USING DATA MINING TECHNIQUES FOR DEVELOPMENT EXPERT SYSTEMS EQUIPPED WITH LEARNING CAPABILITIES FOR USE IN AUTOMATED INDUSTRIAL PLANTS

ALEXANDRE ACÁCIO DE ANDRADE, SERGIO LUIZ PEREIRA, EDUARDO MARIO DIAS, CAIO FERNANDO FONTANA
GAESI Group, Electric Power and Automation Engineering Department
Polytechnic School at São Paulo University
Av. Prof. Luciano Gualberto, Trav. 3, 158 CP. 61548 CEP. 05508-900 – São Paulo
BRAZIL
alacacio@terra.com.br, sergio@pea.usp.br, emdias@pea.usp.br, caio.fontana@poli.usp.br

Abstract: - The use of expert systems gained importance with the growing amount of data that the current plants generate automated, in this paper, the development and tests of a specialist system based on data mining and that possesses the learning capacity is presented. The referred system was termed as NESISES (Neural System of Integration for Supervisory and Expert Systems). It operates in real time with industrial automation supervisory systems and whose aim is to minimize the frequent knowledge engineering procedures for constantly updating the base of knowledge of an Expert System (ES). NESISES was validated in both the laboratory as well as in an automation plant.

Key-Words: -Data Mining, Machine learning, Industrial Automation, Supervisory and Specialist Systems - Interconnection

1 Introduction

Expert systems (ES) operating in real time with Supervisory Systems (SS) [1], are a powerful tool to aid and optimize the human operators' action within large and complex automated industrial plants such as: Seaports, Steel industry, Petrochemical industry, Automotive industry, etc. However, specialist systems depend on knowledge engineering to be updated as along the time the group of variables and equipment of automated complex plants go through some alterations. This article presents the development and the validation tests of NESISES (Neural System of Integration for Supervisory and Expert Systems) performed in both a laboratory and in a seaport automation application at Santos (Brazil).
It is show a data mining approach to get useful knowledge from huge data bases generated for industrial plants.
Figure 1 depicts the required functionality of an ES operating in real time with a SS that has learning ability.

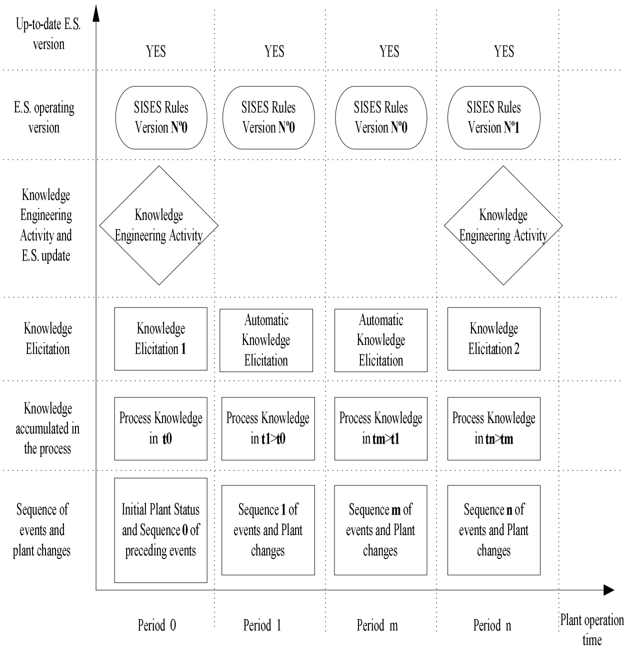

Figure 1 - NESISES functionality along the time.

It can be seen that from period 1 to period n, the knowledge elicitation is automatic. Conventional expert systems do not have this characteristic, thus, they must be constantly updated by the knowledge engineering [3] so as not to become obsolete.

As shown in Fig. 2, NESISES is an ES (Expert System) with learning ability which operates in real time with industrial automation SSs (Supervisory Systems). The learning ability of NESISES uses one of the artificial intelligence science techniques called DM (Data Mining) [4].
The DM algorithm used at NESISES is the Tertius algorithm [6]. Tertius makes use of heuristic approaches so as to make the searching rules more efficient. Through an A* type algorithm [6], a search in the space of possibilities of the association rules is done. The heuristic used is actually a data "fit" utilizing a Qui squared distribution χ^2 that supplies an association type rule, i.e. A => B (α, ß). The algorithm objective was to apply a best-first search (in this case A*) finding the most confirmed k hypotheses and also including a refinement of non-redundant operators and so detach unnecessary searches. Consequently, very efficient searches are obtained.

2 Architecture of the NESISES Hardware and Software

Figure 2 broadly shows the operation and connectivity between NESISES and the plant's automation system. Notice that the several process signals get to the SS through controllers which often are PLCs (Programmable Logical Controllers).

