CPNT-Jade Framework
Developing Agent-based Control of Discrete Event Systems

Andrzej Bożek and Marian Wysocki
Department of Computer and Control Engineering, Rzeszow University of Technology, al. Powstańców Warszawy 12, Rzeszów, Poland

Keywords: Multi-Agent Systems, Jade, CPN Tools, Colored Petri Nets, HTCPN.

Abstract: A framework built as a combination of CPN Tools software and JADE (Java Agent DEvelopment Framework) platform has been proposed. CPN Tools makes it possible to model a controlled system with the use of hierarchical timed colored Petri nets (HTCPN), a powerful formalism for modeling complex discrete event systems. The framework supports real-time mode of simulation of the target system model. Issues of communication and time synchronization between the two software components are transparent to a user. An example of the framework use has been presented. It refers to a dynamic flexible job shop manufacturing system for which an agent-based control has been developed.

1 INTRODUCTION

Multi-agent systems represent important kind of intelligent software architecture characterized by distributed implementation, mobility and scalability. A multi-agent control system consists of controlled (sub)system, e.g. a business process, a factory plant, road traffic, etc., and an agent-based software (sub)system (Candido and Barata, 2007). A development phase is needed for every software system. In this stage new features are added and tested. Therefore, a connection between the developed control system and the controlled system, real or simulated, is required.

The presented solution refers to the scenario where a computational model of the controlled system is used. There exist many formalisms and software tools for simulation of continuous or discrete dynamic systems. In this work the discrete event systems are the focus of interest. Petri net formalism with high-level extensions has been selected for the framework implementation as the powerful tool for modeling and simulation of discrete event systems.

The framework has been composed of two parts. The first part simulates discrete system with the use of a Petri net executable model. The second part is a multi-agent system that performs a control algorithm. The framework provides transparent bidirectional communication between the two subsystems. The discrete system model and the multi-agent control system are built during the development phase. After that, the developed agent-based control system is ready to be reconnected to a real-world target environment.

An exemplary solution based on the framework has also been presented in this work. It refers to a control of production process in a screw factory that has a form of a dynamic flexible job shop problem.

2 RELATED WORK

Many frameworks and generalized architectures related to multi-agent systems have been proposed. There exist multi-agent frameworks dedicated for specific domains. Albashiri and Coenen (2009) have proposed EMADS framework (Extensible Multi-Agent Data Mining System). Vokřínek, Komenda and Pěchouček (2011) have developed abstract architecture of a task-oriented multi-agent problem solver. MABC (Multi-Agent Based Clustering) framework has been developed by Chaimontree, Atkinson and Coenen (2011). Hu, Du and Spencer (2011) have developed Aframe framework for Ambient Systems. An agent-based framework that supports implementation of IPPS system (Integrated Process Planning and Scheduling) has been proposed by Li, Zhang, Gao, Li and Shao (2010). FAST (Flexible and Adaptive Scheduling Tool) is a toolkit

Bożek A. and Wysocki M..
CPNT-Jade Framework - Developing Agent-based Control of Discrete Event Systems.
DOI: 10.5220/0004218001840189
In *Proceedings of the 5th International Conference on Agents and Artificial Intelligence* (ICAART-2013), pages 184-189
ISBN: 978-989-8565-38-9

that provides a powerful agent-based framework to develop flexible, fault tolerant and scalable scheduling systems (Garcia, Valero, Argente, Giret and Julian, 2008). Liu, Abdelrahman and Ramaswamy (2007) have proposed a multi-agent framework for robust and adaptable dynamic job shop scheduling. Using of multi-agent solution in manufacturing process planning, scheduling and control has become more and more common recently. The work of Shen, Wang and Hao (2006) is the good state-of-the-art survey on this subject.

The Petri net formalism is also used for development of multi-agent systems. Miranda and Perkusich (1999) have presented the modeling, analysis and verification of MATHEMA multi-agent system by means of colored Petri nets. Weyns and Holvoet (2002) have proposed using colored Petri net for modeling and analysis of an entire multi-agent system. They have chosen Design/CPN (predecessor of CPN Tools) software and *Packet-World* as a case study. The same authors (Weyns and Holvoet, 2004) have developed a colored Petri net dedicated for regional synchronization in a multi-agent system.

