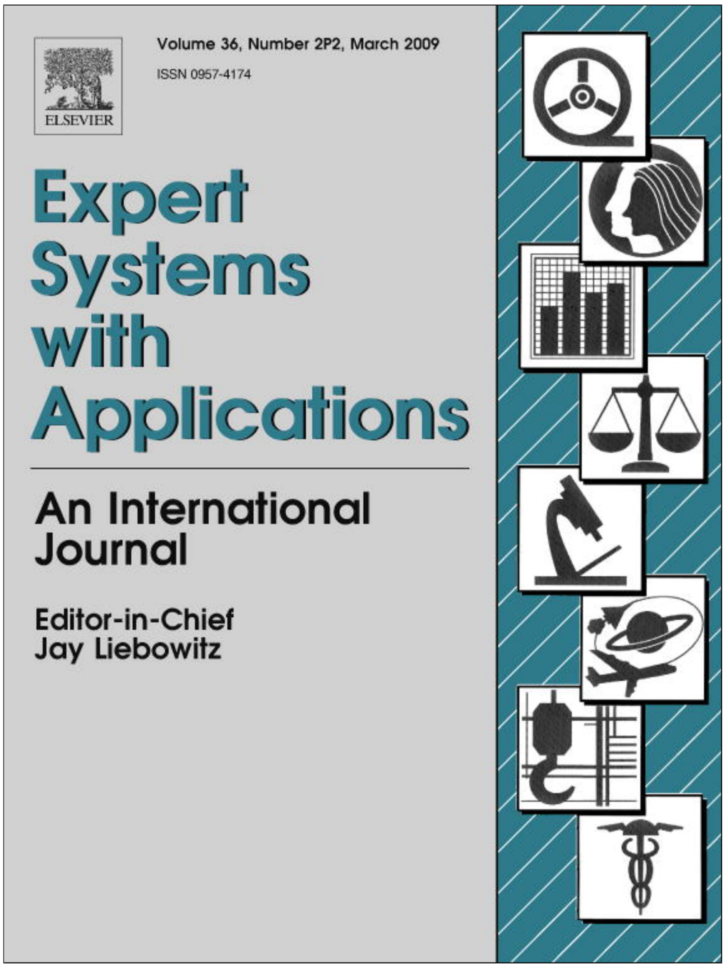


Volume 36, Number 2P2, March 2009
ISSN 0957-4174

ELSEVIER

Expert Systems with Applications

An International Journal

Editor-in-Chief
Jay Liebowitz

Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 36 (2009) 2937–2946
Expert Systems with Applications
www.elsevier.com/locate/eswa

A multi-agent system to construct production orders by employing an expert system and a neural network

Omar López-Ortega [*], Israel Villar-Medina

Universidad Autónoma del Estado de Hidalgo, Instituto de Ciencias Básicas e Ingeniería, Centro de Investigación en Tecnologías de Información y Sistemas, Carr. Pachuca-Tulancingo Km. 4.5 Pachuca, Hidalgo, C. P. 42083, México

Abstract

The authors describe the implementation of a multi-agent system, whose goal is to enhance production planning i.e. to improve the construction of production orders. This task has been carried out traditionally by the module known as production activity control (PAC). However, classic PAC systems lack adaptive techniques and intelligent behaviour. As a result they are mostly unfit to handle the NP Hard combinatorial problem underlying the construction of right production orders. To overcome this situation, we illustrate how an intelligent and collaborative multi-agent system (MAS) obtains a correct production order by coordinating two different techniques to emulate intelligence. One technique is performed by a feed-forward neural network (FANN), which is embedded in a *machine agent*, the objective being to determine the appropriate machine in order to fulfil clients' requirements. Also, an expert system is provided to a *tool agent*, which in turn is in charge of inferring the right tooling. The entire MAS consists of a *coordinator*, a *spy*, and a *scheduler*. The *coordinator agent* has the responsibility to control the flow of messages among the agents, whereas the *spy agent* is constantly reading the Enterprise Information System. The *scheduler agent* programs the production orders. We achieve a realistic MAS that fully automates the construction and dispatch of valid production orders in a factory dedicated to produce *labels*.

Keywords: Multi-agent systems; Production planning; Expert systems; Artificial neural networks

1. Introduction

Manufacturing companies, regardless their size, compete fiercely in the current globalised marketplace. Therefore, supporting software for manufacturing activities must help firms achieving flexibility and adaptation to changes. Production management, particularly, is supported by a system known as production activity control (PAC), which links planning and manufacturing together (Browne, Harhen, & Shivnan, 1992). This union is materialized by the generation of *production orders* addressing values for available machines and right tooling, among other variables. However, as input data is not necessarily homogeneous, classical PAC modules are mostly unfit to deal with the exponential number of combinations to generate the *correct* production order. Hence, flexible approaches open immense opportunities to improve this manufacturing activity, as reported in Wang, Yung, and Ip (2005), where a genetic algorithm has been employed to schedule orders in a dispersed manufacturing setting.

It has been claimed that multi-agent systems (MAS) have emerged as the major swift to enhance performance in manufacturing: Authors such as Mônch and Stehli (2006) report that key activities of industrial enterprises are being performed by multi-agent systems. Such is the case of process planning (López-Morales & López-Ortega, 2005; López-Ortega & López-Morales, 2006), holonic control, and production management (Mar̆ík & Lazanský, in press), which have already incorporated agent technology. The addition of capabilities to emulate intelligence in agents is no longer an aspiration but a reality, although some topics regarding cognitive abilities are left to be

[*] Corresponding author.
E-mail addresses: lopezo@uaeh.reduaeh.mx, olopez27@prodigy.net.mx (O. López-Ortega), villar_israel@yahoo.com (I. Villar-Medina).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.01.070

explored. Such is the case for algorithms that mirror the supervised learning process, which is achieved primarily through feed-forward artificial neural networks (FANN), with the backpropagation algorithm. Even though this technique provides flexible ways of making inferences, it has not been widely exploited in multi-agent systems to make production planning more flexible.

Modelling and implementing a MAS that employs co-ordinately learning and decision making capabilities, becomes a great challenge to provide flexible mechanisms to cope with changing market conditions. In this setting, our main contribution is the design and implementation of a multi-agent system that coordinates an expert system and a FANN, whose global is to construct production orders, hence improving the key activity of production planning.

The article is organized as it is explained next. A brief analysis of the related work is presented in Section 2. Section 3 contains the knowledge engineering process that was carried out. Next, we present the design and implementation of the intelligent and collaborative system. Finally, conclusions and future research perspectives are outlined.

2. Related work

We behold the continuous incorporation of agent technology into manufacturing systems. Marík (*op cit*) coined the revealing term *agentification of manufacturing systems* to summarize this issue. Specifically, agent technology has been implemented to improve communication of data among supporting applications. An example is given by Feng (2005), who created a MAS to improve collaboration between design and manufacturing departments. Planning activities are improved by using AI techniques and agent-based systems. Kornienko, Kornienko, and Priese (2004) presents a MAS aimed at optimizing resource assignments by analyzing process plans. Wang and Shen (2003) tackles the process planning problem, and extends the results to the scheduling problem (Wang, Shen, & Hao, 2006), in an effort to optimize decisions before manufacturing execution takes place. In (Feng, Stouffer, & Jurrens, 2005) a rule-based system is provided to agents to plan manufacturing processes. Frey, Nimis, Worn, and Lockemann (2003) implements a MAS to calculate the lot size, the operations to be performed, and the time-span between jobs. The information is fed to a simulator, which determines the best production plan according to the data provided by the MAS. Also, a multi-agent system including data mining techniques to set up the profile of resources in a supply chain is described thoroughly (Symeonidis, Kehagias, & Mitkas, 2003).

