To a Formal Modeling Approach of Error Recovery in Production Systems Based on Holonic Multi-Agent Systems Specification

Belhassen Mazigh[1*], Ahmed Hammed[1] and Vincent Hilaire[2]

[1] Research Unit MARS, Faculty of sciences, 5000, Monastir, Tunisia
[2] Laboratoire Systèmes Et Transport, Université de Technologie de Belfort-Montbéliard, Rue Thierry Mieg, 90010 Belfort cedex, France

e-mail: belhassen.mazigh@gmail.com, ahmed.hammed@yahoo.fr, vincent.hilaire@utbm.fr

Abstract. Production systems must be reconfigurable and endowed with methods and tools allowing an automatic recovery of expected and unexpected errors. In this domain, error recovery in production systems is always an open problem. Multi-agent systems seem to be adapted for representing this kind of problem in an attempt to arrive at a reconfigurable, adaptive, and "intelligent" manufacturing systems. The objective of this work consists in proposing a specification, modeling and structural analysis approach of error recovery production systems using holonic multi-agent systems and Object Petri Nets (OPN).

Keywords: Error Recovery, Holonic Multi-Agent Systems, Object Petri Nets, Production Systems, Structural Analysis.

1 INTRODUCTION

Researches on intelligent agents in the context of manufacturing have been mostly concentrated on the "production activities" e.g. scheduling, planning, processing and material handling. However the activities related to exception handling such as diagnostics and error recovery have received little attention. Our research aims to propose a holonic multi-agent based approach from the specification using ASPECS [1] until the formal modeling and structural check using object Petri nets of error recovery system.

After a brief presentation of the error recovery problem, an error recovery system context is presented. In section III, the statement of the problem is presented. A quick overview of the ASPECS process and modeling approach will be presented in section IV. The analysis and conception phase of the ASPECS process and their associated activities are then described in Section V. This section also presents the application of ASPECS activities to design error recovery system. Section VI presents the formal specification of the error recovery system with object Petri nets using the renew editor. Section VII consists in the application of error recovery model on the robotic cell for piston assembly example. Section VIII summarizes the results of the paper and describes some future work directions.

2 ERROR RECOVERY SYSTEM CONTEXT

One of the major problems in manufacturing systems is how to effectively recover from both anticipated and unanticipated faults. Traditional techniques have addressed the error recovery problem from the point of view of defining a set of actions for a known set of errors. In this context, the authors of [2] proposed an error recovery approach based on the integration of explanation-based and analogy-based learning. The knowledge representation for the proposed learning technique is based on the object-oriented modeling concept. Zhou and DiCesare [3] developed a formal description of four possible error recovery strategies in terms of Petri net constructs, namely input conditioning, backward error recovery, forward error recovery and alternate path recovery. The authors of [4] proposed a strategy for online recovering of the manufacturing operationality by using a multi-agent based decision support system with fuzzy reasoning.

In an attempt to investigate an “intelligent” manufacturing workstation controller, an approach integrating Petri net models and neural network techniques for preliminary diagnosis was undertaken [5]. More recent works pertain to addressing the issue of monitoring, diagnostics, and error recovery within the context of hierarchical multi-agent systems [6].

3 STATEMENT OF THE PROBLEM

The focus in this research is on recovery of physical error. The chosen approach is to design a reconfigurable, adaptive and intelligent manufacturing system. As such, a specification approach based on holonic multi-agent systems seems to be a promising approach to deal with the unpredictable nature of error due to their decentralization, autonomy, cooperation features and their hierarchical ability to react to unexpected situation. For this specification, we use the agent-oriented software process called ASPECS which is based on a holonic organizational metamodel and provides a step-by-step guide from requirements to code allowing the modeling of a system at different levels of details using a set of refinement methods. We use this methodology to identify holonic organization of a steady system and we propose a Petri Net approach for modeling and analysis specification associated to the case steady.