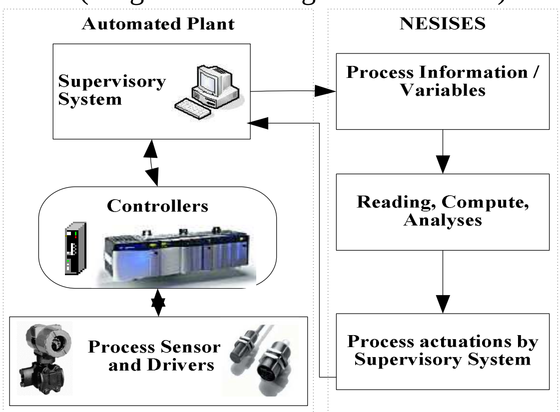

Figure 2 - Operation and connectivity between NEISISES and an automation system of the industrial plant.

The NESISES operability and architecture is depicted in Figure 3. NESISES accesses the plant's data in real time via an interface with the SS. This interface is provided by the MISS (Module of Interface of the Supervisory System) [1]. MISS delivers the necessary data to both the tags of the SISES (System of Integration of the Supervisory and Specialist Systems) and to the observatory module. The observatory module checks the SISES's outputs that can be affecting the plant and also establishes communication with the analytical module.

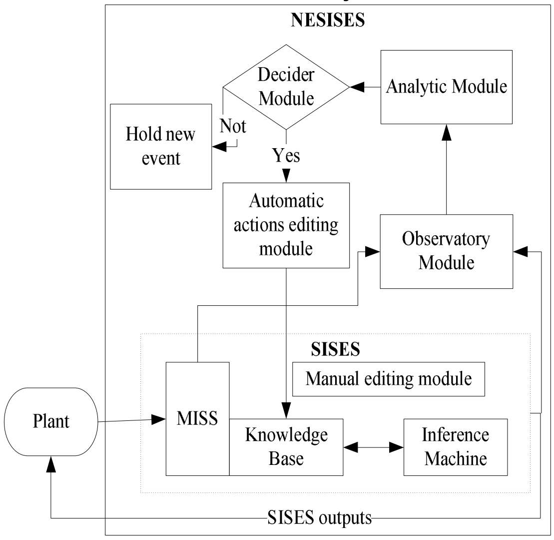

Figure 3 - Operation & connectivity and software architecture between NESISES and the plant's automation system.

In turn, the analytical module establishes communication with the decision module which has the incumbency of deciding on whether modify or not any rule parameter of the ES knowledge base.

2.1 Module interface to supervisory systems (MISS)

The MISS is the communication module of Expert System (ES) with the SS. Like most current SS operates with the Windows ® operating system, and also as the SS was chosen Rockwell Software RSView ®, the SISES was prepared in order to be compatible with these significant products of the world market. However, the scientific foundations and methodological SISES can be applied and developed for other operating environments, and SS. The development environment chosen SS SISES to operate with this research work provides several ways to communicate with Windows. However, more efficient communication is done by (DLLs) Dynamic Link Library Specific RsvApplication.dll and RsvProject.dll. This fact occurs because any other form of communication with RSView indirectly uses these same DLLs, and therefore adds more steps in communication with SISES RSView [7].
Figure 4 illustrates how the various interactions occur and RSView SISES, highlighting the functionality of MISS that is represented by the bond of communication RSView - Objects VCL and Delphi.

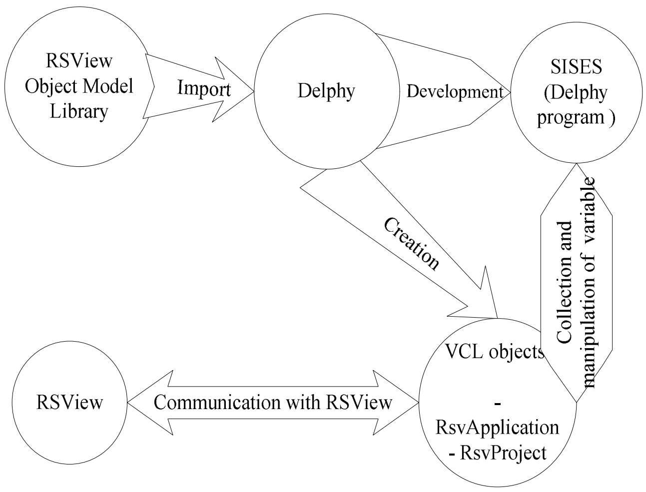

Figure 4 – Communication between the expert system and the supervisory system

2.2 Module Construction Systems Specialists (MCSE)

The MCSE is an environment where has built the ESs The Figure 5 illustrates the organization of the MCSE, which is composed of three modules:

• Knowledge base: the set of rules obtained by the knowledge engineer and stored on file SISES;

• Editor bases: is the tool that allows you to edit and change Knowledge Base;

• Machine inference: SE module that performs the inferences and conclusions about the knowledge base. In the Machine Inference SISES operates using the backward chaining algorithm (backward chaining) [10]

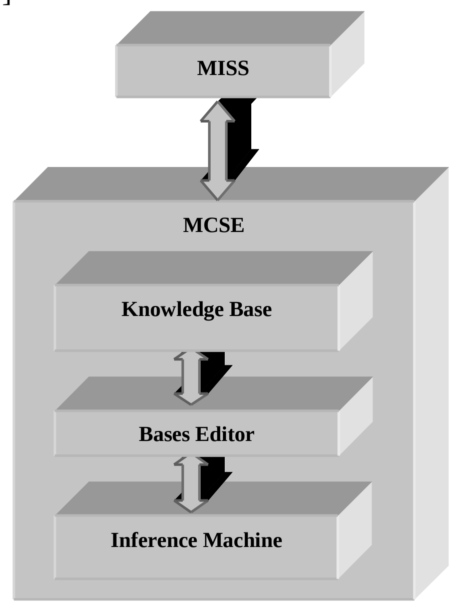

Figure 5 – Basic Architecture os ESs generated in MCSE

2.3 Module Cognitive Meta SISES

The Figure 6 illustrates the software architecture of a cognitive module meta SISES.

