

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Petri Nets Hierarchical Modelling Framework of Active Products' Community

Ahmed Zouinkhi^{1,2}, Eddy Bajic¹,
Eric Rondeau¹ and Mohamed Naceur Abdelkrim²

¹*Research Center for Automatic Control - CRAN - CNRS UMR 7039, Henri Poincaré
University, Nancy, BP 239, 54506 Vandoeuvre-les-Nancy,*

²*Unit of research MACS (Modelling, Analysis and Control of Systems)
National school of engineers of Gabes, Street Omar Ibn Elkhattab, Zrig-Gabes 6029,*

¹*France*

²*Tunisia*

1. Introduction

Nowadays in industrial process area it is necessary to manage in real time all informations related to the interactions between resources, products, processes and operators along the process zone. 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 token by interacting devices eventually exposing people 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 is based on communicating object's concept which represents a physical product equipped with perception, communication, actuation and decision making capabilities. Products and resources when upgraded so to communicate with objects then become active products that are communicating with each others. So to cope with such a complete active products' community we propose to define a Petri nets hierarchical modelling framework in order to analyse and to solve cooperation and communication functionalities of active products.

The communicating object's approach has attracted the interest of several research projects as COBIS project (Collaborative Business Items) (CoBIs, 2008) 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 (Strohbach et al., 2005) conceived cooperative products with perception, analysis and communication capacities that operate by information sharing principle.

Also, (Quanz et al., 2008) 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 neighbours 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 neighbours.

Mainly focused on a security purpose but although extensible to other process management issue, our work involves transforming products with dangerous nature into communicating entities assuming the surveillance of its environment while collecting information from its surrounding. The aim of this work is to propose a Petri nets hierarchical modelling framework with internal cooperation model of active products by using the High Level Petri Nets (HLPN) formalism. Conceptual modelling was validated by the simulation software CPN-Tools from Aarhus University (Ratzer et al., 2003).

Our paper is organized as follows: after the introduction, the second part presents the concept of the active product for the security management. The third part presents the communication between active products. Modeling by Petri nets will be illustrated on section 4. Finally the last part will expose the Petri Nets modelling of a cooperation of active products. Future research developments will be discussed in the conclusion.

2. Active product

The concept of active product consists in endowing a product with the capacities of communicating, informing, acquiring, deciding and reacting to the stimulus and disruptions of its environment in order to allow the product to adjust, to influence, to cooperate, and to transform the behavior of its environment. The product is thus an intelligent and proactive actor in its ambient environment with which it interacts by means of wireless communication and its embarked sensors which allow the data entry of its environment (Zouinkhi et al., 2009).

This concept is shown in our application by integrating a sensor platform in every chemical container with a hazardous substance, therefore, upgrading it with interaction capacities in the middle of its action environment. If two active products are in the same proximity, they communicate through messages sent by radio frequency waves (See figure 1). So, an active product can communicate with the manager and the operator in the same way.

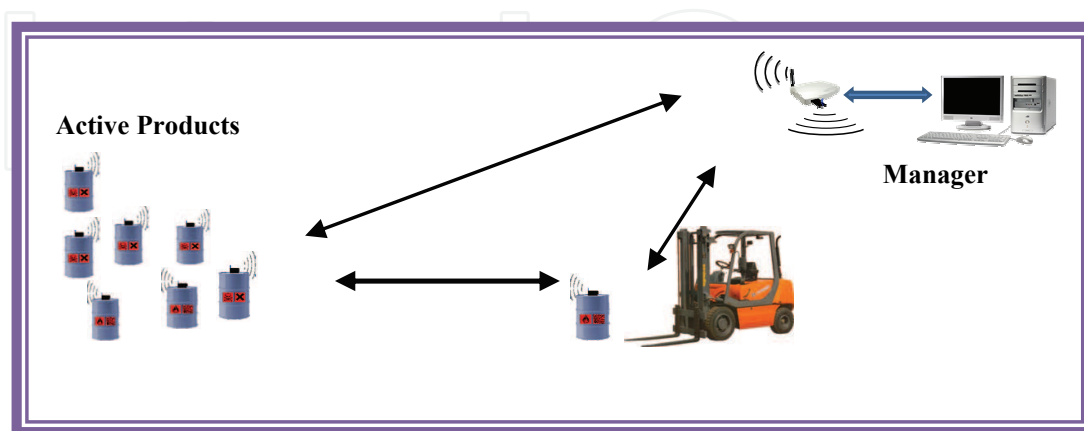


Fig. 1. Active Security management system

To insure a good security monitoring, three safety levels were established from Good to Dangerous (B: Good, M: Average and D: Dangerous). Determining security levels results

after applying security rules (logical and analytical) we have designed which are divided into three categories: (Zouinkhi et al., 2009)

- **Static Rules:** engage the product alone in its environment, this product measures some values defining its safety level (such as temperature, humidity, shock, luminosity ... these can be used e.g. for fire detection) in order to keep itself in a stable sane state, these values should not exceed certain min or/and max limits.

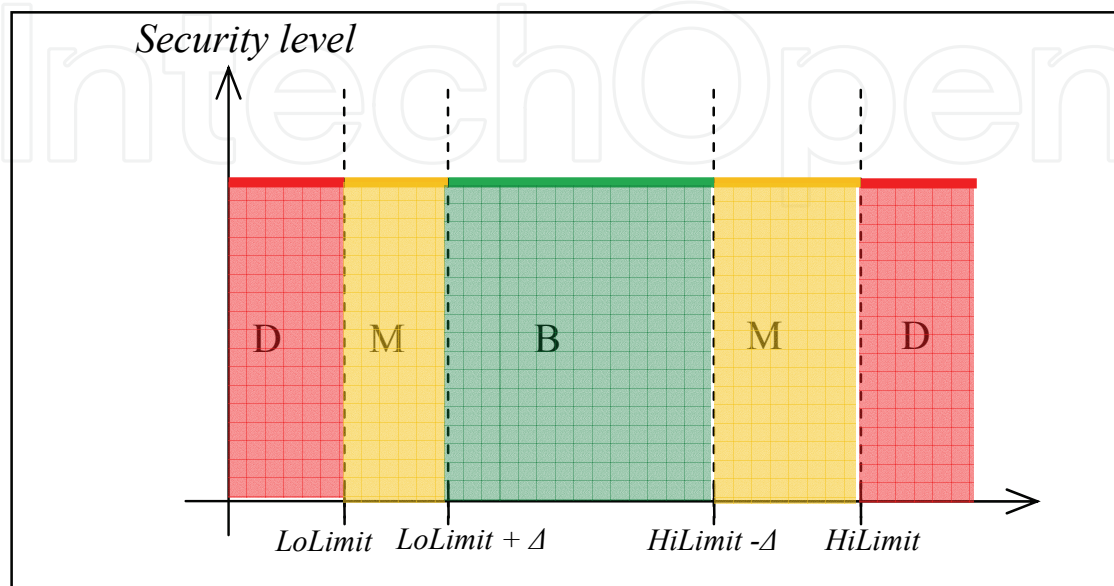


Fig. 2. Static rules

- **Dynamic rules:** these are rules related to the product by itself considering its state evolution through time. For example some product could not be affected if they reach a certain temperature threshold but the fact of reaching it several times in a period of time can bring the product in an alarming state.

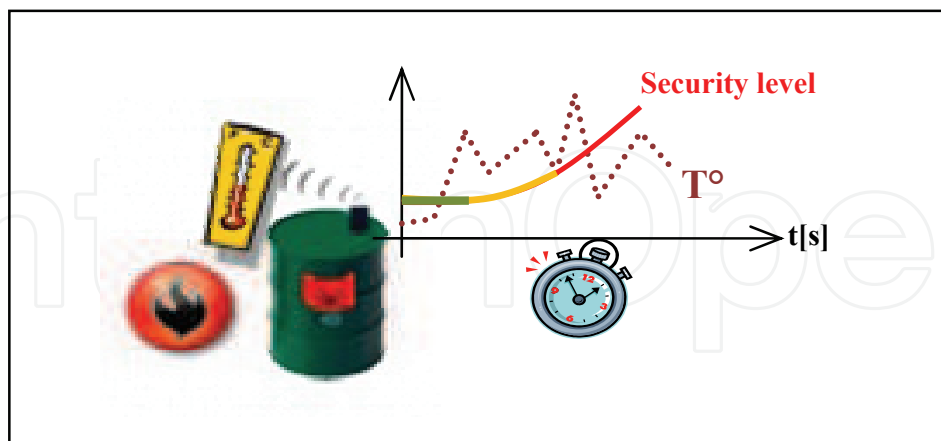


Fig. 3. Dynamic rules

- **Community rules:** depending on compatibility constraints with other products. Incompatibility is established from the security symbols and also from risk and safety phrases according to the European directives 67/548/EEC concerning chemical interaction. Also manipulation of product may require an operator with a specific fitness and aptitude; consequently, a product needs a well determined operator quality.