4 A QUICK OVERVIEW OF THE USED ASPECS PROCESS

As mentioned in [7], the ASPECS Process is a step-by-step requirement to code software engineering process based on a metamodel, which defines the main concepts for the proposed HMAS analysis, design and development [8][9].

The target scope for the proposed approach can be found in complex systems and especially hierarchical complex systems. The main vocation of ASPECS is towards the development of societies of holonic (as well as not holonic) multi-agent systems. ASPECS has been built by adopting the Model Driven Architecture (MDA) [10]. In Cossentino and al. [11] they label the three metamodels “domains” thus maintaining the link with the PASSI metamodel. The three definite fields are:

- The Problem Domain. It provides the organizational description of the problem independently of a specific solution. The concepts introduced in this domain are mainly used during the analysis phase and at the beginning of the design phase.
- The Agency Domain. It introduces agent-related concepts and provides a description of the holonic, multi-agent solution resulting from a refinement of the Problem Domain elements.
- The Solution Domain is related to the implementation of the solution on a specific platform. This domain is thus dependent on a particular implementation and deployment platform.

Our contribution will relate to the consolidation of the Problem Domain and the Agency Domain. We propose a formal specification approach for the analysis of the various organizations and the interactions between them facilitating therefore the Solution Domain.

5 HOLARCHY DESIGN OF ERROR RECOVERY SYSTEM (ERS)

In this section, we use the ASPECS methodology to describe partially the analysis phase, the design of the agent society and to propose a holonic structure of Error Recovery System (Fig.1).

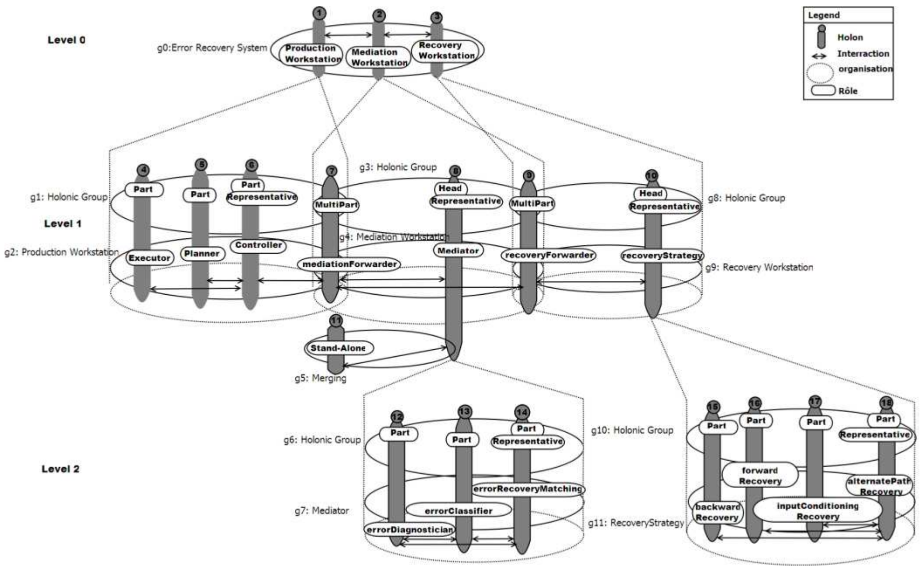

Fig. 1. Holarchy of Error Recovery System

At the level 0 of the Holarchy, three super-holons H1, H2 and H3 play respectively the role of the “Production Workstation”, ”Mediation Workstation” and “Recovery Workstation” in g0 group. H1 contains an instance of “Production Workstation” organization (g2), H2 contains an instance of “Mediation Workstation” (g4) and H3 contains an instance of “Recovery Workstation” (g9). The super-holon H8 of the “Mediation Workstation” Organization plays the role of the “mediator” in g4 group (Mediation Workstation). It contains an instance of “mediator” organization.