This module will be divided into three sub-modules:

• Observatory Module (O.M.). The observatory module must have the ability to observe the outputs of the SE and / or plan and enroll them temporarily change the indexed and boundary conditions of the plant;

• Analytical Module (A.M.). The module makes analytical comparisons and measurements between the desired values and the values obtained over time. Thus, the MA will receive the information from the MO and perform analytical accounts;

• Actuator Module Editor Decision Making (A.M.E.D.). This module, according to the results obtained from the AM decides which rule or consideration of the knowledge base should be changed. The actuator module, in turn, is composed of two modules: the decision module and the module editor's online knowledge base.

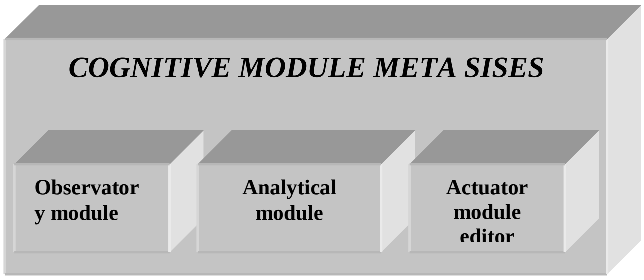

Figure 6 – Software Architecture of cognitive module meta SISES

The functionality and flow of information and actions of NESISES is illustrated in Figure 3. Note that the plant feeds the SISES online data through the MISS (Supervisory System Interface Module), the MISS in turn provides the data needed for the tags SISES and also for the module observatory. The module checks the outputs of the observatory SISES that may be affecting the plant, and also communicates with the analytical module.

The analytical module in turn communicates with the module that is making the task of deciding whether or not the change of some parameter of rule knowledge base, and the variation of performance automatically.

3 Methodology for the NESISES simulation and field tests

In order to carry out the NESISES validation tests, it was used the methodology proposed by the IEEE

standard [8] and adapted to the specifications of this research work.
This norm could be seen in the figure 7.

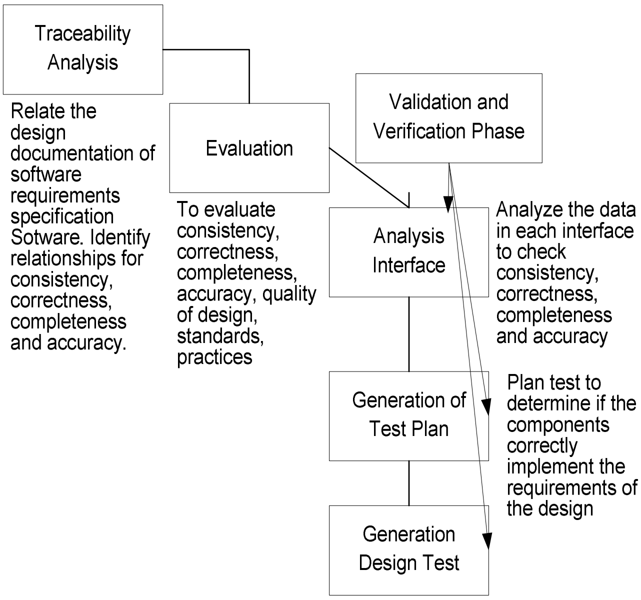

Figure 7 – Methodology for the NESISES simulation and field tests

The NESISES validation tests were divided in three stages:

- ✓ Validation tests to determine its effectiveness or degree for reaching the targets.
- ✓ Simulation tests of the automated process.
- ✓ Field tests.

a) Simulation tests to determine the degree of effectiveness

The Tertius algorithm complexity [6] does not allow, in a simple way, to theoretically determine the statistical confidence degree of the response obtained by Tertius. So, it was generated a mass of data relating Boolean variables, two by two, whose logical correct result was previously known. The objective for proposing this type of test was to create logical reference patterns through which the level of logical success obtained by NESISES can be checked.
The mass of data received by the analytical module of NESISES was created by a ladder language program specifically developed, so that the supervisory system could acquire the data and transfer them to NESISES. Approximately 10000 different data were recorded. Tables 1, 2 and 3 show the reference logical patterns embedded in the automatically generated mass of data.

Table 1– First group simulated with variables 2 by 2

Variable V1	V2
V1 = 1	30% of times V2 = 1
V1 = 0	70% of times V2 = 1

Table 2– Second group simulated with variables 2 by 2

Variable V4	V5
V4 = 1	50% of times V5 = 1
V4 = 0	50% of times V5 = 1

Table 3 – Third group simulated with variables 2 by 2

Variable V7	V8
V7 = 1	70% of times V8 = 1
V7 = 0	30% of times V8 = 1

NESISES delivered the results identified as correlation rules among variables V1 and V2, V4 and V5, V7 and V8 (Table 4).