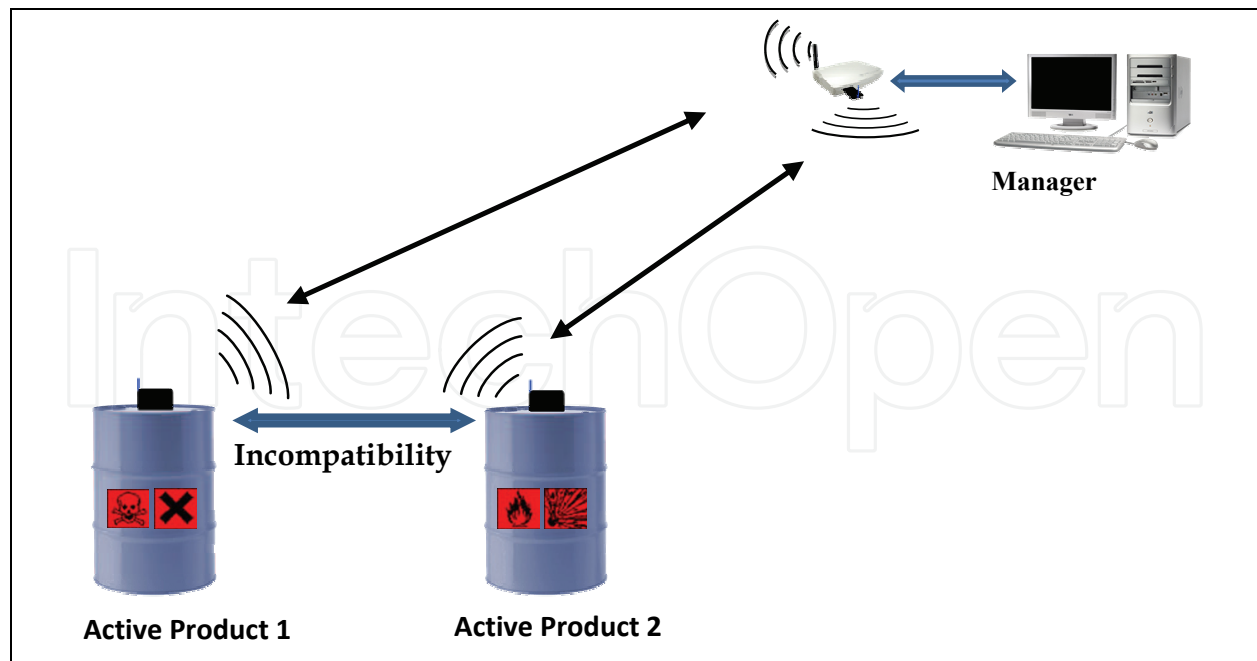


Fig. 4. Community rules

Our works promote a cooperation mechanism that integrates two approaches later centralized administration and decentralized cooperation between active products by the exchange of messages. Active products work together and exchange real-time information about the environment and the security level by a product that can manage asset's safety and security of its environment.

3. Communication between active products

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

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

CTR (Control Timestamp Request): message which declares to the manager the arrival of a new product.

AckCTR: the acknowledgement message from the manager.

After the registration of the product which 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 a product was not configured, a it 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.

NCF2: Product has only security rules configuration.

Then the system manager answers by an appropriate product configuration command message respectively:

CMD1: Configuration of the product classification.

CMD3: Configuration of the security rules.

Once the product is correctly configured; it becomes completely capable of surveying its neighbourhood: it is now an effective Active Product.

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: Greeting Message carrying specific product information (name, safety symbols) and has a further role contributing to the calculation process of the distance separating two active products.

RSI: a message sent after the 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 active product.

CFG: a message emitted by active product after an manager request, contains the specific configuration in the active product.

SER: a broadcast message containing the active product security rules values.

ALE: an alert message to report to the manager about a threat or a defective security state and can be sent as a rapp_D message if the state of active product is dangerous or rapp_M if the state is average.

The manager participates on the communication part by specific command messages.

CMD2: Manager requires the configuration of the active product through this message.

CMD4: Manager asks for Security rules Configurations.

CMD5: Manager asks for specific ambient information of active products.

4. Modeling by Petri Nets

The Petri nets are used for a long time as modeling tools of discrete events systems.

Petri Nets (PN) and particularly, Colored Petri Nets (CPN) (Jensen et al., 1997), are a powerful and a recognized modelling tool, endowed with a big expressiveness and allowing to represent the two aspects of system: static thanks to the PN structure and dynamic thanks to the token distribution evolution (Bouali and al., 2009).

Petri net (Murata, 1989) is an effective tool for modeling manufacturing systems. The advantages of applying Petri nets formalism are summarized as follows. First of all, the graphical nature of Petri nets can visualize sequences of firing via token passing over the net. Second, Petri nets have well-established formal mechanisms for modelling and analysis of manufacturing systems (Rudas et al., 1997) (Hsieh, 2004) (Zurawski, 2005). Third, the mathematical foundation of Petri nets can analyze structural and dynamic behaviours of a system. These advantages make Petri nets a suitable modelling and analysis tool.

(Song et al., 2008) define the Petri net as a tool modelling events that require a special synchronization as wireless sensor network. The system modelled is a safety system (evacuation) used in the mines of a coke which locates the positron of miners in an accident. The modelisation is divided into two phases: modeling of a particle (service and communication) and generalizing model to a large scale (performance evaluation and interaction between particles).

(Fu-Shiung, 2009) presents a concept of verification and resolution problem due to the mechanism of cooperation and interaction of Multiagent systems. These systems are often modeled by Petri nets and the approach consists in controlling the vivacity of network, a character illustrating the efficiency of the interaction between particles.

(Khoukhi et al., 2010) notes that the classic Petri nets is unable to model uncertain systems what motivated the researchers to combine between the Petri nets and fuzzy logic to reach a Fuzzy Petri nets used in various application such as robotics and real-time control systems. We choose High Level Petri Nets (HLPN) formalism to model the cooperation between active products. This model uses a generic and modular approach which requires the use of colouring and hierarchy.

Others authors used this formalism to model the Ethernet switch (Marsal, 2006). (Brahimi and al., 2008) are used HLPN formalism to propose an integrated modelling environment to represent globally the Networked Control Systems behaviour.

Hierarchical Coloured Petri Nets are used for modelling communication because Petri Nets are a formal method enabling to express parallelism, synchronisation, interaction, resource sharing, temporal (and stochastic) properties, and to achieve qualitative analysis (checking of the logic of the non temporal and/or temporal mechanisms) and quantitative analysis (performance evaluation and/or reliability). Then this formalism offers a framework well adapted and progressive for the representation and the analysis of the communication systems (Brahimi and al., 2008).

5. Model of cooperation between active products

The cooperation model has several components (P_1, P_2, \dots, P_i , Manager) that communicate among themselves, within a wireless network. Each element is represented by a transition (hierarchical) which shows the services and the appropriate patches listed in detail in what follows.

As the figure 5, each element has two Capacities: "Net Input msg" and "Net Output msg" which are respectively output buffers of each product and the input.

These buffers have role in order to temporarily store messages received from network before being treated and those issued by products in the network.

The objective of our work is to represent the behaviour of the active product and the stream of messages through a wireless network in order to achieve cooperation 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 sub-models. What is meant here that in the hierarchical Petri net model certain transitions represent another Petri net sub-model.

5.1 Active products level

Figure 6 represents an internal model of an active product P_1 , in this structure the transitions hold a description of a functioning part of the product. The place "net Input msg P_1 " corresponds to the wireless communication network and collects messages emitted and received from all active products; let us note that the messages are sent in broadcasting mode. This characteristic is carried out by a transition which puts tokens in all the places corresponding to the various products. Messages intended for a specific active product will fire its own "net output msg P_1 " place. Circulating messages will be represented by colored tokens.

Each active product represents 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.

