Petri Net Inside RFID Database Integrated with RFID Indoor Positioning System for Mobile Robots Position Control

José Jean-Paul Zanlucchi de Sousa Tavares, Rodrigo Hiroshi Murofushi, Lucas Henriques Silva, and Gustavo Rezende Silva

Manufacturing Automated Planning Laboratory (MAPL), Faculdade de Engenharia Mecânica (FEMEC), Universidade Federal de Uberlândia (UFU), Uberlândia, Brazil.
jean.tavares@ufu.br, {hiroshihm, lucashenriquessilva}@gmail.com,
gustavorezendesilva@hotmail.com

Abstract. The advent of industry 4.0, internet of things and smart products increase the importance of solution focused on machine-to-machine communication. There is a need for a solution that meets these characteristics and the Petri Net integrated with RFID (PNRD) can reach them. There are a lot of papers connect the Petri Net to RFID by creating the network markings based on the reading of the tags. The PNRD uses the Petri net as the formal data structure to be stored in the tag memory, increasing Petri Net and RFID integration. RFID can also be useful as indoor positioning system or IPS. This work proposes to integrate PNRD and IPS in order to store the object process model in the tag, as well as its position obtained by the IPS can become a prerequisite of the process itself. A case study presents a mobile robot position control based on PNRD and IPS integration.

Keywords: Indoor Positioning System, Radio Frequency Identification, Petri Net Inside RFID database – PNRD, Mobile Robot Position Control

1 Introduction

The advent of industry 4.0 [1], the internet of things (IoT) [2, 3], and smart product [4] make solution focused on device-to-device communication. Continuing to use systems development methodologies focused on human-machine interaction is a challenge in relation to machine-machine solutions. There are highlighted technologies with respect to this machine-machine iteration, especially with regard to operational issues. One of them, pointed as product DNA and information key source, is RFID (Radio Frequency Identification) that allows process and product monitoring, tracking and control [5]. Since the initial RFID application with the Walmart initiative in early 2000's, much attention has been pointed out for this technology. The current RFID market shows that it has been overestimated. Nowadays the RFID implementation is below its potential without process information embedded adding automatic data and process capture.

Numerous researchers that relate RFID and Petri Net present solutions in several different areas such as quality management of a process [6], logistic process modelling [7], healthcare [8], control design of flexible cells of manufacturing system [9], monitoring and control of assembly and disassembly systems [10], material management among others [5]. These applications have a low-level connection between Petri Net and RFID, and they focused on the creation of the Petri Net marking generation based on the reading of the tags.

The PNRD [11] provides a formal data structure to tagged objects, which defines and introduces the process activity into a RFID disperse database. In this way, the tag stores its own Petri net (incident matrix and object actual state), and readers have the control vector associated with the reading activity and another conditioning sentence allowing the automatic object Petri net next state calculation, as well as updating its own state vector after the calculation. Since the tag refers to a single object, the PNRD must be a safe Petri net, and the calculation of the next state must be a unitary vector. Any result other than a unit vector identifies an inconsistency in the process of tag, and it is viewed as an exception. A software called DEMIS (Distributed Environment Manufacturing Information System) performs the next PNRD state calculation. The RETIM (Real Time Item Monitoring) software graphically displays the tag corresponding Petri net model and actual state in real time.

Another RFID feature is the object localization through an Indoor Positioning System (IPS), which determines the position of an object in an indoor environment. Recent IPS techniques based on RFID generally uses the Received Signal Strength (RSS) information to estimate the location of a tagged object [12]. IPSs can be used for different applications that can range from detection and tracking of items, production assistance, and process monitoring [13]. With the development of automation and control, different industries rely more on IPSs for their operations such as robotic guidance, industrial robots, robot cooperation, and smart factories [14].

The PNRD approach integrated with IPS increase the value of RFID technology and it provides an example, in which the RFID implementation can be seen as a positioning sensor (IPS), and as an automatic data and process capture tool. In this context the RFID technology cannot only reach intelligent product requirement, as well as, it can be useful as IPS tool inside the smart factory approach, too. This work proposes to integrate PNRD and IPS so that the data and process is stored in the tag data memory, as well as it is possible to obtain the tag position by the IPS. A case study of a mobile robot with a passive tag is presented, in which the mobile robot changes its movement direction depending on the vehicle's distance from the reader antenna. The PNRD stores the Petri net incidence matrix as well as the robot actual state. This state changes according to the calculated distance. The vehicle moves away from the reader antenna whenever the distance is less than 35 cm and it approaches the reader antenna whenever the distance is greater than 70 cm.

This article presents Petri Net and RFID review in section 2. Section 3 shows PNRD and IPS integration purpose. Section 4 describes the mobile robot implementation. Conclusions are presented in section 5, followed by acknowledgments and references.

2 Petri Net, RFID and IPS

2.1 Petri Net and RFID Integration

On one hand, PNs provide the formal foundation formal modeling concurrency and synchronization [15]. PNs have been successfully used to model, control, and analysis discrete event dynamic systems that are characterized by concurrency or parallelism, asynchrony, deadlocks, conflicts and event-driven processes [2]. On the other hand, RFID is an automatic identification and data capture (AIDC) technology with usually presented as composed by three parts, RFID tags that is connected physically to objects; RFID reader that generates an electromagnetic field to stimulates RFID tag response when it is near enough; and RFID middleware that cares about data filtering, reader management, and application connection. There are many papers integrating RFID and PN.

Chen [9] build a CPN models for different modules of FMS (Flexible Manufacturing System), to plan RFID codes rules of FMS and to develop a cell controller for RFID-based on a centralized FMS cell controller. This article provides a suggestion for mapping between color tokens of place in the CPN and the data memory of RFID tags. Petri Net model defines RFID read & write action. RFID tag data is position sensitive, and the implementation used a low-frequency reader and tag.

Sun et al. [10] proposed an assembly executive process Petri net (AEPPN) integrating Petri nets and mobile agent-based complex product assembly framework. This approach describes states of assembly as PN transitions, events in assembly executive process as PN places and mapped to RFID tags states, which are able to trigger dispatching of assembly agents and executive of assembly tasks. AEPPN is a set of places, transitions, color set, input function, output function, initial marking and time delay transition. The mapping relationship between the product set and the color set is 1:1. In each net, the amount of token with an exclusive color is 1 and only 1. RFID tag's states can be used to describe the assembly executive process state. AEPPN can acquire, delete, create or update Tag data; therefore the AEPPN is center-controlled but executed dispersedly. RFID tags can also be regarded as offline communication channels.