Table 4– Rules found by NESISES

Referred to Table	Rule	Confirmation	Frequency of counter-examples	Rule found
1	1	/* 0.332298	0.0027857 */	V2(true) -> V1(true)
		/* 0.332298	0.0027857 */	V1(false) -> V2(false)
	2	/* 0.339633	0.0034568 */	V2(false) -> V1(false)
		/* 0.339633	0.0034568 */	V1(true) -> V2(true)
2	3	/* 0.009803	0.483092*/	V5(false) -> V4(true)
		/* 0.009803	0.483092*/	V4(false) -> V5(true)
	4	/* 0.009803	0.507246 */	V5(true) -> V4(false)
		/* 0.009803	0.507246 */	V4(true) -> V5(false)
3	5	/* 0.745586	0.014928 */	V8(false) -> V7(false)
		/* 0.745586	0.014928 */	V7(true) -> V8(true)
	6	/* 0.745584	0.015174 */	V8(true) -> V7(true)
		/* 0.745584	0.015174 */	V7(false) -> V8(false)

It can be seen that NESISES was able to identify that variable V2 was equal to 1 during 33% of the times when variable V1 was equal to 1 (rules 1 and 2). It was also able to identify that V4 was equal to 1 during 50% of the times when V5 was equal to 1 (rules 3 and 4). Further, NESISES was able to identify that V7 was equal to 1 during 74% of the times when V8 was equal to 1 (rules 5 and 6). The rules found and presented in Table 4 differ slightly from the rules inserted in the mass of data. The maximum difference showed to be 4%. So, it can empirically be concluded that the reliability of NESISES is in the order of 95%.

By using the same methodology, it was also tested the effectiveness of the algorithm for three different variables, the results reached showed approximately 2.3% of accuracy [3] in relation to the probability established in the simulation.

b) Simulation tests of the automated process

In this stage, an SS previously developed for a Nylon manufacturing plant [2] was used. Figure 8 shows the architecture of the simulation tests.

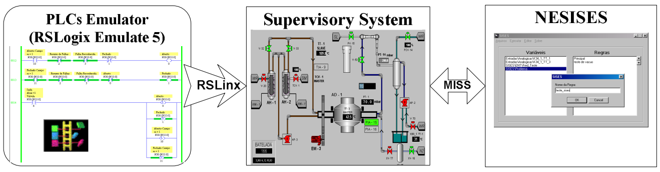

Figure 8 - Basic architecture of the nylon manufacturing process simulation tests

The simulated system that produced the field variables operates automatically generating a great mass of data. However, it has been programmed to create situations and certain logical correlations amidst a great amount of random data. NESISES was able to identify relevant patterns programmed within the mass of data. It was considered as effective learning the capability of NESISES to alter or suggest relevant rules for the operation of the automated system.

Therefore, it was previously necessary establishing that every automated system obeys to an explicit automation algorithm determined in an earnest way (or not) by the automation engineer in charge of programming the control logics (discreet or dynamic) in the PLCs, SDCDs, PACs and the Supervisory Systems. However, the automation of any industrial plant does not always obtain directly all the descriptive features and interactions among the various devices and processes of the plant, for example:

In the nylon production process studied [2], valves V-20, V-32, V-33 and TCV-1 have an associated interlocking logic, as the opening of valves V-20, V-32 characterizes the cooling process whereas the opening of valves V-33 and TCV-1 characterizes the heating process. Should both heating and cooling valves be simultaneously opened, physical damages will occur in this specific process which may lead to a stoppage of the production. The interlocking of the valves is properly programmed within the PLC that controls the process; however, there is a cause-effect relationship not programmed which is: once switched on the cooling process (valves V-20, V-32 opened) the cooling water starts heating, this in turn connects an additional compressor (along with another one already working) in order to maintain the cooling water cool. The connection of the additional compressor occurs some seconds after the valves' sequence process, although this relationship is not directly programmed in the ladder.

Table 5 shows the programmed logic patterns within the PLC emulator. Each sequence is generated by the emulator so that seven Boolean variables are simultaneously modified in each sequence.

Those seven variables are recorded by the SS, each one being tags: V-20, V-32, V-33, PV-15, V-53, AP-2, V-20.

Each of the SS tags is mirrored by a NESISES variable termed as: V1, V2, V3, V4, V5, V6, V7.

The NESISES variables are associated the SS tags which in turn are associated to the seven variables controlled by the PLC emulator. Sequence 1 is automatically altered to sequence 2 and so forth until sequence 6 is reached, which is altered back to sequence 1.

The whole cycle, repeated perpetually, was programmed during the tests and it lasts approximately 20 minutes.