The super-holon H10 of the “Recovery Workstation” Organization plays the role of the “RecoveryStrategy” in g9 group (”Recovery Workstation”). It contains an instance of “RecoveryStrategy” organization. The holon H6 playing the role of the “Controller” in the “Production Workstation” organization, the two super-holons H8and H10, the holon H14 playing the role of the “errorRecoveryMatching” in the “mediation” organization and the holon H18 playing the role of the “alternatePathRecovery” in the “Recovery strategy” organization named Head and Representative of the other members. The holon H7 playing the role of the “mediationForwarder” and the holon H9 laying the role “recoveryForwarder” are named Multi-part as its shared their roles respectively between “Production Workstation” and “Mediation Workstation” organizations and ”Mediation Workstation” and “Recovery Worksation” organizations.

Concerning H11, it’s a Stand-Alone holon. In fact, when the holon H8 will diagnostic the error at the aim to classify it and to choose the adequate recovery policy, this error may be unanticipated and by consequent, the recovery method is unknown.

In this case, H11 is called to find a solution to this error, it can for example try all the recovery policy presented or request a sub-contracted service.

6 FORMAL MODELING

The focus in this section is on the formal modeling of the error recovery system. Petri nets are well suited for use in formal modeling of multi-agent systems. In this context, Mulan [12] (Multi-Agent Nets) is an architecture used for the multi-agent systems. It‘s built on Java and reference nets and can be executed in Renew [13]. However, there exists no released Version of Mulan that allows us to model multi-agent systems in this architecture.

On the other hand, OREDI [14] (ORganization EDItor), a Petri net based tool enables editing organization models as well as deploying such models as multi-agent systems. OREDI is built on top of Renew and relies on SONAR [15] (Self Organizing Net architecture reference), a formal organizational specification for electronic institutions based on Petri nets. However, OREDI has not been released yet.

For the formal modeling of our error recovery system, we use object Petri net with Renew editor. Renew is a high-level Petri net editor and simulator that achieves a seamless integration of Petri nets and the Java programming language. Petri nets can implement methods and can be treated as first-class Java objects. On the other hand, Java code can be accessed from nets easily.

Renew implements bidirectional synchronous channels and multiple instances of a single net, there, by providing powerful abstraction concepts. The simulation engine is capable of executing multiple transitions in separate threads. Renew is well suited for rapid prototyping and creating executable workflow models.