On the other hand, neural networks and agents have been recently applied in production planning. Paternina-Arboleda and Das (2005) reports learning algorithms to schedule multiple products on a single server. The solution is validated via simulation. Fichtner et al. (2006) describes an unsupervised learning technique to determine the appropriate NC machine, having as input unknown design features coming from a CAD system.

Thus, the *agentification* of production planning helps replacing centralized systems for distributed and more flexible architectures. Also, the incorporation of agent technology must be grounded on the ability to include intelligence. Although the related research found in literature is highly valuable, this present paper pioneers on how a MAS employs co-ordinately an Expert System and a FANN to better production planning. Our proposal is not only innovative, but it also shows practical benefits, because it has already been tested on a real-life setting.

3. Knowledge engineering

3.1. Part I. Analysis of the case study

The environment is a factory dedicated to manufacture *labels.* Labels are used almost everywhere: On bottled products such as wine or sodas, candies, jeans, bar-coding, CDs, etc. They provide information about a given product. Consequently, their demand on the market place is high, yet prone to changes. Minor variations in size, colour, or raw material impact on the entire manufacturing system. Few companies have the capacity and expertise to produce them, and those that manufacture labels face tremendous production planning problems, because traditional planning systems are not suitable to handle such a complex task.

The typical flow of data in our case study is presented in Fig. 1. The labels are produced according to clients' requirements. An external application is used by the sales department to acquire clients' data, such as label design, colour, dimensions, material and specific attributes. These data are stored in the Enterprise Information System (EIS). Once the client's order is processed by the sales department, the production planning department must elaborate a *production order* on which the information to produce the label is comprised. The production order must be accurate and error-free so that the manufacturing department can perform adequately.

The possible combinations to produce a label are numerous. For example, the following raw material can be employed: glued-paper, non-glued paper, thermal paper, nylon, polyester, plastic, and cardboard to name but only a few. Then, the raw material is grouped into families to simplify the planning process and to determine how to acquire it, which can be in the form of continuous cylinders or in batches (also called master). Another complexity arises when a client sets the colours to be used (i.e. blue letters on white, glued-paper). Thus, the tints combination must be fixed according to the client's requirements. The production planning manager is responsible for establishing what the operations (SU1, SU2 and SU3) that are to be performed to comply with the client's design. On such information, the manager must determine the number of tools (up to three). Once these pieces of data are available, the

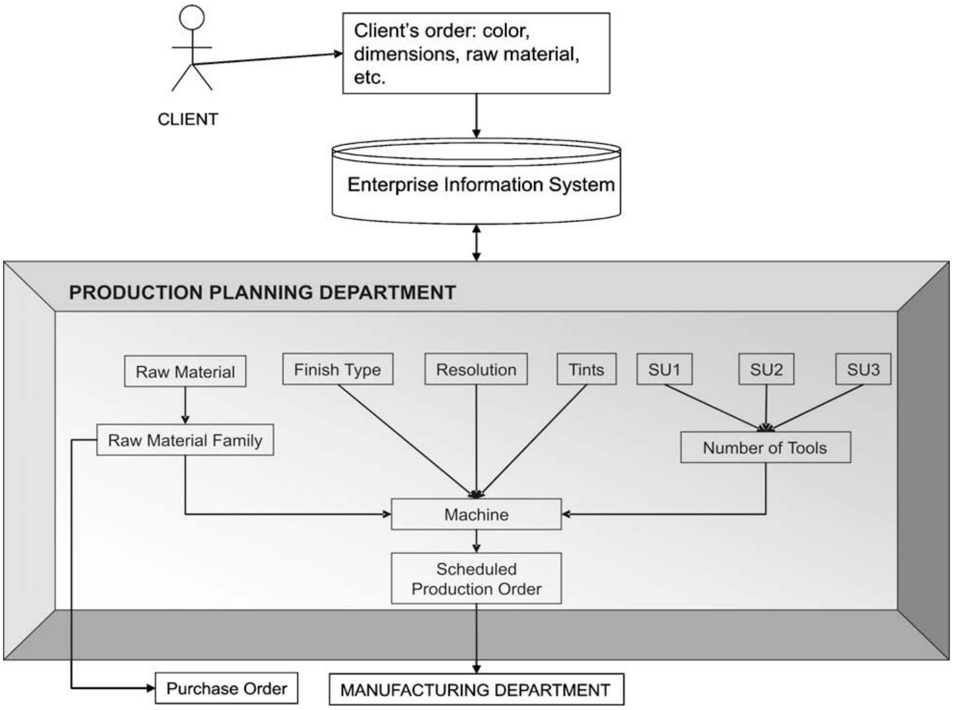

CLIENT
Client's order: color, dimensions, raw material, etc.
Enterprise Information System
PRODUCTION PLANNING DEPARTMENT
Raw Material
Finish Type
Resolution
Tints
SU1
SU2
SU3
Raw Material Family
Number of Tools
Machine
Scheduled Production Order
Purchase Order
MANUFACTURING DEPARTMENT

Fig. 1. Illustration of the case study.

production planning manager must provide the right machine (out of three options) on which the label will be produced. The machine depends on the following variables: raw material family, number of tints, resolution (dots per inch), type of finishing, and number of tools. This creates great difficulty to construct consistent production orders. Normally, the production planning manager relies on his/her experience; however, production orders normally contain imprecise data, as the manager shows inconsistent behaviour about his/her decisions. The production manager either assigns the wrong number of tools for a given machine, or launches a production order for a machine that is not suitable to deal with a specific design requirement. This situation might occur due to fatigue, or the inability of a human-being to manage larger number of combinations. Therefore, more accuracy and flexibility can be achieved by automasing the process of generating a production order.

3.2. Part II. The rule base

The first decision to be made is to establish the number of tools, the raw material family, and the way to purchase such raw material. These decisions are possible to be achieved by setting a rule base. Therefore, the rule base receives the following data: length, width, raw material, and the content of three variables known as SU1, SU2, and SU3. In this way, the rule base provides the number of tools, based on the values of SU1, SU2 and SU3. This is exemplified by the following excerpt:

ruleT1: IF SU1 ! = ““none”” AND SU2 ! = ““none”” AND SU3 ! = ““none”” THEN number_tools = 3
ruleT2: IF SU1 = ““none”” AND SU2 = ““none”” AND SU3 = “none” THEN number_tools = 1
ruleT3: IF SU1 ! = “none” AND SU2 = “none” AND SU3 · = “none” THEN number_tools = 1
ruleT4: IF SU1 = “none” AND SU2 ! = “none” AND SU3 = “none” THEN number_tools = 1
ruleT5: IF SU1 = “none” AND SU2 = “none” AND SU3 ! = “none” THEN number_tools = 1
ruleT6: IF SU1 ! = “none” AND SU2 ! = “none” AND SU3 = “none” THEN number_tools = 2
ruleT7: IF SU1 ! = “none” AND SU2 = “none” AND SU3 ! = “none” THEN number_tools = 2
ruleT8: IF SU1 = “none” AND SU2 ! = “none” AND SU3 ! = “none” THEN number_tools = 2

