Wireless Sensor Network, 2011, 3, 135-146
doi:10.4236/wsn.2011.34016 Published Online April 2011 (http://www.SciRP.org/journal/wsn)

Ambient Intelligence: Awareness Context Application in Industrial Storage

Ahmed Zouinkhi[1, 2], Eddy Bajic[1], Eric Rondeau[1], Mohamed Ben Gayed[2], Mohamed Naceur Abdelkrim[2]

[1]*Research Center for Automatic Control – CRAN – CNRS UMR, Henri Poincaré University, Nancy, France*
[2]*Research Unit MACS, National Engineering School of Gabes, Gabes, Tunisia*
E-mail: {*Ahmed.Zouinkhi*, *eddy.bajic*, *eric.rondeau*}*@cran.uhp-nancy.fr*
Received February 28, 2011; *revised March* 11, 2011; *accepted March* 18, 2011

Abstract

WSNs are designed to efficiently collect data and monitor environments, among other applications. This article describes the concept and realization of an Active Security System for security management of warehousing of chemical substances using WSNs. We present an approach to modeling and simulating cooperation between intelligent products that are equipped with a platform of sensor networks and ambient communication capabilities to increase their security, in a context of ambient intelligence of a deposit for chemical substances. Behavior evolution of every intelligent product is modeled by hierarchical Petri Nets. The simulation of the model is implemented in the Castalia-OMNET++ Tools language.

Keywords: Ambient Intelligence, Intelligent Product, Cooperation, Security, WSN, Petri Nets, Castalia

1. Introduction

Amongst the main constraints and objectives in industrial processes is the security issue. Especially, in industrial environment workers have to deal with unavoidable threats from products, resources and machines that are parts of work risks. Currently, many security systems depend on safety measurements that are taken by interacting devices eventually exposing people's lives to unpredictable situation as an example in storage and transport activities of hazardous chemical substances.

Our research approach to study such fully distributed and discrete industrial environment which is based on communicating object's concept which represents a physical product equipped with perception, communication, actuation and decision making capabilities.

The communicating object's approach has attracted the interest of several research projects as COBIS project (Collaborative Business Items) [1] that has developed a new approach to business processes involving physical entities such as goods and tools in enterprise. The intention is to embed business logic in the physical entities. Also, the computing department at Lancaster University [2] conceived cooperative products with perception, analysis and communication capacities that operated by information sharing principle. Also, [3] is considering the problem of Object Safety: how objects endowed with processing, communicating, and sensing capabilities can determine their safety. He assigned an agent to each object capable of looking out for its own self interests, while concurrently collaborating with its neighbors and learning/reinforcing its beliefs from them. Each product is represented by "an object safety agent", it deals with information from environmental sensors, in a known situation. When the agent detects a threat, it seeks confirmation from its neighbors.

Ambient intelligence and communication technologies bring new visions in creating reliable systems for security management where dangerous products can be turned into smart products to control, prevent and react to security threats in the ambient process. Each product plays the role of an active node of the overall security system by means of an embedded reactive model for the security assurance.

The concept of intelligent products supported by a model that we propose, offers the possibility for objects to interact between them in an autonomous, transparent and intelligent way, without any human help. Indeed, the model presented exploits the advantages offered by a network of sensors to generate active interactions between the products in order to guarantee active security, *i.e.*, an interaction bilateral protected, transparent, autonomous and intelligent.

The aim of this work is to propose a Petri nets hierarchical modeling framework with internal cooperation model of intelligent products by using the High Level Petri Nets (HLPN) formalism. Conceptual modeling was validated by the software CPN-Tools from Aarhus University [4]. An internal model of an active product is implemented and then was validated by the simulation software Castalia based on the OMNET platform.

Our paper is organized as follows: After the introduction, the second part presents the ambient intelligent concept and the intelligent product. The third part presents the ambient security management system and the cooperation mechanisms between products based on messages exchanges. Section 4 exposes the Petri Nets modeling of a cooperation of intelligent products. Finally the last part will expose the simulation results of the system. Future research developments will be discussed in the conclusion.

2. Ambient Intelligence (AmI)

Ambient Intelligence (AmI) [5-7] is growing fast as a multidisciplinary approach which can allow many areas of research to have a significant beneficial influence into our society. AmI has a decisive relation with many areas in computer science. The relevant areas are depicted in **Figure 1**. Here we must add that whilst AmI nourishes from all those areas, it should not be confused with any of those in particular. Networks, sensors, interfaces, ubiquitous or pervasive computing and AI are all relevant but none of them conceptually covers AmI. It is AmI which puts together all these resources to provide flexible and intelligent services to users acting in their environments.

As Raffler succinctly expressed [8], AmI can be defined as: A digital environment that supports people in their daily lives in a nonintrusive way.

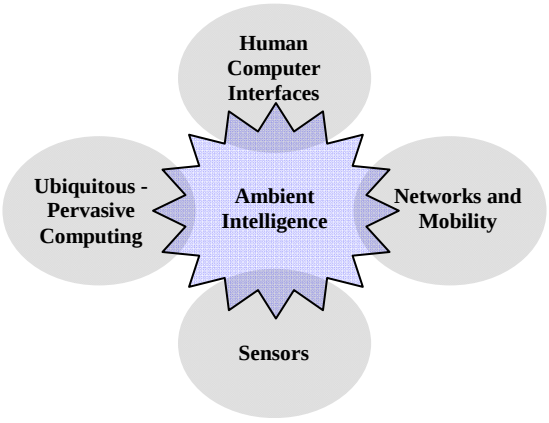

Figure 1. Relation between AmI and other areas.

AmI is aligned with the concept of the disappearing computer [9,10]: the most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.

2.1. Intelligent Product

According to [11,12], an intelligent product is defined as a physical and informational representation of an object offering the following characteristics:

1) It possesses an unique identification;

2) It is capable to communicate effectively with its environment;

3) It can retain or store data about itself;

4) It deploys a language to display its features and its needs over its lifecycle;

5) It is capable of participating in or making decisions relevant to its own destiny;

6) It can survey and control its environment;

7) It can generate interaction by services offering: contextual, personal, reactive services.

It is important to note that in the definition of intelligent product, it is possible to distinguish two levels of complexity: the product that contains the information in its environment and a product that supports decision-making mechanisms [13]. The latter is more complex because in this case it must give the product decision-making mechanisms in implying that the product must have a capacity for integrated analysis to assess and make the best decision according to its condition and context.

According to [14], the concept of intelligent product is associated with the act of managing information of an individual product through its life cycle by integrating the flow of information and equipment to provide services in an internet network.

