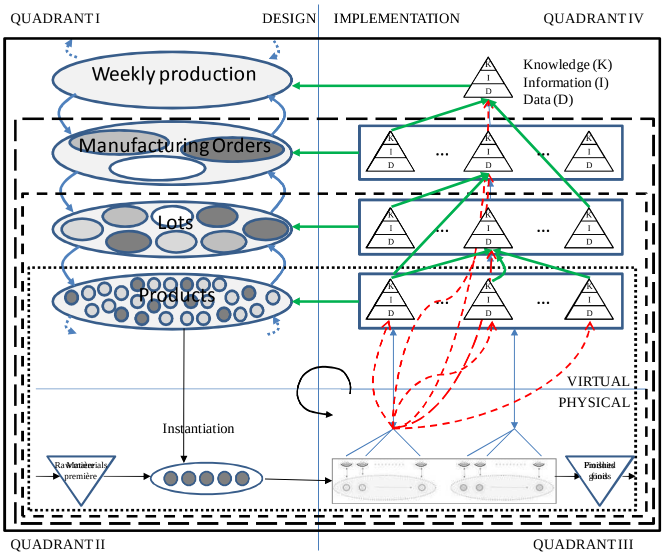

Fig. 1. VSM based product driven control system.

Concerning the figure 1 above, we have made the hypothesis that products are instrumented with RFID technology which allows acquiring, archiving data and communicating and interacting with their environment. The products holarchy is designed in order to represent the decision levels of a MRP2 system. This resulting metamodel uses the holon concept. This figure is subdivided into four quadrants (I, II, III, and IV) in order to simplify the explanations. The horizontal axis distinguishes the physical world to the virtual one. The vertical axis distinguishes the design which is a representation of the system to the implementation which performs the decision making and the knowledge management. Red arrows (dashed) represent the data flows from the shop floor to the quadrant IV (Data Management System – DMS), when the green (bold) arrows represent informational or knowledge flows into the DMS. To avoid overloading the figure, only some examples of these flows are shown.

The quadrant I shows the planning system which may be centralized or distributed. Its decomposition is based on four levels of aggregation of product entities (weekly production, manufacturing orders, lots and products). To each level, entities are modeled as agents. The product entities are agents with a specific control/autonomy level which allow to represent all the hybridization levels of the system from a pure centralized system (product agents transmit information to upper level where decision is made) to a pure heterachical system (agents communicate among themselves in order to make decision). The quadrant II is a conceptual representation of the instantiation phase. It shows the physical implementation and corresponds to the instancing of products in the form of holons, which load intelligence and the functions allowing them to interact with environment and to acquire the desired level of autonomy. The quadrant III shows the physical control in which product holons are able to make decisions according to events concerning their own evolution. The quadrant IV shows the virtual implementation and corresponds to the transformation process of data (coming from the shopfloor) to information allowing knowledge to emerge. It is this knowledge which must be loaded by agents in quadrant I in order to improve their adaptation abilities to events with the principle of experience feedback. The question that needs to be answered is “How perform this experience feedback”?

4. DATA MINING AND PDS

As previously said, in the concept of product driven system (PDS), product must take decisions and interact with its environment thanks to acquired knowledge and information. The synchronization of physical and informational flows inherent to PDS implies that many data may be exploited in order to create this knowledge and this information. These data may be related to product himself, or to the production process. So, the question becomes: how to exploit these data?

Considering figure 1, green arrows (thick) connect the elements of quadrant IV to elements of quadrant I. These connections represent the knowledge loaded in agents. So, the first task is to determine which knowledge is needed by different levels agents. Obviously, a product agent doesn’t need the same knowledge than a manufacturing order agent. So it is necessary to define precisely these needs in order to be able to answer them. When this is done, it remains to determine how to build this knowledge. For this, the recursion of the VSM model may be useful. A first level may be defined which includes the two physical quadrants (II and III) and the product layer of the two virtual quadrants (I and IV). This level is surrounded by dotted line. The second level (Lots) includes the first level and adds the lots layer of quadrants I and IV. It is surrounded by short dashed line. The third level (Manufacturing Order level) includes Lots level and adds the fabrication order layer of the quadrants I and IV. It is surrounded by long dashed line. At last, the fourth and last level (weekly production level) includes fabrication order level and adds the weekly production layer of the two virtual quadrants (I and IV). It is surrounded by solid line. This decomposition is comparable to the concepts of systems and sub-systems of the system engineering which may be used in order to define interfaces between levels. To each level, the knowledge manufacturing process is performed in several steps:

1. Determination of witch knowledge must be loaded in agent. As explained previously, this need is different according to agent level.
2. Determination and collect of information available in order to produce knowledge. This point will be detailed afterwards.
3. The structure of the model (here multilayer perceptron) has to be determined.
4. At last, learning and validation phase must be performed. The failure of the learning process implies a feedback on the second or third points.