Lv et al. [6] developed RFID-based CPN to improve the quality of the system without sacrificing any one of the performance parameters. RFID system elements, which are connected bidirectionally with CPN, can update information promptly with real-time action. RFID tag and color token stored the information of the product in a manufacturing system, and both of them can update the status of product simultaneously. This approach combined RFID and CPN for simulation analysis. CPN simulation results can help update the RFID database, and both databases can be synchronized. This research developed the RFID-based colored Petri net to finish the accurate real-time analysis for the manufacturing system, so as, to realize automatic abnormity handling and enhance decision making. CPN token color remains the same after transition activity if this processing developed smoothly. Otherwise, color changing indicates failure modes happening in the last transition activity. Once the color of the token changes, the reader antenna sensor receives a signal that the status of the product changed. Then the reader rewrites the stored information and sends the new data to host application. Host application feedbacks a corresponding process activity on the colored changed token, for example, the failure part needs rework by reentrant.

Zhang et al. [2] presented a real-time production performance analysis and exception state diagnosis model (PAEDM). By combining RFID, hierarchical-timed-colored Petri Nets (HTCPN) with decision tree algorithm, this paper proposes a real-time production performance analysis and exception diagnosis model. The proposed architecture relies on three modules. The first one is IoT-enabled shop-floor module that is a bridge for information communication between physical manufacturing systems and the process. The second module deals with dynamic behavior model of the manufacturing system and data capture processing. The third module corresponds to decision tree-based exception and cause diagnosis. It presented a case scenario from a collaborative company using high-frequency RFID tags and readers. There was a need for integration with CAD/CAM/CAPP systems to perform the presented case.

Guo et al. [16] proposed a timed colored Petri net simulation-based self-adaptive collaboration method for Internet of Things-enabled production-logistics systems. The method combines the schedule of token sequences in the timed colored Petri net with real-time status of key production and logistics equipment. The proposed framework is composed of three layers, namely physical layer, cyber layer and the application one, where a Timed Colored Petri Net (TCPN) model is developed to depict and control the behavior of key equipment by adjusting the schedule of token sequences. In the simulation, a personal computer, fifteen antennas, four RFID readers, and nine RFID tags were used. The RFID tags were attached to different manufacturing objects, such as machines, AGVs, and WIP. The TCPN model started running at the same time when the production and logistics were executed according to the planned time. Firstly, real-time status information of machines, AGVs, and work in process (WIP) was transmitted to the PC through RFID reader ports. Secondly, based on the collected information, the objective functions were implemented and the results were stored in Standard ML (SML) files. Thirdly, every time the cycle of the TCPN model started, the information in the SML files was updated. By loading SML files, the status of colored tokens was tuned accordingly. Comparing TCPN based self-adaptive collaboration method with an event-driven method, total waiting time reduced 28,8%, makespan decreased 16,5%, and total electricity consumption down 4%.

Jiang et al. [17] presented a Petri-Net model-driven methodology for the development, validation, and operation of a RFID-enabled decentralized FMS. A methodology to define active and passive elements was presented in order to each active resource is equipped with a reader and each passive resource is banded with a tag. Active resources acquire the status of passive ones by analyzing the PN models; they decide the next steps by combining their own status and behavior logic.The Color Petri Net model presented two distinguished places, it means, state-place for real-time status of the equipment, and port-place for an interface of workpiece and storage equipment.

It can be noticed that most of these applications have a low-level connection between Petri Net and RFID usually generated by the color token relationship with RFID tag reading, it means they focused on the creation of the Petri Net marking identification based on the reading of the tags in a centralized PN control model. In the case of Jiang et al. [17], RFID is the product database itself for operational level management; however, it is not clear how strong this connection is.

2.2 PNRD

According to [11], the PNRD - Petri Net Inside RFID Database - is a RFID data structure based on the elementary Petri Net formalism or Low-Level Petri Net (LLPN), and it can be described as a five-tuple (P, T, A, w, M_0), where P is the finite set of places, $P \neq$, T is the finite set of transitions, $T \neq$. $A \subseteq (P \times T) \bigcup (T \times P) \to \mathbb{N}$ is the set of arcs from places to transitions and from transitions to places, $w : A \to \{1\}$ is the unit weight function on the arcs, and $M_0 : P \to \{0, 1\}$ is the PN initial marking. As PNRD has a 1:1 relation with each tag, and PNRD must be a safe Petri net with only one weight function, and a unitary marking. In this approach each tag stores its own incidence matrix and state vector of a Petri net referring to the process part to which the tagged object in question participates; and each reader stores the corresponding control vector list and the triggering conditions. The PNRD operation is based on the capture of the tagId followed by the A^T or incidence matrix, and the tag state vector (M_k). The software responsible for calculating the next state finds the corresponding u_k (control vector) related to the conditioning set composed by tagId, tag state, antennaId, readerId, and other optional additional data, such as time interval, the distance among other. The calculation of the next tag state $M_k + 1$ follows (1).

$$M_{k+1} = M_k + A^T * u_k \quad k = 1 \dots n \ . \tag{1}$$

The next tag state result must be evaluated. If the result is a unitary vector, this means that the Petri net remains elementary and safe, which is consistent with the fact that each tag has a 1: 1 relation with Petri Nets. This result is supposed in agreement with the expected process flow, allowing the record of the M_{k+1} in the tag memory as new tag state. Otherwise, the Petri net is no longer safe, indicating an abnormality in the expected follow-up of the process, which can generate a real-time warning signal. It is able to monitor the process of each tag individually. Even flexible processes can be stored, giving to the tagged object the ability to follow different paths as long as properly planned and modeled previously. One of the possible problems during the execution is the appearance of conflicts. Conflicts occur when the same antenna/ reader is associated with more than one transition relative to the same tagId, tag state, and additional data. A decision algorithm can be applied to choose what transition should be triggered in order to solve the conflict and more details was presented in [11]. Hence, PNRD is based on a previously modeled system, and it is able to check whether the desirable model is followed or not.

It is possible to point out that in the PNRD approach there is a strong connection between RFID and Petri Net, which reduces the need for queries in external databases. In the other hand, it is evident that the PNRD approach uses an additional step of capturing data related to the incident matrix and the control vector. In this direction, the process of the tagged object must be predefined in advance. After this process modeling, an operational management system must attribute a specific PN process to each tagged object.