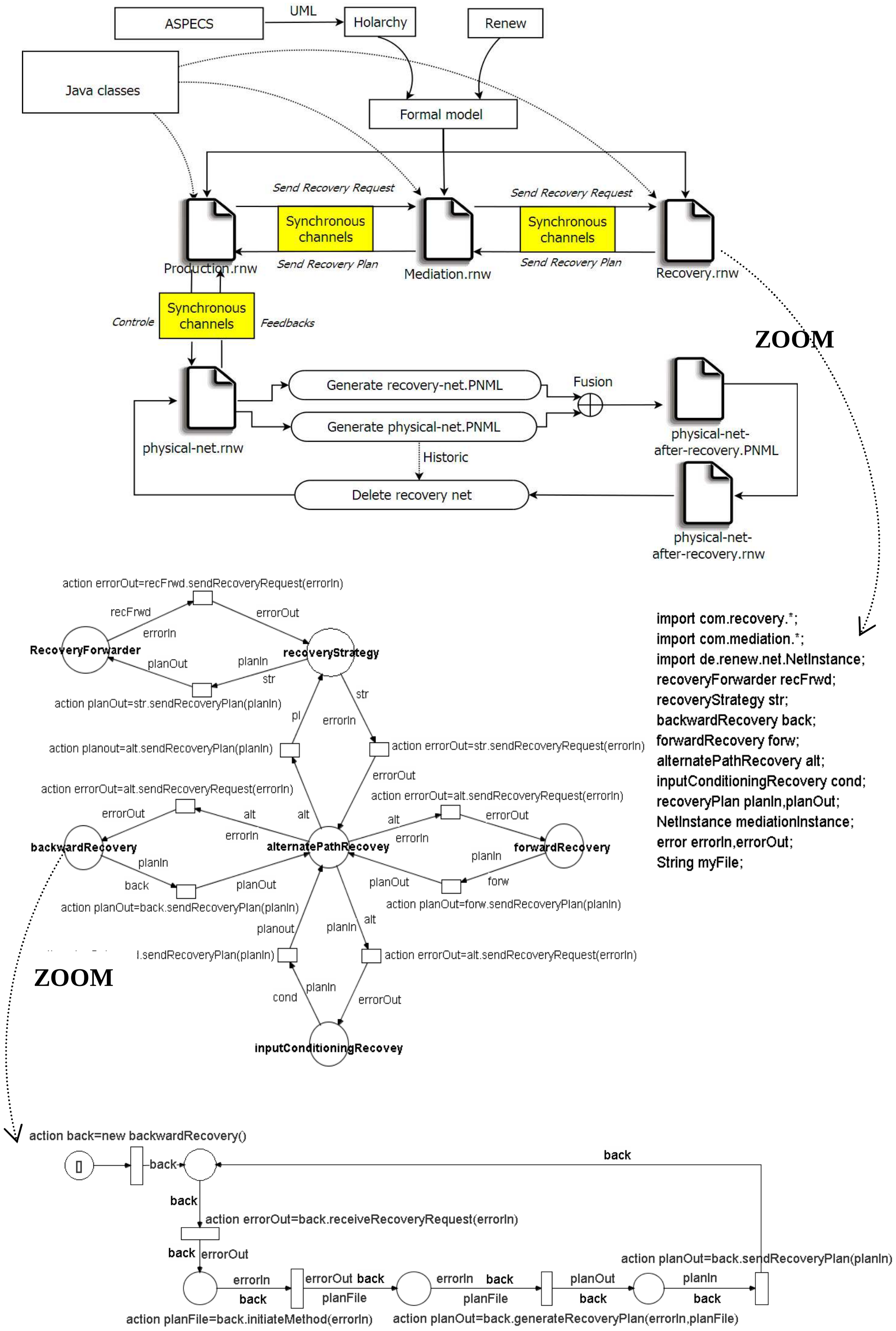

Fig. 2. Formal model of Error Recovery System

The formal model will be deduced from the holarchy presented in the previous section. In fact, the three base organizations: Production Workstation, Mediation Workstation and Recovery Workstation will be represented by three rnw files (rnw: extension of Renew files) and they will be described respectively by the Java packages Production, Mediation and Recovery. Agents are represented by Java classes of which the attributes are the characteristics of these agents and methods are actions which can be executed by these agents.

Classes belonging to the same package represent agents of the same organization. Interactions between organizations are modeled by synchronous channels implemented in Renew. The Fig. 2 describes our formal model for the error recovery system.

7 VERIFICATION AND VALIDATION OF THE MODEL

In this section, we try to verify and validate our formal model on a robotic cell for piston assembly workstation example taken from Zhou and DiCesare's book [3]. The workstation uses two robots to place and pull the pistons into the cylinders of an engine block and attaches the piston rods to the crank shaft.

Fig. 3 demonstrates necessary steps to apply our error recovery model on this example (robotic cell for piston assembly) from error detection to the recovery of this error and the return back of the system to his normal state.

In this example, we treat three types of error with three different recovery policies as mentioned by Zhou and DiCesare: piston-puller-tool-down error with input conditioning error recovery, out-of-tolerance error with alternate path error recovery and incorrect part orientation error using the forward error recovery method.

When an error is detected, only its specification and its error occurrence place are sent by the « Executor » agent to the « Controller » which assigns to this error an identifier "errorId" and forward it to the "Mediation Workstation".

The error is diagnosed by the "errorDiagnostician" agent, classified by the "errorClassified" agent and an error recovery policy is chosen by the "errorRecoveryMatching" agent. After that, the error is forwarded to the "Recovery Workstation" when a recovery plan is generated by the corresponding error recovery agent and sent to the "Production Workstation" via the "Mediation Workstation".