Table 5– Simulated patterns for the nylon fabrication processo

SS variable	**V-20**	**V-32**	**V-33**	**PV-15**	**V-53**	**AP-2**	**V-70**
NESISES variable	*V1*	*V2*	*V3*	*V4*	*V5*	*V6*	*V7*
Sequence 1	1	1	0	1	1	1	1
Sequence 2	0	1	1	0	1	1	1
Sequence 3	1	0	1	0	0	1	1
Sequence 4	0	0	1	0	0	0	1
Sequence 5	1	1	0	1	0	0	1
Sequence 6	0	1	1	1	1	0	0

Table 6 is actually an extension of Table 5. It clearly shows the behavior pattern among variables PV-15, V-33 and V-70 present in sequences 2, 3 and 4, respectively. It can be noticed that there is a formation rule of this pattern. Such a formation rule states: whenever PV-15 is 0 then variables V-33 and V-70 will be equal to 1.

Table 6– Main simulated rule for NESISES

SS variable	**PV-15**	**V-33**	**V-70**
NESISES variable	***V4***	***V3***	***V7***
Sequence 2	0	1	1
Sequence 3	0	1	1
Sequence 4	0	1	1

Initially, the patterns shown were not simulated during the NESISES validation tests for about 594 hours as each 24 hours a new file was generated by the SS. During the 24-hour period this file had approximately 9474 lines and 7 columns. Each line

represented a sequence whereas each column represented one of the tags.
Table 7 shows the 6 more probable rules found in the database generated by the simulation. Notice the two numbers at the first column of the table; the first one represents the confirmation probability so that the rule becomes true, whereas the second one represents the frequency of counterexamples found. In other words, they represent the confirmation degree of the rule and its frequency of counterexamples. Notice also that in this paper's context, a counterexample refers to data contradicting the rule found.
Table 7 also shows that the rules are sorted in descending order with respect to the confirmation degree. By analyzing this table, it can be seen that NESISES was able to effectively deduce the explicit rule presented in Table 6. Further analyzing Table 7, it can be concluded that the first rule found represents the explicit rule of Table 6 with approximately 99% of certainty. Note that each rule presented is unique; thus; there cannot be other rule(s) contradicting it.
Hence, it can be concluded that there was an effective learning of the plant's behavioral patterns simulated by NESISES [3].

Table 7– Main rules found by NESISES

Rule	Confirmation	Frequency of counter examples	Rule found
1	/* 0.992298	0.003750 */	pv_15 = 0 :- v_33 = 1 and v_70 = 1.
	/* 0.992298	0.003750 */	v_33 = 0 or v_70 = 0 :- pv_15 = 1.
2	/* 0.878195	0.003750 */	v_33 = 0 :- pv_15 = 1 and v_70 = 1.
	/* 0.878195	0.003750 */	pv_15 = 0 or v_70 = 0 :- v_33 = 1.
3	/* 0.875885	0.003750 */	v_33 = 0 or pv_15 = 1 or v_53 = 1 :- v_32 = 1.
	/* 0.875885	0.003750 */	v_32 = 0 :- v_33 = 1 and pv_15 = 0 and v_53 = 0.
4	/* 0.875717	0.003864 */	v_32 = 0 :- pv_15 = 0 and v_53 = 0.
	/* 0.875717	0.003864 */	pv_15 = 1 or v_53 = 1 :- v_32 = 1.
5	/* 0.872376	0.006137 */	v_32 = 0 :- v_33 = 1 and v_53 = 0.
	/* 0.872376	0.006137 */	v_33 = 0 or v_53 = 1 :- v_32 = 1.
6	/* 0.691482	0.000114 */	ap_2 = 1 or v_70 = 0 :- v_53 = 1.
	/* 0.691482	0.000114 */	v_53 = 0 :- ap_2 = 0 and v_70 = 1.

c) Field tests at Santos Seaport (Brazil)

The main objective of the field tests were the validation of NESISES in real operative conditions within an industrial atmosphere and whose dynamics and inherent characteristics often extrapolate the limits that the simulations are able to check.
The automated process corresponds to a terminal with solid grain storing and delivery systems at Guarujá seaport terminal (Santos seaport, Brazil). This seaport has complex characteristics that make of it suitable for the NESISES field tests. The analyzed process has the following main parts: three warehouses, a storage patio, four granary containers for truck loading and two granary containers for railway wagons, totaling approximately 6000 automation points.
The automated process was characterized by the vast use of communication networks. A big number of input and output points are communicated with the PLCs through data communication networks, namely: ControlNet, DeviceNet, Modbus and Profibus.
Figure 9 shows both the general architecture of the process and its most important components.