Let us focus on the product level. In quadrant IV, the knowledge to load in the agent has to be built. The main particularity of this level is that the entries of knowledge manufacturing process are only data collected on the workshop. This process is then a classical process of knowledge extraction from data. The main difficulty is to know which data is necessary. It can be noticed that these data may be of continuous or discrete nature, determinist or stochastic one, and the knowledge design process must take into account the hybrid nature of this data.

When the next level is considered, the entries of knowledge manufacturing process may be data (aggregated or not) collected on the workshop, but also information and knowledge built at the product level. So the double challenge is:

- To determine which data, but also which information and which knowledge at the product level are necessary in order to built the knowledge to be loaded in agent at the lot level,
- To define a tool allowing to aggregate entities of different nature (data, information, knowledge) in order to built the desired knowledge.

The desired knowledge to be loaded in agents of upper levels will be built with a recursive approach by using the preceding procedure. This paper highlights knowledge manufacturing process at the product level in order to point out the main difficulties encountered and to propose solutions.

5. ILLUSTRATION

Let us consider a simple production process constituted of sequential work centers presented by figure 2. One of these work centers is a bottleneck. The only knowledge which must be loaded in product agents is the lead time between the release of manufacturing order and the products arrival into input queue of bottleneck.

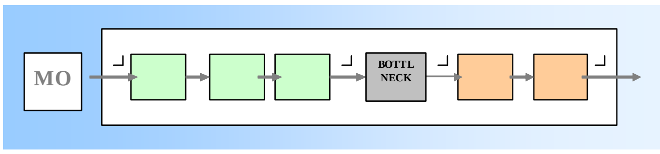

Fig. 2. Considered production system.

The knowledge manufacturing process must be highly automated. So, the data loaded by holons (quadrants II and III) are collected and exploited by using multilayer perceptron which uses supervised learning. Its structure is given by:

$$z = \sum_{i=1}^{n_1} w_i^2 . g\left(\sum_{h=1}^{n_0} w_{ih}^1 . x_h^0 + b_i^1 \right) + b \qquad (1)$$

where x_h^0 are the n_0 inputs of the network, w_{ih}^1 are the weights connecting the input layer to the hidden layer, b_i^1 are the biases of the hidden neurons, g(.) is the activation function of the hidden neurons (here, the hyperbolic tangent),

w_i^2 are the weights connecting the hidden neurons to the output one, b is the bias of the output neuron and z is the network output.

The weights and biases are determined by using a supervised learning which can be performed in two steps (Thomas *et al.* 2011): i) *initialization step*; this initialization may be randomly performed or by using more complex algorithms. This step is crucial in order to avoid local optimum trapping. ii) *learning step*; many learning algorithms exist. One of them is the Levenberg-Marquardt algorithm. It works as a hessian algorithm when solution is distant and as a gradient algorithm when solution is near.

This neural network must model the lead time between the release of manufacturing order and the products arrival into input queue of bottleneck. This lead time is a continuous notion. A first step is to fetch the lead time and all the explanatory variables collected by each product holons. These explanatory variables will become the neural network inputs. They may be continuous, as utilization rates, queues size… or discrete, as routing choice, machine choice… The question that needs to be answered is "How to take into account this discrete data"? Previous works have shown that some discrete variables may be used without particular precautions but other variables may not (Thomas *et al.*, 2011). In order to solve this problem, two approaches may be used.

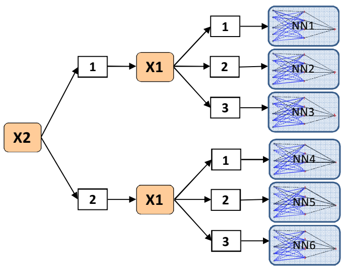

Fig. 3. Taken into account discrete variables – multi-model approach.