The common feature of the three aforementioned works is that the Petri net formalism has been used for modeling and analysis of multi-agent systems. In this work the Petri net formalism has been used for modeling and simulation of an external environment of the multi-agent system that substitutes a target environment during the development phase and does not need to be implemented as a multi-agent structure.

3 FRAMEWORK ARCHITECTURE

3.1 Components of the Framework

The framework is based on two software packages. The first one is CPN Tools (CPN Tools Home Page, 2012), the environment for modeling, simulation, analysis and verification of discrete event systems with the use of hierarchical timed colored Petri nets (HTCPN) formalism (Jensen and Kristensen, 2009). The most important reasons of this selection are:

- Petri nets are convenient formalism for structural representation of typical features and behaviors of discrete event systems, e.g. concurrency, synchronization, resources sharing, mutual exclusion.
- The high-level extensions are very useful. The hierarchy allows to decompose a structure of a modeled system into nested modules. Colored tokens carry information that can be represented by complex data types and can be processed by arc expressions. The time extension allows to model time relations between discrete events.

The second package is JADE (JADE Home Page, 2012). The most important reasons of JADE platform selection are:

- Java is classified as one of the most portable programming languages. A multi-agent system can be developed on one platform and then can be moved to another target platform. The paradigm of agents mobility is also well supported by the Java language portability.
- JADE is compliant with FIPA (The Foundation for Intelligent Physical Agents) (FIPA Home Page, 2012) specifications and supplies implementations of standards defined by this specifications.

Both of the software packages are free and open source. Because of the names CPN Tools and JADE, the proposed framework has been named as CPNT-JADE framework.

3.2 Framework Structure

The framework structure is presented in Figure 1. Both Petri net structure components and multi-agent system components are divided into two sets, namely framework components and solution components. Framework components constitute the base part of the framework that cannot be changed. The solution components are project-related elements.

CPN Tools and JADE are connected by TCP/IP. Information frames exchanged between applications have simple but usable form of list of messages. Each message is a pair `(address, data)`. On JADE side, there are five agents that have been predefined for CPNT-JADE framework. The agents are described in Table 1. A developer that uses the framework has to implement class of *Configurator* agent and class of *DispatchingPolicy* object. *DispatchingPolicy* object defines how to translate each message received from CPN Tools into an instance of FIPA compliant *ACLMessage* routed to other agents and vice versa.

The framework supports the real-time mode of simulation of a target system model. This mode is provided by *Real-time Mode Subsystem*. In the real-