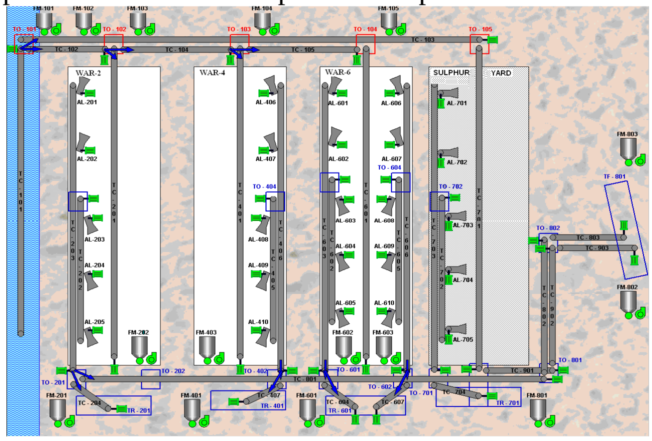

Figure 9 - General architecture showing the seaport automated process

The main section of this process is the reception of materials that is carried out through the marine terminal. In other words it considers the necessary components needed to unload the ships to subsequently store the goods in a warehouse or yard. The other main section is the delivery division from which the stored material in the warehouse or patio is delivered to the trucks or wagons. The latter section is represented in the hardware architecture. Both the reception and the delivery section have independent PLCs.
During the field tests NESISES should at least learn the simple logics of the process, for instance, should transporting conveyor TC-105 be working, the other assisting conveyors should also be working (i.e. TC-104, TC-102 and TC-101). Aside of learning the simple rules, the system should also be able to learn non trivial rules of the process.

1) Data Analysis at the Reception (Warehouse 4):

Once selected the variables pertaining to warehouse 4, it was set Table 8 which has the most probable 8 rules.

Table 8– Rules found by NESISES for the variables of warehouse-4 (Reception)

Rule	Confirmation	Frequency of counter-examples	Rule found
1	/* 0.976151	0.004942 */	TC_102_L(true) OR TC_401_L(true) -> TC_104_L(true)
	/* 0.976151	0.004942 */	TC_104_L(false) -> TC_102_L(false) AND TC_401_L(false)
2	/* 0.976029	0.000000 */	TC_105_L(true) OR TC_401_L(true) -> TC_104_L(true)
	/* 0.976029	0.000000 */	TC_104_L(false) -> TC_105_L(false) AND TC_401_L(false)
3	/* 0.971303	0.004550 */	TC_103_L(true) OR TC_104_L(false) -> TC_102_L(false) AND TC_401_L(false)
	/* 0.971303	0.004550 */	TC_102_L(true) OR TC_401_L(true) -> TC_103_L(false) AND TC_104_L(true)
4	/* 0.969299	0.010119 */	TC_102_L(true) OR TC_105_L(true) -> TC_104_L(true)
	/* 0.969299	0.010119 */	TC_104_L(false) -> TC_102_L(false) AND TC_105_L(false)
5	/* 0.969140	0.000052 */	TC_104_L(true) -> TC_102_L(true)
	/* 0.969140	0.000052 */	TC_102_L(false) -> TC_104_L(false)
6	/* 0.969128	0.015402 */	TC_104_L(false) -> TC_102_L(false)
	/* 0.969128	0.015402 */	TC_102_L(true) -> TC_104_L(true)
7	/* 0.967766	0.000000 */	TC_101_L(false) OR TC_104_L(false) -> TC_102_L(false)
	/* 0.967766	0.000000 */	TC_102_L(true) -> TC_101_L(true) AND TC_104_L(true)
8	/* 0.963946	0.015009 */	TC_102_L(true) -> TC_103_L(false) AND TC_104_L(true)
	/* 0.963946	0.015009 */	TC_103_L(true) OR TC_104_L(false) -> TC_102_L(false)

In order to make more relevant the analysis, segmented tests were carried out, firstly with the complete database and then using variables chosen from pertinent areas, as it was previously known that both warehouses operated independently one to another, but were dependent of the reception.

By analyzing the rules presented in Table 8, it can be noticed that the three more probable rules (1, 2 and 3) relate the reception transporting conveyors to warehouse 4 main conveyor (TC-401).

An analysis on rule 1 leads to establish that, if TC_104 would be operating then, there is 97% of certainty that either TC_102 or TC-401 should also be operating. The first two rules relate TC-102 and TC-104 (reception) to the main conveyor (feeder) of warehouse 4. Also, from the process analysis it can be seen that mats TC-102 and TC-104 operate jointly, with TC-102 feeding TC-104.

Those rules show to be coherent as there is a dependence relationship between them. For instance, rule 1 shows that conveyors TC-102, 401 and 104 operate together which corresponds to the reception operational facts operating together with warehouse 4.

Similar tests were performed for the possible combinations of variables of the other three warehouses, the reception section and the patio; thus, covering the whole process. NESISES was able to learn the process operation in all the tests conducted.

3.1 Analysis of traceability of tests to simulate industrial processes automated

Traceability (ISO 8402) is defined as the ability to describe the history, application, processes or events and location of a product to a particular organization, through records and identification. In simpler track is to keep the records necessary to identify and report data on the origin and destination of a product or process.

Initially, the starting point is a data structure that enables the traceability and was implemented in the software NESIS a tree structure that documents each of his decision making. For the simulation tests in itself, was executed the following procedure to ensure traceability:

a) The events that would be implemented in the simulator were defined and recorded before its implementation language ladder inside the simulator.

b) The application software developed in the creation of SSs RSView [7], this software donated to Covenant EPUSP-Rockwell Automation, is configured so as to acquire event and automatically store all the events that occurred, these events (recorded at base data) were stored in data files tipo.DBF, and these files have the following form:

b.1) Two files for each day of data acquisition, for example: "2009 05 11 0000 (tagname). DBF" and "2009 05 11 0000 (Wide). DBF. Note that the actual file name is the date on which it was generated, in addition there are two types of files, whose names are generated automatically by RSView, and Wide tagname tagname is where are the variables that will be stored and wide data that are stored.

b.2) The wide type files have internal structure in a tabular form, with the first and the second column to store date and hour / minute / second / millisecond in which the event occurred. The remaining columns are for storage of events in the sequence file specified by tagname.

c) Each time the data mining algorithm was run it was noted and saved its parameterization is that the generated file has the extension. RES.

d) All files generated by the simulation, periodically, were copied and stored on a different computer which was running the simulation to avoid the risk of any accidental envelope.