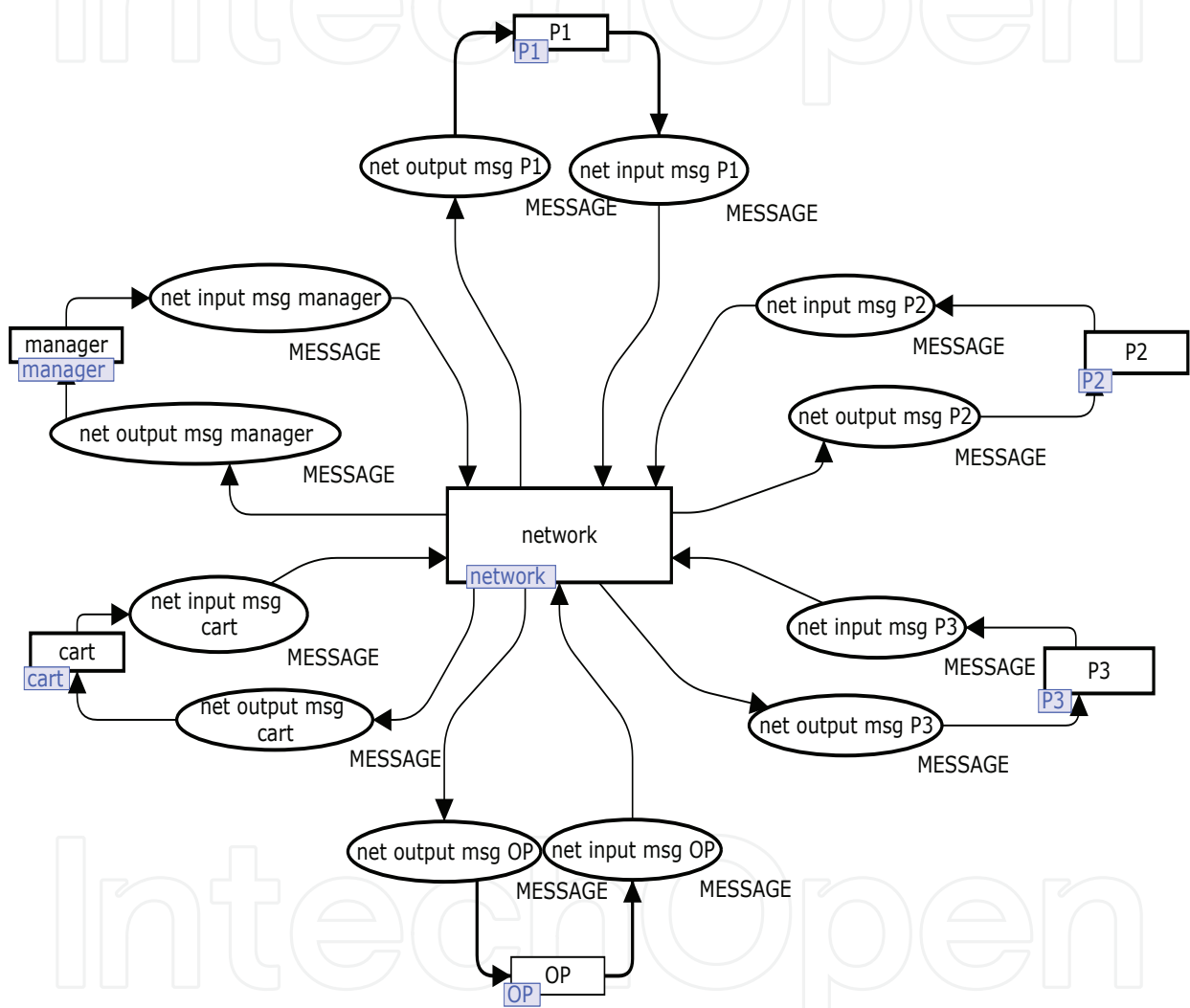


Fig. 5. Global cooperation model

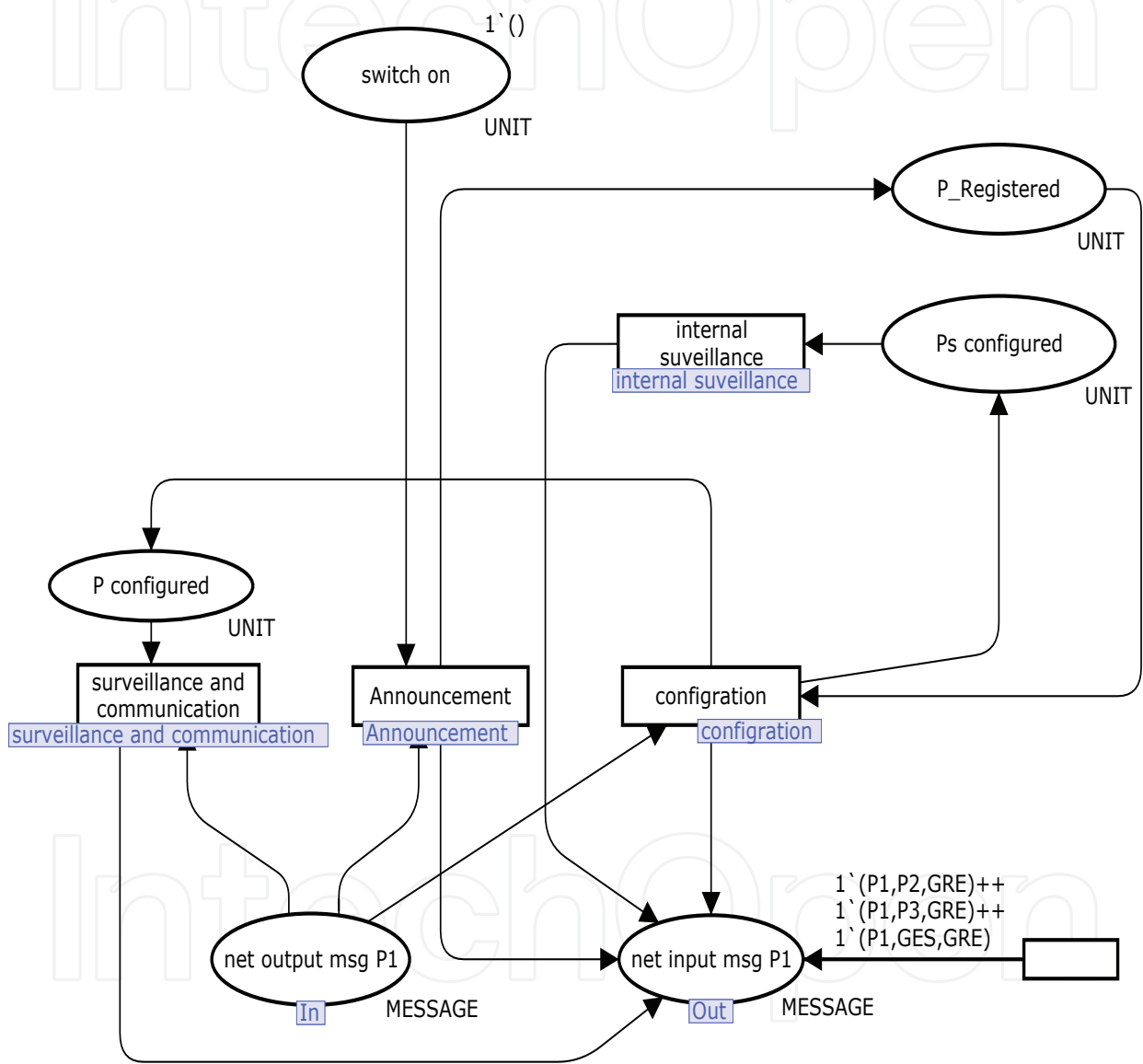


Fig. 6. Active product model

5.2 Product's dependence tasks

Two tasks represented of hierarchical transition: announcement (Figure 7) and configuration (Figure 8), 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).

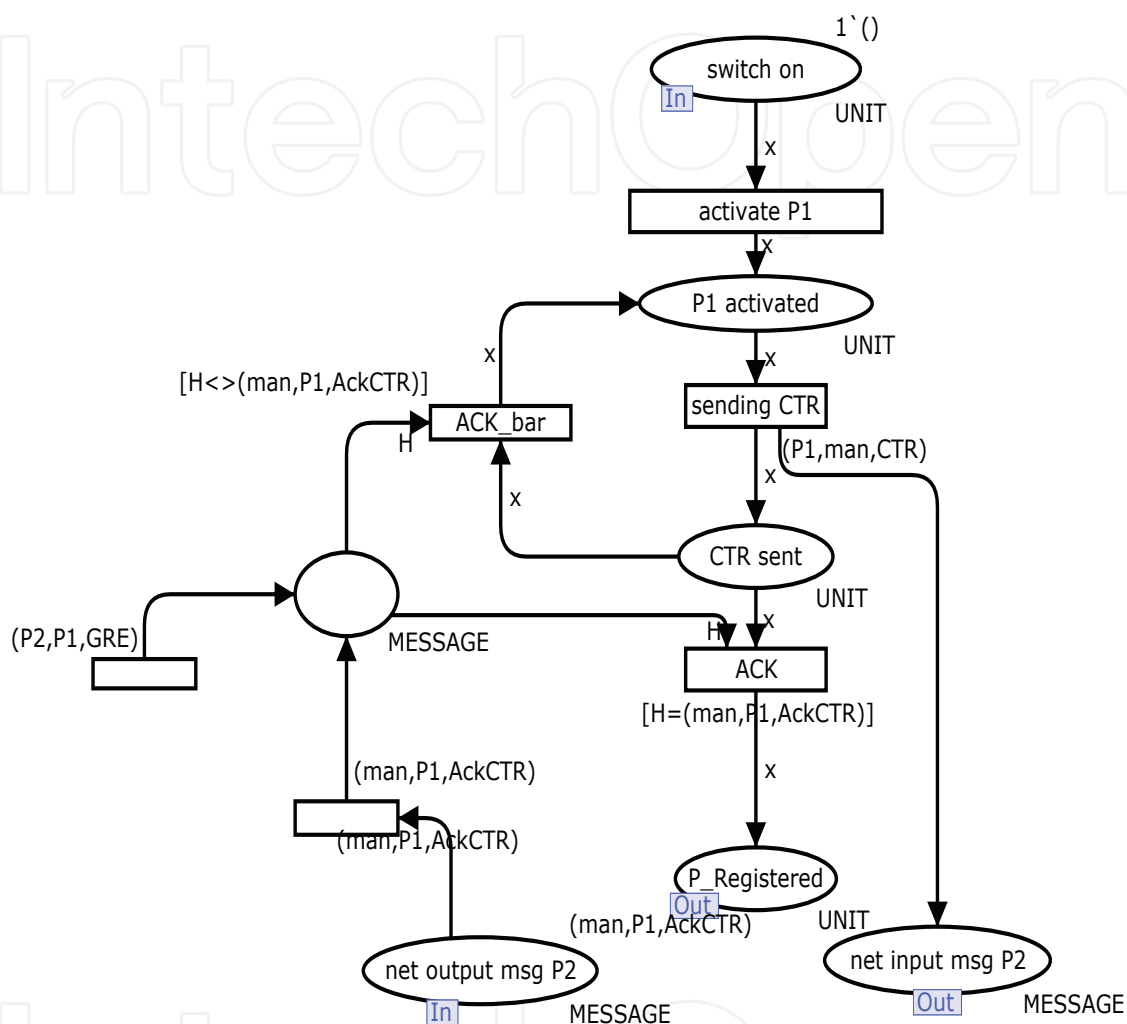


Fig. 7. Announcement model

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 already existing.

The configuration's role is to provide to the intelligent product the necessary configurations enabling him to cooperate in the interaction with the community (the neighborhoods), each product must check that it has its safety rules (its ambient critical variable) as symbol of safety (which are the products that presents a threat to him).

And as shows in the figure ones announced the active product has to be configured by checking if it has a safety rules and safety symbols, in dead we have to notice 4 probable case: c_s (rules + symbols), c_ns (rules + messing symbols), nc_s (messing rules + symbols) and nc_ns (messing rule + messing symbols) so in each case the active product has to react in order

to get messing feature from manager by sending a request for that. So we notice that we have a classification stage (c_s, c_ns, nc_s, nc_ns) to pick out in each case we are, ones classified and depending on the messing feature one token is going to be sent to the manager.