The first approach is similar to the multi model philosophy. In fact, if learning cannot take into account some discrete variables, this is due to the system comportment changes when these variables change of state. So, these discrete variables define different operating areas of the system, and so, it is necessary to design one neural model for each operating area. As example, if two discrete variables X1 and X2 may take 3 and 2 states respectively, the considered system can be found into 2*3=6 operating areas and so, it needs 6 neural models to learn (figure 3). The advantages of this approach are that the neural networks have only continuous inputs and so, the learning is much simpler. Moreover, the structures of the networks include less input and hidden neurons and so the computational time decreases during the learning and exploitation steps. However, the main drawbacks are the number of neural networks to learn and the need to design a models selection system in function of states of discrete variables (Thomas *et al.*, 2011).

The second approach consists to transform the discrete variables into binary ones and to use these binary variables as inputs of the network. With the same example as above, five binary input variables must be created, each of them may take the states 0 and 1: "X2=1"; "X2=2"; "X1=1"; "X1=2"; "X1=3". The main advantage of this approach is that only one neural network models the entire system. However, this model includes more inputs and hidden neurons and so the computational times increase during the learning and exploitation steps (Thomas and Thomas, 2009). These two approaches are not paradoxical. An optimal solution may be to mix these two approaches in order to limit both, the growing number of neural models to learn and the size of each of these models.

Following the explicative variables are collected and defined. In a second step, the structure of the neural network must be designed. The works of Cybenko and Funahashi have proved that a multilayer perceptron with only one hidden layer (using a sigmoïdal activation function) and an output layer (using a linear activation function) can approximate all nonlinear function with any desired accuracy. However, nothing is said about the number of hidden neurons.

The simplest approach would be to choose a very great number of hidden neurons which permits to obtain the best accuracy. However, we are not in front of a regular approximation problem but in front of a function adjustment to a finite number of points (Dreyfus, *et al.* 2002). The risk is to learn the noise and not the function. This risk is called *overfitting*. In order to avoid it, different techniques have been proposed as regularization methods, early stopping or penalty methods. However, the determination of the optimal structure of the network allows to avoid the overfitting and to optimize the calculation times. For this, two approaches exist. The first one is a constructive one where the hidden layer is iteratively built. Another way is to start from a structure including too many hidden neurons and to remove the spurious neurons. The main advantage of this approach is to allow to some algorithms to determine simultaneously the hidden neurons number and the feature selection (Engelbrecht, 2001; Hassibi *et al.*, 1993; Setiono and Leow, 2000).

Three algorithms have been tested and compared on the lead time model in a sawmill (Thomas and Thomas, 2008). These three algorithms are OBS (Hassibi *et al.*, 1993), N2PFA (Setiono and Leow, 2000) and the one proposed by Engelbrecht (Engelbrecht, 2001). The obtained results have shown that the association of two algorithms (an extension of Engelbrecht algorithm and N2PFA) gives the optimal structure of the network quickly. In fact, a first step of pruning with the fastest algorithm (extension of Engelbrecht) allows finding the number of hidden neurons. In a second step, the N2PFA algorithm works with a smaller structure than the initial one (less hidden neurons) allows to design the network structure by pruning the spurious inputs.

5. CONCLUSIONS

In this paper, in a first step, we have summarized the new advances in product driven system approach and a VSM model has been presented. In a second step, we have investigated the knowledge manufacturing process. In summary, we can say that data are essential sources of knowledge but they are often unclaimed treasure because their exploitation may become time consuming. However, some tools exist, as learning overall and neural network in particular, allowing automating this exploitation. Nevertheless, we need to be careful and to make sure to respond positively to the questions: i) all the necessary information is included in the data? ii) all the collected data is well necessary? iii) how associate data of different natures (continuous or discrete) in the same model?

If the learning fails, this implies that the answer to the first question is negative. In this case, in order to supplement database, the product holons must collect other variables which may require improving the instrumentation. For the two other questions, some tools and methods have been presented succinctly here.

The implementation of this approach has two sides. The learning phase of a neural model is a complex task which must be designed off line. However, the resulting model is a simple equation which may be loaded in an agent.

However, one open question remains to be addressed: How to perform the knowledge manufacturing process for upper levels? This point represents a double challenge: i) how to determine which data, but also which information and which knowledge at the lower level are necessary in order to perform this knowledge manufacturing process? ii) which tool allows to aggregate entities of different natures (data, information, knowledge)? iii) These challenges will be addressed in our future works.