To explain PNRD didactically, Fig. 1 shows an aerospace product selection example. In this process, the product must be tested twice. The first selection defines whether the product can be sent to industrial Companies by $Test1Acceptance$ transition or not, it means, it must be sold as a replacement item by $Test1Rejection$ transition. The industrial supply item must be selected as aerospace supply by $Test2Rejection$ transition or military one by $Test2Acceptance$ transition. Since the PNRD is the Petri Net of the tagged object point of view and this object is one and only one, physically, it is not possible to split an object identification as an AND-split transition. In this example, there is no AND-split transition, so, the original Petri net model is identical to the PNRD model. In the PNRD model, there is a need for identifying the reader antenna associated with each transition. In this case, transitions $Test1Acceptance$ and $Test2Rejection$ are connected with Reader1 – Antennas 1/2, and other transitions have a distinguish reader antenna, for instance, $Test1Rejection$ is connected with Reader1 – Antenna 3, and $Test2Acceptance$ is connected with Reader1 – Antenna4. An initial marking is included in $AerospaceProduct$ state, as presented in Fig. 2.

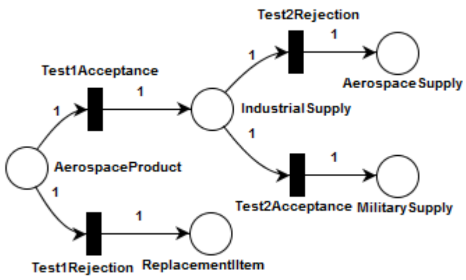

Fig. 1. The Petri Net Model of the Aerospace Product Test Process

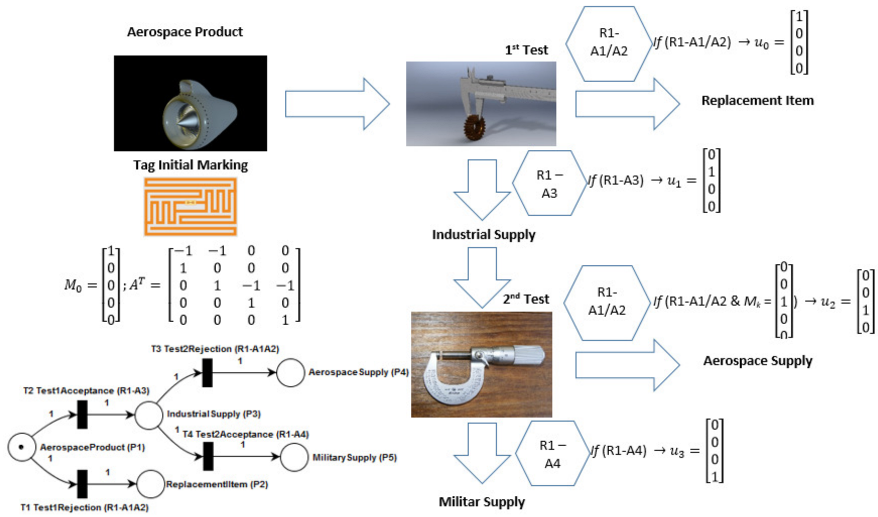

Fig. 2. The Aerospace Product Test Process Schema With PNRD Model, Initial Marking, Incident Matrix, Readers/Antennas and Control Vector Conditions

Figure 2 also presents the schema of the physical layout with one reader1 and four antennas installation. Each tagged Aerospace Product stores tagId, PNRD incident matrix, and tag state (represented by the tag initial marking M_0). Each control vector has specifics conditions as if reader1 − antennas1/2 captures a tag data, the control vector must be $[1, 0, 0, 0]^T$ ($Test1Rejection$ activity), unless tag state is equal to $[0, 0, 1, 0, 0]^T$ where the control vector changes to $[0, 0, 1, 0]^T$ ($Test2Rejection$ activity). This distributed data allows the automatic next state calculus during tag data capture by a specific reader. As an example, if the reader1 − antenna 1/2 captures one tag in the state M_0 ($AerospaceProductstate$), then the next state calculation result is the $ReplacementItem$ state as presented in (2).

$$M_1 = M_0 + A^T * u_1 = (0, 1, 0, 0, 0)^T \tag{2}$$

As the PNRD model must be a one-safe Petri Net, the tagged Aerospace Product cannot be in more than one state. This feature allows an automatic exception state detection. For instance, if an $AerospaceProduct$ is in the $ReplacementItem$ state and reader1 − antenna 3 is triggered, the result of the next state calculation identifies and absent of token in the $AerospaceProduct$ state, one token in the $IndustrialSupply$ state and a remaining token in the $ReplacementItem$ state (3).

$$M_2 = M_1 + A^T * u_1 = (-1, 1, 1, 0, 0)^T \tag{3}$$

DEMIS – Distributed Environment Manufacturing Information System DEMIS or Distributed Environment Manufacturing Information System is an implementation of PNRD in software based on Java technology. The DEMIS has two modules: the PNRD core and the interfaces one. The interface module is responsible for communicating with the various devices, such as RFID readers, PLCs - Programmable Logic Controller, and the interpretation of the data sent and received. The DEMIS Core has the next state calculation algorithm or PNRD Engine; an Inference Machine with a knowledge base to solve conflicts; and configuration files (ips, port, the number of readers' antennas, control vector list and tags state pre and post conditions). Figure 3 presents, in a simplified way, the DEMIS architecture.

ReTIM – Real Time Item Monitoring Since each tag stores its own process data and process, it is possible to visualize the object operational state and process, graphically. The ReTIM software integrates the concept of process remote monitoring related with individual tagged object. After DEMIS calculates the next state, it sends a message to the ReTIM with reader/antennaId, tagId, incident matrix and tag state. Then ReTIM can graphically display the Petri Net of the object and its respective state [18]. Figure 4 shows an example of a tag data capture from DEMIS integrated with ReTIM to graphically visualize a Petri Net with four places, four transitions, and the marking in the P4 state.

