DOI:10.2298/CSIS111223031P

Modeling a Holonic Agent based Solution by Petri Nets*

Carlos Pascal and Doru Panescu

“Gheorghe Asachi” Technical University of Iasi,
Department of Automatic Control and Applied Informatics,
Blvd. Prof. Dr. Doc. Dimitrie Mangeron 27, Iasi 700050, Romania
{cpascal, dorup}@ac.tuiasi.ro

Abstract. One of the key design issues for distributed systems is to find proper planning and coordination mechanisms when knowledge and decision capabilities are spread along the system. This contribution refers holonic manufacturing execution systems and highlights the way a proper modeling method – Petri nets – makes evident certain problems that can appear when agents have to simultaneously treat more goals. According to holonic organization the planning phase is mainly dependent on finding an appropriate resource allocation mechanism. The type of weakness is established by means of the proposed Petri net models and further proved by simulation experiments. A solution to make the holonic scheme avoid a failure in resource allocation is mentioned, too.

Keywords: HMES, Petri nets, multiagent systems, planning, resource allocation.

1. Introduction

Design and implementation of appropriate mechanisms to control manufacturing execution systems are still open problems, and research in Artificial Intelligence (AI) has had an important impact for these subjects [2-4]. With respect to this, an example is the holonic approach, which is considered in this paper; it combines benefits of hierarchical and heterarchical manufacturing control architectures. Holonic Manufacturing Execution Systems (HMESs) are clearly influenced by planning and coordination mechanisms established in AI, primarily in the field of multi-agent systems [3-6]. An HMES regards a control scheme for the shop-floor level of a manufacturing company that is developed around autonomous, co-operative, intelligent entities, named holons. The most often used holonic

* This is a revised and extended version of a paper originally presented at the ICSTCC 2011, Sinaia, Romania [1].

taxonomy is derived from the PROSA reference architecture [7]. This takes into account four types of holons: product, resource, order and staff.

Without giving all details of the proposed holonic scheme construction, this paper aims at showing how an appropriate modeling and analysis method can reveal certain problems for the HMES functioning. Namely, due to the distributed nature of control within HMES, special coordination mechanisms are needed. Though some protocols from multi-agent systems can be considered, these can determine definite drawbacks for manufacturing control systems, requiring an appropriate tuning.

Regarding our paper organization, after presenting some related works, a generic structure of a holon and a basic Petri net model for the holonic decisional component, the holonic agent, are discussed. Then, the Petri nets modeling the inter-holonic communication are presented. About the internal holonic agent operation, this comprises two distinct phases: planning and execution. These are also modeled by Petri nets, which are used to reveal certain drawbacks possible to appear during the planning stage. The theoretical points are illustrated through experiments that were conducted by means of a complex HMES model, obtained as a coloured Petri net. A solution to eliminate the holonic faulty operation is sketched, too.

2. Related Work

While modeling of automated manufacturing systems by different classes of Petri nets is described and commented in a great number of papers and concentrated in some books [8, 9], fewer articles are dedicated to the use of Petri nets for HMES modeling. Nevertheless, the interest for implying Petri nets in holonic and multiagent systems modeling is justified, as they represent a powerful tool for dealing with concurrent processes, which is the case of HMESs. Thus, in [10] some Petri net models of holons were proposed in order to explain the holonic interaction mechanism in PROSA. These models are specific, being provided for certain types of holons (resource, order) and for different kinds of interactions. They highlight some aspects regarding cooperation (synchronization of holons, progress of parallel activities), being restricted to the relation between two holons, mainly between an order and a resource holon. Some benefits are got by the Petri nets application, as these can describe the structure of PROSA holonic components, conducting to logical and temporal analysis of their behavior. For example, by using Petri nets, it was possible to model and evaluate the coupling between a reactive scheduler holon and a holon with special tasks, the on-line manufacturing control holon. In this way an improved adaptability to disturbances was obtained. It results that the proposed Petri net models of holons ensure some guiding points for a PROSA based holonic system design and implementation.

In ADACOR holonic architecture the dynamic behavior of holons is modeled by Petri nets, too [11]. In the same way as in PROSA, models are developed for each kind of holon and for various holonic behaviors in correspondence with the considered manufacturing environment. The proposed models catch the coordination process based on the Contract Net Protocol (CNP), too. As an advanced possibility, a top-down approach is used in ADACOR. Specifically, some transitions of a Petri net providing a first abstraction can be replaced by more detailed Petri nets in order to assure the description of additional aspects, as needed for the holonic system deployment. Such a successive decomposition in Petri nets and sub-Petri nets allows the incorporation into holonic models of details concerning the production plans and resource allocation.

Another methodology involving Petri nets is described in [12-15]. A formalism on holarchy formation and optimization, as well as on the management of coordination process is facilitated by the use of Petri nets. As a main point, an aggregated Petri net model of a holarchy is proposed, which is augmented with cost functions so that some conditions on holarchy feasibility could be formulated. Enhancements on handling by Petri nets the order constraints and reconfiguration abilities of holonic systems are developed, too. All these aspects are discussed only with regard to the execution phase, without considering details on the planning process and the link between planning and execution.

It thus results that Petri nets were already employed in HMES modeling. Even so, this paper together with the research published in [16, 17], aim at fulfilling some new, distinct goals: to underline all types of events that appear during the operation of any type of holon, to obtain a model of holonic communication so that, in conjunction with the holons' models, the complete HMES model should be obtained and to better reveal the dependence between planning and execution phases.