The flowchart shown in Figure 10 describes the sequence of events that allow the traceability of software testing.

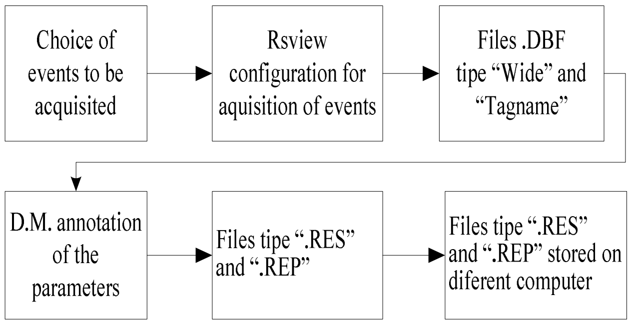

Figure 10 – flowchart of traceability of tests

The data were evaluated on each item cited in traceability, followed by a white box-type method (Pressman, 2001), where each piece of software is evaluated in the sense of having to do it properly your specification. In simulation tests, the internal routine of the simulator generated events, arquivos.DBF were generated daily, the data from these files were checked in the sense of correctness, consistency, completeness and accuracy.
At this stage they were also verified the results of step data mining where the data were analyzed in files.rar in what was a black box type approach [11], ie the focus is only on the results without going into merit of the intermediate steps.
25 files were generated between 02/04/2009 to 26/04/2009 totaling 594h and the failure rate in each of the requirements prescribed by the standard was zero.

3.2 Generation of test plan and design of simulation tests of automated industrial process

These steps in IEEE 1059 - 1993 were two distinct stages were grouped according to the needs of this research work.
This step was performed according to the flowchart shown in Figure 11, note that the process is interactive.

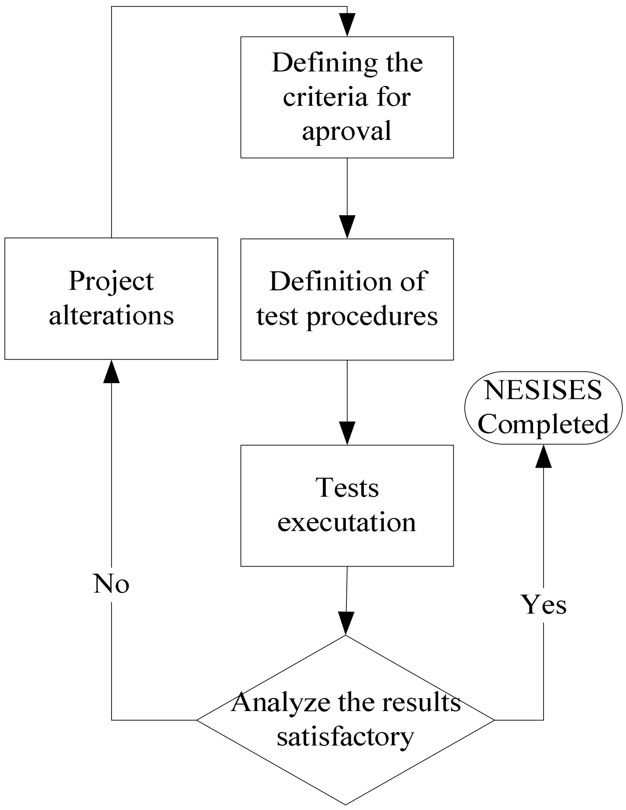

Figure 11 – Definition of the criteria for approval of field tests with NESISES

3.3 Definition of the criteria for approval of field tests with NESISES

To validate the tests, both the simulation as the field, we adopted the following criteria:
1. Assertiveness learning: the NESIS must have learning assertive.
2. Performance SS: SS should not suffer appreciable drop in performance.
3. Runtime: NESIS the time to learn something from the SS cannot be an order of magnitude higher than the predetermined time that engineers have the knowledge to reach the same conclusions.
4. Compatibility: the NESIS may not require as many features of the operating system (Windows) to impair their performance, or the functioning of NESIS not affect the normal operation of SS on their tasks.