Based on the raw material, the raw material family is established. Some rules read:

ruleRMF01: IF RAW_MATERIAL = “kimno” THEN RAW_MATERIAL_FAMILY = “kimdura”
ruleRMF02: IF RAW_MATERIAL = “kimsi” THEN RAW_MATERIAL_FAMILY = “kimdura”
ruleRMF03: IF RAW_MATERIAL = “kimte” THEN RAW_MATERIAL_FAMILY = “kimdura”
ruleRMF04: IF RAW_MATERIAL = “valer” THEN RAW_MATERIAL_FAMILY = “fabric”
ruleRMF05: IF RAW_MATERIAL = “poliester” THEN RAW_MATERIAL_FAMILY = “fabric”

ruleRMF06: IF RAW_MATERIAL = “nylon” THEN RAW_MATERIAL_FAMILY = “fabric”
ruleRMF07: IF RAW_MATERIAL = “auttr” THEN RAW_MATERIAL_FAMILY = “paper”
ruleRMF08: IF RAW_MATERIAL = “orono” THEN RAW_MATERIAL_FAMILY = “paper”
ruleRMF09: IF RAW_MATERIAL = “oroag” THEN RAW_MATERIAL_FAMILY = “paper”
ruleRMF10: IF RAW_MATERIAL = “cartd” THEN RAW_MATERIAL_FAMILY = “cardboard”
ruleRMF11: IF RAW_MATERIAL = “carts” THEN RAW_MATERIAL_FAMILY = “cardboard”
ruleRMF12: IF RAW_MATERIAL = “bopbc” THEN RAW_MATERIAL_FAMILY = “bopp”
ruleRMF13: IF RAW_MATERIAL = “boppm” THEN RAW_MATERIAL_FAMILY = “bopp”
ruleRMF14: IF RAW_MATERIAL = “boptr” THEN RAW_MATERIAL_FAMILY = “bopp”

Then, the type of purchase is set. Should raw material be purchased on continuous cylinders or on batches, it all depends on the raw material family. An excerpt of the rules to decide what type of purchase must be made, follows here:

ruleTOP1: IF RAW_MATERIAL_FAMILY = “fabric”AND RAW_MATERIAL ! = “nylon”
THEN TYPE_OF_PURCHSE = “master”
ruleTOP2: IF RAW_MATERIAL_FAMILY = “fabric” AND RAW_MATERIAL = “nylon”
THEN TYPE-OF_PURCHASE = “continuous cylinder”
ruleTOP3: IF RAW_MATERIAL_FAMILY = “cardboard” AND RAW_MATERIAL ! = “cartd”
THEN TYPE_OF_PURCHASE = “master”
ruleTOP4: IF RAW_MATERIAL_FAMILY = “cardboard” AND RAW_MATERIAL = “cartd”
THEN TYPE_OF_PURCHASE = “continuous cylinder”
ruleTOP5: IF RAW_MATERIAL_FAMILY = “kimdura”
THEN TYPE_OF_PURCHASE = “master”
ruleTOP6: IF RAW_MATERIAL_FAMILY = “various”
THEN TYPE_OF_PURCHASE = “master”
ruleTOP7: IF RAW_MATERIAL_FAMILY = “paper” AND RAW_MATERIAL ! = “dual”
THEN TYPE_OF_PURCHASE = “master”
ruleTOP8: IF RAW_MATERIAL_FAMILY = “paper” AND RAW_MATERIAL = “dual”
THEN TYPE_OF_PURCHASE = “continuous_cylinder”
ruleTOP9: IF RAW_MATERIAL_FAMILY = “bopp” THEN TYPE_OF_PURCHASE = “master”

Consequently, one agent must possess the previous knowledge, receive input data, and provide values for the number of tools, the way to purchase raw material, and the raw material family. This agent is called *tool agent* (see Section 4 for more details). The purchasing department is informed about the type of purchase, whereas the number of tools and the raw material family are used along with other data to determine the machine that will be in charge of producing the label.

3.3. Part III. The FANN and the machine

We provide details of the feed-forward artificial neural network to determine the right machine. As it has been stated before, obtaining the appropriate machine depends on the combination of five different data sets, as follows:

$$Raw\_Material\_Family\mathbf{X}Tints\mathbf{X}Resolution\mathbf{X}Finishing\_Type\mathbf{X}Number\_Tools- > Machine$$

Each input data set contains the following elements:

$$Raw\_Material\_Family = \{Fabric, Paper, Kimdura, Cardboard, BOPP, Other\}$$
$$Number\_Tools = \{1, 2, 3\}$$
$$Tints = \{1, 2, 3, 4, 5, 6\}$$
$$Resolution = \{notspecified, 280, 360, 400, 500, 600, 700, 800\}$$
$$Finishing\_Type = \{notspecified, laminated, sulfated\}$$

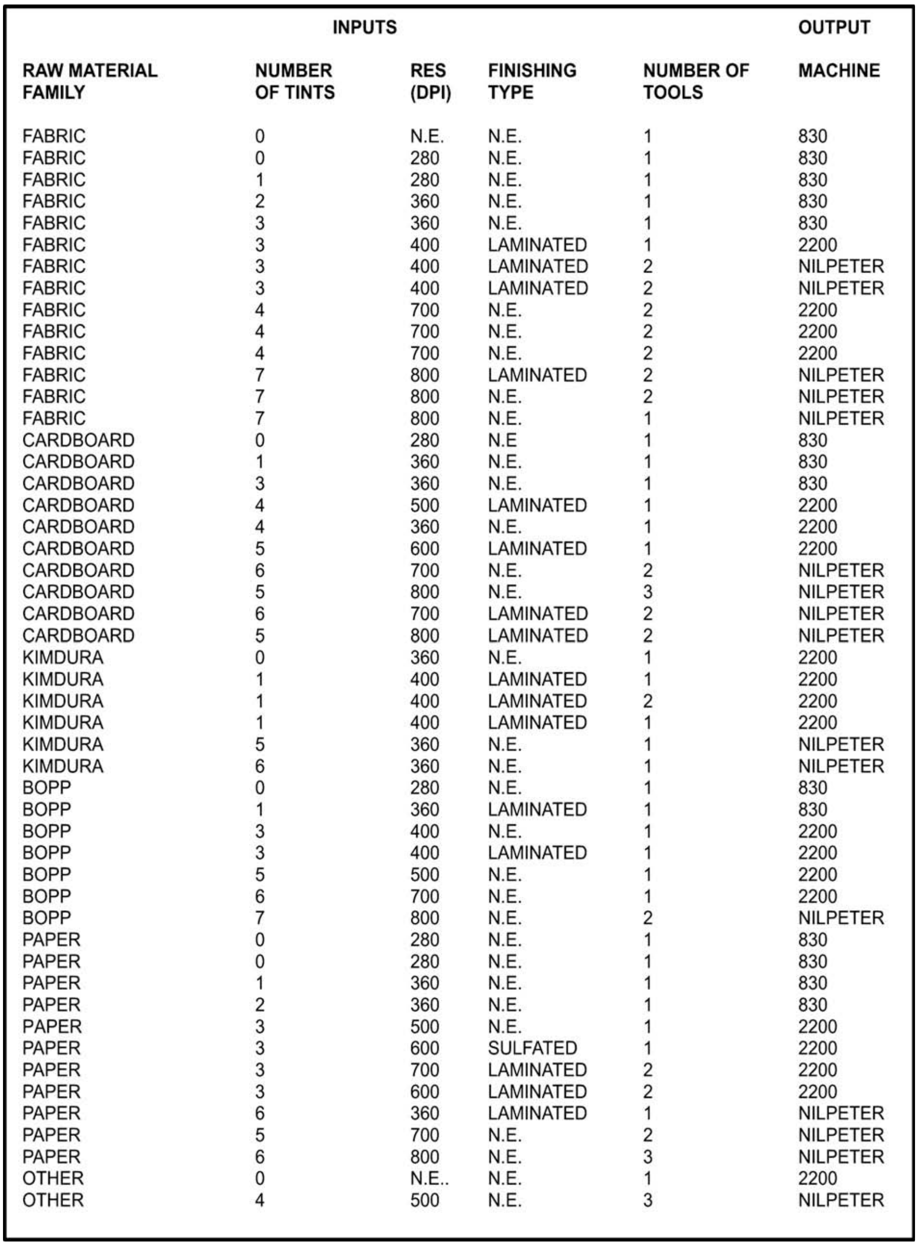
The output data set is formed by

$$Machine = \{830, 2200, Nilpeter\}$$

The raw material family and the number of tools are obtained by the rule base previously described; the rest of the input variables are provided by the planning department based on the client’s specifications.