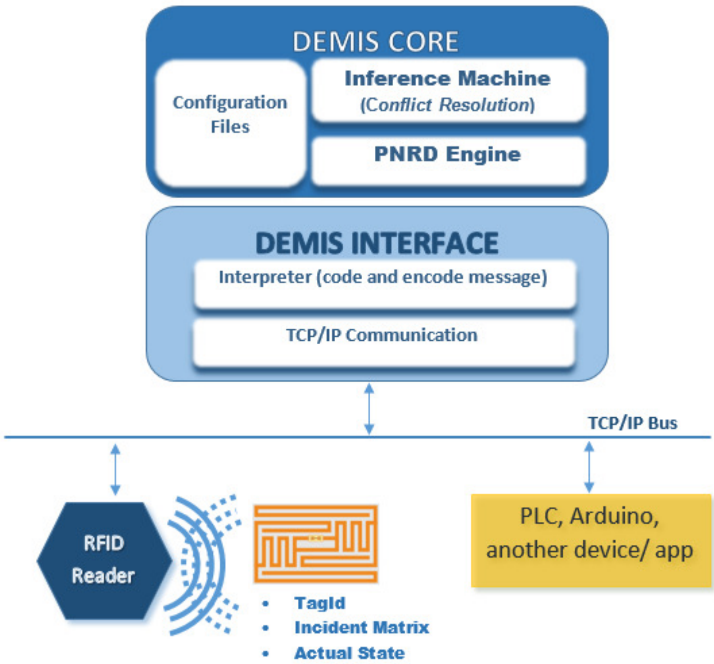

Fig. 3. DEMIS Architecture adopted from [11]

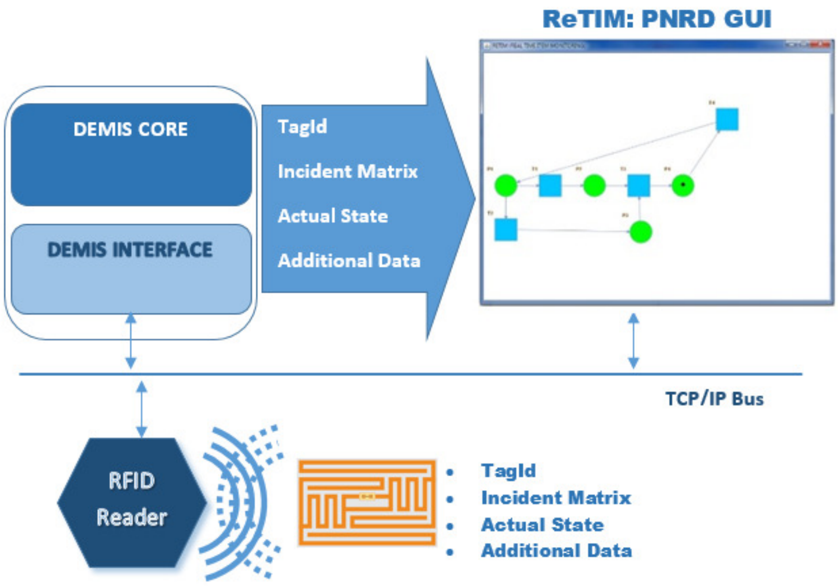

Fig. 4. Example of ReTIM interface with DEMIS

2.3 IPS (Indoor Positioning System)

There are two types of RFID indoor positioning system, i.e., reader localization, and tag localization depending on what, between reader and tag, needs to be localized. In the reader localization, the accuracy of the RFID system is highly depending on the density of tag deployment and the maximal reading ranges. In a probable localization context, a large number of RFID tags, which contain its own location information, can be deployed to cover an entire indoor environment. The disadvantage of this approach is the large number of RFID tags, which need to be applied, and prerecorded in advance with location information. Obviously, this method is more expensive and the cost increases with the increase in the number of used RFID readers [19].

Related with IPS algorithms, there are four types, it means, Time of Arrival (TOA), Time Difference of Arrival (TDOA), Angle of Arrival (AOA) and Received Signal Strength (RSS). RSS estimates the distance of an unknown node to reference node from some sets of measuring units using the attenuation of emitted signal strength. This method can only be possible with radio signals. RSS localization method could be using either a propagation model algorithm or a fingerprinting algorithm. Propagation Model Algorithm (PMA) establishes the model between RSS and the distance. Generally, the higher of the RSS values the closer from the Access Point (AP) the tagged object is. Attenuation of the received signal strength is inversely proportional to the distance from AP in the outdoor. In contrast, it is complex in the indoor environment because of the existence of obstacles (furniture, equipment windows, doors etc.) may cause multipath propagation, such as reflection, refraction, and diffraction [14].

Indoor localization of autonomous vehicles (or mobile robots) is a challenging and lively subject because of the complexity of the indoor scenarios, the diversity of technologies involved, and the commercial and industrial interests [20] and one possible way of estimating a robot position makes use of the RFID. This subject has received a considerable attention in the last few years and in many cases, the localization system is realized by installing a reader on the robot and by providing the environment with a certain number of tags placed in known position [13, 21–23]. Applying formal methods to model robot tasks like Petri net provides a systematic approach to modeling, analysis, and design, scaling up to realistic applications, and enabling analysis of formal properties, as well as design from specifications [24].

There are also many papers about IPS based on RFID technology and the main purpose of this subsection is to present some works related to mobile robot localization.

DiGiampaolo and Martinelli [22, 23] propose a global localization system combining odometry data with RFID readings. The RFID tags are placed on the ceiling of the environment and can be detected by a mobile robot unit traveling below them. The detection of the tags is the only information used in the proposed approach (no distance or bearing to the tag is considered available), but only a small number (about one each square meter or less) of tags are used. This is possible using a suitable tag's antenna in a ultrahigh frequency band, expressly designed to obtain regular and stable RFID detection regions. A satisfactory performance is achieved, with an average position error of about 0.1 m.

Martinelli [20] proposes a global positioning system based on the received signal strength and the phase shift of UHF-RFID signals coming from a set of passive tags deployed on the ceiling of the environment together with odometry provides the position of a mobile robot. A multi-hypothesis extended and unscented Kalman filter is proposed to localize the robot and to simultaneously improve the initial estimate on the tag coordinates.

Murofushi et al. [25] and Murofushi and Tavares [26] designed a real time unidimensional indoor positioning using passive tags based on the RSS of the backscattered signal. The IPS design was based in the system calibration, and the distance estimation phase. The IPS accuracy achieved is 4.7 cm for a mobile robot moving at constant velocity.

Errington et al. [27] investigate the concept of using an array of RFID tags placed at fixed known positions to provide the initial position to the Simultaneous Localization and Mapping (SLAM) algorithm. The mobile vehicle has a RFID tag reader coupled to it and the antenna is used to detect the tags. The application of interest here involves determining the initial position of a stationary vehicle in an underground mine using an array of RFID tags placed at known positions to provide the initial position of the vehicle. The results suggest that

RFID-based positioning, using the Least Square approach, has the potential to provide relatively accurate and low-cost initial position estimation.