As concrete example of applying the concept of intelligent product we quote the traceability [15] in its life cycle product automatic identification of each individual product to link a product with its physical representation of information in a distributed information system. The goal in this case is to record and update all information associated with a dynamic product (such as his statements, the operations he has endured ···). Indeed, the introduction of an automatic identification system allows the physical product to be recognized as providing the information to influence decisions and operations that a system performs with him. This involves assigning a more active role in a physical product. In this vein, [11] states that a product is an intelligent article of manufacture, that has the ability to monitor, analyze and reason about its current or future, and if it is necessary to influence his destiny.

3. Active Security Management System

3.1. General Context

In order to present the general situation of subject, we have defined the elements which constitute the global framework of cooperation. A warehouse is an environment where we stoke dangerous chemicals products. In order to ensure the safety of these products, we will check only brightness, moisture and temperature. The follow-up of these variables can help to ensure the wellness of products.

For example, starting from a value of temperature rather high one can note that the product in subject undergoes poor circumstances from where a critical condition is announced. Each containing chemicals must be equiped with a node of sensor containing: a temperature gauge, a sensor of moisture and a sensor of light, have fine to collect the variables of environment, the cycle of operation of each intelligent product is the following: to acquire the values of the sensors, to evaluate these values by consulting the clean knowledge base and decision making after having to compare the variables of environment with the critical variables. All intelligent products communicate with the manager who has the level higher (as shown in **Figure 2**). On the other hand, the objective consists with stage the mechanisms of interaction in which the intelligent product are able to communicate, acquire information, to decide and react to the stimuli and disturbances of its environment in order to make it possible that the product to deal with its intrinsic safety and total safety in its interactions with other products or people finally touching a decentralized aspect. But it is necessary to highlight which the decentralized aspect is not single, because, for example, the suitable knowledge base for have are sent by an administrator who also has like role of configured remotely these intelligent products. In made, our management system of active safety includes/understands a whole of product initially intelligent being the subject of the mutual interactions and sharing between them a flow of information and in second place a manager who undertakes to initialize and to gather the data coming from each intelligent product. Cooperation between intelligent products takes place by the exchange of messages.

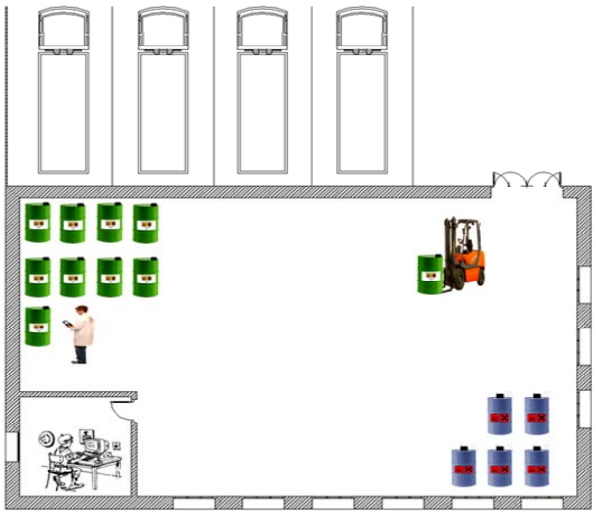

Figure 2. Intelligent product interactions in an AmI environment.

3.2. Exchanged Messages

Communication between products works by using several types of messages which are sent by a broadcasting mode and classified according to their.

Product's announcement in the products' community is of great importance for the overall security management. For this, we propose two types of messages:

CTR (Control Timestamp Request): message which declares to the administrator the arrival of a new product. Ack_CTR: the acknowledgement message from the administrator.

After registration the product needs a setup configuration to allow it to interact within the community. This configuration concerns the type of product regarding its hazardous classification (safety symbols) and its static, dynamic and community related rules as well. When not configured, a product announces its status with three types of messages: NCF0: Product has no hazardous classification and no security rules configuration, NCF1: Product has only hazardous classification configuration and NCF2: Product has only security rules configuration.

Then the system administrator answers by an appropriate product configuration command message respectively: CMD1: Configuration of the product classification and CMD3: Configuration of the security rules.

Once the product is correctly configured; it becomes completely capable of surveying its neighborhood: it is now an effective Intelligent Product (IP).

Any environment modification or event that break individual or mutual security rules must be detected by products diagnosed and has to generate external actions allowing to recover the normal safety level by actions or directed information of the ambient environment. These interactions are made by means of the following messages: GRE: a greeting message carrying specific product information (name, safety symbols) and has a further role contributing to the calculation process of the distance separating two IP. RSI: a message sent after reception of a GRE message, indicates the APs Inter-distance value calculated with the power loss of received signal.

INA: this message carries the ambient sensors values embedded in the product. CFG: a message emitted by IP after an administrator request, contains the specific configuration in the IP. SER: a broadcast message containing the IP security rules values. ALE: an alert message to report to the administrator about a threat or a defective security state.

The administrator participates in the communication part by specific command messages: CMD2: Administrator requires the configuration of the IP through this message, CMD4: Administrator asks for Security rules Configurations and CMD5: Administrator asks for specific ambient information of IPs.

3.3. Interaction Mechanism

3.3.1. Centralized Tasks

Then in order to get through chemical community, any foreign product has to introduce itself to the community manager, this product has to be announced to the manager by sending a CTR message which is an empty message that affirms to the manager the product being into the network, this message is sent continuously in broadcast mode until the manager answers by a Ack_CTR message which represents the acknowledgment of the manager after the reception of the CTR message. After having to finish the phase d' inscription with the network, the product must ask for to the manager his rules and its symbols of safety by the sending of messages NCF. The manager in his turn already identified the product (by message CTR), can provide him these needs by consulting his database by sending CMD1 containing to him the symbols for safety or CMD3 containing the safety regulations. It is noticed that the two spots announcement and configuration obey centralized approaches because each time the IP must refer to the manager for s' to identify or to update its knowledge base. The second spot is the surveillance and communication where an IP must communicate with the products of vicinity. The communication between IPs is done by the greeting message GRE. It represents a message of greeting carrying information clean of the product (name, symbols of safety, ···), its current security level; and has as a role to contribute later on to the computing process of the distance separating two IPs. As soon as an IP receives a message GRE it will transmit a Message RSSI (Received Signal Strength Indicator): The information of this type of message contains mainly the difference in power of the signal. This method of measurement is used to estimate compatibility with minimal distance between IPs.

3.3.2. Ubiquitous Tasks

Equipped knowledge base (rules and symbols of safety) and of a capacity of collecting and decision, the IP can carry out two spot essence to be well as shown in **Figure 3**. The first spot is the internal monitoring where it becomes able to supervise its vicinity, whereas any modification of its environment, violating the individual or mutual safety regulations must be detected, analyzed and finally, following a difference between the variables of environment and those basic of knowledge, with the reactions are associated such as the sending of Rapp_D to the manager announcing a state of danger. The second spot is the monitoring and communication where an IP must communicate with the products of vicinity.