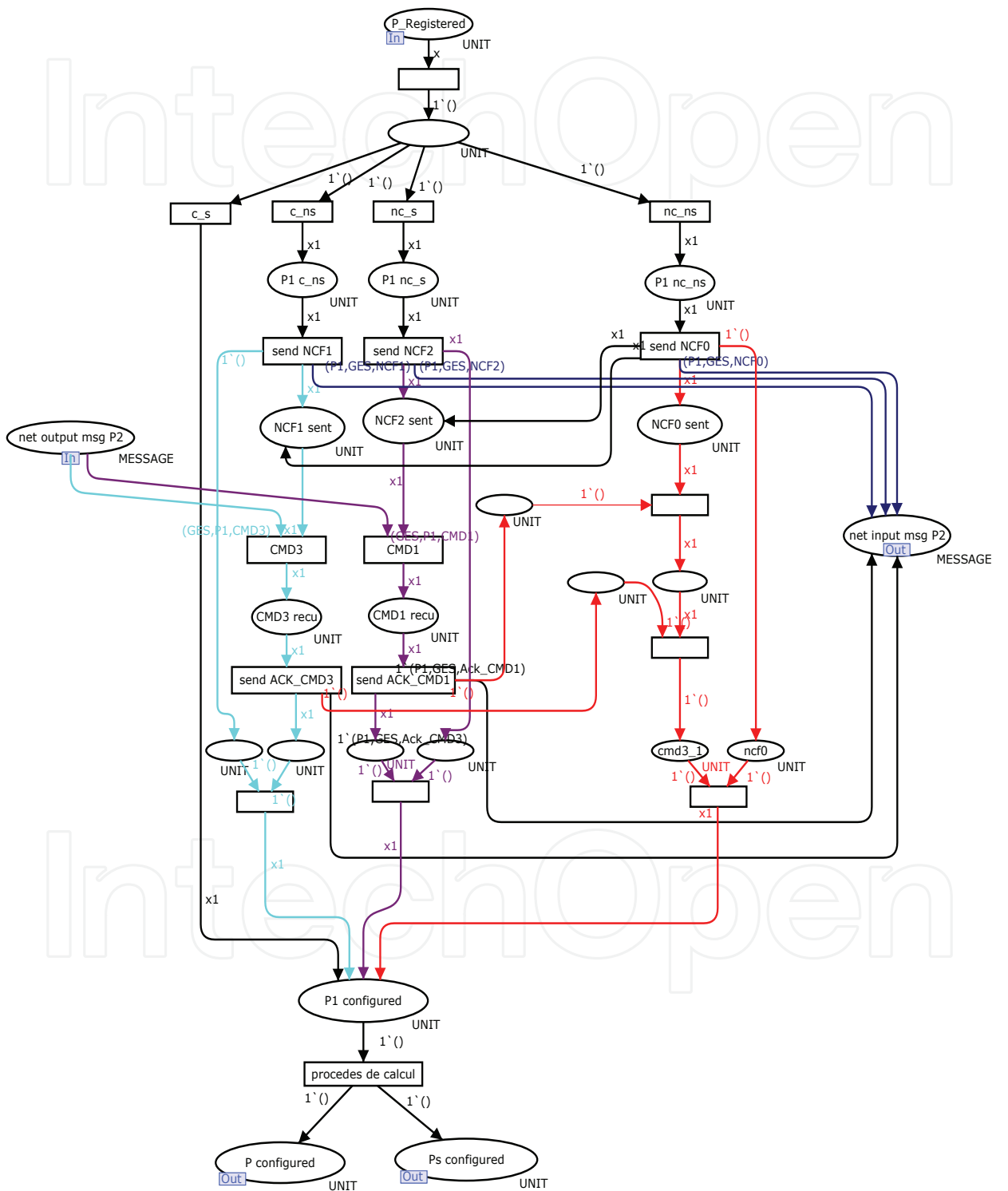


Fig. 8. Configuration model

5.3 Product's autonomic tasks

The two other hierarchical transitions represented in figure 6: surveillance and communication and internal surveillance, follow the distributed approach where each product is equipped with a decision capacity (autonomy) which illustrates the concept of reactivity.

The surveillance and communication model represented in figure 9, also illustrates the distributed intelligence by the concept of sociability.

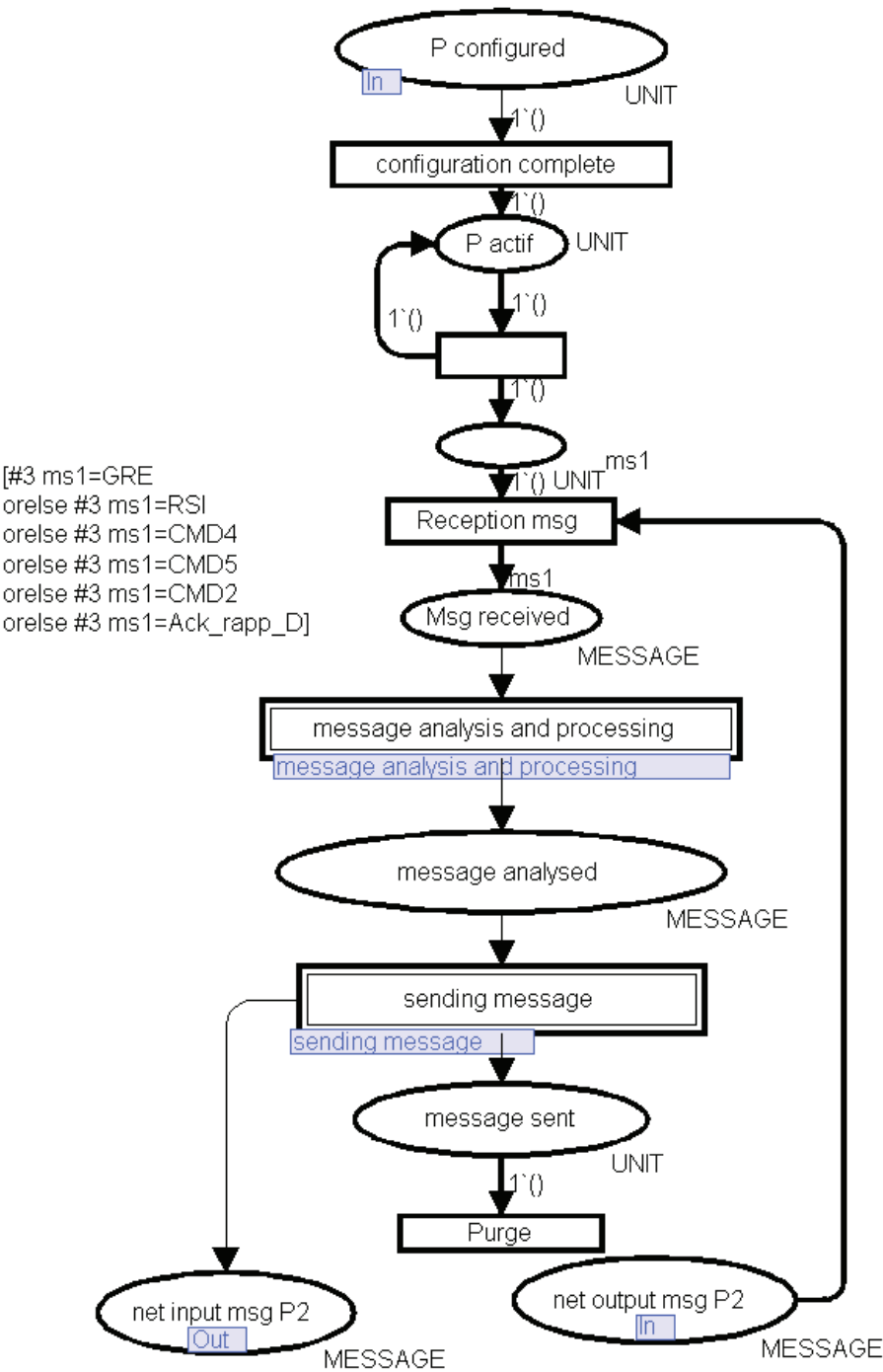


Fig. 9. Surveillance and communication model

In this model the accepted messages are CMD2, CMD4 and CMD5: received from manager (proactive concept) and RSSI messages: received from other products neighbourhood (sociability concept). RSSI Messages illustrate the collaboration between products: each time a product receives a GRE message and due to a module RSSI (Received Signal Strength Indicator) that product will estimate the distance that separates it from the sender product. This distance is compared to two values: L_{inf} and L_{sup} (received during the configuration). These messages are illustrated in figure 10. After the reception of these messages, a knowledge base serves for treating the different messages.

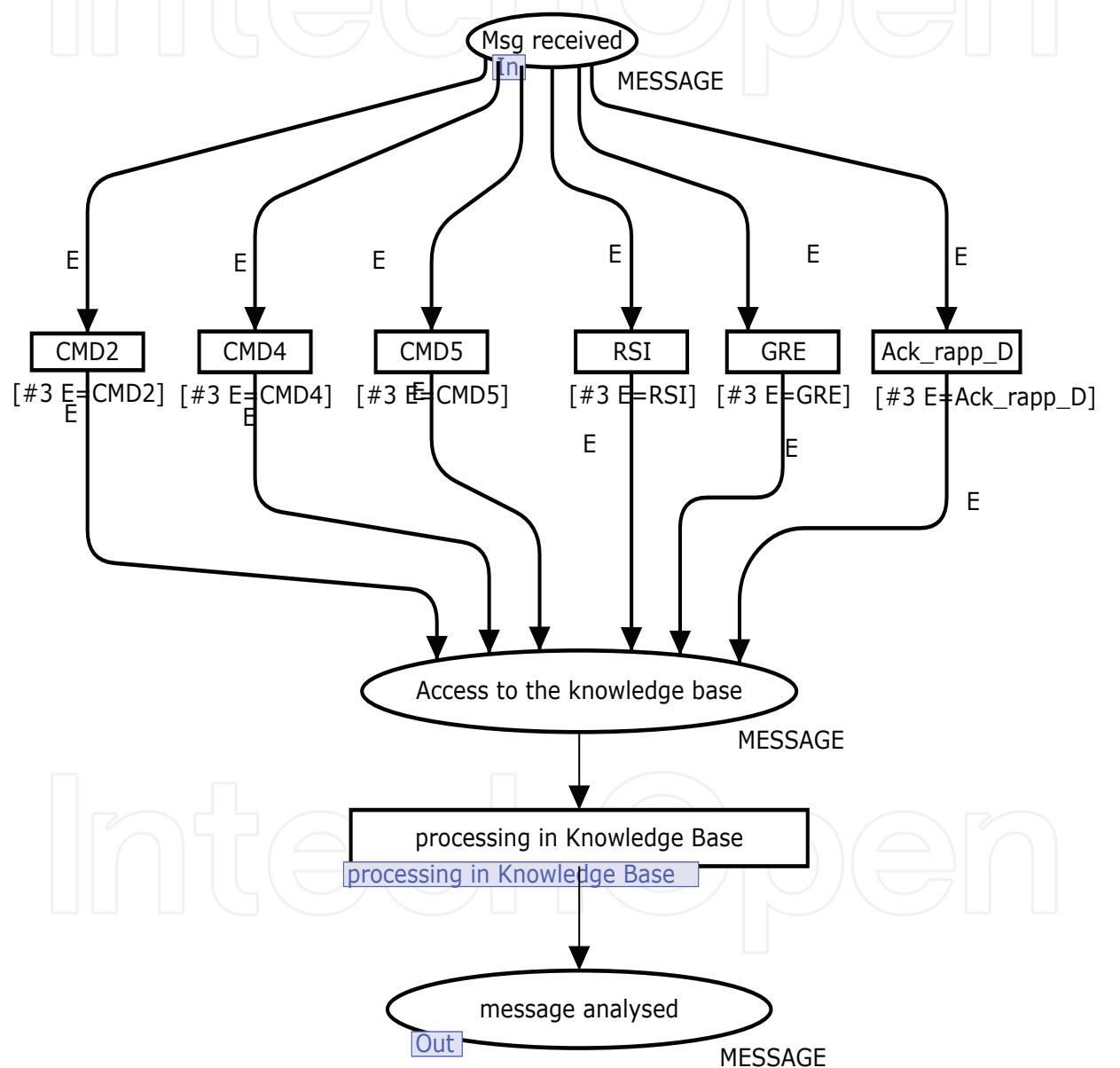


Fig. 10. Analysis and message processing model

The processing in knowledge base transition of the figure 10 represent a sub model that is represented by figure 11.