The entire range of possibilities to determine a right machine represents an NP-Hard combinatorial problem. We chose the feed-forward artificial neural network architecture because such neural networks are known to be good classifiers. They are trained by backpropagating the error signal, based on the gradient descent technique: the error value diminishes as the gradient of a given function i.e. the logistic function, is calculated (Jang, Sun, & Mizutani, 1997). In each training pass, the weights of the net are updated. When the error value is less than a given target value, the training phase is complete and the resultant weights are stored. A fiable training matrix is employed to train the feed-forward artificial neural network (Fig. 2), which is a subset of the entire range of possibilities to set a machine.

The actual configuration of the FANN is related to the training matrix. Specifically, one processing unit is created for each value contained in the training set. For example,



INPUTS					OUTPUT
RAW MATERIAL FAMILY	NUMBER OF TINTS	RES (DPI)	FINISHING TYPE	NUMBER OF TOOLS	MACHINE
FABRIC	0	N.E.	N.E.	1	830
FABRIC	0	280	N.E.	1	830
FABRIC	1	280	N.E.	1	830
FABRIC	2	360	N.E.	1	830
FABRIC	3	360	N.E.	1	830
FABRIC	3	400	LAMINATED	1	2200
FABRIC	3	400	LAMINATED	2	NILPETER
FABRIC	3	400	LAMINATED	2	NILPETER
FABRIC	4	700	N.E.	2	2200
FABRIC	4	700	N.E.	2	2200
FABRIC	4	700	N.E.	2	2200
FABRIC	7	800	LAMINATED	2	NILPETER
FABRIC	7	800	N.E.	2	NILPETER
FABRIC	7	800	N.E.	1	NILPETER
CARDBOARD	0	280	N.E	1	830
CARDBOARD	1	360	N.E.	1	830
CARDBOARD	3	360	N.E.	1	830
CARDBOARD	4	500	LAMINATED	1	2200
CARDBOARD	4	360	N.E.	1	2200
CARDBOARD	5	600	LAMINATED	1	2200
CARDBOARD	6	700	N.E.	2	NILPETER
CARDBOARD	5	800	N.E.	3	NILPETER
CARDBOARD	6	700	LAMINATED	2	NILPETER
CARDBOARD	5	800	LAMINATED	2	NILPETER
KIMDURA	0	360	N.E.	1	2200
KIMDURA	1	400	LAMINATED	1	2200
KIMDURA	1	400	LAMINATED	2	2200
KIMDURA	1	400	LAMINATED	1	2200
KIMDURA	5	360	N.E.	1	NILPETER
KIMDURA	6	360	N.E.	1	NILPETER
BOPP	0	280	N.E.	1	830
BOPP	1	360	LAMINATED	1	830
BOPP	3	400	N.E.	1	2200
BOPP	3	400	LAMINATED	1	2200
BOPP	5	500	N.E.	1	2200
BOPP	6	700	N.E.	1	2200
BOPP	7	800	N.E.	2	NILPETER
PAPER	0	280	N.E.	1	830
PAPER	0	280	N.E.	1	830
PAPER	1	360	N.E.	1	830
PAPER	2	360	N.E.	1	830
PAPER	3	500	N.E.	1	2200
PAPER	3	600	SULFATED	1	2200
PAPER	3	700	LAMINATED	2	2200
PAPER	3	600	LAMINATED	2	2200
PAPER	6	360	LAMINATED	1	NILPETER
PAPER	5	700	N.E.	2	NILPETER
PAPER	6	800	N.E.	3	NILPETER
OTHER	0	N.E..	N.E.	1	2200
OTHER	4	500	N.E.	3	NILPETER

Fig. 2. The training matrix.

as input 1 of the training matrix has six different values, six processing units are built. However, the actual value is represented by a string, which is not a suitable input type for the FANN. Thus, such values are converted to a stream of 0's and 1's. Table 1 illustrates the codification for the raw material family.

The prior codification is necessary because FANNs only handle values within the closed interval [0, 1]. Therefore, the actual input and output values that the net receives and obtains are 0's and 1's. This codification–decodification is done by the encoder class attached to the *machine agent* (see Fig. 5). Therefore, the FANN has six processing units in the input layer, which are in charge of dealing exclusively with the raw material family. The totality of discrete values contained in the input and output sets were codified in a similar way. Tables 2–6 show the resultant codification.

Table 1
Codification of the *raw material family* set

Input stream						Raw material family
0	0	0	0	0	1	Paper
0	0	0	0	1	0	Fabric
0	0	0	1	0	0	Kimdura
0	0	1	0	0	0	Cardboard
0	1	0	0	0	0	BOPP
1	0	0	0	0	0	Other

Consequently, the number of processing units in the input layer of the FANN equals the number of codified input values. For this case, 25 processing units in the input

Table 2
Codification of the *tints* set

Input stream								Number of tints
0	0	0	0	0	0	0	0	
0	0	0	0	0	0	1	1	
0	0	0	0	0	1	0	2	
0	0	0	0	1	0	0	3	
0	0	0	1	0	0	0	4	
0	0	1	0	0	0	0	5	
0	1	0	0	0	0	0	6	
1	0	0	0	0	0	0	7	

Table 3
Codification of the *resolution* set

Input stream								Resolution
0	0	0	0	0	0	0	1	Not specified
0	0	0	0	0	0	1	0	280
0	0	0	0	0	1	0	0	360
0	0	0	0	1	0	0	0	400
0	0	0	1	0	0	0	0	500
0	0	1	0	0	0	0	0	600
0	1	0	0	0	0	0	0	700
1	0	0	0	0	0	0	0	800

Table 4
Codification for the *finishing type* set

Input stream		Finishing type
0	0	Not specified
0	1	Laminated
1	0	Sulfated

Table 5
Codification for the number of tools set

Input stream		Number of tools
0	0	1
0	1	2
1	0	3

Table 6
Codification of the *machine* set

Input stream			Machine
0	0	1	830
0	1	0	2200
1	0	0	Nilpeter

layer are fixed. Once the processing units for the input layer are generated, then the intermediate layer is set. The Java code assigns the same number of processing units in the input layer to the intermediate layer. Only one intermediate layer is created, since a FANN with a single intermediate layer is powerful enough to deal with non-linear problems. The output layer is then constructed.

The number of processing units in the output layer also depends on the number of values that are provided as valid outcomes in the training set. In our training matrix, the output layer is given three different values, and three processing units are constructed.