3 PNRD and IPS Integration Proposal

The proposal of this article relies on increasing RFID and Petri Net operational potential application based on IoT approach, it means, to use RFID system as a process aware based on PNRD approach integrated with an IPS.

In this sense, RFID IPS sensor must become a pre or post condition enabling or inhibiting one or more PNRD transitions. This arrangement changes PNRD from ordinary to a high-level PN. It is necessary to complement the PNRD formalism with the pseudo-box concept, which denotes an observable condition that is not controlled by the modeled PN; and disabling arc [28].

As presented in [28], pseudo-box is a hierarchical resource embedded in Petri Net, that is, elements that only propagate information and preserve the marking in its original place. It could be an enabling gate, that is, one that sends information if is marked, or an inhibitor gate, if propagates information when is not marked. Thus, these gates must always have an original place in Petri Net graph, a special place called pseudo-box.

Pseudo-boxes denotes an observable condition that is not controlled by the modeled system. During the course of the modeling, pseudo-boxes could also stand for control information external to the hierarchical components and could be collapsed when components are put together. Thus, pseudo-boxes must be considered in the structure of the net but should not affect its properties or the rank of the incidence matrix.

This high-level Petri Net is a five-tuple (L, T, A, w, M_0), where L is the finite set of places and pseudo-box, $L = B \bigcup P$, $L \neq$, T is the finite set of transitions; $T \neq$, $A \subseteq (L \times T) \bigcup (L \times P) \rightarrow \mathbb{N}$ is the set of arcs from places or pseudo-box to transitions and from transitions to places or pseudo-box; $w : A \rightarrow \{1\}$ is the unit weight function on the arcs; and $M_0 : P \rightarrow \{0, 1\}$ is the PN initial marking.

Hence, PNRD extended to distance is the original PNRD with pseudo-box and disabling arc. This new information is calculated and storage at the reader and Fig. 5 shows its correspondent sequence diagram. If the precondition response identifies a required distance range, this generates a new operation in order to determine tag distance and check if it is inside transition disabling the rule. If so, next state calculation is realized. In this case, the internal application runs PNRD and IPS algorithm.

Next section presents the case study of a mobile robot position control using PNRD extended to distance approach.

4 Case Study: Implementation of PNRD and IPS in Mobile Robot Position Control

The case study presented in this work is about a mobile robot controlled by the PNRD. The vehicle moves forward or backward from 35 to 70 cm in an oscillating

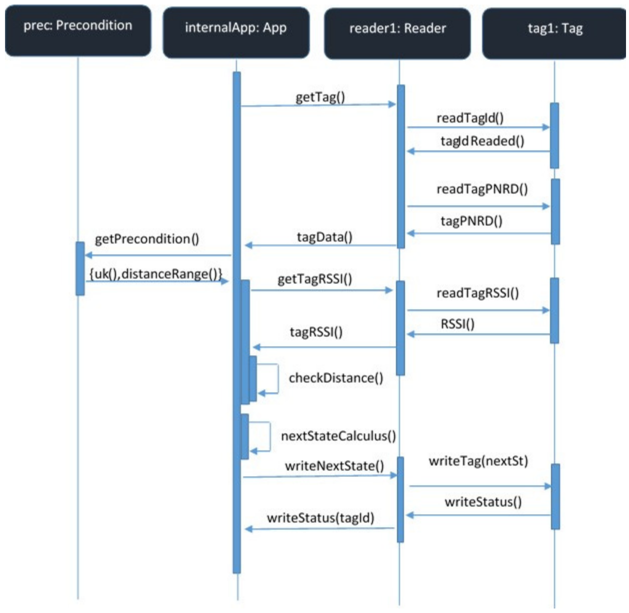

Fig. 5. Sequence diagram of proposed PNRD extended to distance

cycle. Figure 6 shows a scheme of the mobile robot position control. The mobile robot uses two stepper motors, it has a short dipole tag attached, and an Arduino Uno R3 controls it. The reader is a reader M6e micro with a monostatic antenna. DEMIS and IPS were implemented in java programming language in an Intel core I5 750, 2.66GHz, 8 GB RAM DDR3 in Windows 10 Pro 64 bit platform. The algorithm takes about 200ms for each position estimation (data processing operation) and the vehicle was programmed to move at a constant velocity of 100 mm/s. Therefore, the mobile robot positioning error is lower than the IPS accuracy of about 57 mm. Therefore, the identification of the position of the robot by the IPS can be identified as in real time.

4.1 Mobile robot position control Petri net model

Figure 7 presents mobile robot Petri net model with places (white circle) and pseudo-box (gray circles), transitions, arcs and disabling arcs. There are four places, $InitialMarking$ ($P1$), $MobileRobotStandBy$ ($P2$), $ForwardMovement$ ($P3$) and $BackwardMovement$ ($P4$) and five pseudo-box $DistanceMeasurement$ ($Ps1$), $FwdMovMessage$ ($Ps2$), $BckMovMessage$ ($Ps3$), $StopFwdMovMessage$ ($Ps4$) and $StopBckMovMessage$ ($Ps5$). Places define mobile robot dynamic behavior and pseudo-box deals with external sensing ($Ps1$) and command messages ($Ps2$ to $Ps5$). The estimated distance "d" is calculated, and, depending on "d" value, a transition may be triggered. For instance, when the vehicle is in the $P2$ state, $Ps1$ receives "d" from IPS; and, if "d" is less or equal to 50cm, $T2$ triggers

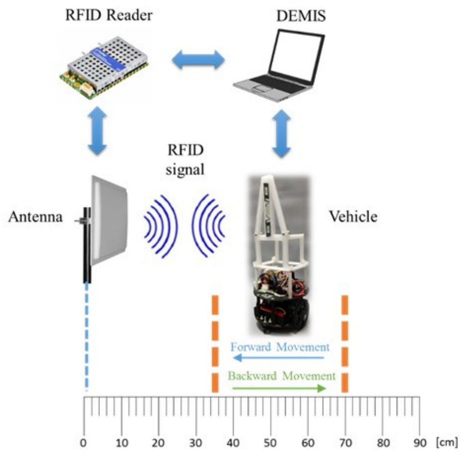

Fig. 6. Scheme of the mobile robot position control

changing the vehicle state from $P2$ to $P3$ and $Ps2$ sends a message related to the triggered transition. Each place, pseudo-box, and transition have a corresponding with RFID system. Table 1 describes places and pseudo-box, and Table 2, transition. In the mobile robot example, pseudo-boxes are applied as the robot distance control as IPS interface. It can be noticed that the configuration file stores distance setup to be reached by the robot and represents robot control pre-conditions to change its own direction.