3. A Petri Net Model of Holonic Agent Operation

As the main operation unit of HMESs, a holon is composed of three components [18] (see Fig. 1): a decisional part in charge with managing the received goals and finding solutions for them; this is materialized under the form of a holonic agent. It has to apply a combination of planning and coordination procedures, as within HMESs a goal is always solved by the cooperation of several holons. The holonic agent's decisions are put into practice by the holon's structural component. For a resource holon this is a proper physical device: a robot or machine tool controller, a PLC commanding a conveyor, etc. In the case of an order or a product holon the structural component becomes a holarchy, which is a temporary construction, namely a group of holons that are conducted by the respective holon in order to solve a goal by cooperation. It is also possible for a resource holon to extend its structural component with a holarchy, when the holon asks the collaboration of other holons. The information is changed between the holonic agent and the structural component by means of a proper communication interface.

The mechanism for formation of holarchies is based on the CNP, the common coordination method of multi-agent systems [19, 20]. It supposes that a holon not able to solve a goal by itself becomes a manager, asking the cooperation of the other holons, by sending appropriate goals/sub-goals. Those holons able to provide a solution reply with corresponding bids, the best one being selected by manager. The holon that made the respective offer receives a contract from the manager holonic agent, to put into practice the solution it proposed. The common multi-agent CNP has to be adapted and enhanced for a holonic use, in order to obtain a reliable and near to optimum HMES operation [21]. Moreover, a further tuning is needed when the inference mechanism of agents is based on the Belief Desire Intention (BDI) architecture [22, 23]. Certain details regarding the way planning and coordination are supported by holonic agent inference process will be presented in sections 4 and 5.

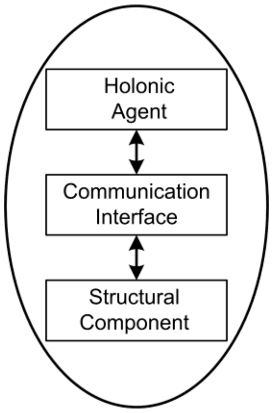

Fig. 1. Generic structure of a holon

HMES operation is both goal and event driven: it has to solve all the goals received at the shop floor level, taking into account events happening in the manufacturing environment (states of various devices, raw parts that are supplied, etc.). It is clear that the whole system operation is determined by the behavior of holons, which is dictated by holonic agents. All these conduct to the necessity of a model for the holonic agent operation and indicate the Petri net formalism as a good choice [12, 24].

The proposed Petri net basic model is a general one (see Fig. 2), being applicable independent of the type of holon, be it an order, product or resource one. The model shows the two main processes that the holonic agent has to pass through as the decisional component, namely planning when it finds a plan for solving a goal, and execution when it applies the decided plan and monitors its carrying out. Besides these, an idle state of agent is present, which shows its availability, allowing switching between the holonic agent's processes, too.

In this Petri net model the corresponding places are: P_0 for the idle state, P_1 for the planning phase and P_2 represents the execution one. Transitions that determine passing to planning are t_1 representing a goal receiving and t_3 modeling the receipt of a set of bids. This is explained by the way a goal is solved: besides the easy case when the goal can be worked out by a single resource holon through its own physical device activity, the other cases imply cooperation between several holons. Knowing the main steps of CNP, it is clear that the planning stage is interrupted or finalized at two types of transitions: t_2 regards sending of a goal (sub-goal) and t_4 appears when the agent releases an internal contract allowing the start of execution phase or when it issues a bid. Because the model aims at being a general one, it considers the case when the same holonic agent can be contractor and manager, too. That is way the agent can receive and announce goals, while it can also propose bids and issue internal contracts. There are two cases when internal contracts are used: when the holonic agent is only manager, as it can be for an order holon, and when the holonic agent of a resource holon can solve a goal by commanding its structural component. Usual contracts, those sent to other holons, are named external contracts or simply, contracts.

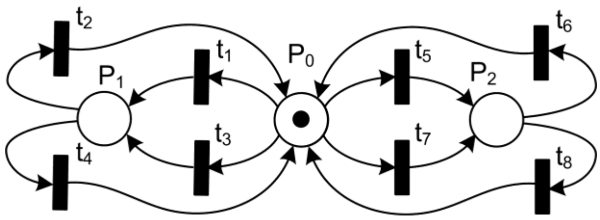

Fig. 2. Basic Petri net model of holonic agent operation

This Petri net model reflects both above mentioned cases, when the agent is able to solve by itself a goal (the corresponding succession is: $P_0t_1P_1t_4P_0$), and when it has to apply for collaboration, this evolution being modeled by the succession: $P_0t_1P_1\{t_2P_0t_3P_1\}t_4P_0$. The notation within curly brackets means that the sequence $t_2P_0t_3P_1$ can appear no time, when there is no goal considered for collaboration, or it can be used one or more times, depending on the number of goals that the agent issued and for which it is waiting cooperation.

Execution stage represents the carrying out of a previous holonic agent's commitment, represented by a bid it made. This is started from the agent's idle state when it receives a contract. The respective event is modeled by the transition t_5. After that, according to the transition t_8 the holonic agent sends the contracts to its sub-contractors or, for a resource holon, commands towards its physical device. Transition t_7 models the feedback from contractors or from the controlled device regarding the ending (with success or failure) of a contract or an action.

When an entire contract is ended, transition t_6 designates the finalization of execution phase. If execution regarded an external contract, according to t_6 the holonic agent provides a feedback towards manager, and this includes the case when the contract could not be accomplished and the respective holon is not able to solve the failure by itself. When the holon can try a further solution to a failed contract or action, during transition t_6 the holonic agent issues an internal (that is not sent by another holonic agent) goal, thus re-starting its planning phase. To conclude, the execution sequence is reflected in the proposed model as follows: $P_0t_5P_2t_8P_0t_7P_2\{t_8P_0t_7P_2\}t_6P_0$. It is to further underline that the proposed model catches (through transitions t_1 and t_3, respectively t_5 and t_7) all the cases when a planning/execution stage is started or resumed for the holonic agent, thus being a general one. The devised Petri net model reflects holonic agent behavior, but the whole HMES activity must be supported by a proper communication mechanism, as shown in the next section.