3.4. Security Rules

To insure a good security surveillance of the product, three safety levels were established: (G) good level, (A) average level, (D) dangerous level. Determining security levels results after applying some security rules which are divided into three categories: Static rules, Dynamic rules and Community rules.

3.4.1. Static Rules

Each intelligent product (IP) has environment values (light, moisture and temperature), in order to avoid bad reactions, the static rule requires that these variables didn't have to exceed breaking values (min or max), following the environment constraints of product (chemical characteristics). For that, the sensor's values must be ones memorized, be compared with a static rules appropriate to the IP. The rules which we defined are founded on a whole of limits for each size to measure. The temperature has a high limit (HiLim) and low limit (LoLim) thus defining the safety intervals.

$$V_s \in \{V_{si},\ \text{Sensor } i \text{ value}\}$$

and $$i = \{\text{Temperature, Humidity, Light}\}$$

Each sensor value V_{si} is characterized by two critical

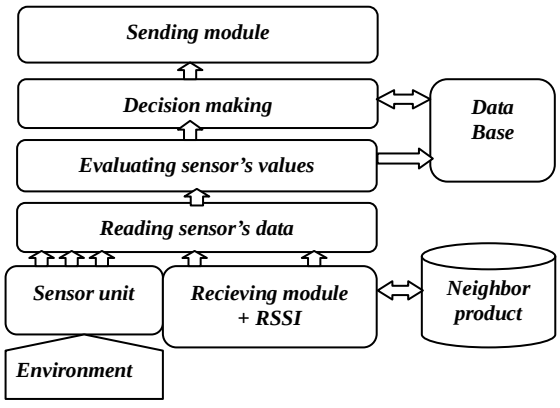

Figure 3. Autonomous behavior of an intelligent product.

values $V_{si\ \min}$ and $V_{si\ \max}$ associated also with a safety margin ΔV_{si}. For each sensor i, we have to evaluate his security level S_i which is between 3 states: S_{Gi} if the value of the sensor i defines a good state, S_{Ai} if this value announces an average or bad state and S_{Di} if the value of sensor indicates a dangerous state. From where

$$S_i = \begin{cases} S_{Gi} \text{ if } V_{si} \in \left]V_{si\min} + \Delta V_{si}, V_{si\max} - \Delta V_{si}\right[\\ S_{Ai} \text{ if } V_{si} \in \left[V_{si\min}, V_{si\min} + \Delta V_{si}\right] \cup \left[V_{si\max} - \Delta V_{si}, V_{si\max}\right] \\ S_{Di} \text{ if } V_{si} \in \left]-\infty, V_{si\min}\right[\cup \left]V_{si\max}, +\infty\right[\end{cases} \tag{1}$$

S_{sr} is the state to be concluded starting from the static rules, this state is between good (G), average (A) or dangerous (D).

$$S_{sr} = \begin{cases} D \text{ if } \exists\, i,\ S_i = S_{Di} \\ A \text{ if } \begin{cases} \forall\, i,\ S_i \neq S_{Di} \\ \exists\, i,\ S_i = S_{Ai} \end{cases} \\ G \text{ if } \forall i \begin{cases} S_i \neq S_{Di} \\ S_i \neq S_{Ai} \end{cases} \end{cases} \tag{2}$$

3.4.2. Dynamic Rules

The purpose of these rules is to develop the temporal, non-existent variable at the static rules. Because, if a bad condition persists for one considerable period, this state must be announced as dangerous state. For example, if the temperature persists in average state for considerable period, a dangerous state must be announced, also if we notice a swing between good state and bad condition, counter must be present to announce a state of danger if the number of swings exceeds a critical value.

In the same way the dynamic rules S_{dr} can conclude a dangerous (D) state described by:

$$S_{dr} = D \text{ if } \begin{cases} \exists t_1,\ \forall t \in \left[t_1,\ t_1 + T_{cr}\right]\ S_{sr}(t) = A \\ ou \\ \text{Occur}(S_{sr}(t)) \geq n_c \end{cases} \tag{3}$$

With T_{cr} is a critical period fixed according to the product at not overcome when an average state is reached, and n_c is the number of swing of S_{sr} authorized between states G and A and Occur is a function initialized with zero that is incremented when S_{sr} swings from state G to state A.

3.4.3. Community Rules

According to their chemical characteristics, certain products can have constraints of compatibility with other products according to the compatibility matrix between them. So the need of a procedure which seeks to determinate the level of compatibility between products stored in the same warehouse.

In the same way the community rules S_{cr} can conclude one of the 3 states describes previously by:

$$S_{cr} = D \text{ if } \begin{cases} \left(\begin{array}{l} \exists S_{yi} \in \{\text{Symbol of product } i\} \\ \exists S_{yj} \in \{\text{Symbol of product } j\} \end{array}\right) \\ \text{And} \\ \left(\begin{array}{l} F_{Comp}\left(S_{yj}, S_{yj}\right) = \text{Incompatible} \\ D(\text{product } i,\ \text{product } j) < D_{\min} \end{array}\right) \end{cases} \tag{4}$$

$$S_{cr} = A \text{ if } \begin{cases} \left(\begin{array}{l} \exists S_{yi} \in \{\text{Symbol of product } i\} \\ \exists S_{yj} \in \{\text{Symbol of product } j\} \end{array}\right) \\ \text{And} \\ \left(\begin{array}{l} F_{Comp}\left(S_{yj}, S_{yj}\right) = \text{Incompatible} \\ D(\text{product i},\ \text{product } j) \in \left[D_{\min}, D_{\min} + \Delta D\right] \end{array}\right) \end{cases} \tag{5}$$

$$S_{cr} = G \text{ if } \begin{cases} \left(\begin{array}{l} \exists S_{yi} \in \{\text{Symbol of product } i\} \\ \exists S_{yj} \in \{\text{Symbol of product } j\} \end{array}\right) \\ \text{And} \\ \left\{\begin{array}{l} \left(\begin{array}{l} F_{Comp}\left(S_{yj}, S_{yj}\right) = \text{Incompatible} \\ D(\text{product } i, \text{product } j) \in \left[D_{\min}, D_{\min} + \Delta D\right] \end{array}\right) \\ \text{Or} \\ F_{Comp}\left(S_{yj}, S_{yj}\right) = \text{Compatible} \end{array}\right. \end{cases} \tag{6}$$

Where $F_{comp}(S_{yi}, S_{yj})$ is a function which studies the compatibility between the safety symbols of two products i and j, D (product i, product j) is the distance which separates these two products, $D_{\min}$ is a critical distance to respect when the incompatibility between products and ΔD is a margin of distance fixed by the nature of product.