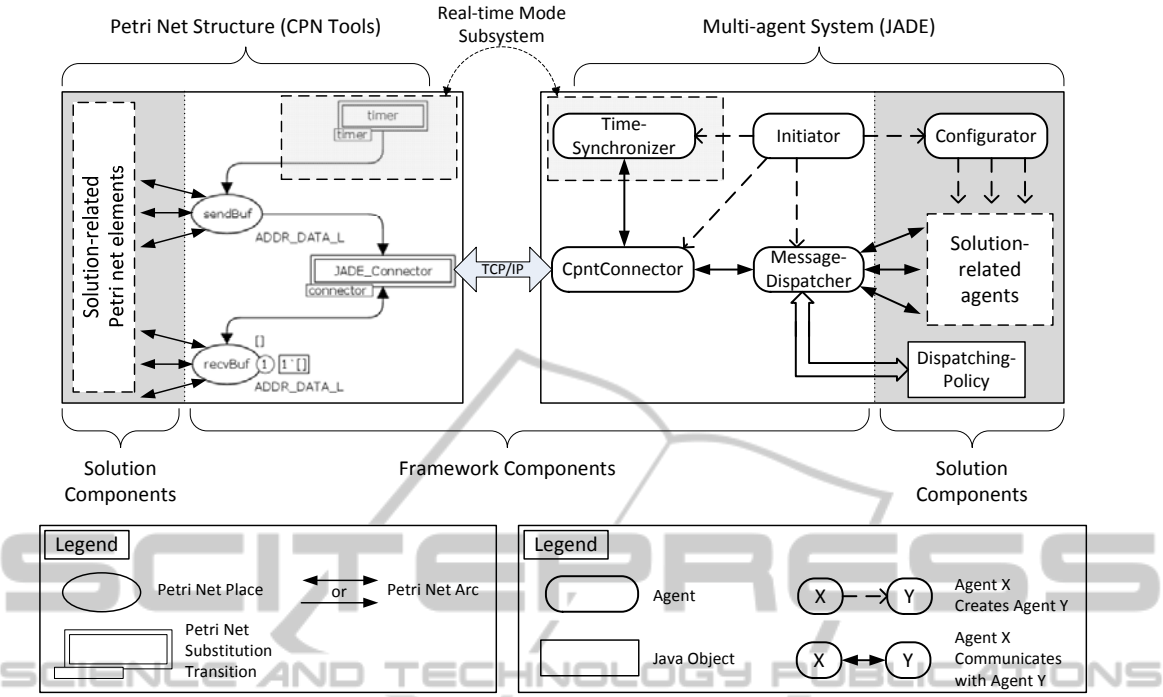

Figure 1: The structure of the CPNT-JADE framework.

time mode timestamps of HTCPN model are mapped into real time intervals. Thus, model parameters include data that defines locations and relations of events in real time axis.

Table 1: The predefined agents of the CPNT-JADE framework.

Agent class name	Description
Initiator	Created by JADE. It is responsible for creation other predefined agents.
Cpnt-Connector	Performs low-level TCP/IP communication with CPN Tools and translates data between low-level frames and high-level messages.
Message-Dispatcher	Redirects messages between *CpntConnector* and other agents. Routes are defined on the basis of messages addresses and rules defined by *DispatchingPolicy* object.
Time-Synchronizer	Optional, needed if the framework runs in the real-time mode. Details are given in subsection 3.3.
Configurator	Created by the framework but its behavior is solution-specific and has to be defined by a user. This agent creates and administers all solution-related parts of the multi-agent system.

4 EXAMPLE – DEVELOPING A PRODUCTION CONTROL SYSTEM

4.1 Problem Formulation

As an example solution, a prototypical production control system is presented. Organization of shop-floor level production in the factory can be classified as a kind of *dynamic flexible job shop manufacturing system* (Rajabinasab and Mansour, 2011), with detailed characteristics as follows:

- Jobs arrive continuously in time. No parameters of a job are known until it arrives.
- Arrival (release) date r_k and due date d_k are defined for each job J_k.
- A non-empty sequence of operations ($O_{1,k}$, $O_{2,k}$, ..., $O_{o(k),k}$) is defined for each job J_k.
- A non-empty set of machines $\{M_{1,i,k}, M_{2,i,k}, ..., M_{m(i,k),i,k}\}$ is defined for each operation $O_{i,k}$, any machine can alternatively execute the operation.
- A processing time $p_{m,i,k}$ is defined for each pair machine-operation $O_{m,i,k}$.
- Each machine can perform at most one operation at a time.