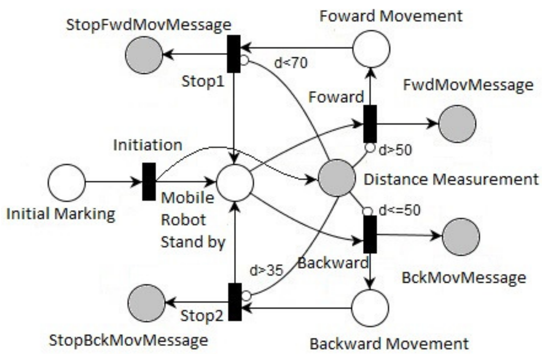

Fig. 7. Mobile robot Petri net model

Table 1. Description of places and pseudo-box

Meaning	Corresponding RFID System
Initial marking	Tag(incid. matrix and state vector)
Mobile robot stand by	Tag (incid. matrix and state vector)
Forward movement	Tag (incid. matrix and state vector)
Backward movement	Tag (incid. matrix and state vector)
Distance measurement	Reader (DEMIS Position algorithm)
Forward movement message	Reader (DEMIS Core Conf. File)
Backward movement message	Reader (DEMIS Core Conf. File)
Stop forward movement message	Reader (DEMIS Core Conf. File)
Stop backward movement message	Reader (DEMIS Core Conf. File)

Table 2. Description of transitions

#	Meaning	Corresponding RFID System
T1	Position control initiation	Reader (DEMIS Core Conf. File)
T2	Distance less or equal than 50	Reader (DEMIS Core Conf. File)
T3	Distance more than 50	Reader (DEMIS Core Conf. File)
T4	Distance more or equal 70	Reader (DEMIS Core Conf. File)
T5	Distance less or equal 35	Reader (DEMIS Core Conf. File)

4.2 IPS deployment

The IPS must be calibrated first so that an expression of the distance in function of RSS is estimated. Then the estimated distance can be calculated. The calibration is made by collecting 500 samples of RSS values every 3 cm in the range from 30 to 90 cm, measured from the antenna. Since the antenna varies the frequency of the emitted signal, in a range of 50 distinct frequencies between 902 and 928 MHz, it is also important to associate the signal frequency with the respective RSS value and distance.

After data collection, a second-degree calibration curve is fitted for each individual frequency, associating a RSS value with a distance in centimeters. Then, utilizing those curves, a mean distance is computed for each position (and the confidence interval for the each location was estimated for a significance of 5%). Equation (4) shows the calibration expression for the distance in function of the RSS, where c_0, c_1, and c_2 are constantly obtained from calibration process, and they depend on signal frequency.

$$Distance = \frac{-c_1 - \sqrt{{c_1}^2 - 4 * c_2 * (c_o - rss)}}{2 * c_2} \tag{4}$$

Figure 8 shows a graphic of the experimental data and adjusted curve. The real distance curve is, evidently, non-linear. This is due to electromagnetic waves present in the environment which may cause an interference in the RSS values, such as, reflection; another reason is the low resolution of the RSS reader (1 dBm) [27].

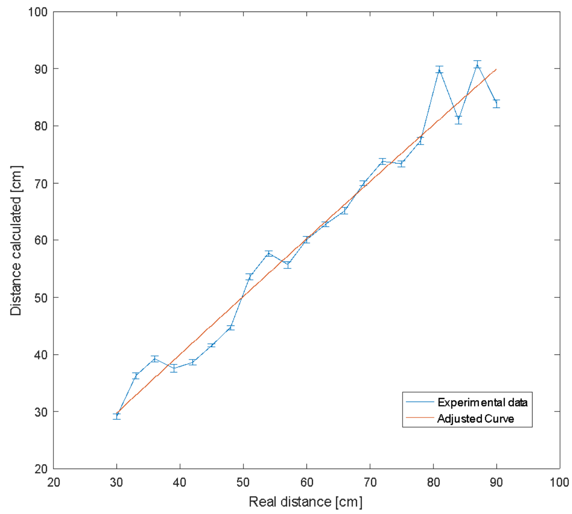

Fig. 8. Experimental data and adjusted curve graph

4.3 PNRD Extended to Distance Implementation

This case study requires a specific configuration file, which identifies state transition depending on mobile robot state and antenna distance. Figure 9 presents configuration file code, and, it can be notice the transition 1 has no distance preconditioning, only state 1, the initial marking. As the mobile robot is connected physically by USB port, Java communication is "serial" type. The transition 2 has a distance pre-condition, it means, this transition fires only if the distance is more than 50. Transitions 3 to 5 are similar with a different distance requirement. As the distance range is unique to each transition, there's no need of state identification for transition 2 to 5 in order to avoid conflict. Each "distance" label requires the petri Net pseudobox Ps1 external sensing. The "outputType" label starts the petri Net pseudobox Ps2 to Ps5 external communication.

The Fig. 10 shows DEMIS process log example when mobile vehicle is in state P2 with distance of 44 cm. As this distance is less than 50 cm, the transition T2 fires, it sends the message "2" to the mobile robot, changing mobile robot state to P3.

Figure 11 (a) and (b) presents RETIM graphical example of this state changing. Depending on the mobile robot state, it changes its own direction (move for-

```
[{"state": 1,"transition": 1,"outputType": "serial",},
{{"distanceMin": 50,"transition": 2,"outputType":
"serial",},
{"distanceMax": 50,"transition": 3,"outputType":
"serial",},
{"distanceMax": 35,"transition": 4,"outputType":
"serial",},
{"distanceMin": 70,"transition": 5,"outputType":
"serial",}]
```


Fig. 9. Mobile robot position control configuration file

ward or backward) according to the flowchart presented in Fig. 12. For instance, if the transition T2 triggers, the mobile robot moves forward. This logic control is implemented in the mobile robot Arduino. Figure 13 shows an implementation site photo.

5 Conclusion

Increase Petri net and shop floor integration, higher the opportunity to develop a complete methodology of discrete event system design, model checking, and deployment.