In end after having gathered the states of each rule (static, dynamic and community) it is necessary of formalized a global state S_G which describes the absolute circumstances of the product.

$$S_G = f\left(S_{sr}, S_{dr}, S_{cr}\right)$$

whether $\alpha = \{S_r, d_r, C_r\}$

$$S_G = \begin{cases} D \text{ if } \exists \alpha,\ S_\alpha = D \\ A \text{ if } \forall \alpha \begin{cases} S_\alpha \neq D \\ \exists \alpha,\ S_\alpha = A \end{cases} \\ G \text{ if } \forall \alpha \begin{cases} S_\alpha \neq D \\ S_\alpha \neq A \end{cases} \end{cases} \tag{7}$$

4. Modeling by Petri Nets

Petri Nets are used for a long time as modeling tools of discrete events systems. Several works opted for the Petri Nets modeling in fields like communication systems, flow shop and logistic chain. [16] proposed a model of TCP/IP communication behavior; [17] presented a model of a network controlled system.

The major advantages that promote the use of Petri Nets are, first the possibility to verify the system behavior have good properties and to give specifications in formal way and to provide graphic of system, and then, the possibility to model and to simulate the system [18].

The objective of our work is to represent the behavior of the active product and the stream of messages through a wireless network in order to achieve interaction between products; we opted for colored Petri Nets models designed, validated with CPN-Tools software. CPN-Tools allow creating hierarchical models in order to simplify complex ones and divide it into other submodels. This means that in the Hierarchical Petri Net model certain transitions represent another Petri Net submodel.

4.1. Global Model

The model of cooperation is equipped with six elements (P1, P2, P3, manager (Administrator), Operator, Cart) which communicate between them, in order to form a community of wireless cooperation. Each element is represented by a transition (hierarchical) which presents the services and suitable task quoted in details in what follows. As the **Figure 4** shows, each node presents two places: Net Input and Net Output which respectively presents the output buffers of each element and input one. These aims of this buffers is to memorize the messages temporarily received from network before being treated (in the processing unit) and those emitted by the elements in the network.

4.2. Network Level

In this part, we will present the network's model where the sensor's nodes interact; firstly, we are going to model the hierarchical transition network which is represented by the **Figure 4** as a perfect network (without any disturbance) to evaluate the impact of progressive increasing of node's number existing in this network, and thereafter, we will create a disturbance in this network to check the robustness of allover the system.

The **Figure 4** indicates the lower level of the Network: the higher places Net input indicate the output's buffers of the node where the messages are stored before being emitted in the network; these messages pass by a classification's stage which classify them according to their transmitting nodes before being stored in the place "message transmitted in network". The network being perfect (without any disturbance), then all the messages will pass directly through the transition network (which is not simple transition) towards the buffers from exit of the network messages received thus, all the messages will be to reclassify again according to their destination before being emitted towards the entry's buffers of the nodes.

The network presented to the **Figure 5** defines a disturbed network where there is risk of loss of message. Each token (message), which is presented in the place

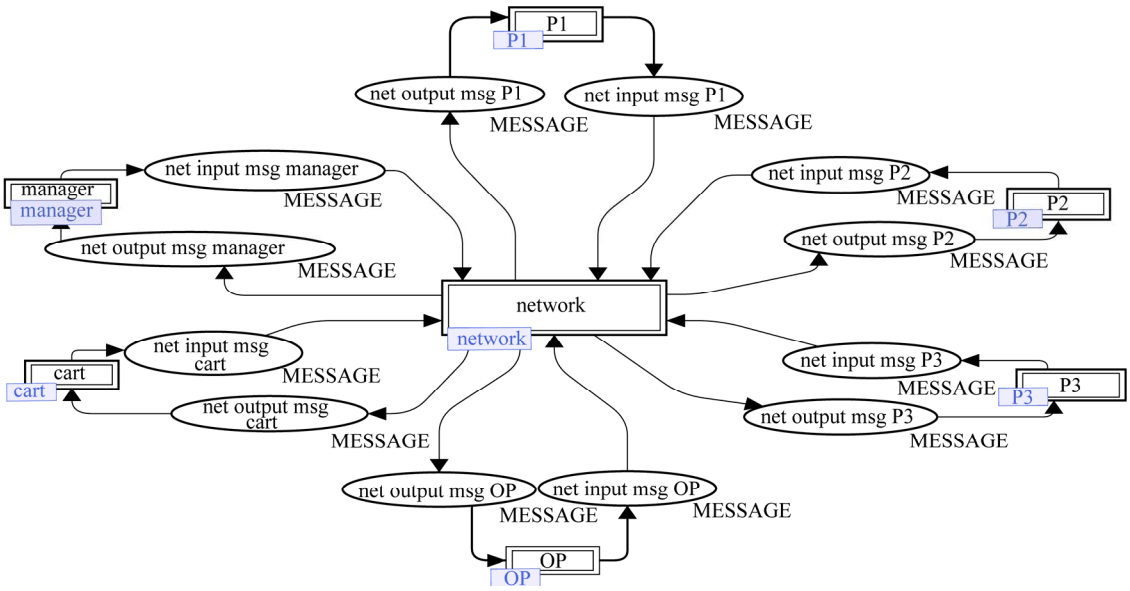

Figure 4. Global cooperation model.

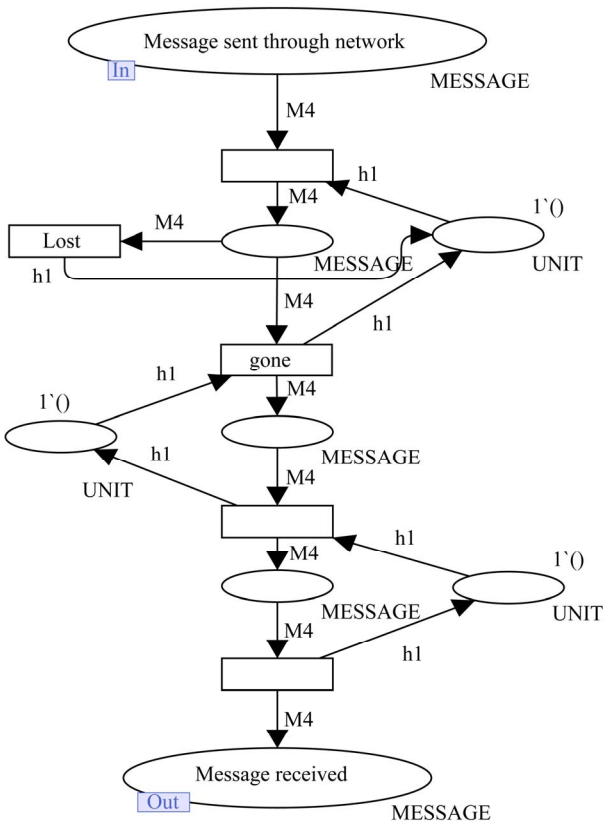

Figure 5. Network model.

("message sent through network") must cross the transition where it will be to assign to another place, in this moment this token will be lost or gone, after this passage (transition "gone"), this token enters a buffer of entry and afterwards enters a buffer of exit to be finally in the place "message received".

4.3. Intelligent Product Level

As indicated in the **Figure 6**, each product presents some internal and external tasks of which some conform the centralized approach however the others follow the approach of omnipresence, each transition in this network has a hierarchical structure described explicitly later.

4.3.1. Product's Dependence Tasks

Two tasks represented of hierarchical transition: announcement P1 and P1 configuration, illustrate the centralized approach where each IP must refer to the manager initially to announce themselves (to enter in the network and to have an ID) and also to configure themselves (to ask the manager for the safety rules).

Announcement is used to introduce a foreign product into the community of the other intelligent products. The need to launch out in this community requires announcement towards the manager so that this last detects it and adds it in its database which contains the products

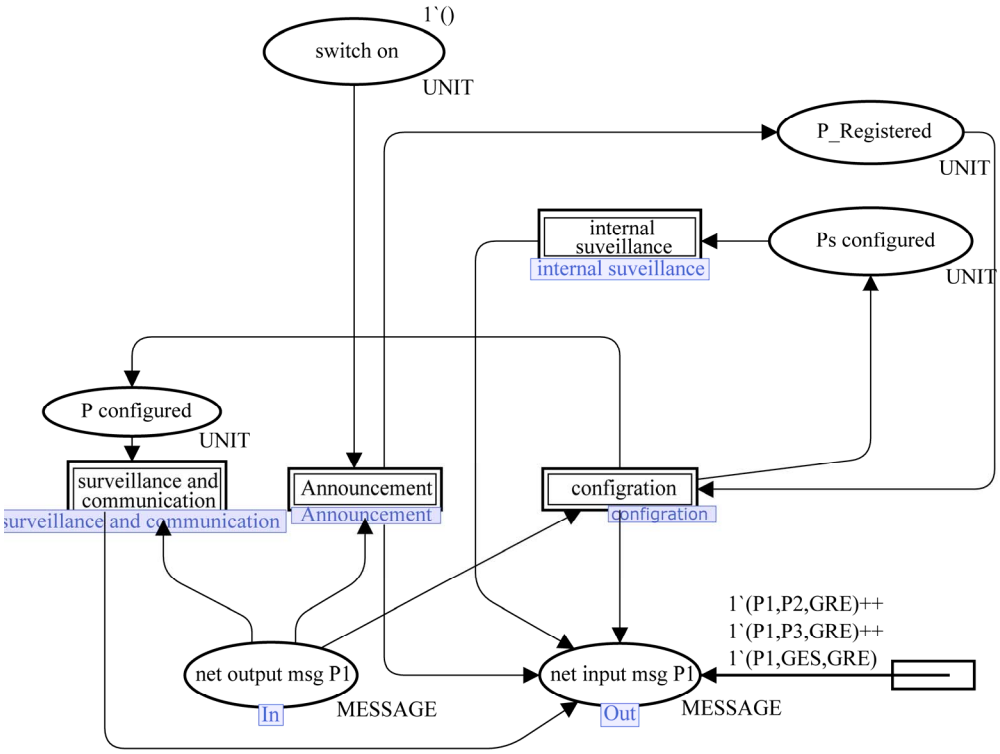

Figure 6. Intelligent product model.

already existing. This model manages the registration of products that announce them self in the community by sending continuously a CTR message to the manager. After switching on the IP, a token (P1, Man, CTR) will be put in the net input msg P1 place indicating this way the fact of sending a CTR message from P1 to the manager, the transition Ack will be valid if a token (Man, Ackctr, P1) is sent back. The absence of acknowledgement token will lead to the validation of the Ack bar transition and the same process will be repeated over again till haven an answer from manager.

Two tasks represented of hierarchical transition: announcement and configuration, illustrate the centralized approach where each product must refer to the manager initially to announce themselves (to enter in the network and to have an ID) and also to configure themselves (to ask the manager for the safety rules).

The configuration's role is to provide to the IP the necessary configurations enabling him to cooperate in the interaction community, each IP must check that it has its safety rules (its ambient critical variable) as symbol of safety (which are the IPs that presents a threat to him). And ones announced the IP has to be configured by checking if it has a safety rules and safety symbols, so in each case the IP has to react in order to get messing feature from manager by sending a request for that.

4.3.2. Product's Autonomic Tasks

Two other hierarchical transitions: surveillance and communication and internal surveillance, follow the distributed approach where each IP is equipped with a decision capacity (autonomy) which illustrates the concept of reactivity.

For the internal surveillance, the IP each time collects information from the sensor (temperature, light and moisture) and evaluates (for each variable) the safety level, so that each time, if a dangerous level is reached, the IP sends a rapp_D message to the manager to inform him that one of its sensor's variables reached a critical level [19].

The transition surveillance and communication also illustrates the distributed intelligence by the sociability concept. In this place the accepted messages are CMD2 and CMD4 and CMD5: received from manager (proactive concept) and RSSI messages: received from other IPs neighborhood (sociability concept). RSSI Messages illustrates the collaboration between IPs: each time an IP receives a GRE message and due to a module RSSI that IP will estimate the distance that separates it from the sender IP. This distance is compared to two values: L_inf and L_sup (received during the configuration).

The surveillance and communication model represented in **Figure 7**, also illustrates the distributed intelligence by the concept of sociability. In this model the accepted messages are CMD2, CMD4 and CMD5: received from manager (proactive concept) and RSSI messages: received from other products neighborhood (sociability concept). RSSI Messages illustrate the collaboration between products: each time a product receives a GRE message and due to a module RSSI that product will estimate the distance that separates it from the sender product. After the reception of these messages, a knowledge base serves for treating the different messages.

For the internal surveillance model represented in **Figure 8**, each time, the product collects information from the sensors (temperature, light and moisture) and evluates (for each variables) the safety level, so that, if a dangerous level is reached, the product sends a rapp_D message (dangerous report) to the manager to inform him that one of its sensor's variables reached a critical level [19].