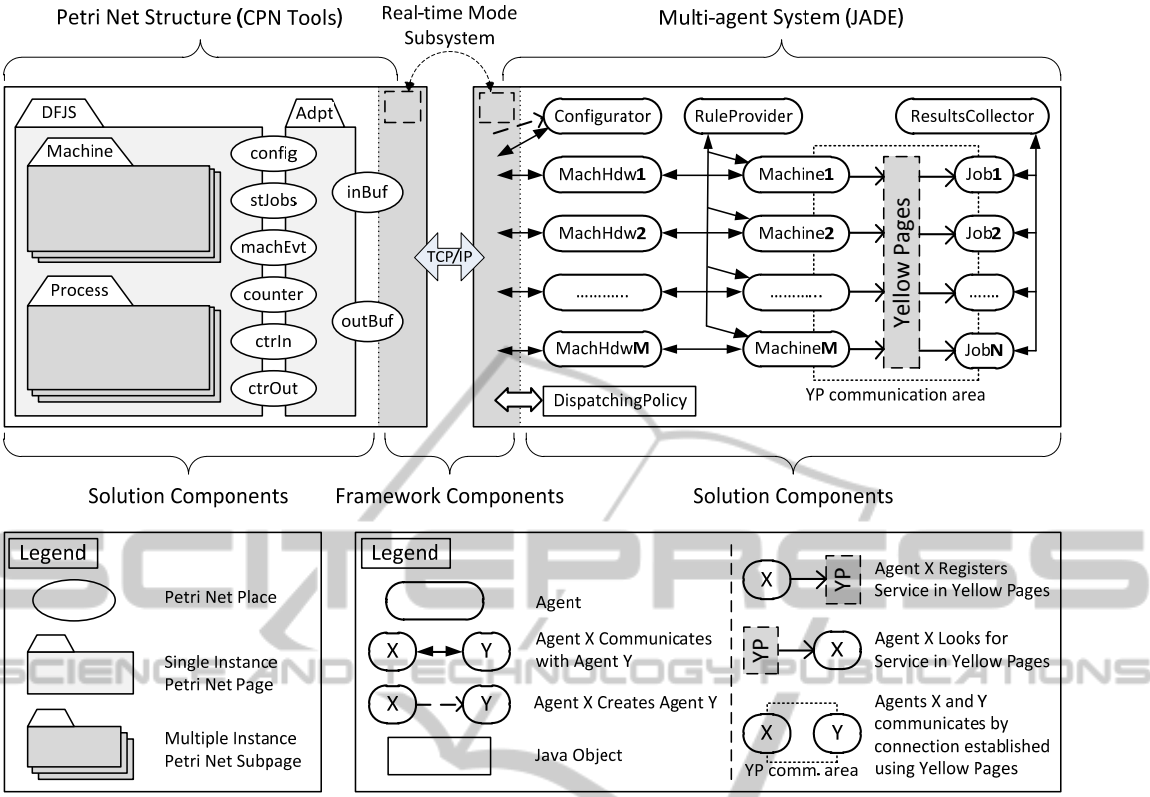

Figure 2: The structure of the prototypical dynamic production scheduling system.

The production in the screw factory is partially automated. Machines are equipped with embedded PC controllers (Żabiński and Mączka, 2012). These controllers are connected in a network and each one can be a base for a container of a multi-agent platform.

4.2 HTCPN Plant Model

The structure of the prototype production control system is presented in Figure 2. The left part represents a Petri net plant model based on the HTCPN formalism. The model consists of four pages (substructures): *Adpt* (*JADE Adapter*), *DFJS* (*Dynamic Flexible Job Shop*), *Machine*, *Process*. *DFJS* is the main page of the model. It models dynamic flexible job shop system structure on high-level. The *Machine* subpage represents features of a real machine important for the prototyped solution. Its implementation is presented in Figure 3. The modelled properties are as follows:

- Progress of operations is simulated (place *counter*).
- A machine can be stopped over arbitrarily defined time intervals (place *brkdwns*).
- Start/stop/breakdown events are registered (place *machEvt*).
- The multi-agent subsystem can control a queue of processed operations (place *ctrIn*).

In the real production environment PLC controllers have a connection with item counters and the progress is delimited by comparison of the numbers of produced and planned items. PLC controllers also register start/stop/breakdown events in the system.

4.3 Agent-based Control System

The agent-based control system is presented in the right part of Figure 2. *Configurator* agent creates all other solution-related agents, but this is not shown in Figure 2 to keep it readable.

There are M *MachHdw* (*Machine Hardware*) and *Machine* agents in the system, one pair of the agents for each machine. There are N *Job* agents in the system, where N is the number of currently processed jobs. The control is performed by choosing one operation in advance for each machine.

Selecting of operations for processing is main behavior of the *Machine* agent. When a discrete state of the production system changes, i.e. a new job