The "Planner" agent transforms the recovery plan to a PNML file and adds it to the PNML file of the physical system. Finally, the "Executor" agent generates a Petri net corresponding to the physical system and the recovery module which will be deleted when the recovery procedure is achieved.

Until achieving this work, Renew does not allow to make structures check, but it can generate the format PNML [16] (Petri Net Markup Language) which is a proposal of an XML-based interchange format for Petri nets.

For this, and since validation of object Petri nets and colored Petri nets are the same, the idea is to transform our Petri nets modules to ordinary Petri nets and to use Tina as Petri net editor that can make structural analysis (liveness, safeness, reversibility and boundedness) and support PNML format.

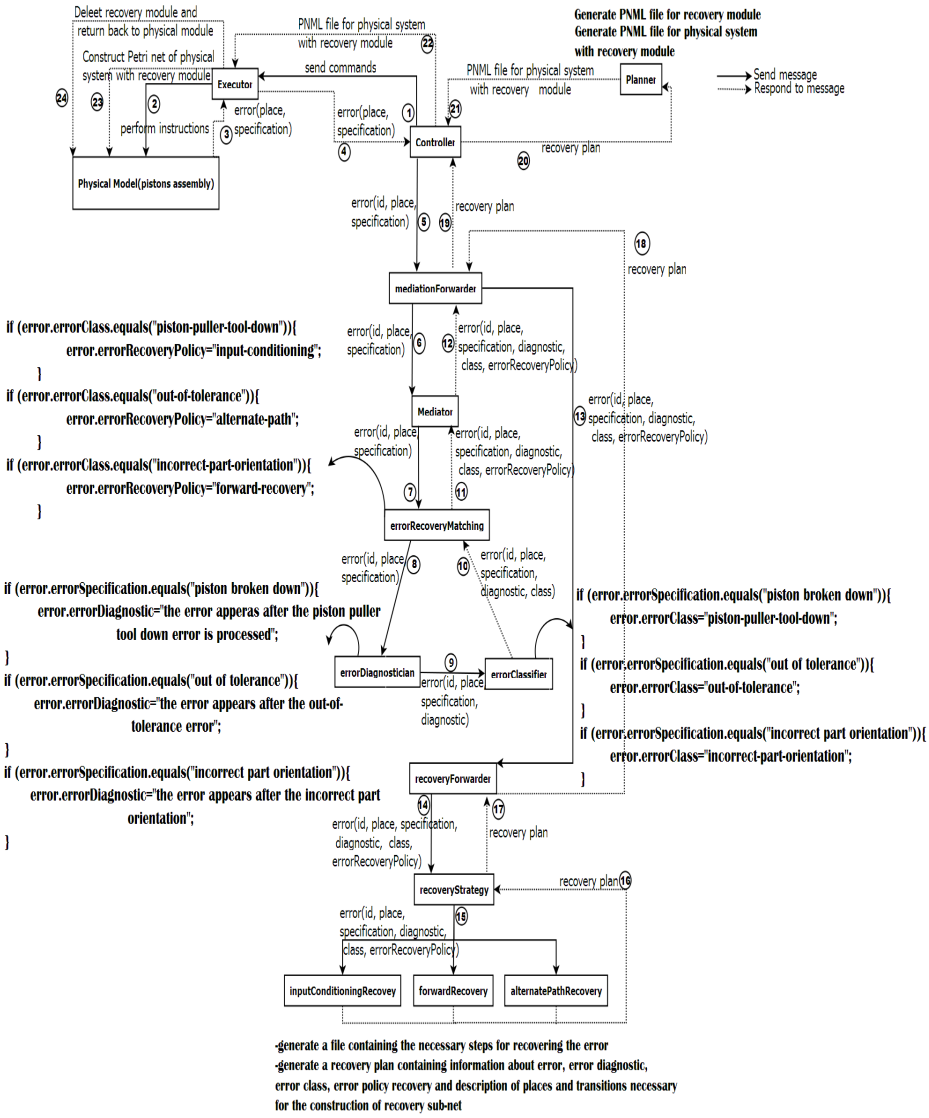

Fig. 3. The error recovery model for piston assembly workstation