After the determination of the distance between incompatible products, three cases can be distinguished:

- If $\text{distance} > L_{\text{sup}}$: indicates that we have a comfort distance between the two products.
- If $L_{\text{inf}} < \text{distance} < L_{\text{sup}}$: indicates that the product is in bad condition which resulted in the sending of a message rapp_M (bad report).
- If $D < L_{\text{inf}}$: illustrates a state of danger because the distance between the two products are a critical distance where one has a risk of a dangerous chemical reaction, therefore a message rapp_D will be sent to the manager.

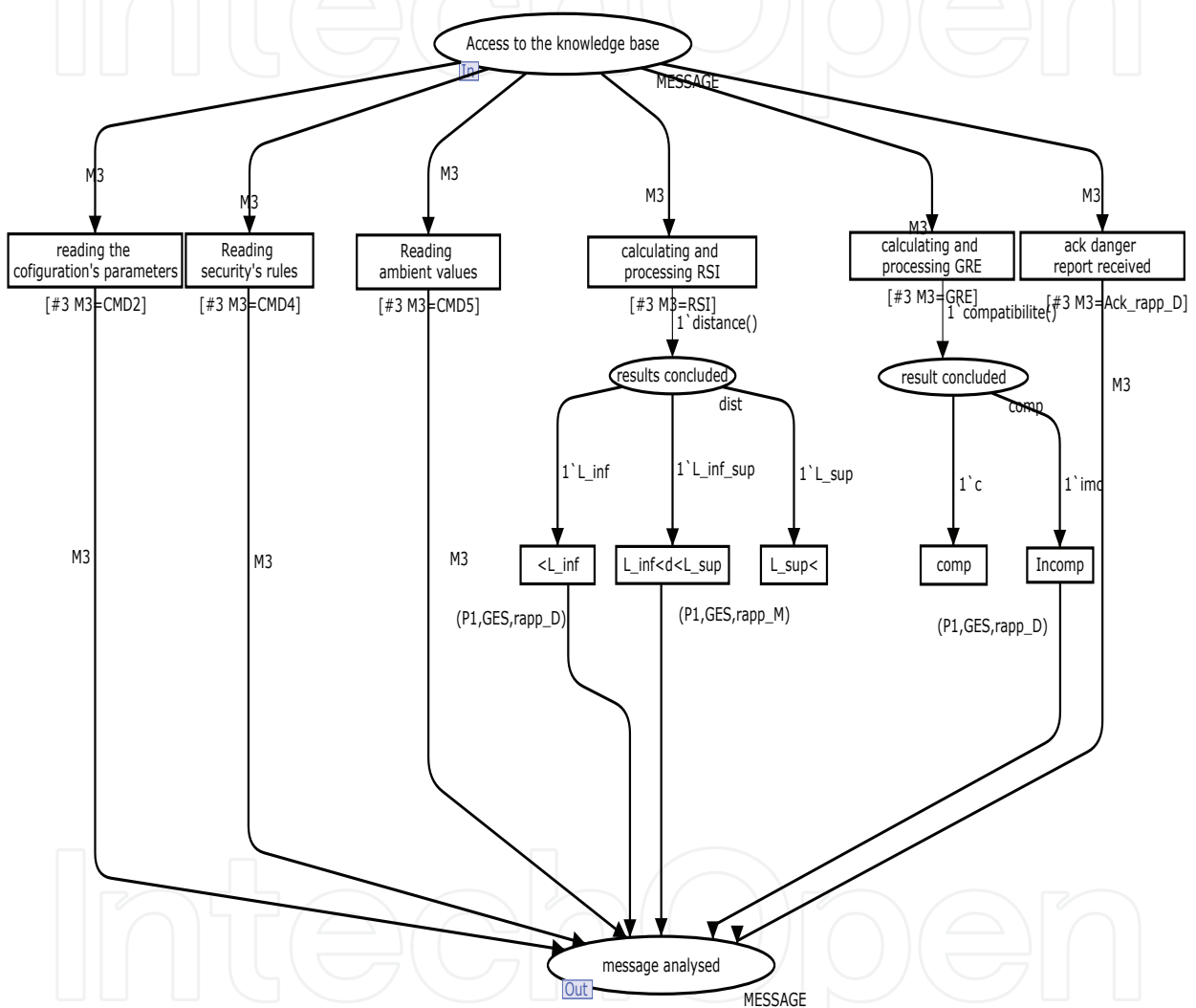


Fig. 11. Processing in knowledge base model

The sub model of sending message transition represented in figure 9 as follow; After analyzing the different messages received, the product reacts by sending a corresponding message;

- If the product receives a CMD2 message, it sends a CFG message.
- If the product receives a CMD4 message, it sends a SER message.
- If the product receives a CMD5 message, it sends an INA message.

If the product is in a danger state, a report will be sent to the manager and an alarm is activated until it receives an acknowledgment of the manager.

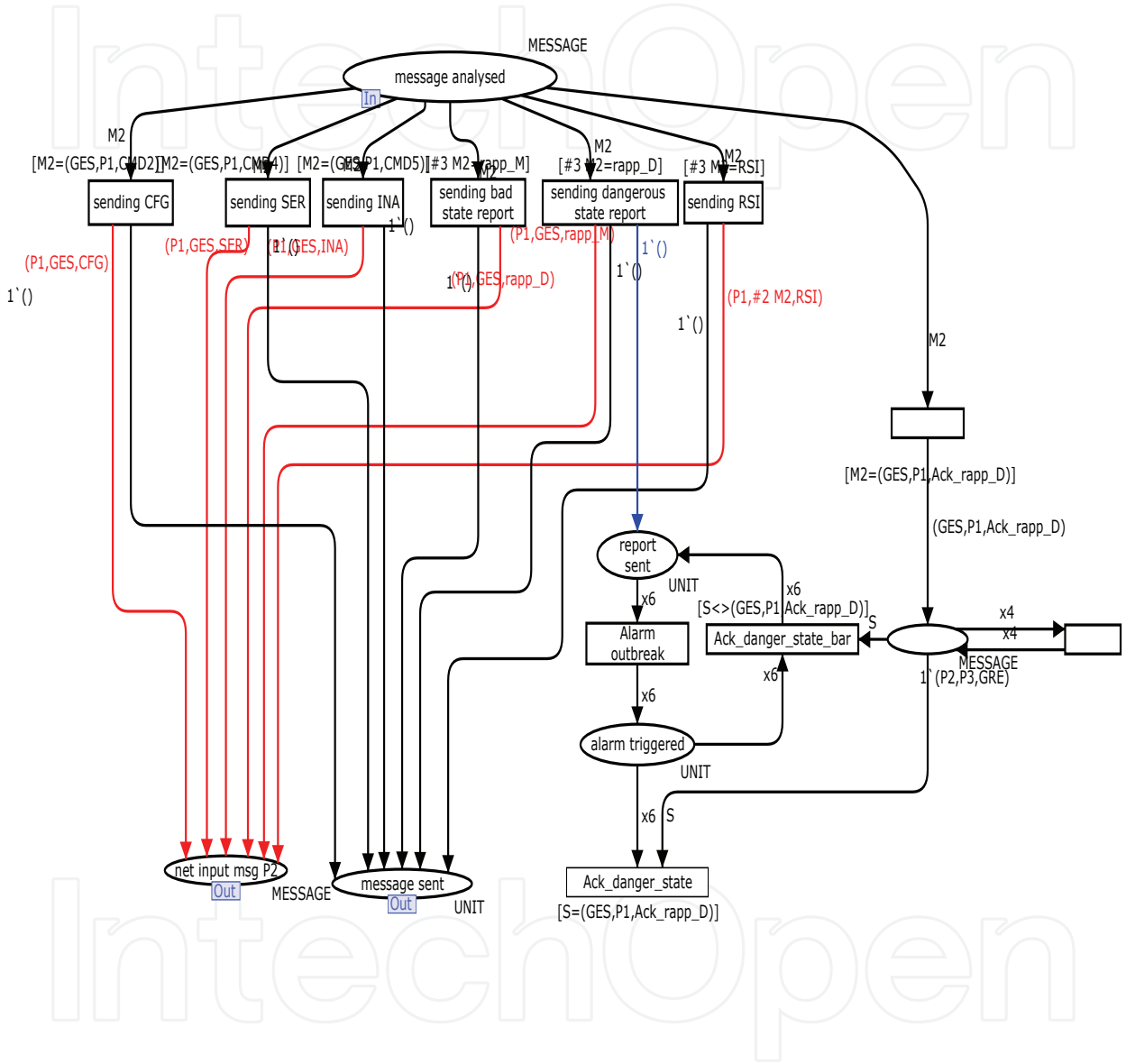


Fig. 12. Sending message in Surveillance and communication model

For the internal surveillance model represented in figure 13, each time, the product collects information from the sensors (temperature, light and moisture) and evaluates (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 (Zouinkhi et al., 2009).