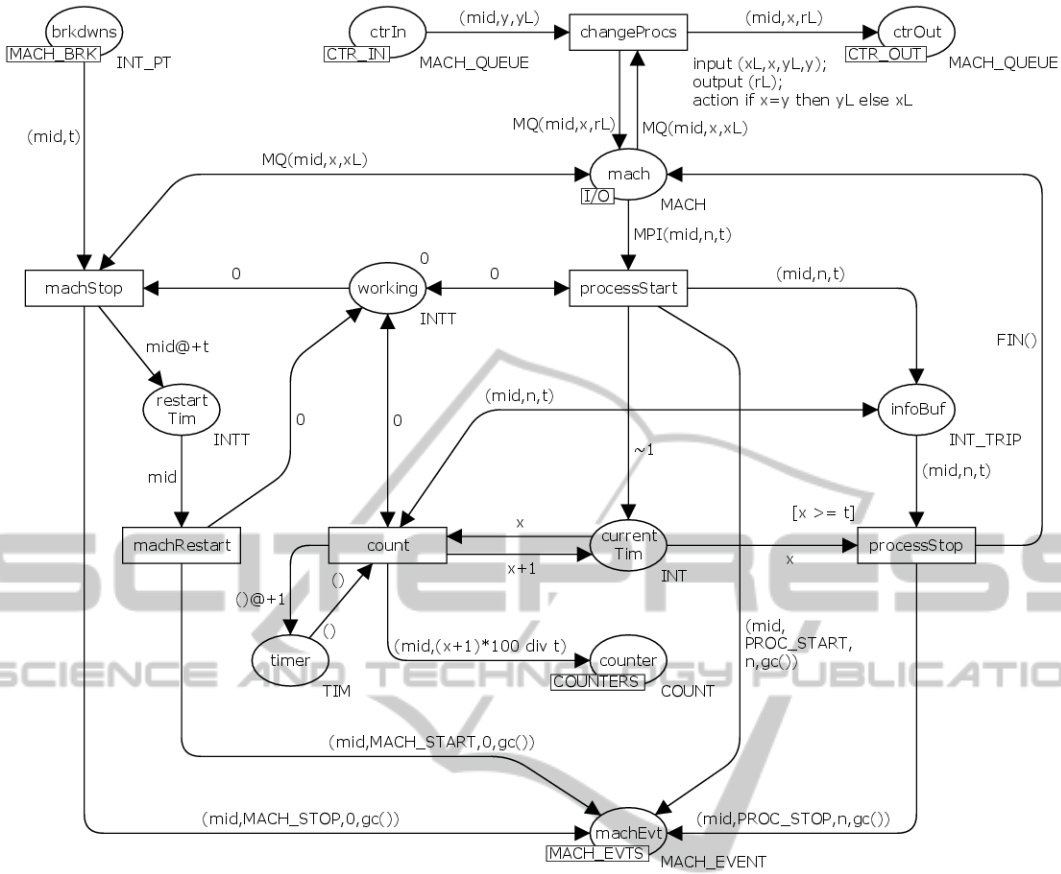

Figure 3: The machine subpage of the HTCPN model of the production system.

arrives or some operation finishes, a special message is broadcasted to all *Machine* agents. After receiving this message, each *Machine* agent checks whether it needs to reserve a next operation for processing. If so, the agent starts the *JobSelection* interaction protocol.

The sequence diagram of this protocol is presented in Figure 4. The *JobSelection* protocol has been patterned after FIPA-CNIP (*contract net interaction protocol*) (FIPA Home Page, 2012). A scheduling decision is made by *Machine* agent after receiving *operationParameters* messages from all asked *Job* agents. The parameters represent expected start and stop times of waiting operations. The *Machine* agent compares parameters of waiting operations and selects at most one operation. The *RuleProvider* agent stores selection criteria. For tests, several typical scheduling dispatching rules have been implemented: first-come first-served (FCFS), earliest arrival date (EAD), earliest due date (EDD) and critical ratio (CR - ratio of the remaining processing time to the allowance time of a job).

When a machine breaks down, an operation reserved for this machine is removed from the processing queue and marked as unreserved by related *Job* agent. Thus, it is possible that the operation will be reserved and processed by other machine. It is a form of automatic rescheduling.

5 CONCLUSIONS

The main goal of the presented work is to propose a solution that can be helpful in developing agent-based control systems designed for real-world applications. A user of the framework should at first build a model of a controlled system using HTCPN formalism. Then, the user can develop a multi-agent control system with the use of JADE. When the control system is fully ready or safe enough it can be excluded from the framework and moved to a target environment.

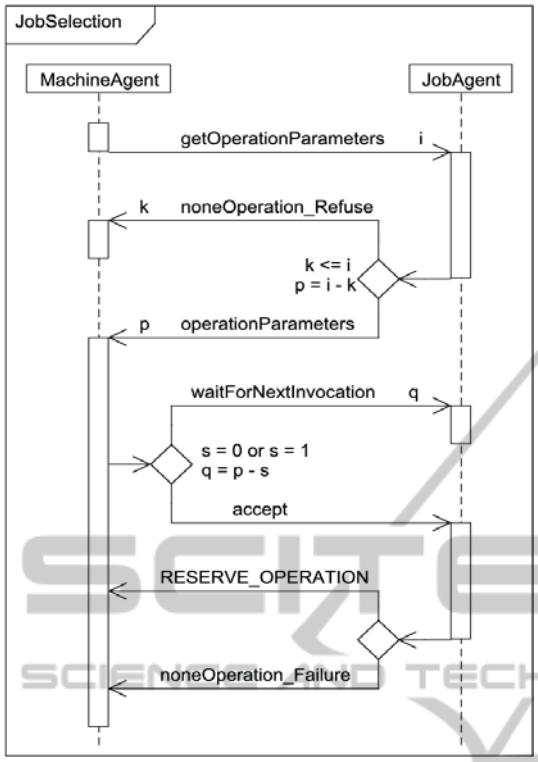

Figure 4: The sequence diagram of the JobSelection interaction protocol.