According to the previous codification, the resultant FANN consists of 3 layers, 25 processing units in both, the input and intermediate layers, and three processing units in the output layer. This is represented in Fig. 3. The processing units on the input layer receive streams of 0's and 1's, which result after codifying the incoming symbol. The reverse process is carried out for the output layer, where the processing units provide 0's and 1's, all of which

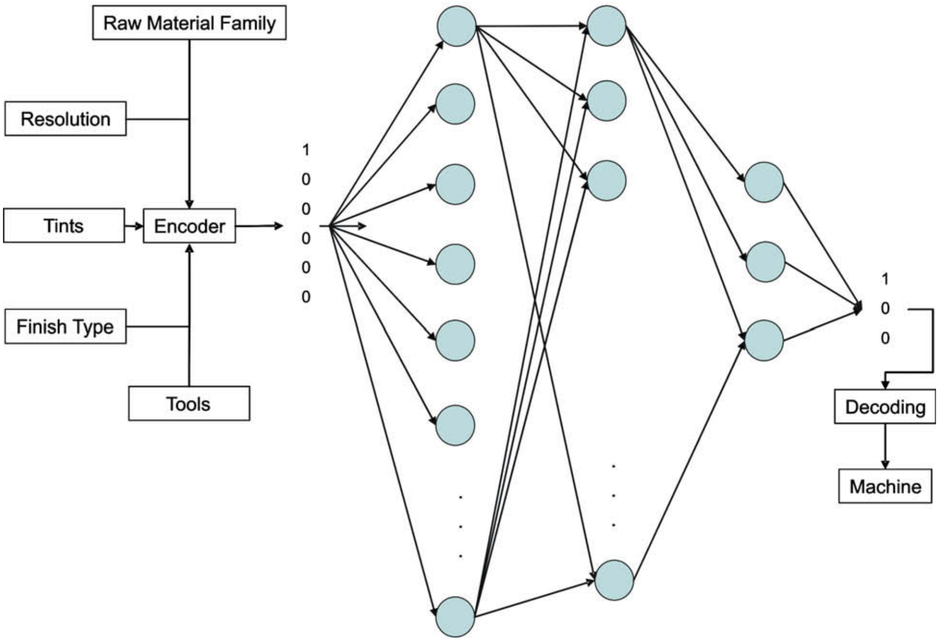
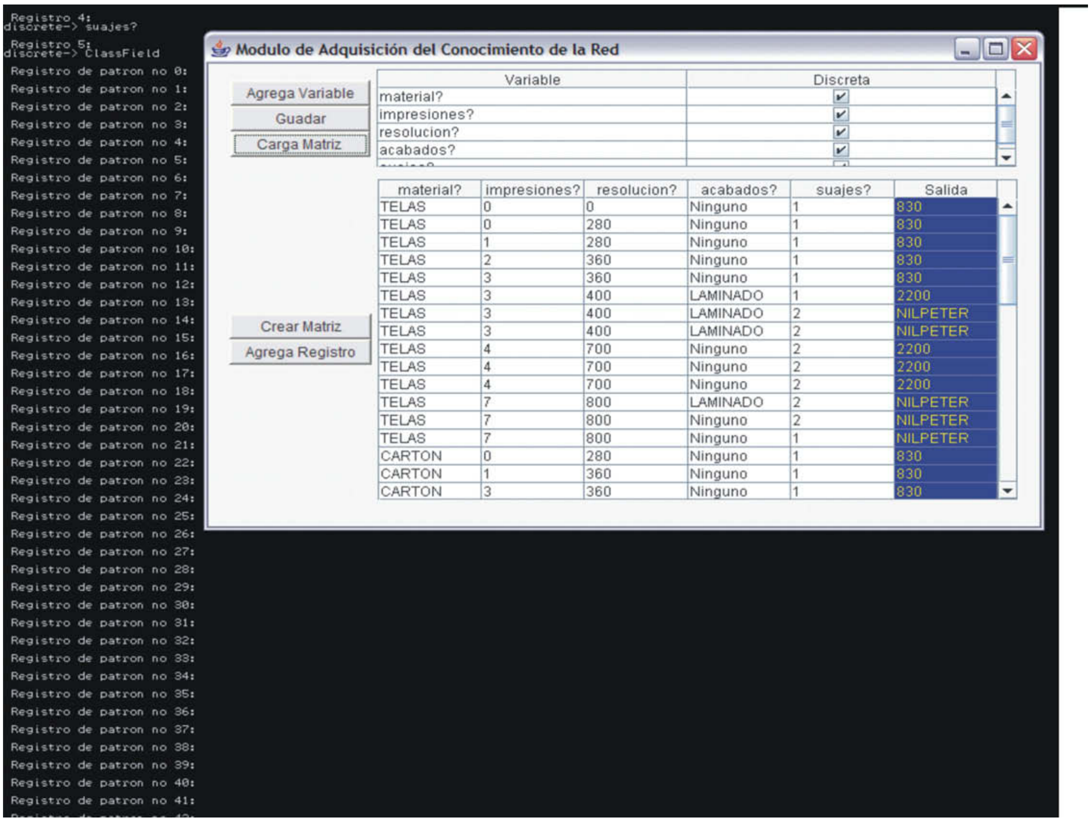


Fig. 3. Configuration of the resultant FANN.



```
 Registro 4:
discrete-> suajes?
 Registro 5:
discrete-> ClassField
 Registro de patron no 0:
 Registro de patron no 1:
 Registro de patron no 2:
 Registro de patron no 3:
 Registro de patron no 4:
 Registro de patron no 5:
 Registro de patron no 6:
 Registro de patron no 7:
 Registro de patron no 8:
 Registro de patron no 9:
 Registro de patron no 10:
 Registro de patron no 11:
 Registro de patron no 12:
 Registro de patron no 13:
 Registro de patron no 14:
 Registro de patron no 15:
 Registro de patron no 16:
 Registro de patron no 17:
 Registro de patron no 18:
 Registro de patron no 19:
 Registro de patron no 20:
 Registro de patron no 21:
 Registro de patron no 22:
 Registro de patron no 23:
 Registro de patron no 24:
 Registro de patron no 25:
 Registro de patron no 26:
 Registro de patron no 27:
 Registro de patron no 28:
 Registro de patron no 29:
 Registro de patron no 30:
 Registro de patron no 31:
 Registro de patron no 32:
 Registro de patron no 33:
 Registro de patron no 34:
 Registro de patron no 35:
 Registro de patron no 36:
 Registro de patron no 37:
 Registro de patron no 38:
 Registro de patron no 39:
 Registro de patron no 40:
 Registro de patron no 41:
```

Variable	Discreta
material?	☑
impresiones?	☑
resolucion?	☑
acabados?	☑

material?	impresiones?	resolucion?	acabados?	suajes?	Salida
TELAS	0	0	Ninguno	1	830
TELAS	0	280	Ninguno	1	830
TELAS	1	280	Ninguno	1	830
TELAS	2	360	Ninguno	1	830
TELAS	3	360	Ninguno	1	830
TELAS	3	400	LAMINADO	1	2200
TELAS	3	400	LAMINADO	2	NILPETER
TELAS	3	400	LAMINADO	2	NILPETER
TELAS	4	700	Ninguno	2	2200
TELAS	4	700	Ninguno	2	2200
TELAS	4	700	Ninguno	2	2200
TELAS	7	800	LAMINADO	2	NILPETER
TELAS	7	800	Ninguno	2	NILPETER
TELAS	7	800	Ninguno	1	NILPETER
CARTON	0	280	Ninguno	1	830
CARTON	1	360	Ninguno	1	830
CARTON	3	360	Ninguno	1	830

Fig. 4. The software module to train the FANN.

are further converted to the symbol that represents a machine.

Although there are countless simulators of neural networks, none of them is suited to be used by a multi-agent system. Hence, it was necessary to incorporate supervised learning in the *machine agent* by means of Java programming. To comply rightly with the requirements to implement backpropagation, a number of technological innovations were carried out, adapting the work of (Bigus & Bigus, 2002) on Java coding of backpropagation. Fig. 4 presents the module that we created to set training matrices for FANNs.

4. Design and implementation of the multi-agent system

The primary goal of the MAS is to construct a production order. Our solution is to decompose the decision process carried out by the human manager in a series of tasks performed by software agents. These tasks are:

1. To read the EIS for newly created sales orders, and to acquire relevant variables.
2. To determine all the necessary tooling information.
3. To determine the correct machine.
4. To establish a sequence for launching production orders to the manufacturing department.
5. To maintain data consistency and robustness along the process.