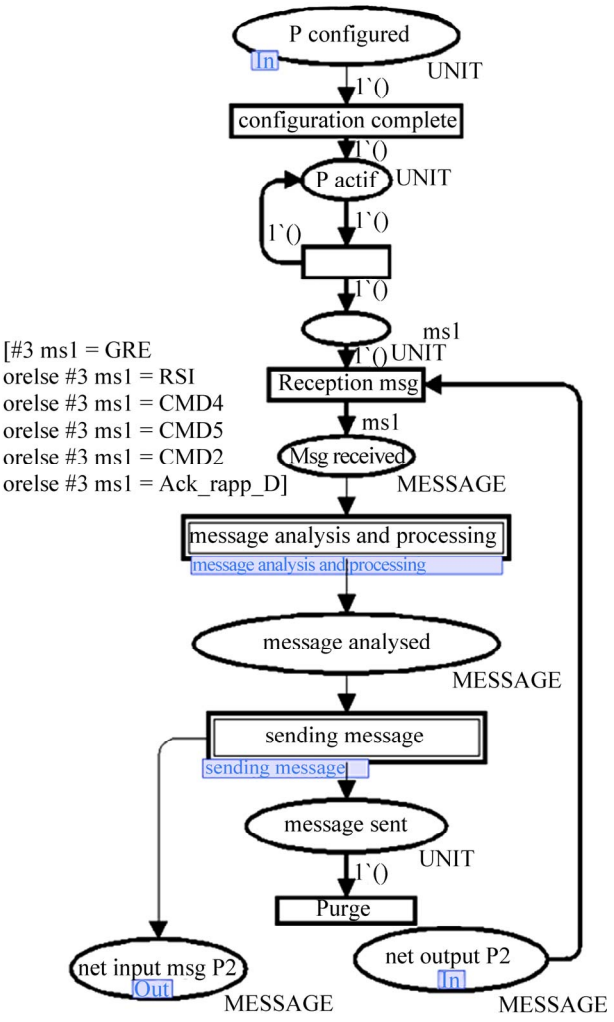

Figure 7. Surveillance and communication model.

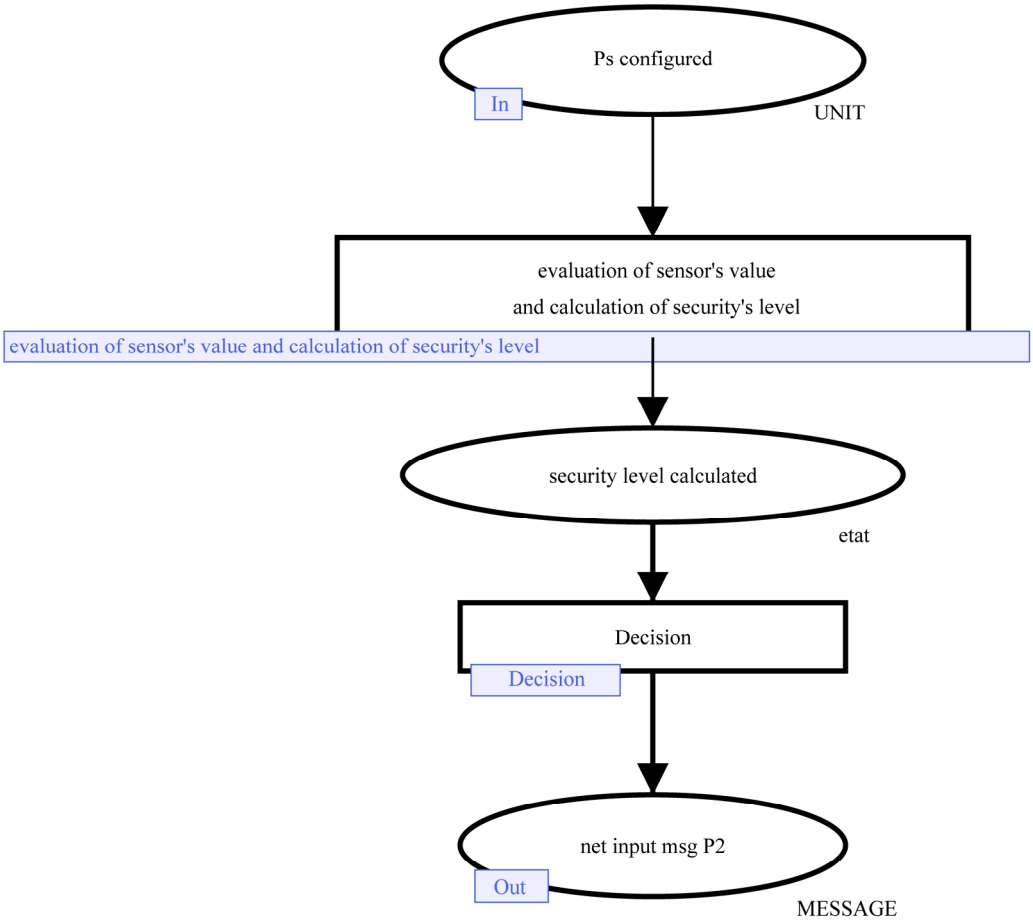

Figure 8. Internal surveillance model.

After determining the security level, a state of the product is evaluated. If the state is average (bad), a GRE message is sent in broadcast in which the security level is indicated. If the state is dangerous, a rapp_M message is sent to the manager.

4.4. Manager Level

The manager's model can be subdivided in two parts according to the concept characterizing the node: reactif or pro-actif.

The manager's reactivity [2]: when a token containing a message arrives to the entry's buffer of manager, this message passes by a stage of classification according to the nature of message (INA, CFG, NCF0, NCF1, Ack_CMD1, NCF2, Ack_CMD3, CTR, RAPP_D, RAPP_M). According to each message received the manager must react either by updating his database or by sending messages to provide information to the other IPs (safety rules, acknowledgment of the received reports ···).

Pro-activity of manager [20]: The manager anticipates sometimes by asking randomly for the variables's information of IP's environment by sending (CMD5, CMD4 and CMD2) to a hazardous chosen IPs.

With Petri Nets, we have verified the consistency and non-blocking states of our model. In addition, we have simulated the cooperation between active products with Petri Nets in terms of packets exchange and setup configuration time [19]. But the simulation becomes more difficult when the number of products becomes increasingly important. On the other hand, we cannot estimate energy consumption which is important in the Wireless Sensor Network context. For these reasons, we have to use a tool that can meet our needs as Castalia.

5. Castalia Simulation

For evaluation purposes we have implemented the IPs model into Castalia 2.0 a state of the art WSN simulator based on the OMNet++ platform.

Castalia is a Wireless Sensor Network (WSN) simulator based on the OMNet++ platform that can be used by researchers and developers who wants to test their distributed algorithms and protocols within a realistic wire-

less channel and radio model which takes account of the physical characteristics of the radio [21].

Several works opted for the Castalia simulation in fields like communication systems. [22] used Castalia/OMNET++ to demonstrate that model-based techniques (like the model checker of UPPAAL) can be used as an alternative approach to the design and analysis of WSNs to complement traditional simulation-based. They compared simulation results by UPPAAL for two medical scenarios with traditional simulation techniques. The comparison shows that their analysis results coincide close-ly with simulation results by OMNeT++.