The prototypical agent-based control system, presented as an example, is a subject of continuous development and improvement. The HTCPN model is intended to be augmented with additional features, as representation of setup and transport times, limitation of resources, etc. New functionalities are going to be added to the control system, among other things, a possibility of interactive changing of operation sequences by machine operators, scheduling many operations in advance for each machine, cooperation with a long-term planner.

The work has been supported by the program of the Polish Ministry of Science and Higher Education for young scientists (U-8603/DS/M).

REFERENCES

Albashiri, K., A., Coenen, F. (2009). A Generic and Extensible Multi-Agent Data Mining Framework. *Hybrid Artificial Intelligence Systems. LNCS*, 5572, 203-210.

Candido, G., Barata, J. (2007). A Multiagent Control System for Floor Assembly. *HoloMAS 2007*, 4659(9-12), 293-302.

Chaimontree, S., Atkinson, K., Coenen, F. (2011). A framework for Multi-Agent Based Clustering. *Auton Agent Multi-Agent Syst*, 25(3), 425-446.

CPN Tools Home Page (2012). Retrieved November 5, 2012, from http://cpntools.org

Miranda, M., C., Perkusich, A. (1999). Modeling and Analysis of a Multi-Agent System using Colored Petri Nets, *Workshop on Applications of Petri Nets to Intelligent System Development*. Williamsburg.

FIPA Home Page (2012). Retrieved November 5, 2012, from http://www.fipa.org

Garcia, M., E., Valero, S., Argente, E., Giret, A., Julian, V. (2008). A FAST Method to Achieve Flexible Production Programming Systems. *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, 38(2), 242-252.

Hu, X., Du, W., Spencer, B. (2011). A Multi-Agent Framework for Ambient Systems Development. *The 2nd International Conference on Ambient Systems, Networks and Technologies. Procedia Computer Science*, 5, 82-89.

JADE Home Page (2012). Retrieved November 5, 2012, from http://jade.tilab.com

Jensen, K., Kristensen, L.M. (2009). *Coloured Petri Nets. Modelling and Validation of Concurrent Systems*. Berlin: Springer.

Li, X., Zhang, Ch., Gao, L., Li, W., Shao, X. (2010). An agent-based approach for integrated process planning and scheduling. *Experts Systems with Applications*, 37(2), 1256-1264.

Liu, N., Abdelrahman, M., A., Ramaswamy, S. (2007). A Complete Multiagent Framework for Robust and Adaptable Dynamic Job Shop Scheduling. *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, 37(5), 904-916.

Rajabinasab, A., Mansour S. (2011). Dynamic flexible job shop scheduling with alternative process plans: an agent-based approach. *Int. J. Adv. Manuf. Technol.* 54, 1091-1107.

Shen, W., Wang, L., Hao, Q. (2006). Agent-Based Distributed Manufacturing Process Planning and Scheduling: A State-of-the-Art Survey. *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, 36(4), 563-577

Vokřínek, J., Komenda, A., Pěchouček, M. (2011). Abstract Architecture of Task-oriented Multi-agent Problem Solving. *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, 41(1), 31-41.

Weyns, D., Holvoet, T. (2002). A Colored Petri Net for a Multi-Agent Application. *Second Workshop on Modelling of Components, Objects and Agents, MOCA'02*. Aarhus.

Weyns, D., Holvoet, T. (2004). A Colored Petri Net for Regional Synchronization in Situated Multi-Agent Systems. *Proceedings of First International Workshop on Petri Nets and Coordination*. Bologna.

Żabiński, T., Mączka, T. (2012). Implementation of Human-System Interface for Manufacturing Organizations. *Advances in Intelligent and Soft Computing*, 98, 13-31.