IoT changes to control system paradigm from centralized to distribute one. This new paradigm requires new implementation approaches in order to deal with micro processed operational level. In this direction, RFID is a cornerstone technology of IoT [2]. In a distributed environment, there is a need for a distributed modeling technique. Petri nets are commonly viewed as modeling and controlling tool; but, in control field, Petri net is usually applied as system dynamic model. PNRD is a method that fits distributed modeling technique requirement, splitting Petri net structure and storing it in RFID components readers and tags. Several examples show RFID and Petri net integration, but most of them rely on a centralized model in the control level, far away from the sensor itself [2, 6, 9, 10, 16]. Jiang et al [17] stores Petri net model in the sensor level of RFID tag, although this model is stored completely, it remains centralized.

The RFID technology is more than a distributed database, and this paper presents an IPS based on RFID RSS signal. Most of IPS research fixes signal frequency [20, 22, 23, 27]. In regular RFID application, readers have a frequency range to generate more than one communication channel.

This article presents an extended PNRD integrated with an IPS application in mobile robot positioning control. In this context, there is a need to deal with external communication between the reader and the mobile vehicle; transition triggering preconditions; and a frequency range of RFID signal. To reach these requirements, the IPS deals with the whole reader frequency range; and the PNRD approach requires a high-level Petri net structure with pseudo-box and disabling arcs. To implement this model in RFID system, pseudo-box and disabling arcs are stored in the reader configuration file; and the IPS algorithm is

```
Listening Port: 7070
---------------------Reading
tagId-----------------------------------------Reading
tagData--------------------Processing tag data:
Number of States: 4
Number of Transition: 5
Incident Matrix:
-1  0  0  0  0
 1 -1 -1  1  1
 0  1  0 -1  0
 0  0  1  0 -1
Actual State: P2
------------------Configuration File ------------------
Reading configuration file pre-condition…
------------------Getting Preconditions----------------
Distance requesting:
RSSI: -36
Calculated distance: 44.24500617599104 [cm]
----------------Control Vector definition-----------
Transition T2 pre-condition satisfied!
-------------------Next state calculation--------------
Transition T2 fired!
Sending message: 2
New state: P3
Writing new state in tag…
Writing status: Ok
```

Fig. 10. DEMIS process logging next state calculation example from state P2 to state P3

embedded in PNRD engine. An example is presented where the PNRD is the control logic algorithm, and the integrated IPS is the positioning measurement of the vehicle at the operational level. This paper demonstrates the feasibility of integration between Petri nets and RFID technology through PNRD and IPS. This approach increases RFID value generation, creating opportunities for new improvement in several areas.

The PNRD approach must be extended to time with time Petri nets [28], continuous process with continuous Petri nets [29], process mining [30], color Petri nets [15], and the Petri net next state writing process may have technical issues, which means that an internal communication procedure must deal with cases of recording problems. RFID hardware must be improved to find a more reliable and accurate positioning results, a new position algorithm using RSS and phase signal, treating the full range of radio frequency, is demanded.

6 Acknowledgement

CAPES, CNPQ, FAPEMIG and UFU supported this research.

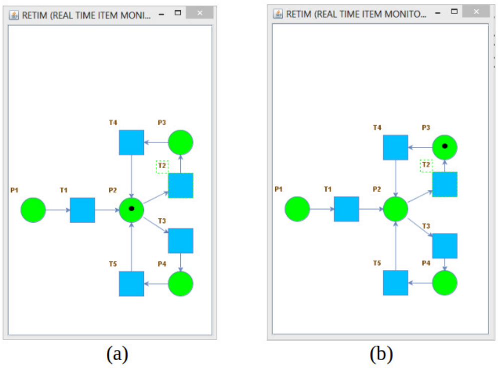

Fig. 11. RETIM graphical example mobile robot state changing from P2 (a) to P3 (b)