[23] used Castalia/OMNET++ to evaluate their solution for distributed node monitoring called DiMo (Distributed Node Monitoring in Wireless Sensor Networks), which consists of two functions: Network topology maintenance, and Node health status monitoring.

In order to measure the responsiveness of the system, we have created a scenario. This scenario represents an IP of the community, which has critical value for a temperature of 200˚C. We have programmed the value of this parameter so that it exceeds the limit at $t = 200$ s.

Subsequently we are going to measure the detection time of danger by the manager after the transmission of the critical condition of the IP. Then we are going to measure the relative error in % (E_r) is defined by:

$$E_r(\%) = \frac{t_m - t_r}{t_r} \times 100 \quad (8)$$

t_m is the time of alert detection by the manager and the very occurrence moment of the alert that is equal to 200s.

Some applications of wireless sensor networks and primarily in the area of surveillance requires that the data collected must reach the base station for a time limit so that the data is useful and acceptable [24]. On the other hand, a sensor node consumes, generally, the majority of its energy during the exchange of data [25]. For this we dedicate our study mainly on system responsiveness and power consumption. To properly adjust the values of the period readings of background values (T_{cap}), we made a series of simulation.

5.1. Regulating of Reading Period Sensors

At first we changed the reading period of sensors and we were measured the responsiveness for each value system. To measure the responsiveness of the system, we have created a scenario. This scenario represents a community of IP (IP 3) which has the maximum value of ambient setting is 200 (for temperature 200℃) and was programmed so that the value of this parameter exceeds the limit at the $t = 200$ s. The following figure shows the effect of the reading period sensors on the relative error of system responsiveness. As it is shown in the **Figure 9**, the error on the reactivity of the system is minimal for a period of reading sensors equal to 0.5 seconds. On the other hand, we have studied the loss of packets based on number of intelligent products in a warehouse 25 m × 25 m surface and for each value of the reading period sensors during a simulation period equal 1000s.

In the following we will fix the sensor reading period and the period of sending messages to 0.5 s.

In our case, we fixed three aims for the simulation step that are: reactivity: The validation of all models proposed of supervision and communication, scalability: studying the model behaviour in a large-scale network and energy consumption.

5.2. Studying Reactivity

When the IP is configured, and it has the security rules, it will start to send the salutation message (GRE).

Value is the value captured by the sensor and VMax is threshold value for the sensor. (see **Figure 10**)

In the scenario of triggering alert, we simulate the sensed value as a value initialized by 7 and it would be after increased by 2 each sensing period. So, at 41.637 175 s this value reaches the threshold value (14), and in this case, it sent an ALE message on broadcast.

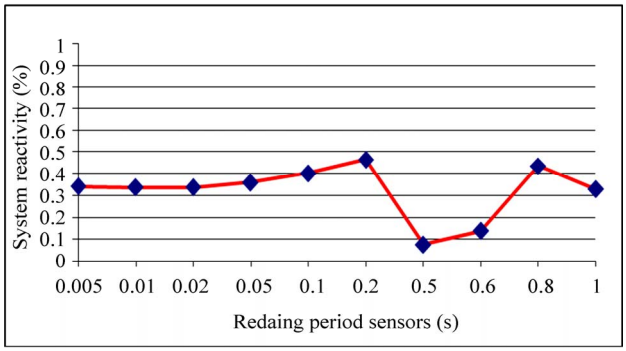

Figure 9. Influence of the reading period sensors on the system reactivity.

IP: 3 Value = 7.038 390 V_{Max} = 14.000 000
IP: 3 Value = 7.840 240 V_{Max} = 14.000 000
IP: 3 Value = 8.806 861 V_{Max} = 14.000 000
IP: 3 Value = 10.787 591 V_{Max} = 14.000 000
IP: 3 Value = 11.902 659 V_{Max} = 14.000 000
IP: 3 Value = 12.927 368 V_{Max} = 14.000 000
IP: 3 Value = 15.015 800 V_{Max} = 14.000 000
IP: 3 -> sent ALE from Value on BROADCAST at 41.637 175

Figure 10. Scenario of triggering alert.

5.3. Studying Scalability

In order to verify the influence of the adding of the IPs model in the node application under Castalia, we run multiple simulations and in each simulation, we modify the IPs number. After that, we extract from each simulation the probability of lost packets.

The histogram in **Figure 11** shows that the probability of lost packets exceeds 0.5 when the number of IPs in the warehouse surpasses the 278. In addition, it exceeds 0.2 when tthe number of IPs surpasses the 38.

5.4. Energy Consumption

Resource management is of overriding importance for Wireless Sensor Networks because the corresponding resource budgets need to be guaranteed in order to achieve certain requirements. This is particularly true for energy resources that are naturally limited. Our model should respect this particularity. The **Table 1** shows the value of the spent energy for each IP in the network. When we calculate the rate of the spent energy, we find that each IP consumes 0.85% of its initial energy (18720 joules) in a simulation time fixed to 1000s.

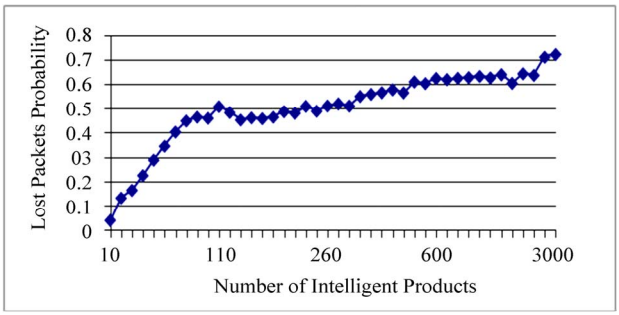

Figure 11. Influence of the number of products on the lost packets probability.

Table 1. Spent energy for each state.

Request	*Spent energy (J)*
Initialisation(CTR/ACKCTR)	0.011 764
Configuration(NCF0/CMD1/CMD3)	0.013 863
Reading security rules (CMD4/SER)	0.051 75
Reading parameters(CMD2/CFG)	0.051 75
Reading ambiant information(CMD5/INA)	0.051 75

6. Conclusions

In this work, we define a concept of an active security distributed management system, with modelling of IP's behaviour dedicated to security management of hazardous products. We proposed an IP's behavior model represented by hierarchical colored Petri nets. This hierarchy includes sub-models where each one allows displaying the evolution of every state of the IP (registration, configuration, surveillance and communication and internal surveillance). With Petri Nets, we have verified the consistency and non-blocking states of our model. Cooperation between IPs is provided by exchange of messages in order to manage and control dynamically in real-time the global active security level. To implement our approach, we are using self developed simulation test bed, designed using Castalia and OMNET++ simulators. We are currently implementing our approach in various real time scenarios to check its adaptiveness but the success and robustness of our model. Certainly, we only broke the surface of the problems associated with more realistic simulation and correspondence of real deployment data with simulation. As perspective of this work, one will develop an experimental platform in order to compare these simulation results of with the experimental results.