6. ACKNOWLEDGEMENTS

This paper is based on many works that have been achieved and supervised by authors. Many thanks to researchers that have worked with us on this subject. Particular thanks to Mr. C. Herrera for its contribution during its works of Ph.D. Moreover, the authors gratefully acknowledge the financial support of the CPER 2007-2013 Competitiveness Fibre Cluster (Structuration du Pôle Compétitivité Fibres Grand'Est), local (Conseil Général des Vosges), regional (Région Lorraine), national (DRRT and FNADT) and European (FEDER) funds.

REFERENCES

Babiceanu R. and Chen F. (2006). Developpement and applications of holonic manufacturing systems: a survey. *J. of Int. Manufacturing*, 111-131.

Beer S., (1984). The viable system model: Its provenance, development, methodology and pathology. *J. of the Op. Research Society*, 7-25.

Créput J.C. (2008). *Hybridation de métaheuristiques pour la résolution distribuée de problèmes d'optimisation spatialisés.* Habilitation à diriger les Recherches, Université de Bourgogne.

Dreyfus G., Martinez J.M., Samuelides M., Gordon M.B., Badran F., Thiria S., Hérault L. (2002). *Réseaux de neurones : Méthodologies et applications.*, Paris, France, Editions Eyrolles.

Engelbrecht A.P. (2001) A new pruning heurisitc based on variance analysis of sensitivity information. *IEEE trans. on Neural Networks*, 1386-1399.

Goldratt E., Cox J. (1992). *The goal: A process of ongoing improvement, 2nd revised edition*, Great Barrington, North River Press.

Hassibi B., Stork D.G., Wolff G.J. (1993). Optimal brain surgeon and general network pruning. *IEEE Int. Conf. on Neural Networks.* San Francisco, USA, 293-299.

Herrera C., Belmokhtar S., Thomas A. (2011). Viable system model approach for holonic product-driven manufacturing systems. *1st workshop on Service Orientation in Holonic and Multi Agent Manufacturing Control SOHOMA'11*. Paris, France.

Leitao P. (2009). Agent-based distributed manufacturing control: a state-of-the-art survey. *Eng. Application of Artificial Intelligence*, 979-991.

McFarlane D., Sarma S., Chirn J., Wong C., Ashton K. (2003). Auto ID systems and intelligent manufacturing control. *Eng. App. of Artificial Intelligence*, 365-376.

Morel G., Panetto H., Zaremba M., Mayer F. (2003). Manufacturing enterprise control and management system engineering: paradigms and open issues. *Annual Reviews in Control*, 199-209.

Setiono R., Leow W.K. (2000). Pruned neural networks for regression. *6th Pacific RIM Int. Conf. on Artificial Intelligence PRICAI'00,* Melbourne, Australie, 500-509.

Thomas A. (2004) *De la planification au pilotage pour les chaines logistiques*, Habilitation à diriger les recherches, Nancy Université, 2004.

Thomas P., Thomas A. (2008). Sélection de la structure d'un perceptron multicouches pour la réduction d'un modèle de simulation d'une scierie. *5ème Conf. Int. Francophone d'Automatique CIFA'08,* Bucarest, Roumanie.

Thomas P., Thomas A. (2009). How deals with discrete data for the reduction of simulation models using neural network. *13th IFAC Symp. on Information Control Problems in Manufacturing INCOM'09,* Moscou, Russie.

Thomas P., Thomas A., Suhner M.C. (2011). A neural network for the reduction of a product driven system emulation model. *Prod. Planning and Control*, 767-781.

Van Brussel H., Wyns J., Valckenaers P., Bongareerts L., Peeters P. (1998). Reference architecture for holonic manufacturing systems: PROSA. *Computers in Industry,* 255-274.

Verstraete P., Valckenaers P., Van Brussel H., Saint Germain B., Hadeli K., Van Belle J. (2008). Towards robust and effcient planning execution. *Engineering Applications of Artificial Intelligence*, 304-314.

Vollman T., Berry W., Whybark D. (1997). *Manufacturing planning and control systems.* New-York: McGraw-Hill.

Yoshikawa H. (1995). Manufacturing and the 21st century-intelligent manufacturing systems and the renaissance of the manufacturing industry. *Technological Forecasting and Social Change*, 195-213.