The Fig. 4 represent a part of our formal model, it describes the Petri net model of the example physical system with the sub Petri net of error recovery module. The sub Petri net module describes an alternate path error recovery policy. It assumes that an out-of-tolerance condition for a nut is built up and detected in the place P16. The error recovery procedure consists of three transitions t13, t14 and t15 and three places: P21: M-1 discards the nuts, P22: Two nuts available and P23: M-1 picks up the nuts. This Petri net is live, bounded, reversible and safe. In fact, the analyze check of corresponding PNML file with Tina editor gives results described by Fig. 5:

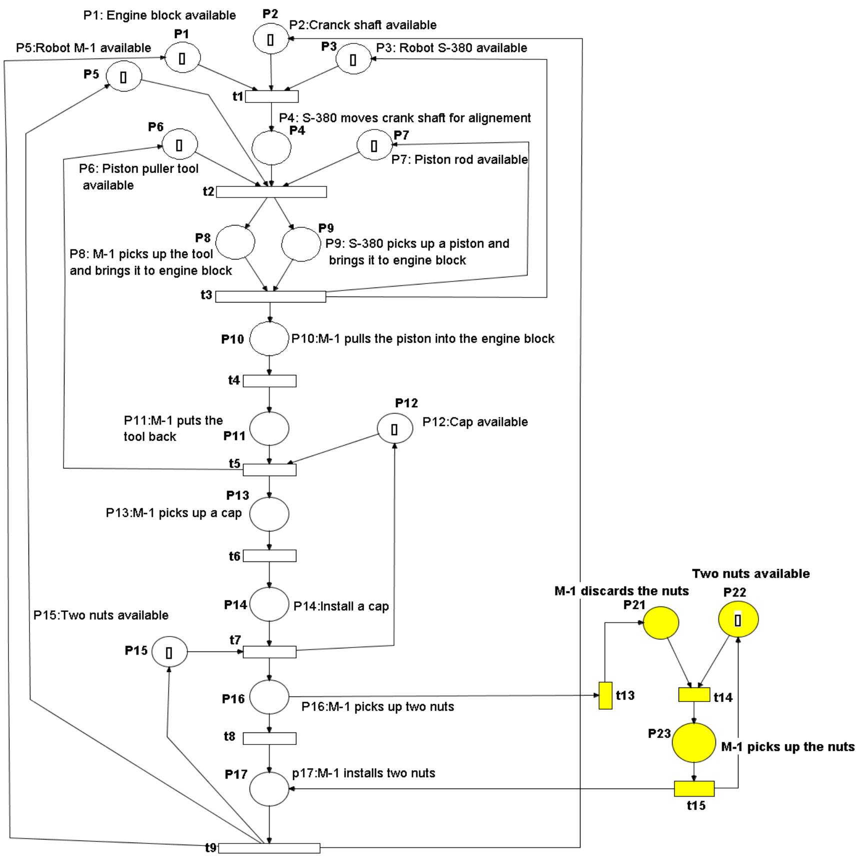

Fig. 4. Physical system model with error recovery module

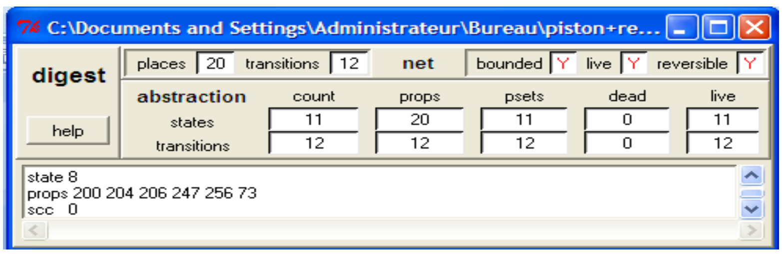

Fig. 5. Analyze check results of Physical system model with error recovery module