7. References

[1] Collaborative Business Items, European Community FP6 STREP Project, IST 004270, Technical Report, 2008. URL: www.cobis-online.de

[2] M. Strohbach, G. Kortuem and H. Gellersen, "Cooperative Artefacts — A Framework for Embedding Knowledge in Real World Objects," International Workshop on Smart Object Systems at UbiComp, Tokyo, 11-14 September 2005, pp. 91-99.

[3] B. Quanz and C. Tsatsoulis, "Determining Object Safety Using a Multiagent Collaborative System," In ECOSOA, *Workshop at the 2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, Venice, 20-24 October 2008, pp. 25-30.

[4] A. V. Ratzer, L. Wells, H. M. Larsen, M. Laursen, J. F. Qvortrup, M. S. Stissing, M. Westergaard, S. Christensen and K. Jensen, "CPN-Tools for Editing, Simulating and Analysing Coloured Petri Net," *Proceedings of the 24th International Conference on Applications and Theory of Petri Nets*, Eindhoven, Vol. 2679, 23-27 June 2003, pp. 450-462.

[5] IST Advisory Group. The European Union Report, "Scenarios for Ambient Intelligence in 2010," 2001. URL: ftp://ftp.cordis.lu/pub/ist/docs/istagscenarios2010.pdf

[6] J. C. Augusto and D. Cook. "Ambient Intelligence: Applications in Society and Opportunities for AI," *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, Hyderabad, 6-12 January 2007.

[7] J. C. Augusto, "Ambient Intelligence: The Confluence of Pervasive Computing and Artificial Intelligence," In: A. Schuster, Ed., *Intelligent Computing Everywhere*, Springer Verlag, Berlin, 2007, pp. 213-234. doi:10.1007/978-1-84628-943-9_11

[8] H. Raffler, "Other Perspectives on Ambient Intelligence," 2006. URL:www.research.philips.com/password/archive/23/pw23_ambintel_other.html

[9] M. Weiser. "The Computer for the Twenty-First Century," *Scientific American*, Vol. 165, No. 3, 1991, pp 94-104. doi:10.1038/scientificamerican0991-94

[10] N. Streitz and P. Nixon, "The Disappearing Computer-Introduction," *Communications of the ACM*, The Disappearing Computer (Special Issue), Vol. 48, No. 3, March 2005, pp. 32-35.

[11] D. McFarlane, S. Sarma, C. J. Lung, C. Y. Wong and K. Ashton, "The Intelligent Product in Manufacturing Control and Management," *Proceedings of the* 15*th IFAC Triennial World Congress*, Barcelona, 21-26 July 2002.

[12] E. Bajic, "A Service-Based Methodology for RFID-Smart Objects Interactions in Supply Chain," *International Journal of Multimedia and Ubiquitous Engineering*, Vol. 4, No. 3, July 2009, pp. 37-54.

[13] Wong C. Y., McFarlane D., Zaharudin A., Agarwal V., "The Intelligent Product Driven Supply Chain," *IEEE International Conference on Systems, Man and Cybernetics*, Hammamet, 6-9 October 2002, pp. 6-11.

[14] M. Kärkkäinen, J. Holmström, K. Främling and K. Artto, "Intelligent Products — A Step towards a More Effective Project Delivery Chain," *Journal of Computer in Industry*, Vol. 50, No. 2, 2003, pp. 141-151.

[15] M. H. Jansen-Vullers, C. A. Van Dorp, A. J. Beulensb, "Managing Traceability Information in Manufacture," *International Journal of Information Managemen*, Vol. 23, No. 5, 2003, pp. 395-413. doi:10.1016/S0268-4012(03)00066-5

[16] M. Bitam and H. Alla, "Performance Evaluation of Communication Networks for Distributed Systems," *International Journal of Computer Applications in Technology*, Vol. 25, No. 4, 2006, pp. 218-226.

[17] B. Brahimi, C. Aubrun and E. Rondeau, "Modelling and Simulation of Scheduling Policies Implemented in Ethernet Switch by Using Coloured Petri Nets," *Proceedings of the* 11*th IEEE International Conference on Emerging Technologies and Factory Automation*, Prague, 20-22 September 2006, pp. 667-674. doi:10.1109/ETFA.2006.355373

[18] A. El Fallah-Seghrouchni, S. Haddad and H. Mazouzi, "Protocol Engineering for Multi-agent Interaction," *Proceedings of the* 9*th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, Valencia, 30 June-2 July 1999, pp. 89-101

[19] A. Zouinkhi, E. Bajic, R. Zidi, M. Ben Gayed, E. Rondeau and M. N. Abdelkrim. "Petri Net Modelling of Active Products Cooperation for Active Security Management," *Proceedings of the* 6*th IEEE International Multi-Conference on System Signals and Devices*, Djerba, 23-26 March 2009, pp. 1-6.

[20] S. Kashif, F. Norsheila, H. Sharifah, K. Sharifah and R. Rozeha "Autonomously Intelligent WSN Routing Protocol Based on Ant Colony Optimization," *Proceedings of the* 7*th International Conference on Robotics, Vision, Signal Processing & Power Applications*, Langkawi, 19-20 December 2009.

[21] H. N. Pham, D. Pediaditakis and A. Boulis, "From Simulation to Real Deployments in WSN and Back," *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Espoo, 18-21 June 2007, pp. 1-6.

[22] S Tschirner, X. D. Liang and Y. Wang, "Model-Based Validation of QoS Properties of Biomedical Sensor Networks," *Proceedings of the* 8*th ACM International Conference on Embedded Software*, Atlanta, 19-24 October 2008, pp. 69-78.

[23] A. Meier, M. Motani, S. Q. Hu and S. Künzli, "DiMo: Distributed Node Monitoring in Wireless Sensor Networks," *Proceedings of the* 11*th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Vancouver, 27-31 October 2008, pp. 117-121. doi:10.1145/1454503.1454526

[24] K. Sohraby, D. Minoli and T. Znati, "Wireless Sensor Networks Technology, Protocols, and Applications," John Wiley & Sons, Inc., New York, 2007, ISBN 978-0-471-74300-2. doi:10.1002/047011276X

[25] G. J. Pottie and W. J. Kaiser. "Wireless Integrated Network Sensors," *Communications of the ACM*, Vol. 43, No. 5, 2000, pp. 51-58. doi:10.1145/332833.332838