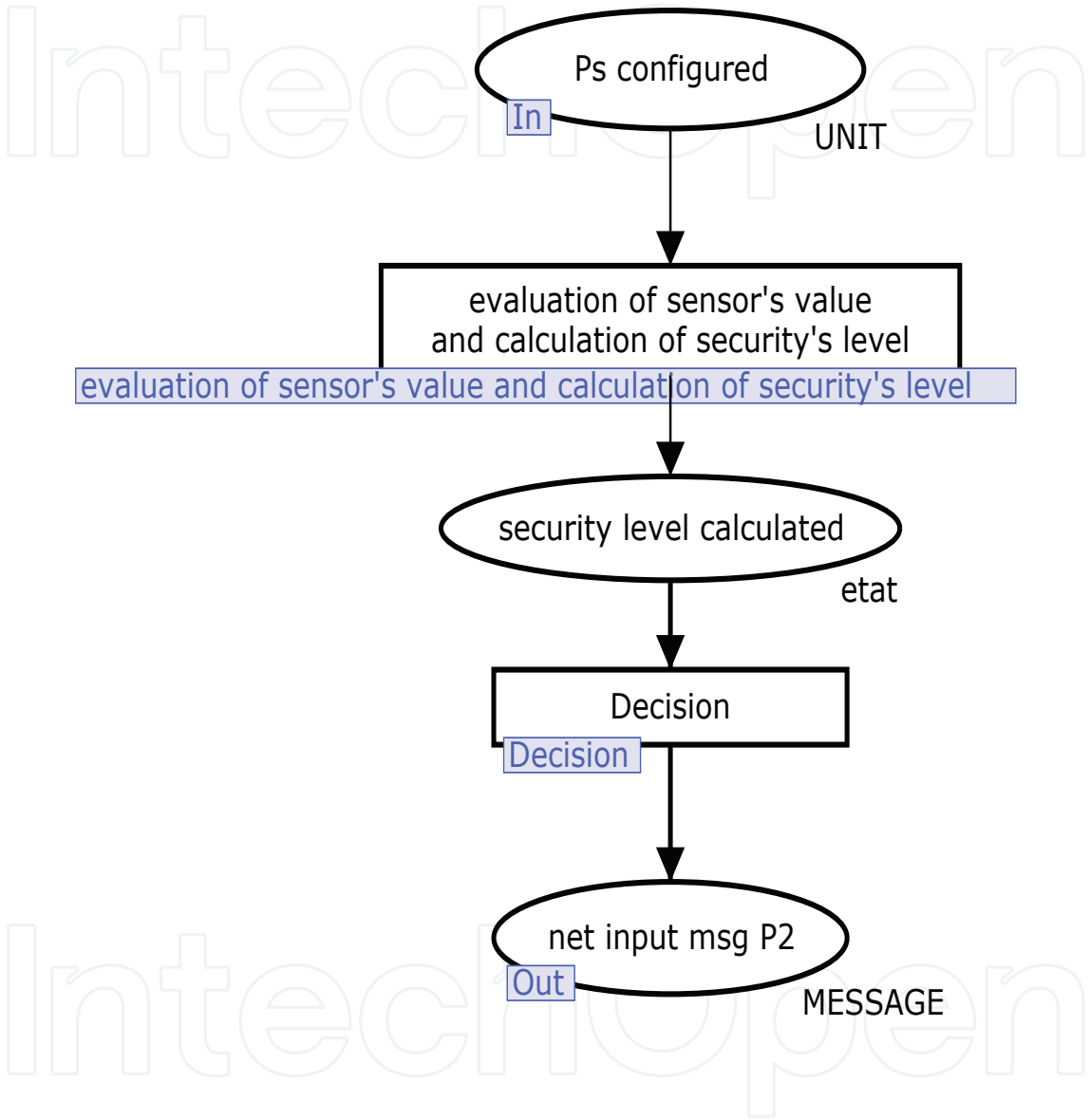


Fig. 13. Internal surveillance model

The decision transition can be represented by figure 14. 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.

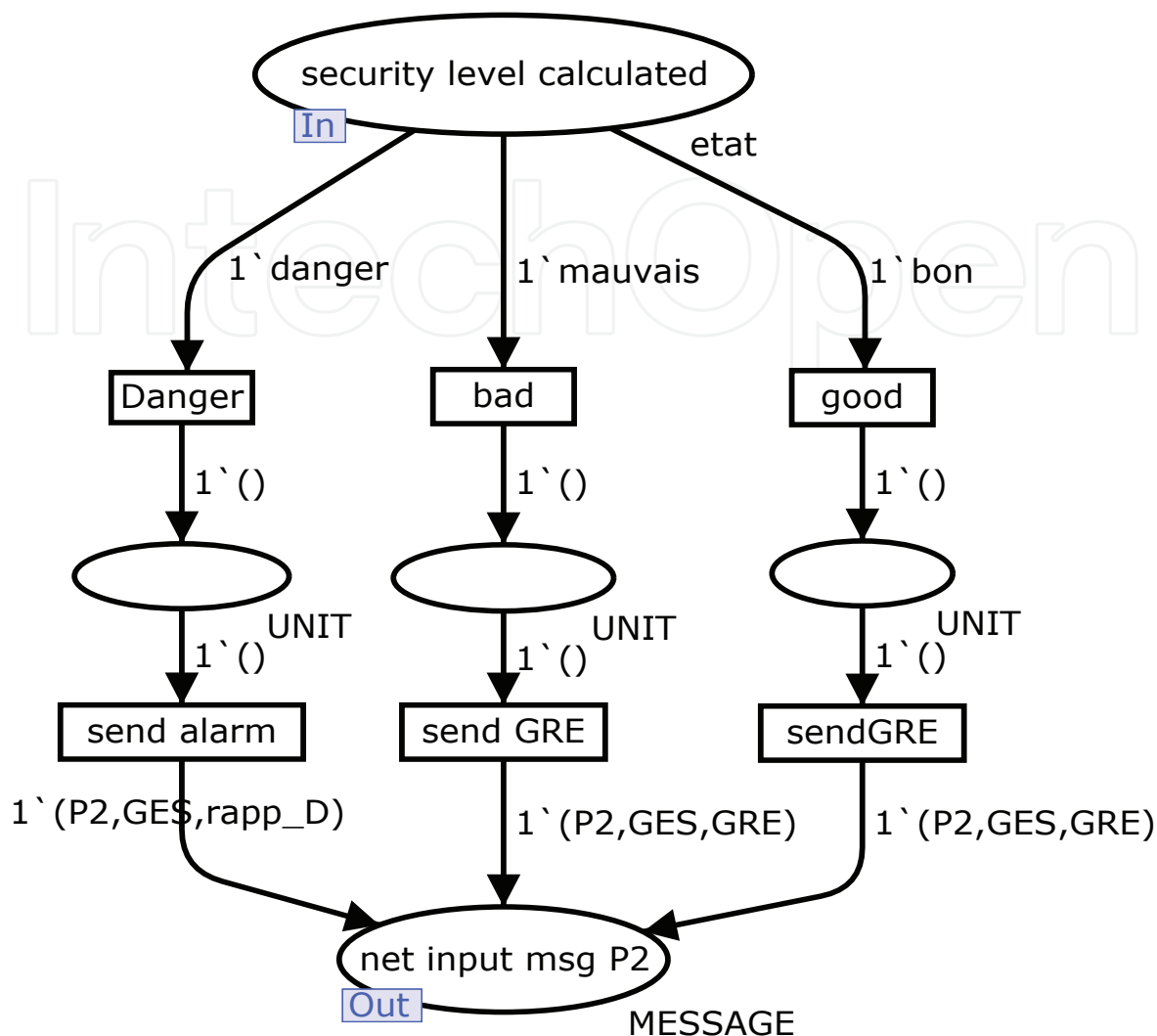


Fig. 14. Decision level in internal surveillance model

5.4 Manager model

The manager’s model (Figure 15) can be subdivided in two parts according to the concept characterizing the product: reactif or pro-actif.

The manager’s reactivity (Strobach et Al., 2005): when a token containing a message arrives to the entry’s buffer of manager, this message passes by a stage of classification as the figure indicates it according to the nature of message (INA, CFG, LIKING, NCF0, NCF1, Ack_CMD1, NCF2, Ack_CMD3, CTR, RAPP_D, RAPP_M, NCFOP, CMP). According to each message received the manager must react either by updating his database or by sending messages to provide informations to the other products (safety rules, acknowledgment of the received reports...).

Pro-activity of manager (KASHIT et al., 2009): As figure 15 indicates it, the manager anticipates sometimes by asking randomly for the variable’s information of product’s environment by sending (CMD5, CMD4 and CMD2) to a hazardous chosen products.