4. Model of Holonic Agent Communication

When each holonic agent is represented by the Petri net model of Fig. 2, the communication between holons can be modeled according to Fig. 3a and b. It is considered a holonic interaction with three holons, named H_1, H_2 and H_3 (see Fig. 3a). In this example, communication starts when H_1, as manager within the CNP, issues a goal according to the transition $t_{2(H1)}$ (notations of Fig. 2 are respected, too). The result of this transition, as the agent's message, is placed in a buffer represented by the place P_{OUT2} in Fig. 3a. From here, by means of a communication network, it is transmitted to all the possible contractor holonic agents, being inputted into their buffers, marked as the places P_{IN1}. Thus, transition t_1 regarding the presence of a new goal can be fired (in our example, two holons, H_2 and H_3, activate the corresponding transitions: $t_{1(H2)}$ and $t_{1(H3)}$). After that, these contractor holons start their planning process, which is abstracted at the manager holon level by the place $P_{W(H1)}$ (see Fig. 3b).

Each contractor enters in the planning phase only when the holonic agent is freed from other activities. As an example of the proposed model application, for the holon H_3 this means a token is present in the place $P_{0(H3)}$ (see Fig. 3b). Planning phase will have as result a message containing a bid (an output buffer is used), corresponding in our model with the placement of a token in the place P_{OUT4} when the transition t_4 (the one for bid's sending) is fired ($t_{4(H3)}$ in Fig. 3a, b). From this buffer, the communication network transfers the message to the buffer (the place P_{IN3}) of the manager agent; thus, the transition t_3 can be fired for this agent ($t_{3(H1)}$ in our example). In Fig. 3b the places $P_{1\text{->}3}$ and $P_{3\text{->}1}$ abstract information (goals and bids) transfer from one holon to the other by means of their buffers and the communication network.

A similar mechanism exists for transmission of contracts and feedbacks issued at the end of contracts. Agents' communication buffers should allow several goals/contracts to be received, and thus it can happen that a holonic agent has to treat more goals and/or contracts. Therefore, the issue of treating several goals by the same holon has to be discussed, and also the case when the solutions provided to a set of goals by some holons interfere; these issues are discussed in the next sections.

To understand these problems for a BDI based holonic agent (this kind of agents was used in our approach), it is necessary to be aware of the BDI mechanism operation principle.

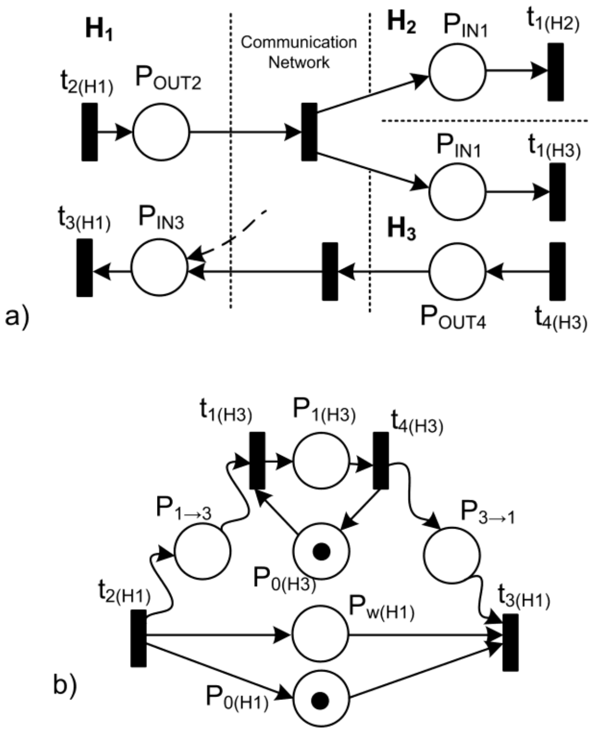

Fig. 3. Model of inter-holonic communication

5. Planning and Execution Processes for a Holonic Agent

As already discussed in section 2, the holonic agent operation covers two distinct phases: planning and execution. The holonic agent functioning starts with planning. The use BDI agents determines a certain influence on the way planning is treated. A specific aspect regards the library of plans that endows the BDI holonic agents (see, for example, JACK software platform dedicated to implementing BDI agents, which allows the definition and use of plans [23]). For each type of goal the BDI holonic agent must possess a set of plans that it can try when faced with the respective type. In the proposed approach these plans are un-instantiated execution workflows. Here, we name as workflow the whole sequence of actions that a holonic agent uses for an entire execution phase. An un-instantiated execution workflow specifies the sequence of actions that can solve a certain goal, but with the actors to carry out the actions being un-specified. Thus, during the planning process the holonic agent tries to validate an execution workflow by establishing (mostly through the means of CNP) which will be the entities (other holons or its own physical device) to perform the actions of execution workflow. Taking into account all these aspects, the planning cycle for the holonic agent is conducted according to a sequence consisting in the following steps:

Step 1. Choose an execution workflow that could solve the goal. If there is no further choice in the agent's library of plans, then the cycle is ended by declining the goal.

Step 2. Try to validate the selected execution workflow (this means to find resources able to carry out the actions of workflow).

Step 3. If the selected execution workflow has been validated then the planning phase is successfully ended (the corresponding bid/internal contract is sent), else the cycle is restarted with the Step 1.