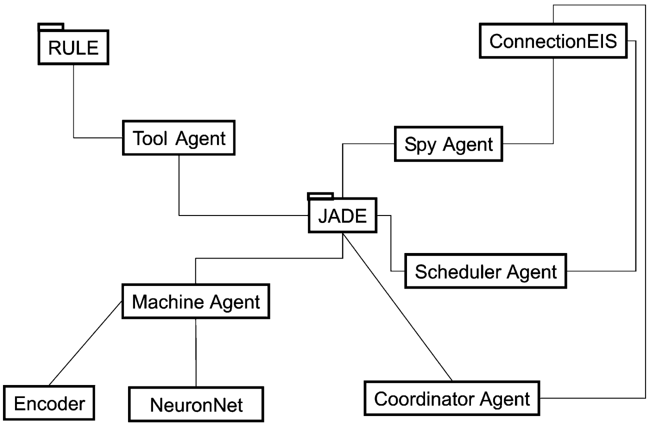


Fig. 5. Structure of the MAS.

According to the previous requirements the general design of the multi-agent system is presented. The AUML (Bauer & Odell, 2005) has been employed extensively to model the MAS, whereas JADE is employed as the implementation platform. The class diagram of the system is presented in Fig. 5. The MAS has the following agents:

- coordinator agent
- machine agent
- tool agent
- spy agent
- scheduler agent

In order to actually implement intelligent agents on the JADE platform, external tools must be employed. In the proposed design both, the *machine agent* and the *tool agent* possess intelligent capabilities. The *tool agent* has a rule-based system to determine the right tooling, whereas, the *machine agent* incorporates a FANN. In the previous section we already discussed about the specifics of the rule base and the FANN.

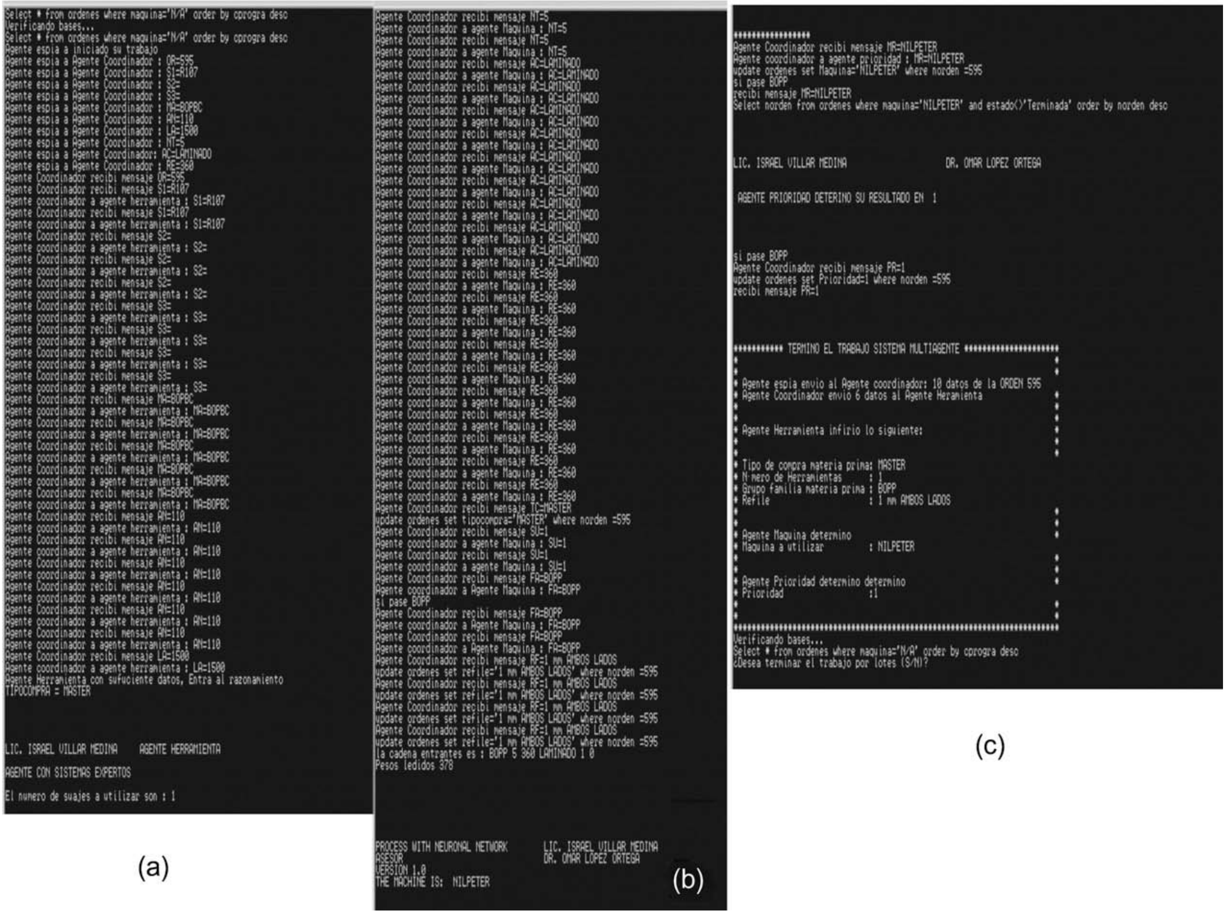
The *coordinator agent* is responsible for maintaining data consistency during the process by controlling the flow of messages. A *spy agent* must read the Enterprise Information System in order to acquire clients' orders, and send the information to the *coordinator*. The *machine agent* obtains the appropriate machine, whereas the *tool agent* is responsible for providing the right tooling. As soon as the machine, tools and other data are established, a priority is assigned to the production order. To do so, we include a *scheduler agent*. When the production order gets the priority, it is then launched to the manufacturing department. The *behaviours* of the agents were set up properly in order to avoid interferences while the agents are exchanging information. Interested readers may consult (Bellifemine, Caire, & Greenwood, 2007) for insights on how to program agents on the JADE platform. In the following section we present the results achieved by the multi-agent system.

4.1. The MAS exemplified

We illustrate the series of tasks and partial decisions achieved by the software agents. We deliberately use the command line to illustrate the entire process of message exchange between the agents. The process starts as soon as the sales department enters a client's requirement. The *spy agent*, which possesses a *cyclic behaviour*, realizes that a new order has just entered the EIS, and it sends a series of messages to the *coordinator*, informing about the values of the following variables:

In the example, the client demands labels made of a material called "BOPBC", with a laminated finishing type, and a resolution of 360 dots per inch. The planning department feeds the following values to the EIS:

Sales order: 595
Width: 110
Length: 1500
Raw material: BOPBC
SU1: R107
SU2: " "
SU3: " "
Number of tints: 5
Resolution: 360
Type of finishing: laminated



```
Select * from ordenes where maquina='N/A' order by cprogra desc
Verificando bases...
Select * from ordenes where maquina='N/A' order by cprogra desc
Agente espia a iniciado su trabajo
Agente espia a Agente Coordinador : OR=595
Agente espia a Agente Coordinador : S1=R107
Agente espia a Agente Coordinador : S2=
Agente espia a Agente Coordinador : S3=
Agente espia a Agente Coordinador : MA=BOPBC
Agente espia a Agente Coordinador : AN=110
Agente espia a Agente Coordinador : LA=1500
Agente espia a Agente Coordinador : NT=5
Agente espia a Agente Coordinador: AC=LAMINADO
Agente espia a Agente Coordinador : RE=360
Agente Coordinador recibi mensaje OR=595
Agente Coordinador recibi mensaje S1=R107
Agente coordinador a agente herramienta : S1=R107
Agente Coordinador recibi mensaje S1=R107
Agente coordinador a agente herramienta : S1=R107
Agente Coordinador recibi mensaje S2=
Agente coordinador a agente herramienta : S2=
Agente Coordinador recibi mensaje S2=
Agente coordinador a agente herramienta : S2=
Agente Coordinador recibi mensaje S2=
Agente coordinador a agente herramienta : S2=
Agente Coordinador recibi mensaje S3=
Agente coordinador a agente herramienta : S3=
Agente Coordinador recibi mensaje S3=
Agente coordinador a agente herramienta : S3=
Agente Coordinador recibi mensaje S3=
Agente coordinador a agente herramienta : S3=
Agente Coordinador recibi mensaje S3=
Agente coordinador a agente herramienta : S3=
Agente Coordinador recibi mensaje MA=BOPBC
Agente coordinador a agente herramienta : MA=BOPBC
Agente Coordinador recibi mensaje MA=BOPBC
Agente coordinador a agente herramienta : MA=BOPBC
Agente Coordinador recibi mensaje MA=BOPBC
Agente coordinador a agente herramienta : MA=BOPBC
Agente Coordinador recibi mensaje MA=BOPBC
Agente coordinador a agente herramienta : MA=BOPBC
Agente Coordinador recibi mensaje AN=110
Agente coordinador a agente herramienta : AN=110
Agente Coordinador recibi mensaje AN=110
Agente coordinador a agente herramienta : AN=110
Agente Coordinador recibi mensaje AN=110
Agente coordinador a agente herramienta : AN=110
Agente Coordinador recibi mensaje AN=110
Agente coordinador a agente herramienta : AN=110
Agente Coordinador recibi mensaje AN=110
Agente coordinador a agente herramienta : AN=110
Agente Coordinador recibi mensaje LA=1500
Agente coordinador a agente herramienta : LA=1500
Agente Herramienta con sufuciente datos, Entra al razonamiento
TIPOCOMPRA = MASTER



LIC. ISRAEL VILLAR MEDINA    AGENTE HERRAMIENTA

AGENTE CON SISTEMAS EXPERTOS

El numero de suajes a utilizar son : 1
```

```
Agente Coordinador recibi mensaje NT=5
Agente coordinador a agente Maquina : NT=5
Agente Coordinador recibi mensaje NT=5
Agente coordinador a agente Maquina : NT=5
Agente Coordinador recibi mensaje AC=LAMINADO
Agente coordinador a agente Maquina : AC=LAMINADO
Agente Coordinador recibi mensaje AC=LAMINADO
Agente coordinador a agente Maquina : AC=LAMINADO
Agente Coordinador recibi mensaje AC=LAMINADO
Agente coordinador a agente Maquina : AC=LAMINADO
Agente Coordinador recibi mensaje AC=LAMINADO
Agente coordinador a agente Maquina : AC=LAMINADO
Agente Coordinador recibi mensaje AC=LAMINADO
Agente coordinador a agente Maquina : AC=LAMINADO
Agente Coordinador recibi mensaje AC=LAMINADO
Agente coordinador a agente Maquina : AC=LAMINADO
Agente Coordinador recibi mensaje AC=LAMINADO
Agente coordinador a agente Maquina : AC=LAMINADO
Agente Coordinador recibi mensaje AC=LAMINADO
Agente coordinador a agente Maquina : AC=LAMINADO
Agente Coordinador recibi mensaje AC=LAMINADO
Agente coordinador a agente Maquina : AC=LAMINADO
Agente Coordinador recibi mensaje RE=360
Agente coordinador a agente Maquina : RE=360
Agente Coordinador recibi mensaje RE=360
Agente coordinador a agente Maquina : RE=360
Agente Coordinador recibi mensaje RE=360
Agente coordinador a agente Maquina : RE=360
Agente Coordinador recibi mensaje RE=360
Agente coordinador a agente Maquina : RE=360
Agente Coordinador recibi mensaje RE=360
Agente coordinador a agente Maquina : RE=360
Agente Coordinador recibi mensaje RE=360
Agente coordinador a agente Maquina : RE=360
Agente Coordinador recibi mensaje RE=360
Agente coordinador a agente Maquina : RE=360
Agente Coordinador recibi mensaje RE=360
Agente coordinador a agente Maquina : RE=360
Agente Coordinador recibi mensaje RE=360
Agente coordinador a agente Maquina : RE=360
Agente Coordinador recibi mensaje RE=360
Agente coordinador a agente Maquina : RE=360
Agente Coordinador recibi mensaje TC=MASTER
update ordenes set tipocompra='MASTER' where norden =595
Agente Coordinador recibi mensaje SU=1
Agente coordinador a agente Maquina : SU=1
Agente Coordinador recibi mensaje SU=1
Agente coordinador a agente Maquina : SU=1
Agente Coordinador recibi mensaje FA=BOPP
Agente coordinador a Agente Maquina : FA=BOPP
si pase BOPP
Agente Coordinador recibi mensaje FA=BOPP
Agente coordinador a Agente Maquina : FA=BOPP
Agente Coordinador recibi mensaje FA=BOPP
Agente coordinador a Agente Maquina : FA=BOPP
Agente Coordinador recibi mensaje RF=1 mm AMBOS LADOS
update ordenes set refile='1 mm AMBOS LADOS' where norden =595
Agente Coordinador recibi mensaje RF=1 mm AMBOS LADOS
update ordenes set refile='1 mm AMBOS LADOS' where norden =595
Agente Coordinador recibi mensaje RF=1 mm AMBOS LADOS
update ordenes set refile='1 mm AMBOS LADOS' where norden =595
Agente Coordinador recibi mensaje RF=1 mm AMBOS LADOS
update ordenes set refile='1 mm AMBOS LADOS' where norden =595
la cadena entrantes es : BOPP 5 360 LAMINADO 1 0
Pesos ledidos 378



PROCESS WITH NEURONAL NETWORK      LIC. ISRAEL VILLAR MEDINA
ASESOR                             DR. OMAR LOPEZ ORTEGA
VERSION 1.0
THE MACHINE IS:  NILPETER
```

```
*****************
Agente Coordinador recibi mensaje MR=NILPETER
Agente coordinador a agente prioridad : MR=NILPETER
update ordenes set Maquina='NILPETER' where norden =595
si pase BOPP
recibi mensaje MR=NILPETER
Select norden from ordenes where maquina='NILPETER' and estado<>'Terminada' order by norden desc



LIC. ISRAEL VILLAR MEDINA                 DR. OMAR LOPEZ ORTEGA


 AGENTE PRIORIDAD DETERINO SU RESULTADO EN  1



si pase BOPP
Agente Coordinador recibi mensaje PR=1
update ordenes set Prioridad=1 where norden =595
recibi mensaje PR=1



*********** TERMINO EL TRABAJO SISTEMA MULTIAGENTE *********************
*                                                                      *
*                                                                      *
* Agente espia envio al Agente coordinador: 10 datos de la ORDEN 595
* Agente Coordinador envio 6 datos al Agente Heramienta                *
*                                                                      *
*                                                                      *
* Agente Herramienta infirio lo siguiente:                             *
*                                                                      *
*                                                                      *
* Tipo de compra materia prima: MASTER
* N·mero de Herramientas      : 1
* Grupo familia materia prima : BOPP
* Refile                      : 1 mm AMBOS LADOS
*                                                                      *
*                                                                      *
* Agente Maquina determino                                             *
* Maquina a utilizar          : NILPETER
*                                                                      *
*                                                                      *
* Agente Prioridad determino determino                                 *
* Prioridad                   :1
*                                                                      *
*                                                                      *
************************************************************************
Verificando bases...
Select * from ordenes where maquina='N/A' order by cprogra desc
¿Desea terminar el trabajo por lotes (S/N)?
```

Fig. 6. Inferences and communication occurring inside the MAS.

Fig. 6a shows how the *spy agent* informs the *coordinator agent* about the previous data, organized in the sales order 595. When the *coordinator* acknowledges reception of messages from the *spy*, then it informs the *tool agent* regarding width, length, raw material, SU1, SU2 and SU3.

The *tool agent* assesses that it has enough data to initiate its reasoning process (“Agente herramienta con suficientes datos”), then it runs the expert system and obtains a conclusion (Fig. 6a, bottom). The following rules are triggered according to the input data of the current example:

ruleRMF12: IF RAW_MATERIAL = “bopbc” THEN RAW_MATERIAL_FAMILY = “bopp”
ruleTOP9: IF RAW_MATERIAL_FAMILY = “bopp” THEN TYPE_OF_PURCHASE = “master”
ruleT3: IF SU1 ! = “none” AND SU2 = “none” AND SU3 = “none” THEN number_tools = 1

Therefore, the *tool agent* reaches the next conclusion:

RAW MATERIAL FAMILY: BOPP
NUMBER OF TOOLS: 1
TYPE OF PURCHASE: Master (batch)

The *tool agent* informs the *coordinator agent* about its conclusions. On the reception of messages from the *tool agent*, the *coordinator* sends the following data to the *machine agent* (Fig. 6b, bottom – “la cadena entrante es: BOPP 5 360 Laminado 1”):

RAW MATERIAL FAMILY: BOPP
NUMBER OF TINTS: 5
RESOLUTION: 360
TYPE OF FINISHING: laminated
NUMBER OF TOOLS: 1

Based on this type of information, the FANN embedded within the *machine agent*, which is set into execution mode, obtains a valid machine, and sends a message to the *coordinator* (Fig. 6c – “Agente coordinador recibi mensaje MR = Nilpeter”). In this example, the obtained machine is known as “Nilpeter”, which is a valid machine for this combination of values. For this production order, the assigned priority is 1, since it was the only order set the day of executing this example. The bottom side of Fig. 6c summarizes the results. This process is repeated entirely every time the *spy agent* reads a new sales order. Thus, the construction and release of a valid production order is fully automized by the intelligent and collaborative system that we developed.

5. Conclusions and research perspectives

Based on theoretical and practical results, we sustain that multi-agent systems represent a major swift in supporting systems for manufacturing. We contribute to agent-based production planning with the collaborative and intelligent MAS presented here. Marík (*op cit*) argues that the *agentification* process of the production planning department provides an elegant mechanism for system integration, and supports the migration from centralized planning towards distributed and flexible architectures. The MAS we developed contributes to achieve this flexibility. To the best of the authors' knowledge, this is the first report on how to coordinate a rule-based system and supervised learning, making more flexible the production planning activity.

Although production planning is improved by the collaborative and intelligent system presented here, it is necessary to develop a solid scheduling policy to provide more realistic plans. Our *scheduler agent* employs a FIFO policy, even though it is the simplest ordering/queuing mechanism. We can asses that integrating planning and scheduling is not a trivial task, yet it opens up opportunities to actually close the loop between production planning and manufacturing execution. The current configuration of the MAS that we developed enhances production *planning*, yet production *control* is not fully covered. Thus, a dynamic control policy might be obtained when relevant variables, such as Work In Process (WIP) inventory, backorder penalty cost, setup time and cost, processing and transportation cost and time, or machine disruption time, are sent back to planning for achieving production control. The loop is closed when events within the manufacturing department, such as machine failure or a delay in the setup time, are communicated in real-time to the planning department. Therefore, communication must be bidirectional, from production planning to manufacturing execution and vice versa. We suggest to design a hierarchy of agents, or a *holarchy* (Walter, Brennan, & Norrie, 2006), to accomplish agent-based PAC. Communication, intelligence, and the capability to integrate data from external applications are key features that must be exploited thoroughly. Achieving such a necessary feedback is an opportunity to continue research in the field of agent-based manufacturing.

References

Bauer, B., & Odell, J. (2005). UML 2.0 and agents: How to build agent-based systems with the new UML standard. *Engineering Applications of Artificial Intelligence, 18*, 141–157.

Bellifemine, F. L., Caire, G., & Greenwood, D. (2007). Developing multi-agent systems with JADE. USA: John Wiley and Sons.

Bigus, J., & Bigus, J. (2002). Constructing intelligent agents using java (2nd ed.). NY: John Wiley and Sons.

Browne, J., Harhen, J., & Shivnan, J. (1992). *Production management systems. A CIM perspective*. UK: Addison – Wesley.

Feng, S. C. (2005). Preliminary design and manufacturing planning integration using web-based intelligent agents. *Journal of Intelligent Manufacturing, 16*, 423–437.

Feng, S. C., Stouffer, K. A., & Jurrens, K. K. (2005). Manufacturing planning and predictive process model integration using software agents. *Advanced Engineering Informatics, 19*, 135–142.

Fichtner, D., Nestler, A., Dang, T., Schulze, A., Carlsten, U., Schreiber, S., et al. (2006). Use of agents and neural networks for acquisition and preparation of distributed NC information to support NC planning.

International Journal of Computer Integrated Manufacturing, 19, 581–592.

Frey, D., Nimis, J., Worn, H., & Lockemann, P. (2003). Benchmarking and robust multi-agent production planning and control. *Engineering applications of Artificial Intelligence, 16*, 307–320.

Jang, J.-S. R., Sun, C.-T., & Mizutani, E. (1997). *Neuro-fuzzy and soft computing. A computational approach to learning and machine intelligence*. USA: Prentice – Hall.

Kornienko, S., Kornienko, O., & Priese, J. (2004). Application of multi-agent planning to the assignment problem. *Computers in Industry, 54*, 273–290.

López-Morales, V., & López-Ortega, O. (2005). A distributed semantic network model for a collaborative intelligent system. *Journal of Intelligent Manufacturing, 16*, 515–525.

López-Ortega, O., & López-Morales, V. (2006). Cognitive communication in a multi-agent system for distributed process planning. *International Journal of Computer Applications in Technology, 26*, 99–107.

Marík, V., & Lazanský, J. (in press). Industrial applications of agent technology. *Control Engineering Practice*, doi: doi:10.1016/j.conengprac.2006.10.001.

Mônch, L., & Stehli, M. (2006). ManufAG: A multi-agent system framework for production control of complex manufacturing systems. *Information Systems and Electronic Business*. doi:10.1007/s10257-005-0030-5.

Paternina-Arboleda, C. D., & Das, T. K. (2005). A multi-agent reinforcement learning approach to obtaining dynamic control policies for stochastic lot scheduling problems. *Simulation Modelling Practice and Theory, 13*, 389–406.

Symeonidis, A. L., Kehagias, D., & Mitkas, P. (2003). Intelligent policy recommendations on enterprise resource planning by the use of agent technology and data mining techniques. *Expert Systems with Applications, 25*, 589–602.

Walter, S. S., Brennan, R. W., & Norrie, D. H. (2006). Experience and reflection on the development of a holonic job-shop scheduling system. *International Journal of Computer Applications in Technology, 26*, 15–27.

Wang, Q., Yung, K.-L., & Ip, W.-H. (2005). An order dispatcher for dispersed manufacturers solved by a mixed variable hybrid genetic algorithm. *International Journal of Computer Integrated Manufacturing, 18*, 41–52.

Wang, L., & Shen, W. (2003). DPP: An agent-based approach for distributed process planning. *Journal of Intelligent Manufacturing, 14*, 429–439.

Wang, L., Shen, W., & Hao, Q. (2006). An overview of distributed process planning and its integration with scheduling. *International Journal of Computer Applications in Technology, 26*, 3–14.