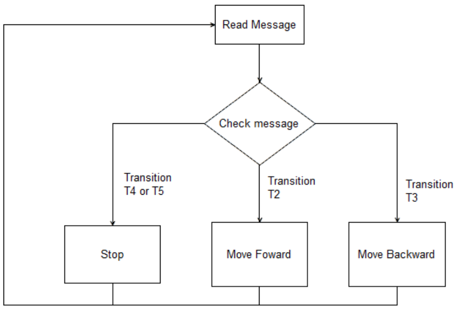

Fig. 12. Mobile robot Arduino code flowchart

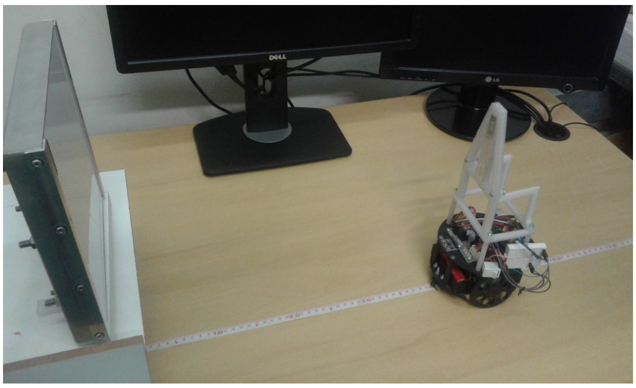

Fig. 13. Implemented mobile robot in front of the reader antenna

References

1. GTAI – German and Trade Investment. (2016, Nov. 6): Industrie 4.0 – Smart Manufacturing for the Future, Online. Available:https://www.gtai.de/GTAI/Content/EN/Invest/_SharedDocs/Downloads/GTAI/Brochures/Industries/industrie4.0-smart-manufacturing-for-the-future-enpdf?v=8
2. Zhang, Y., Wang, W., Wu, N.: IoT-Enabled Real-Time Production Performance Analysys in Exception Diagnosis Model. IEEE Transaction on Automation Science and Engineering, Vol 13, no 3, pp. 1318-1332, July 2016.
3. Borgia, E.: The Internet of Things vision: Key features, applications and open issues. Computer Communication, 54, pp. 1-31, 2014. Available at http://www.sciencedirect.com/science/article/pii/S0140366414003168.
4. McFarlane, D., Sarma,, S., Chirn, J. L., Wong, C. Y., Ashton, K.: The intelligent product in manufacturing control. In Proc. 15th IFAC World Congress, 2002.
5. Ngai, E.W.T., Moon, K.K.L., Riggins, F.J., Yi, C.Y.: RFID research: An academic literature review (1995-2005) and future research directions. International Journal Production Economics, 112 (2008), pp. 510-520.
6. Yaquiong Lv, Lee, C.K.M., Chan, H.K., Ip: RFID-based colored Petri net applied for quality monitoring in a manufacturing system. Int J Adv Manuf Technol 60, pp 225–236, 2012.
7. Oberwies, A., Zhang, H.: Modelling and facilitating RFID-based collaborative logistics processes. International Journal of Organizational Design and Engineering, Vol. 2, No. 1, 2012.
8. Fernandez-Llatas, C., Lizondo, A., Monton, E., Benidi, J.M., Traver, V.: Process-Mining Methodology forHealth Process TrackingUsingReal-Time IndoorLocation Systems. Sensors, 15, pp. 29821-29840, 2015.
9. Chen, K.: Cell controller design for RFID based flexible manufacturing systems. International Journal of Computer Integrated Manufacturing, Vol. 25, No. 1, January 2012, pp. 35-50, Taylor & Francis.
10. Sun, H., Chang, Z., Mo, R.: Monitoring and controlling the complex product assembly executive process via mobile agent and RFID tags. Assembly Automation, Vol 29, Issue 3, pp. 263-271, Emerald Insight, 2009.
11. Tavares, J.J.P.Z.S., Saraiva, T.A., Elementary Petri Nets inside RFID Database (PNRD). Journal International Journal of Production Research. Vol. 48, 2010 - Issue 9: RFID Technology and Applications in Production and Supply Chain Management.
12. Zhou, J., Zhang, H. Mo, L.: Two-dimension localization of passive RFID tags using AOA estimation. In Proc. IEEE Instrumentation and Measurement Technology Conf. (I2MTC), pp. 1-5, 2011.
13. Zhou, J., Shi, J., "RFID localization algorithms and applications, a review". Journal of Intelligent Manufacturing, pages 1–13, 2008.
14. Al-Ammar, M. A., Alhadhrami, S., Al-Salman, A., Alarifi, A., Al-Khalifa, H. S., Alnafessah, A., Alsaleh, M.: Comparative survey of indoor positioning technologies, techniques, and algorithms. In Proc. International Conference on Cyberworlds (CW), pp. 245-252, 2014.
15. Jensen, K., Kristensen, L.M.: "Colored Petri Nets: A graphical language for formal modeling and validation of concurrent systems". Communication of the ACM, Vol. 58, no. 6, June 2015.

16. 16. Guo, Z., Zhang, Y., Zhao, X., Song, Z. "A Timed Colored Petri Net Simulation-Based Self-Adaptive Collaboration Method for Production-Logistics Systems". Applied Sciences, 2017, 7, 235. Doi:10.3390/app7030235.
17. Jiang, Z., E, M., Liu, Y., Liu, J., Li, Y.: "Study of manufacturing resource perception and process control of a radio-frequency-identification-enabled decentralized flexible manufacturing system". Advances in Mechanical Engineering, 2017, Vol. 9(I) pp. 1-12.
18. Molina, F. S. ; Marco, W. P. ; Sousa, A. R. ; Tavares, J. J. Z. S. . "Algoritmo para visualização de processos de manufatura baseado na PNRD". In: Congresso Brasileiro de Automática, 2012, Campina Grande. XIX CBA - Congresso Brasileiro de Automática, 2012. Available at www.eletrica.ufpr.br/anais/cba/2012/Artigos/99831.pdf. Accessed in April 2, 2017.
19. Mainetti, L., Patrono, L., Sergi, I.: A survey on indoor positioning systems. In Proc. 22nd International Conference of Software, Telecommunications and Computer Networks (SoftCOM), pp. 111-120, 2014.
20. Martinelli, F.: A Robot Localization System Combining RSSI and Phase Shift in UHF-RFID Signals. 2015.
21. Shen, J., Wang, A., Wang, C., Ren, Y., Sun, X.: A RFID Based Localization Algorithm for Wireless Sensor Networks. Cloud Computing and Security. Vol. 10040 of the series LNCS pp 275-285, 2016.
22. DiGiampaolo, E., Martinelli, F.: "A Passive UHF-RFID System for the Localization of an Indoor Autonomous Vehicle" Ieee Transactions On Industrial Electronics, [s.l.], v. 59, n. 10, p.3961-3970, oct. 2012. Institute of Electrical & Electronics Engineers (IEEE).
23. DiGiampaolo, E., Martinelli, F.: "Mobile Robot Localization Using the Phase of Passive UHF RFID Signals" IEEE Transactions On Industrial Electronics, [s.l.], v. 61, n. 1, p.365-376, Jan. 2014. Institute of Electrical & Electronics Engineers (IEEE).
24. Costelha, H., Lima, P.: Robot task plan representation by Petri nets: modeling, identification, analysis and execution. Autonomous Robots, Springer, Volume 33, Issue 4, pp 337-360, 2012 Springer Nature.
25. Murofushi, R. H., Gonçalves, R. F., Sousa, A. R., Tavares, J. J. P. Z. S., "Indoor Positioning System Based on the RSSI Using Passive Tags." XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (lars/sbr), [s.l.], p.323-327, out. 2016. Institute of Electrical and Electronics Engineers (IEEE).
26. Murofushi, R. H., Tavares, J. J. P. Z. S., "Towards Fourth Industrial Revolution Impact: Smart Product Based on RFID Technology", to be published in April 2017. IEEE Instrumentation and Measurement Magazine.
27. A. Errignton, F. C. Angus, B. L. F. Daku, A. F. Prugger, "Initial Position Estimation Using RFID Tags: A Least-Squares Approach". Ieee Trans. Instrum. Meas., [s.l.], v. 59, n. 11, p.2863-2869, Nov. 2010. Institute of Electrical & Electronics Engineers (IEEE).
28. Silva J.R., del Foyo P.M.G. "Timed Petri Nets". Pawel Pawlewski (Ed.), Petri Nets: Manufacturing and Computer Science, InTech (2012), pp. 359-378.
29. Julvez, J, Vazquez CR, Mahulea C, Silva M. "Continuous Petri Nets: Controllability and Control" in Control of Discrete-Event Systems London U.K.: Springer vol. 433 pp. 407-428 2013. DOI: 10.1007/978-1-4471-4276-8.
30. Van der Aalst, W. "Process mining" Communication of ACM, Vol. 55, n. 8, August 2012, pp. 76, 83. DOI: 10.1145/2240236.