Both planning and execution can be modeled by Petri nets. As an example, Fig. 4 shows the Petri nets of the execution workflow and the related planning process for solving a goal *g* (this appears as superior index for the entities of models). The execution workflow contains two actions – a_1 and a_2, for which the holon that received the goal, acting as manager, has to find actors (contractors). These can be viewed as resources and they are modeled by the places ${}^gP_{Rai}$ added to the basic Petri net model (see Fig. 4a). The places ${}^gP_{Rai}$ abstract the state of contractors; for example, the token in ${}^gP_{Ra1}$ marks the commitment made at the moment of bidding, regarding the engagement of achieving the action a_1. The execution workflow is chosen at the beginning of planning process (selection is based on the relevance for a goal and on certain optimum criteria). By that time, no tokens are present in the places ${}^gP_{Rai}$.

The planning process can be modeled according to Fig. 4b, where the same notations of the models of Figs. 2 and 3 are used. The model highlights the two possible outcomes for planning. When the manager holon receives at least a bid for an action of the goal *g*, the appropriate contractor is allocated and a token will be present in the corresponding place ${}^gP_{Rai}$. If the manager receives only negative bids for a proposed goal, the planning process is abandoned; this case is indicated through the transitions ${}^gt^a_{4(i)}$ in Fig. 4b, and the manager continues its activity in accordance with the above presented cycle. The planning process of Fig. 4b regards a sequential planning of two actions, and the succession of states and transitions is: $P_0t_1P_{1(1)}t_{2(1)}P_0t_{3(1)}P_{1(2)}t_{2(2)}P_0t_{3(2)}\ P_{1(3)}t_4P_0$, which is in accordance with the general expression presented in section 2 for solving a goal by cooperation. By the places $^gP_{W(i)}$ in Fig. 4b, the planning process at the level of contractors is modeled.

After the planning phase, the instantiated execution workflow (tokens are present in the places $^gP_{Ra1}$ and $^gP_{Ra2}$) is used to entirely guide the execution phase, as the model of Fig. 4a shows. One has to notice how this complies with the Petri model of Fig. 2. During execution no decision points appear, except for the case of an action failure, when planning phase must be restarted. The transition $^gt_5$ starts a first execution stage (marked as $^gP_{2(1)}$ in

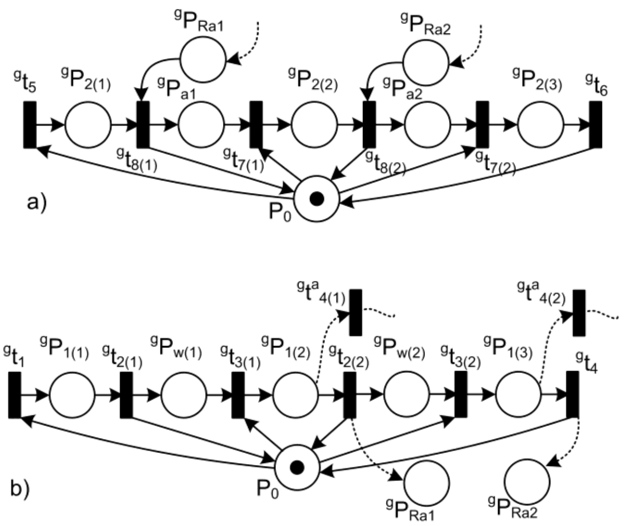

Fig. 4. Petri net models of planning and execution processes for a goal to be solved by cooperation; a) Model of execution workflow; b) Model of planning process

Fig. 4a) as result of contract awarding (see t_5 in Fig. 2). According to the validated workflow, execution continues with awarding a contract towards the chosen sub-contractor – transition $^gt_{8(1)}$. The place $^gP_{a1}$ reflects the execution of the needed action (a_1) by sub-contractor, being ended at the transition $^gt_{7(1)}$, when the holonic agent receives the feedback concerning action end. The place $^gP_{2(2)}$ represents a second execution stage, with the same progress as the first one. The place $^gP_{a2}$ and transitions $^gt_{8(2)}$, $^gt_{7(2)}$ have the same meaning as for the previous execution stage, this time regarding the action a_2. The considered example illustrates the general case of a holon being both contractor and manager: it is contractor for the received goal g, and it is manager with respect to finding solutions to achieve the actions a_1 and a_2. One has to note that the Figs. 4a and b are to be regarded together. This means that there is a single place P_0 in the holonic model and the places

${}^{g}P_{Rai}$ are common for planning and execution. All these models allow the HMES analysis, as shown in the next section.

6. Holonic Interaction Analysis; Experimental Results

6.1. Possible drawbacks for the holonic agent planning activity

The planning and execution activities are not continuous: after sending a goal to potential contractors the agent has to wait for bids, after sending a bid the agent waits for manager's answer, after sending a contract the agent has to wait its accomplishing. Thus, it can happen that processes regarding several activities are interleaved for the same holonic agent. Three types of combinations are possible: two planning processes are interleaved, two execution workflows are simultaneously undertaken, or one planning and one execution process are handled by the holonic agent. The last two cases do not need a special attention, because as long as an execution workflow was validated by planning phase it cannot faultily influence another process.

The significant combination is when two planning processes of the same holon are in progress in the same time. Each planning process is started by a distinct goal. If the agent has received two goals and their treatment is interleaved, then the case of Fig. 5 can happen, where the superior indices *1* and *2* refer the two goals (see the index *g* in Fig. 4). The agent works with two execution workflows, which it tries to validate for the two goals. With respect to this, the agent has announced goals (sub-goals) in order to find contractors for the actions of execution workflows (in our example all these actions need other holons to carry them out). The problem is that the actions represented by the places ${}^{1}P_{a1}$ and ${}^{2}P_{a1}$ need the same type of resources, the same condition being true for the places ${}^{1}P_{a2}$ and ${}^{2}P_{a2}$. If the contractors managing the two types of resources happen to make bids for the two execution workflows as shown by the tokens placed in Fig. 5, both planning processes fail, as they cannot transform the corresponding execution workflows into live Petri nets.

The above case happens even the HMES could provide a solution at least for one goal. This occurs if a single planning process is allowed to start and only after its finalization the second planning process begins. This rule is not to be always applied, because it reduces HMES flexibility: it can also be possible for two planning processes to be treated in the same time without deadlock. The solution is to restrict the simultaneous activation for validation (for planning) of more execution workflows that could determine a blockage, as they refer to common resources. This can be implemented by correspondingly marking plans (execution workflows) within the agent's planning library. The proposed Petri net model can help this marking operation, by using it in a simulation developed before the start of holon's activity within HMES, during which the interaction of plans can be revealed.

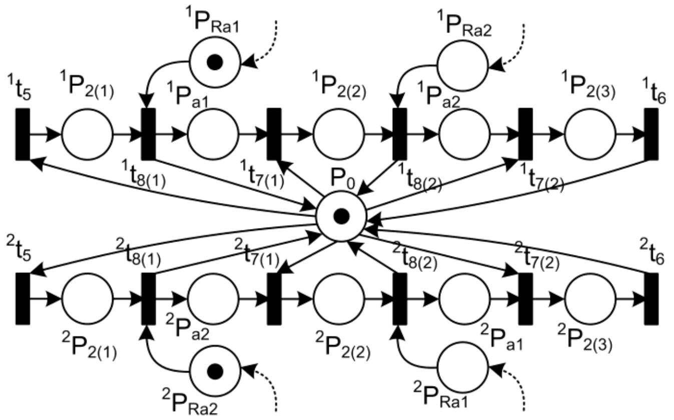

Fig. 5. An example of interaction resulting from the planning process

A similar situation can happen when the two goals that create a conflict are handled by two distinct holonic agents. The case presented in Fig. 5 reflects these circumstances too, except for the fact that two places P_0 will exist, one for each holon. A solution for such situations is beyond the product/order holon possibilities, as they possess local knowledge. There is the need of another component, and this can be a staff holon in the PROSA architecture, which is supposed to take the decision on the management of goals [16]. This type of interaction must be taken into account only between product and order holons, because for resource holons the solution can be obtained at their own level, as they can distinguish goals so that deadlock is avoided [24]. When product/order holons need to announce goals towards resource holons, they should require the acceptance of a dedicated staff holon. This will send the approval only to one requesting holon and keep in a queue the other enquiries that refer to the same type of resources. At the moment of planning finalization, an agent of an order or product holon has to announce the staff holon about this, so that it can consider the next request. In this way two planning processes that refer some common resources are never interleaved and any deadlock is avoided.

In order to conduct significant experiments for proving these theoretical points, a complex model and a simulation environment able to represent an entire HMES were developed [17], by the means of Coloured Petri Nets (CPNs) [25].

6.2. The HMES developed model

The constructed simulation environment appears as a hierarchical CPN, with the highest layer abstracting various entities of the HMES in the form of transitions and places. These are expanded on successive layers, taking into account the proposed basic Petri net model (the one of Fig. 2), but with tokens that can carry different information, according with the formalism of CPNs. An important propriety of CPNs is the way they combine the capabilities of monochrome Petri nets with the support of a high-level programming language, such as the Standard ML [26]. So, it was possible to obtain a highly configurable model-prototype, close to the real HMES implementation.

The top layer of the HMES model is presented in Fig. 6. It comprises one product holon and some resource holons; a staff holon was also introduced in certain experiments. The network necessary to handle the communication between holons is included, too. These entities are represented by transitions that hide the models of lower layers; these materialize by the proposed models for planning and execution (as the ones in Fig. 4). In comparison with the elements presented in Fig. 3, all the input places of a holon were integrated in a single input position. One has to understand that the transitions t_1 - t_8 (see Fig. 2) have attached either an input or an output place (buffer). By using CPNs all the input buffers of a model are represented by a single position, which regards a buffer, marked as *In_k* in Fig. 6. The same is true for the output positions, named *Out_k*. In this way in the CPN model each transition is fired when its attached condition is satisfied, according to the information of its buffer. The transfer of information between holons is achieved through the transition *Communication Network*. The model that this transition is substituting is presented in Fig. 7.

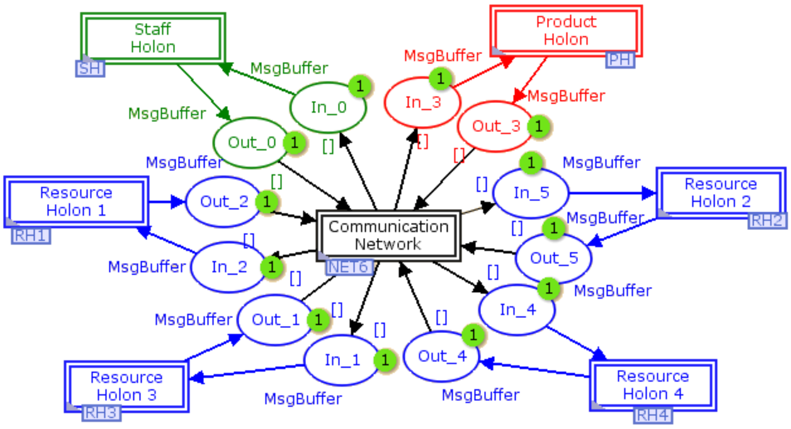

Fig. 6. Top layer of HMES model

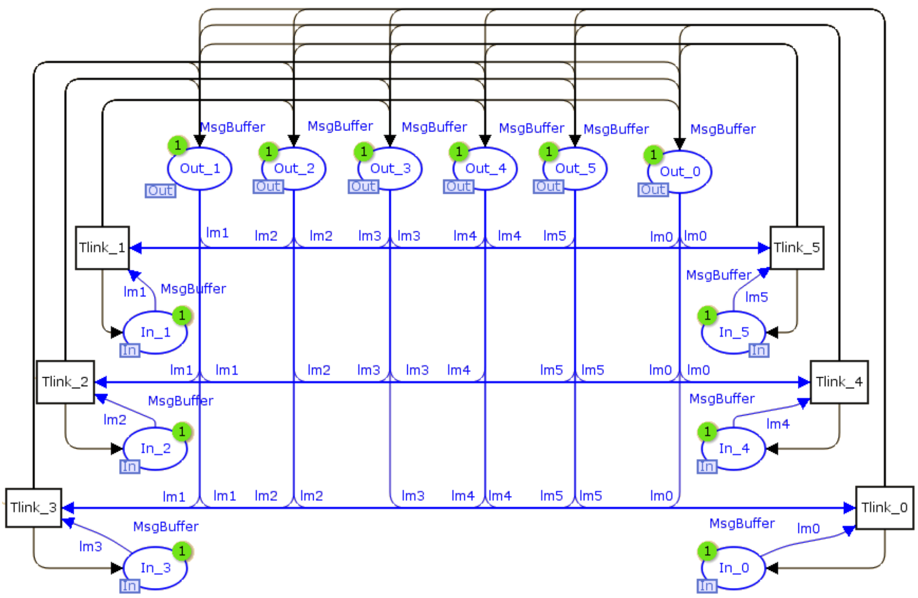

Fig. 7. Communication network model

Thus the part regarding the communication network that was not shown in Fig. 3a is displayed. It is to understand that *Out_k* in Fig. 6 and *In_k* in Fig. 7 are the same positions in the constructed model, according to the use of hierarchical CPNs. The transitions *Tlink_k* in Fig. 7 transfer information at their firing, when a message is present in the output buffer of an entity. Such a buffer is a queue, so that messages are processed in the order they are received. To model the normal operation when the messages are treated in the same order as they were sent, the transitions *Tlink_k* have the greatest salience. In this way, the influence of the communication network on the reachability graph is minimized; this is important because the analysis of HMES performance is obtained by using it.

In principle, a simulation experiment can be used to explore only a finite number of HMES executions. By using the reachability graph of a Petri net a possibility to surpass this difficulty is offered. Indeed, this covers the entire state space of the modeled system, starting from a given initial state [25]. It means the reachability graph can be a powerful tool for assessing the properties of HMESs, and that is way it was chosen. More specific, regarding the problem of establishing the output of holonic agents' planning process, the leaf nodes of reachability graph contain information about the results of resource allocation process. The leaf nodes represent the final or dead markings of the analyzed Petri net, and by their examination the outcome of holonic planning process can be understood.

6.3. Experiments for a holonic system with a product holon

The experiments carried out to prove the above analysis considered the interaction between two planning processes of the same holon (see the explanations concerning Figs. 4b and 5). Five distinct cases were evaluated; the difference between them is given by the availability of resource holons and the treatment of received goals. In cases 1 and 2, the product holon has to face a manufacturing environment with limited resources, allowing only one goal to be fulfilled; this restriction is removed in cases 3 and 4. On the other hand, the product holon tries similar plans in cases 1 and 3, respectively 2 and 4 (it attempts to instantiate similar execution workflows). The final case, the fifth one, regards the HMES operation when a staff holon is also included into the holonic architecture.

The results of the considered experiments are presented in Tables 1-5. These contain three data types. The first two rows give the initial HMES state: the goals received by the product holon, the actions included into the execution workflows that the product holon uses to solve goals and the resource holons able to carry out the respective actions. This information is used to establish the initial marking of the Petri net model. Data on the reachability graph (number of nodes, arcs and dead markings) are presented in the next two rows of the tables, while the following rows summarize the results of planning process. Having two goals to solve, the planning results, reflected in resource allocation, are classified into three classes. These mean the planning process succeeded to instantiate two execution workflows, one of them (with two sub-cases) or none. For each class, the indicated percentage represents the number of final states (dead markings) that correspond to the respective class from the total number of final states.

Table 1. An experiment with two goals and reduced resources

Product Holon's State		Goal 1 (a1-a2)	Goal 2 (a1-a2)
Resource Holons' State		$RH_1 \rightarrow a_1$, $RH_2 \rightarrow a_2$	
Nodes	13525	Arcs	23030
Dead Markings		80	
Goal 1		Goal 2	%
successful		successful	0
successful		unsuccessful	50
unsuccessful		successful	50
unsuccessful		unsuccessful	0

When both goals are solved by the same execution workflow (the succession of actions is a_1, a_2) and two resource holons (RH_1, RH_2) are present in the HMES, only one goal can be fulfilled, as Table 1 shows. In this first experiment no conflict between the planning processes appears. When the HMES context is the same, except for the execution workflows used to treat the goals, the results of Table 2 are obtained. These indicate that a conflict is possible: there are 3.23% of the total number of dead markings that represent cases when neither of the two goals is solved. It means the product holon fails to accomplish at least one goal, despite the fact that a solution exists. This is the case of Fig. 5, when during the planning process two execution workflows are simultaneously treated and it happens that resource holons make bids for the first action in each workflow; thus, no resource is available to complete a planning process. This situation does not appear in the experiment considered in Table 1, because in that case one of the execution workflows is already abandoned when the product holon does not receive a bid for its first action, and thus the failure situation is avoided.

Table 2. An experiment with two goals and different execution workflows

Product Holon's State		Goal 1 (a1-a2)	Goal 2 (a2-a1)
Resource Holons' State		$RH_1 \rightarrow a_1$, $RH_2 \rightarrow a_2$	
Nodes	55271	Arcs	116622
Dead Markings		186	
Goal 1		**Goal 2**	**%**
successful		successful	0
successful		unsuccessful	48.39
unsuccessful		successful	48.39
unsuccessful		unsuccessful	3.22

Table 3. An experiment with two goals and enough resources

Product Holon's State		Goal 1 (a1-a2)	Goal 2 (a1-a2)
Resource Holons' State		$RH_1 \rightarrow a_1$,$RH_2 \rightarrow a_2$, $RH_3 \rightarrow a_1$ $RH_4 \rightarrow a_2$	
Nodes	63257	Arcs	139552
Dead Markings		164	
Goal 1		**Goal 2**	**%**
successful		successful	39.02
successful		unsuccessful	30.49
unsuccessful		successful	30.49
unsuccessful		unsuccessful	0

The case in Table 3 is similar to the first experiment, but this time there are dead markings representing the fulfillment of both goals, because more resource holons exist in the HMES. When the same experiment is made with different workflows used in planning, the results are worse: the percentage of 23.26% dead markings representing the accomplishment of both goals in Table 4 is less than the value in Table 3. The explanation for the difference between the results of Table 3 and Table 4 is the same as for the cases in Table 1 and Table 2. It is to notice that the number of dead markings is much higher than the number of planning results, which is explained by the internal mechanism that labels with distinct identifiers messages sent between holonic agents, conducting to different markings in the Petri net. Anyhow, the proper reading of the results obtained with these simulation experiments proves the theoretical points.

Table 4. An experiment with enough resources and different execution workflows

Product Holon's State		Goal 1 (a1-a2)	Goal 2 (a2-a1)
Resource Holons' State		$RH_1 \rightarrow a_1, RH_2 \rightarrow a_2$, $RH_3 \rightarrow a_1$ $RH_4 \rightarrow a_2$	
Nodes	174575	Arcs	396964
Dead Markings		324	
Goal 1		**Goal 2**	**%**
successful		successful	23.26
successful		unsuccessful	38.37
unsuccessful		successful	38.37
unsuccessful		unsuccessful	0

As already indicated from the theoretical point of view in section 6.1 and practically by the second case, the HMES can fail in solving both received goals when these regard common resources. As mentioned, the introduction of a staff holon can eliminate these drawbacks. Thus, Table 5 presents the result of a further experiment that is conducted for the same case as in Table 2, but with the presence of the staff holon. The interaction diagram in Fig. 8 shows how the product holon communicates with the staff holon when it has to solve two distinct goals. The messages labeled CI (Contractor Information) are those by which the product holon requests from the staff holon the list of available contractors for the plans it has chosen for the two goals. Because the staff holon detects a possible conflict between the contractors of the two plans, it provides a positive answer for one request (the message $CI_{203\text{-}401}$) and the second answer is given only after the finalization of the first planning

Table 5. An experiment with the staff holon

Product Holon's State		Goal 1 (a1-a2)	Goal 2 (a2-a1)
Resource Holons' State		$RH_1 \rightarrow a_1$, $RH_2 \rightarrow a_2$	
Nodes	11243	Arcs	15486
Dead Markings		76	
Goal 1		**Goal 2**	**%**
successful		successful	0
successful		unsuccessful	50
unsuccessful		successful	50
unsuccessful		unsuccessful	0

process (the message EI_{401} in Fig. 8). The obtained result (see Table 5) shows that in this approach the drawback of both goals' failure is eliminated. More details and experiments for the operation of an HMES including a staff holon are presented in [16].

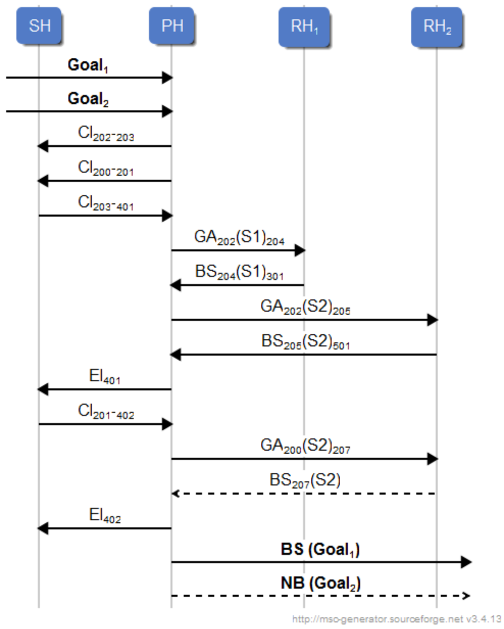

Fig. 8. Interaction diagram for the experiment with the staff holon

7. Conclusion

The work described in this paper addresses modeling and analyzing techniques capable of revealing certain planning and coordination issues of multi-agent systems included in HMESs. The proposed Petri net models describe the internal and external behavior of holons. It is used to construct the reachability graph, which provides important data on the states the HMES can pass through. The analysis has shown the necessity of an appropriate protocol that holonic agents should use for handling of plans, and moreover the need of a centralized component to manage the possible conflicts among planning processes of different holons. These results led us to considering the staff holon as a required entity into an HMES, with the ability to protect the system against potential conflicts. A general planning cycle appropriate for resource, product and order holons was settled, while the staff holon should have a distinct operation, coordinating the other types of holons when a planning conflict is detected; thus, the whole operation of an HMES is covered.

The planned future work aims at better formalizing and evaluating the BDI mechanism when this is involved in the operation of holonic agents. Thus we are supposed to complete a systematic method for the application of multi-agent systems in holonic manufacturing control.

References

1. Pascal, C., Panescu, D.: On Resource Allocation in a Holonic Manufacturing Execution System. In Proceedings of the 15th International Conference on System Theory, Control, and Computing. Politehnium, Iasi, Romania, 427-432. (2011)
2. Candido, G., Barata, J.: A Multiagent Control System for Shop Floor Assembly. In Proceedings of 3nd International Conference on Industrial Application on Holonic and Multi-Agent Systems. Springer-Verlag, Regensburg, Germany, 293-302. (2007)
3. Jarvis, J., Jarvis, D., Ronnquist, R., Jain, L. (eds.): Holonic Execution: A BDI Approach, Studies in Computational Intelligence, Vol. 106. Springer-Verlag, Berlin, 1-32. (2008)
4. Morel, G., Valckenaers, P., Faure, J. M., Pereira, C., Diedrich, C.: Manufacturing plant control challenges and issues. Control Engineering Practice, Vol. 15, 1321-1331. (2007)
5. Giret, A., Botti, V.: Engineering Holonic Manufacturing Systems. Computers in Industry, Vol. 60, 428-440. (2009)
6. Cheng, F. T., Chang, C. F., Wu, S. L.: Development of holonic manufacturing execution systems. Journal of Intelligent Manufacturing, Vol. 15, 253-267. (2004)
7. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P.: Reference architecture for holonic manufacturing systems: PROSA. Computers in Industry, Vol. 37, 255-274. (1998)
8. David, R., Alla, H.: Discrete, continuous, and hybrid Petri nets. Springer Verlag. (2005)
9. Li, Z. W., Zhou, M. C.: Deadlock resolution in automated manufacturing systems: a novel Petri net approach. Springer Verlag. (2009)
10. Bongaerts, L.: Integration of scheduling and control in holonic manufacturing systems. Ph.D. Thesis, Katholieke Universiteit Leuven. (1998)
11. Leitao, P.: An Agile and Adaptive Holonic Architecture for Manufacturing Control. Ph.D. Thesis, Faculty of Engineering of University of Porto. (2004)
12. Hsieh, F. S.: Holarchy formation and optimization in holonic manufacturing systems with contract net. Automatica, Vol. 44, 959-970. (2008)
13. Hsieh, F. S.: Model and control holonic manufacturing systems based on fusion of contract nets and Petri nets. Automatica, Vol. 40, 51-57. (2004)
14. Hsieh, F. S.: Collaborative reconfiguration mechanism for holonic manufacturing systems. Automatica, Vol 45, 2563-2569. (2009)
15. Hsieh, F. S.: Dynamic composition of holonic processes to satisfy timing constraints with minimal costs. Engineering Applications of Artificial Intelligence, Vol. 22, 1117-1126. (2009)
16. Panescu, D., Pascal, C.: HAPBA – a Holonic Adaptive Plan-Based Architecture. In: Borangiu, T., Thomas, A., Trentesaux, D. (eds.): Service orientation in holonic

and multi-agent manufacturing control. Studies in Computational Intelligence, 402, Springer, Berlin, 61-74. (2012)

17. Panescu, D., Pascal, C.: On a holonic adaptive plan-based architecture: planning scheme and holons' life periods, International Journal of Advanced Manufacturing Technology, Springer. (2012)
18. Panescu, D., Sutu, M., Pascal, C.: On the Design and Implementation of Holonic Manufacturing Systems. In Proceedings of the WRI World Congress on Computer Science and Information Engineering. IEEE, Los Angeles, CA, 456-461. (2009)
19. Smith, R. G.: The contract net protocol: High level communication and control in a distributed problem solver. IEEE Transactions on Computers, Vol. C-29, No. 12, 1104-1113. (1980)
20. Hsieh, F. S.: Analysis of contract net in multi-agent systems. Automatica, Vol. 42, 733-740. (2006)
21. Panescu, D., Pascal, C., Sutu, M., Varvara, G.: Collaborative Robotic System Obtained by Combining Planning and Holonic Architecture. In Proceedings of Advanced Technologies for Enhanced Quality of Life. IEEE, Iasi, Romania, 138-143. (2009)
22. Rao, A. S., Georgeff, M. P.: Modeling rational agents within a BDI-architecture. In: Huhns, M. N., Singh, M. P. (eds.): Readings in agents. Morgan Kaufmann, 317-328. (1997)
23. Evertsz, R., Fletcher, M., Jones, R., Jarvis, J., Brusey, J., Dance, S.: Implementing Industrial Multiagent Systems using JACKTM. In: Dastani, M. M., Dix, J., Fallah-Seghrouchni, A. E. (eds.): Programming multi-agent systems. Lecture Notes in Computer Science, Vol. 3067. Springer-Verlag, Berlin Heidelberg New York, 18-48. (2004)
24. Panescu, D., Pascal, C.: Some Issues on Holonic Systems Analysis, Design and Implementation. In Proceedings of the 19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010), Budapest, 199-204. (2010)
25. Jensen, K., Kristensen, L.: Coloured Petri Nets: Modeling and Validation of Concurrent Systems. Springer-Verlag, New York. (2009)
26. Harper, R.: Programming in Standard ML. Carnegie Mellon University. (2005)

Carlos Pascal is a young researcher mainly interested in intelligent manufacturing systems. As an early adopter, he tries to bring up a new architecture for manufacturing systems based on autonomous and cooperative entities, by applying several concepts, like holons, Petri nets and BDI agents. With respect to this, in 2012, he received a PhD degree in Systems Engineering at “Gheorghe Asachi” Technical University of Iasi, Romania.

Doru Panescu is professor within the Department of Automatic Control and Applied Informatics, “Gheorghe Asachi” Technical University of Iasi, Romania. He received two master degrees (in Electrical Engineering and in Computer Science), and a PhD degree in Systems Engineering at “Gheorghe Asachi” Technical University of Iasi, Romania. His present teaching and research activities are in the fields of Robotics and Artificial Intelligence, with a focus on intelligent manufacturing systems.

Received: December 23, 2011; Accepted: April 9, 2012.