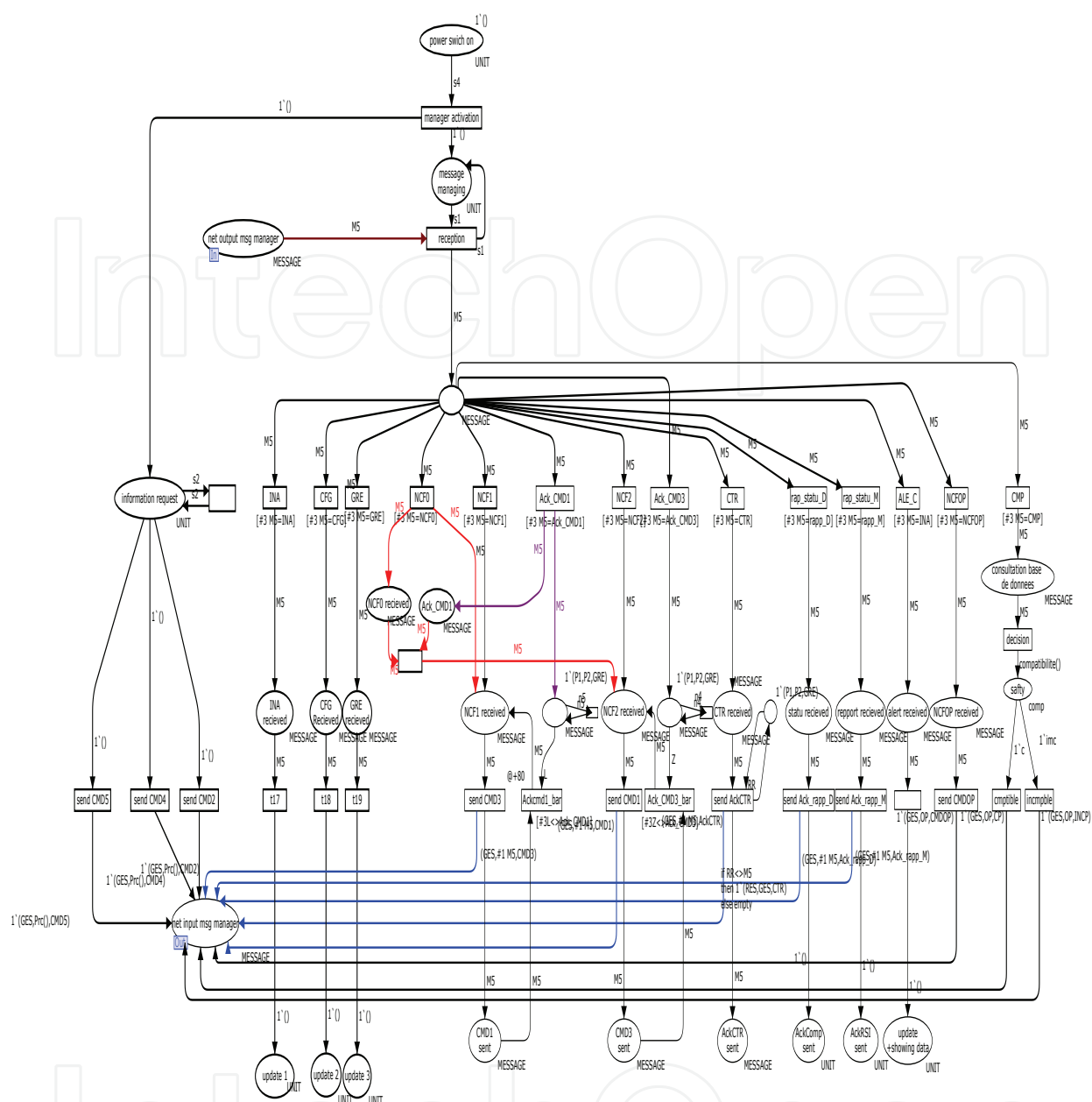


Fig. 15. Manager model

5.5 Network level

In this part, we will present the network's model where the sensor's products interact; firstly, we are going to model the hierarchical transition network which is represented by the figure 5 as a perfect network (without any disturbance) to evaluate the impact of progressive increasing of product's number existing in this network, and thereafter, we will create a disturbance in this network to check the robustness of allover the system.

5.5.1 Perfect network

The figure 16 indicates the lower level of the Network: the higher places Net input indicate the output's buffers of the product 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 products before being stored in the place “message sent through 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 products.

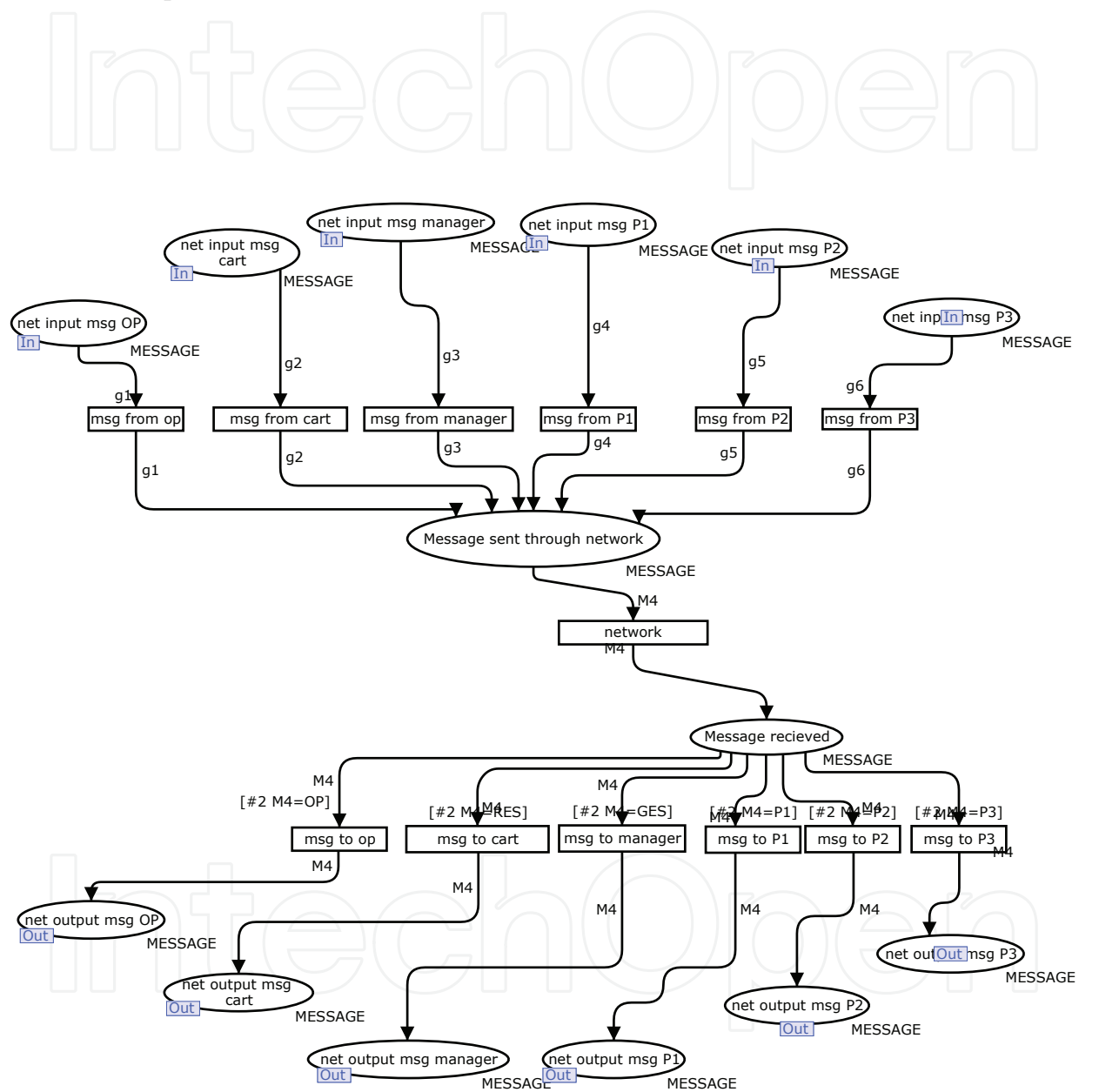


Fig. 16. Perfect Network model

5.5.2 Disturbed network

The network model presented on the figure 17 defines a disturbed network where there is a risk of loss of message. As figure indicates, each token (message), which is presented in the place (" message sent through network") of figure 16 must cross the transition where it will be to assign to another place, in this moment this token will be lost or passed, after this passage, this token enters a buffer of entry and afterwards enters a buffer of exit to be finally in the place " message received". This disturbance in network was presented by (Bitam et al., 2006) where they modeled a line of transmission with disturbance.

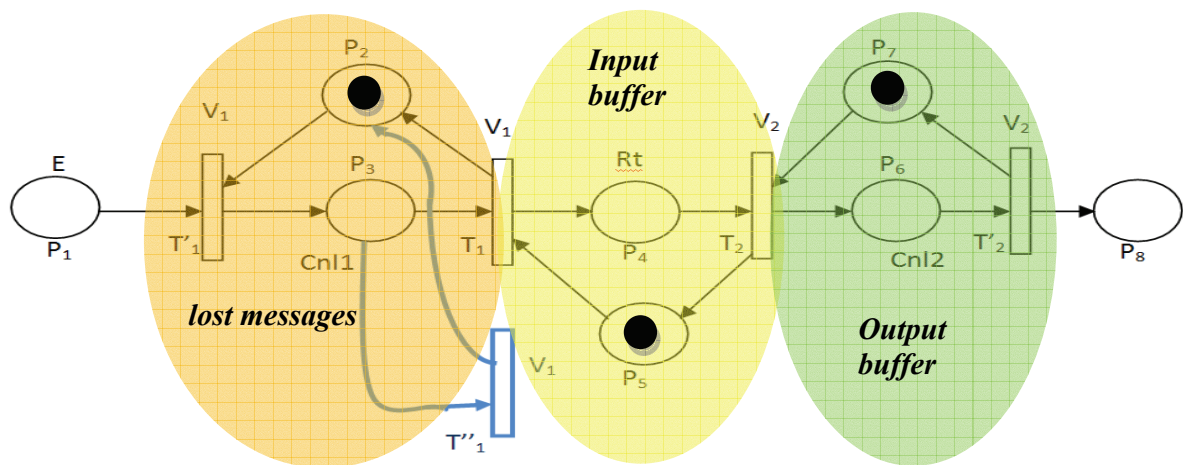


Fig. 17. Disturbed network model

This modeling of loss in a line of communication indicated in figure 17: when a message (token) is in place P_1 and another is in place P_2 , the transition T'_1 may fire (is passable), So this message will go to the place P_3 , at this moment there are two directions crossing T'_1 (this message will be lost) or passing of T_1 (the message will be issued) and after it passes to input buffer then that to output buffer (after crossing T_2). This phenomenon is explained as follows: at times the processing speed of an active product is much slower than receiving. So some messages will not get the chance to be treated by the active product of the limitation of the input place E.

The hierarchical model with different level is represented by figure 18. In this figure are shown different levels namely; Network level, product level, internal functions of an active product and message functions of an active product.