3.4 Analysis of results according to the criteria for approval of field tests with NESISES

After you run the tests, they were analyzed according to the criteria set out in item 4.6. The results of this analysis were the following:

1. Assertiveness learning: the learning assertive NESIS have shown (see Section 4.5.3).
2. Performance SS: SS suffered no appreciable drop in performance.
3. Runtime: Time for NESIS learn something from the SS was a few minutes to 36 hours, which was not as deterrent for large volumes of information an knowledge engineer could take much longer time to reach a conclusion.
4. Compatibility: the NESIS proved fully compatible with the operating system (Windows), and the only restriction is that the step of data mining to occur on a computer other than that is running the SS

4 Conclusions

The development of this study demanded a wide research of the Artificial Intelligence (AI) field in order to verify the possibilities offered by systems with learning capacity. Several AI techniques were studied, mainly the data mining (DM) technique which demanded a more deep study. While studying the DM technique, it was verified that due to its knowledge finding characteristics in large databases, it would be extremely useful to attend the objective of this research which was the learning ability with the experience of the automation system. Thus, this experience could be stored in a database and later used to elucidate the operation of a certain system or even propose operation rules whose weights can be modified along the time.
The next step was the development of a computational tool that could interconnect an industrial automation supervisory system with an expert system. This tool should also have learning capacity, thus, it could be joined to the dynamic reality of the automation system (plant) by means of the SS acquired knowledge, so that it could teach the resulting system how to take inferences over the system and occasionally propose new rules.
Such development required a thorough study of the existing computational tools to execute DM or even start directly from the concept of any DM algorithm. It was found a software implementation of the Tertius algorithm that satisfied the needs of the DM requirements linked to association rules which was flexible enough to incorporate it to SISES. So, it was chosen to develop the software termed NESISES, which is based on the original SISES, and further improve it and provide it a learning DM module.
The NESISES software was experimented in several simulation tests which enabled its improvement. After the simulation tests, it went to a real world trial within an automated industrial plant, where it was checked its learning capacities. NESISES proved to be satisfactory in all the tests conducted.
While developing this study, some other questions appeared which require futures improvements, namely:

- ✓ Implementation of the NESISES decision module based on new artificial intelligence researches;
- ✓ Researches of new AI techniques to improve the NESISES inclusion and effectiveness. Particularly, more efficient data mining algorithms than that of Tertius;
- ✓ Research of new AI techniques to develop both methodologies and construction tools for NESISES and ES;
- ✓ Inclusion of time control within the heuristics in order to better assess a certain action of NESISES;
- ✓ Improvement of its interface with the user making the operation more user friendly;
- ✓ Study the possibility of the NESISES interconnection with other business management software's such as ERPs.

References:

[1] Pereira S.L.; Andrade A. A. . *Ambiente de Desenvolvimento de Sistema Especialista para Operar em Tempo Real com Sistema Supervisório de Automação Industrial*, Anais do Congresso CBA 2004 - XV Congresso Brasileiro de Automática Gramado RS , 2004
[2] Andrade A. A. *Emprego de sistemas especialistas em sistemas supervisórios de automação industrial* 2001, 154 f. Dissertação (Mestrado em Engenharia Elétrica) Escola Politécnica da Universidade de São Paulo, 2001.
[3] Andrade A. A. *Desenvolvimento de Sistema Especialista com Operacionalidade de Aprendizado para Operar em Tempo Real com Sistemas Industriais Automatizados* 2007, 158 f. Tese (Doutorado em Engenharia Elétrica) Escola Politécnica da Universidade de São Paulo, 2007
[4] Witten, I. H. Eibe F., *Data mining: practical machine learning tools and techniques*,. 2005 ; 2nd ed. Morgan Kaufmann series in data management systems.
[5] Pereira, S. L. ; *Andrade, Alexandre Acácio de . Linking Expert Systems to Supervisory Systems in Automations Process*. In: V INDUSCON 2002, 2002, Salvador. Anais from V INDUSCON 2002, 2002
[6] Flach Peter A., Lachiche Nicolas *Confirmation-Guided Discovery of First-Order*

Rules with Tertius, Machine Learning, vol 42, n 1/2, pp. 61--95, 2001.
[7] Rockwell Software Inc.. RSView 32 – User's Guide -USA, 2003.http://literature.rockwellautomation.com /idc/groups/public/ documents/ webassets/ external_info.hcst?dID= 47301&webSize=5671
[8] Russell S. Norvig P, *Artificial Intelligence a Modern Approach*. Prentice Hall 2ed -2004, p384.
[9] The Institute of Electrical and Electronics Engineers, Inc. IEEE Std 1059-1993 IEEE *Guide for Software Verification and Validation Plans* ; 1994 ISBN 0-7381-0410-8, SS16923
[10] Buchanan B.G. , Shortliffe E. H. Rule-Based Expert Systems. Addison-Wesley, 1985. 737p.
[11] Pressman R. S. . Software Engineering A practitioner' s approach . Mc Graw Hill 5ed - 2001, p437.
[12] Kirkby R. , Frank E. WEKA Explorer User Guide for Version 3-5-5, January 26, 2007 Available: http://opensource.cqu.edu.au/oswins/datamining/ weka/ExplorerGuide-3.5.5.pdf
[13] Fayyad U.,Piatetsky-Shapiro G., Smyth P. , From Data mining to Knowledge discovery in Databases , American Association for Artificial Intelligence, 1996.
[14] Larose, Daniel T. Discovering knowledge in data : an introduction to Data mining -2005 , John Wiley & Sons, 2003, Lawrence Erlbaum Associates, Inc.,
[15] Vapnik, Vladimir. The Nature of Statistical Learning Theory, Springer, Berlin 1995