8 CONCLUSION

In this article, we presented an approach of a specification, modeling and structure check of error recovery system. This approach is based on ASPECS to design the holarchy of our system and Renew editor to construct a formal model corresponding to the holarchy using object Petri nets. Recently, a new plug in called Lola was added to Renew tool and it integrates verification capabilities of Petri nets. Although, it's an experimental plug in and ignores all java inscriptions, guards and synchronous channels. So, it's like the work that we have done with Tina. Our future works will focus on the implementation of a tool which could generate holonic multi-agent systems with objects Petri nets.

REFERENCES

1. http://www.aspecs.org
2. Chang, S.J., Goldbogen, G., Dicesare, F.: Automated error recovery in manufacturing systems through learning and reasoning, Proceedings of the 5th IEEE International Symposium on Digital Object Identifier. 10.1109/ISIC.1990.128578 (1990).
3. Zhou, M., DiCesar, F.: Petri Net Synthesis for Discrete Event Control of Manufacturing Systems, Kluwer Academic Publishers. USA (1993).
4. Ulieru, M., Norrie, D.: A multi-agent system for on-line fault recovery of intelligent manufacturing systems, Fuzzy Information Processing Society. NAFIPS. 18th International Conference of the North American on Digital Object Identifier. 10.1109/NAFIPS.1999.781814 (1999).
5. Ma, Y., Odrey, N.: On the application of Neural Networks to a Petri net –based intelligent workstation controller for manufacturing, Proceedings of the Artificial Neural Networks in Engineering (ANNIE '96) conference, pp. 829-836, Vol. 6, St. Louis, MO, ASME Press, NY (1996)
6. Odrey, N.: Error Recovery in Production Systems: A Petri Net Based Intelligent System Approach, in Petri Net, Theory and Applications, Book edited by: Vedran Kordic, ISBN 978-3-902613-12-7, pp. 534, I-Tech Education and Publishing, Vienna, Austria (2008)
7. Mazigh, B., Hilaire, V., Koukam, A.: Formal Specification of Holonic Multi-agent Systems: Application to Distributed Maintenance Company, in proceeding, 9th International Conference on Practical Applications of Agents and Multiagent Systems (2011)
8. Gaud, N. : Systèmes Multi-Agent Holoniques : de l'analyse à l'implantation, PhD thesis, Université de Technologie de Belfort-Montbéliard (2007)
9. Cossentino, M., Gaud, N., Hilaire, V., S. Galland, S., Koukam, A.: A holonic metamodel for agent-oriented analysis and design, 3rd International Conference on Industrial Applications of Holonic and Multi-Agent Systems in LNAI, 4659, Springer-Verlag (2007)
10. Object Management Group. MDA guide, v1.0.1, OMG/2003-06-01 (2003)
11. Cossentino, M., Gaud, N., Hilaire, V., S. Galland, S., Koukam, A.: ASPECS: an agent-oriented software process for engineering complex systems How to design agent societies under a holonic perspective, Auton Agent Multi-Agent Syst 20:260–304 (2010)
12. Koehler, M., Moldt, D, Roelke, H.: Modeling the structure and behaviour of Petri net agents. In Proceedings of the 22nd International Conference on Application and Theory of Petri Nets (ICATPN'01).LNCS Vol. 2075. Springer, Berlin, pp 224–241 (2001)
13. Kummer, O., Wienberg, F., Duvigneau, F.: Renew – User Guide," University of Hamburg, Computer Science Department, Vogt-Kölln Str. 30, 22527 Hamburg, Deutschland, 1.6 edition (2002)
14. Delui, E., Kohler, M.: SONAR/OREDI: A tool for creation and deployment of organization models, Germany (2010)
15. Kohler, M.: A formal model of multi-agent organizations, Fundamenta Informaticae 79, 415-430 (2007)
16. http://www.pnml.org