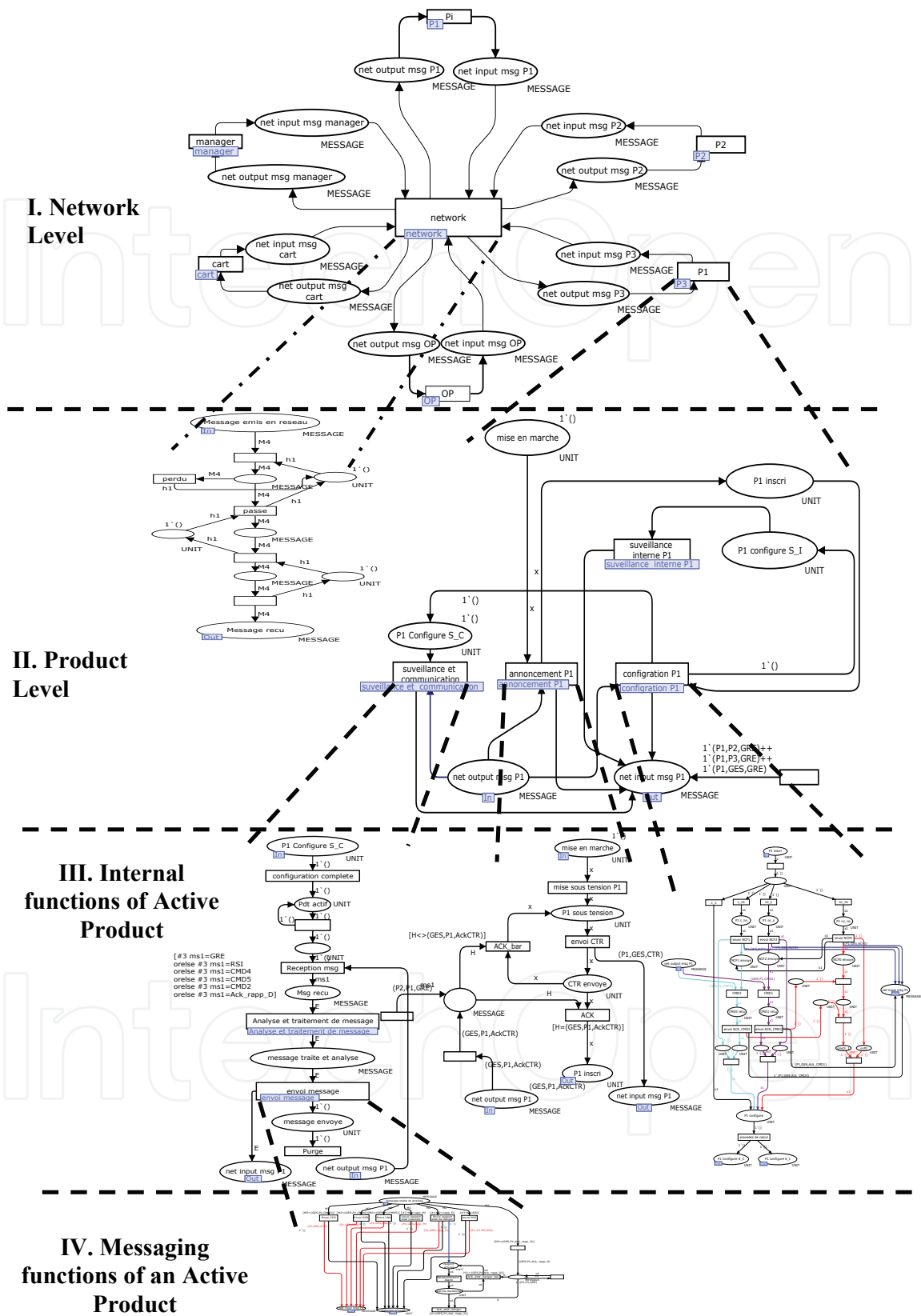


Fig. 18. Hierarchical models framework of an active product’s community

6. Conclusion

In this work, we define the concept of an active security management in a distributed system, with Hierarchical Petri nets modelling of active product's behaviour. The target application is dedicated to security management of hazardous products but the concept is extensible to other application areas.

We proposed an active product's behaviour model represented by hierarchical coloured Petri nets. This hierarchy includes sub-models where each one allows displaying the evolution of every state of the active product (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 active products is provided by exchange of messages in order to manage and control dynamically in real-time the global active security level. The proposed Active Product model will help in future steps of our work to study by large simulation the influence of the communication network on the system functioning (bounded time, messages loss, ...).

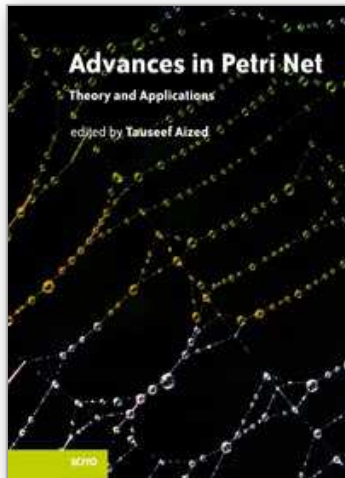
The proposed approach is very promising for the study and analysis of distributed system using the communicating object's approach or active product concept.

7. References

- Bouali, M.; Barger, P. & Schon, W. (2009). Colored Petri Net inversion for Backward Reachability Analysis, In: Second IFAC Workshop on Dependable Control of Discrete Systems, Bari, Italy.
- Brahimi, B. ; Rondeau, E. & Aubrun, C. (2008). Integrated Approach based on High Level Petri Nets for Evaluating Networked Control Systems, In: 16th Mediterranean Conference on Control and Automation, MED'08, Ajaccio, France.
- CoBIs (2008). Collaborative business items. European Community FP6 STREP Project, IST 004270, Technical report.
- Fu-Shiung, H. (2009). Developing cooperation mechanism for multi-agent systems with Petri nets. *Engineering Applications of Artificial Intelligence* vol. 22 (2009) pp. 616–627.
- Hsieh, F.S.(2004). Fault tolerant deadlock avoidance algorithm for assembly processes, *IEEE Transaction on System, Man and Cybernetics, Part A* 34 (1), 65–79
- Jensen, K. (1992 à 1997). Colored petri nets - basic concepts, analysis methods and practical use. Basic Concepts, EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Vol. 1 à 3, 1992 - 1997.
- Kashif S.; Norsheila F.; Sharifah H.; Sharifah K. & Rozeha R. (2009). Autonomously Intelligent WSN Routing Protocol based on Ant Colony Optimization. In: *Proceedings of 7th International Conference on Robotics, Vision, Signal Processing, & Power Applications (RoViSP 2009)*. Awana Porto Malai, Langkawi, Kedah, Malaysia.
- Khoukhi, L. & Cherkaoui, S. (2010). Intelligent QoS management for multimedia services support in wireless mobile ad hoc networks. *Journal of Computer Networks*, Elsevier Edition.
- Marsal, G. (2006). Evaluation of time performances of Ethernet-based Automation system by simulation of High Level Petri Nets, PhD dissertation de l'Ecole Normale Supérieure de Cachan et de l'Université de Kaiserslautern, Décembre 2006.

- Murata, T. (1989). Petri nets: properties, analysis and applications, *Proceedings of the IEEE* 77 (4), 541-580.
- Quanz, B. & Tsatsoulis, C. (2008). Determining Object Safety using a Multiagent Collaborative System, *Workshop at Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, Venice, Italy, October 20-24.
- Ratzer, A.V.; Wells, L.; Larsen, H.M.; Laursen, M.; Qvortrup, J.F.; Stissing, M.S.; Westergaard, M.; Christensen, S. & Jensen, K. (2003). Cpn-tools for editing, simulating, and analysing coloured petri net. *LNC*, 2679, pages 450- 462.
- Rudas, I.J. & Horvath, L. (1997). Modeling of manufacturing processes using a Petri net representation, *Engineering Applications of Artificial Intelligence* 10(3), 243-255.
- Song, C., Qi-Wei, G., Qian-Ming, S. & Qian, Z. (2008). Modeling and performance analysis of wireless sensor network systems using Petri nets. *The 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2008)*, July 6-9, 2008, Kaikyo Messe Shimonoseki, Shimonoseki City, Yamaguchi-Pref., Japan, pp 1689-1692
- Strohbach, M.; Kortuem, G. & Gellersen, H. (2005). Cooperative artefacts-a framework for embedding knowledge in realworld objects. *International Workshop on Smart Object Systems, UbiComp*, pages 91-99, Tokyo, Japan.
- Zouinkhi, A.; Bajic, E.; ZIDI, R.; Ben Gayed, M.; RONDEAU, E. & Abdelkrim, M.N. (2009). Petri Net Modeling of active products cooperation for active security management. *Sixth IEEE International Multi-Conference on System, Signals & Devices (SSD'09)*, Tunisia.
- Zurawski, R. (2005). Petri net models, functional abstractions, and reduction techniques: applications to the design of automated manufacturing systems. *IEEE Transactions on Industrial Electronics* 52 (2), 595-609.

IntechOpen



Advances in Petri Net Theory and Applications

Edited by Tauseef Aized

ISBN 978-953-307-108-4

Hard cover, 220 pages

Publisher Sciyo

Published online 27, September, 2010

Published in print edition September, 2010

The world is full of events which cause, end or affect other events. The study of these events, from a system point of view, is very important. Such systems are called discrete event dynamic systems and are of a subject of immense interest in a variety of disciplines, which range from telecommunication systems and transport systems to manufacturing systems and beyond. There has always been an intense need to formulate methods for modelling and analysis of discrete event dynamic systems. Petri net is a method which is based on a well-founded mathematical theory and has a wide application. This book is a collection of recent advances in theoretical and practical applications of the Petri net method and can be useful for both academia and industry related practitioners.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ahmed Zouinkhi, Eddy Bajic, Eric Rondeau and Mohamed Naceur Abdelkrim (2010). Petri Nets Hierarchical Modelling Framework of Active Products' Community, Advances in Petri Net Theory and Applications, Tauseef Aized (Ed.), ISBN: 978-953-307-108-4, InTech, Available from: <http://www.intechopen.com/books/advances-in-petri-net-theory-and-applications/petri-nets-hierarchical-modelling-framework-of-active-products-community>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen