

Dario Medić,¹ Anita Gudelj,² Maja Krčum³

University of Split
Faculty of Maritime Studies
¹e-mail: dmedic@pfst.hr
²e-mail: anita@pfst.hr
³e-mail: mkrcum@pfst.hr

A formal model for planning and controlling search and rescue actions at sea

JEL codes: R41, C02

Keywords: UAV, Petri net modeling, conflict avoidance

Abstract. Recently, during search and rescue actions at sea, Unmanned Aerial Vehicles (UAVs) have been used. Onboard decision capabilities allow an UAV vehicle to reach the entity that is in distress at sea. UAVs are launched within a few minutes to begin search actions. When the exact location of the injured entity is detected, a rescue action should begin. According to the collected information about the vessel's position, manoeuvrability, and velocity, the control centre determines which vessel is to be engaged in the rescue action.

This highly autonomous system can be described as a discrete event system. Certain states of such systems, such as collisions, are undesirable. This paper presents implementation of information flow to supervise, control, and monitor the behaviour of the UAVs during the search, to avoid collisions and to communicate with computational onboard sub-systems. Planning algorithms and coloured Petri nets are used to specify different phases of the mission execution. When a certain UAV detects an injured entity, alternative encoded reactions are triggered and a control centre starts implementing the rescue plan.

Introduction

Unmanned Aerial Vehicles (UAVs) are becoming a very important tool for Search and Rescue (SAR) operations at sea (Skrzypietz 2010). As it is known, time is critical for saving human life at sea, so any delay can result in potential losses of human

life. For SAR operations, UAV can be fitted with high resolution cameras, multispectral sensor, thermal sensor, infrared sensors (IR), and hyperspectral sensors. According to the development of UAVs to this date, they can be airborne from one hour to more than 24 hours. It is a known fact that UAVs can be controlled in two ways: at distance and by their previously set route. For the purpose of this paper, UAVs that fly according to their previously set route will be analyzed.

Systems with multiple UAVs present significant advantages in different applications by increasing efficiency, performance, and robustness (Alejo et al. 2013, Maza et al. 2011).

Some of the problems that need to be solved are: a) How to control the traffic in a way that UAVs moving in opposite directions make as few stops as possible during the passage through the cells in the space? b) How to resolve possible conflicts in case that more vehicles try to acquire the same cell at the same time? c) How to avoid possible deadlocks in the dense traffic?

The vehicle's moving through the cells in the space can generally be described as a set of discrete states and events (discrete event dynamic systems – DEDES). These events and states are normally observed by the UAV management system (UAVMS) which receives data from Air Traffic Management systems (ATM), using wireless data communication. Some of these states, such as conflicts and deadlocks are undesirable. In Kezić et al. (2010), the authors used DEDES and Petri net (PN) theory, a well-known tool for analyzing DEDES to resolve some of the above-mentioned problems.

In this paper, we present a timed colored Petri net-based hierarchical architecture for UAV traffic system simulations. Central to these simulations is the design issue of conflict free structures of UAV behaviors. Moreover, timed colored Petri net is used for modelling UAV control structure. For the construction of the network and the simulation, Design/CPN tool was used.

1. Problem formulation

Searching and routing are essential elements in many operations, especially in sea environment. The flying of UAVs usually takes place by following a set of predetermined, physical, or virtual guide orbit of airspace. UAV has, as mentioned before, a lot of sensors which transmit data about its position to a central microcomputer. Thus in every timestamp, UAV's position, its speed, and traffic conditions in specific zone of airspace are known. Suppose that each orbit is to be parameterized by the number of cells that the UAV passes through. It is assumed that the size of the cells is a parameter and the safety distance is given by a number of cells. The problem considered in this paper concerns conflict detection and resolution between UAVs in a common search space.

The conflict resolution is based on changing the velocity of the UAVs involved in the potential collision, and it is necessary to execute a simple set of the following rules:

- if no UAV in collision zone, then *remain* in cruise speed;

- if UAV in collision zone, then *transmit* a wait signal; and
- if wait signal received, *decelerate* to slow speed.

For example, as it is shown in Fig. 1 (a), the cells through which UAV₁ passes are: 1, 2, 3, 4, and 5. UAV₁ passes through cells 9, 2, 3, 10, and 11. Each UAV flies from time $t = 0$ to time $t = t_1$ passing through the cell i . When $t = t_1$, both vehicles try to enter cell 2, and then cell 3 at the same time. In that case, there would be a collision between these vehicles. The conflict cells are cell 2 and cell 3. It is necessary to ensure that only one UAV is in the cell 2 and cell 3, respectively.

By following the given set of rules for avoiding collisions, Fig. 1 (b) shows the outcome of situation (a) in which UAV₂ enters cell 3 at the same time as UAV₁ leaves it. Actually, the solution is decreasing the velocity of UAV₂ passing through cell 9 and in this way delays its stay in cell 9 to avoid the collision with UAV₁ in cell 2.

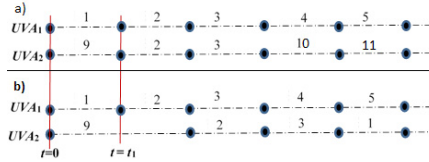


Fig. 1. The example of applied collision avoidance resolution

Source: own elaboration.

2. Timed CPN formalism

PN is a kind of network information flow model. Basically, PN is a bipartite graph consisting of two types of nodes, places, and transitions, connected by arcs. Direct arcs can exist only between change nodes and location nodes. Adding token distribution indicates information status on the base of the directed bipartite graph. According to the firing rule, the event-driven state can evolve, and this will reflect the dynamic process of the system (Murata 1989). Under normal circumstances, transitions were represented by small rectangles, and places were represented by small rounds.

A coloured Petri net (CPN) extends the formalism of the PN by equipping each token with an attached value called a token colour. A CPN allows for a more compact representation of a behaviourally equivalent PN. Timed CPN extends the CPN formalism with the concept of time. This will allow us to evaluate how efficiently the system performs its operations, and it also allows us to model and validate real-time systems.

The formal definition of the CPN, as given in Jensen (1992) and Ribarić (2005), uses an expression representation, and therefore presupposes that certain syntax exists together with a well-defined semantics. A timed coloured Petri net (TCPN) is defined as a bipartite directed graph represented by a ten-tuple $CPN = (P, T, A, \Sigma, N, C, G, E, FT, \mathbf{m}_0)$, where:

- P is a set of places, T is a set of transitions, A is a finite set of arcs such that $P \cap T = P \cap A = T \cap A = \emptyset$;
- Σ is a finite set of non-empty types called colour sets;
- $N: A \rightarrow (P \times T) \cup (T \times P)$ is a node function which connects places to transitions and vice versa;
- $C: P \rightarrow \Sigma$ is a colour function that associates with each element in $P \cup T$;
- $G: T \rightarrow [\text{expr}]$ is a guard function defined from T into expressions such that $\forall t \in T: [\text{Type}(G(t)) = B \dot{\cup} \text{Type}(\text{Var}(G(t))) \dot{\cap} \Sigma]$, B is Bool type;
- $E: A \rightarrow [\text{expr}]$ is a logical arc expression function such that $\forall a \in A: [\text{Type}(E(a)) = C(p(a))_{MS} \wedge \text{Type}(\text{Var}(E(a))) \subseteq \Sigma]$, where $p: A \rightarrow P$ maps each arc a to the component of $N(a)$ which is a place;
- FT is a timing vector; and
- \mathbf{m}_0 is initial marking.

The concepts of places, transitions, and arcs in the above definition have the same significance, and they are represented by the same semantics, as in the ordinary PN framework.

Each place $p_i \in P$ is associated with the set of possible token colours $C(p_i) = \{a_{i,1}, a_{i,2}, \dots, a_{i,u_i}\} \subseteq \Sigma$, where $u_i = |C(p_i)|$ is the number of possible colours of tokens in p_i . Analogously, C maps each transition $t_j \in T$ to the set of possible occurrence colours $C(t_j) = \{b_{j,1}, b_{j,2}, \dots, b_{j,u_j}\} \subseteq \Sigma$ with $u_j = |C(t_j)|$. In order for transition t to be enabled, there must exist a binding in which guard expression $G(t)$ evaluates to $T \in B$. In the same binding, the multi-sets of tokens of appropriate colours, as specified by the input arcs expressions, must be present in all input places of transition t . When enabled, transition may fire. Firing of transition t generates multi-sets of tokens in output places of t , and consumes multi-set of tokens from input places of t , as specified by the arc expressions.

The temporization of a coloured PN can be achieved by attaching time either to places, to transitions, or to the expression functions of arcs (Jensen 1997). In this work, the second option is chosen, and timed transitions and immediate transitions are considered. Precisely, FT denotes a timing vector, and the firing time of each transition t_j is the positive number $FT(j)$ specifying the deterministic duration of the firing of t_j . In this method, a token has a time stamp attached to it, in addition to token colours. The time stamp is described by the function $s: C \rightarrow \mathcal{R}^+$ where $s(c)$ indicates the earliest delay after which the token of colour $c \in C$ becomes available and can be removed by an enabled transition. Hence, as soon as the c -colour token arrives to the place p_i enabling transition t_j , $s(c)$ is set to FT_j . Accordingly, after FT_j time instants the enabled transition t_j becomes ready to fire with respect to colour c . If FT_j is equal to zero, the transition is said to be an immediate transition.

3. Event-driven simulation model of the UAV traffic

This chapter deals with the task net and a supervisor design for the UAV traffic system (Fig. 2). The presented case study is relatively simple. The intention of this example is to clarify the theory in the previous section. However, the above theory is applicable to the more complex systems.

Let us assume that UAV's initial speeds are the possible maximum ones, respectively. Let us also assume that all the UAVs are the same in their functions or behaviours. These UAVs will fly and search for the target along a predefined orbit. To increase the efficiency of UAVs, it is envisaged that there are two circular orbits of the moving of UAVs. It is assumed that the tracks are divided into cells and the traffic is controlled by cells. These cells are specified as areas with a certain width and length, and should be large enough to accommodate the entire body of an aerial vehicle. The move of an UAV is controlled with a capacity of an individual cell. System cell's capacity is determined with a number of UAVs allowed to enter a cell. To achieve a primary goal of the UAV system – a collision and collision free traffic – we assumed that only a single vehicle is allowed to enter a cell. The cells of the space are enumerated, they are the same size, and the time needed to fly an aerial vehicle from one to another is 2 time units.

Suppose that the considered UAV system consists of three UAVs and two orbits. The first orbit contains seven cells, while the other one contains six cells. All stations and UAVs represent resources of the considered system. The UAV in the first orbit must pass the following cells: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 1$, and another UAV in the second orbit must pass $8 \rightarrow 9 \rightarrow 2 \rightarrow 3 \rightarrow 10 \rightarrow 11$.

The traffic in the UAV system can be fully controlled using a computer system which controls traffic lights at the entrance into the conflict zone. Task processing are defined from generated firing sequence which has to be checked for conflict existence. In PN formalism, this can be achieved by adding additional control places and transitions conditions which restrict the number of tokens in critical subzones.

Timed CPN model will be applied to the UAV traffic system simulation. Multiple UAVs are modelled as tokens in CPN. The tokens in input place START represent the set of aerial vehicles waiting to enter the system. The variables *uav* (1) and *uav* (2) are token colour for that place and represent three available UAVs. The aerial search space includes the set of orbit which is represented by the tokens in input place ORBIT. The set of all places that represent cells in the system (the number of tokens in a cell place represents the number of vehicles in a particular resource) is $\{1-11\}$. The colour set inscription WP has been added to those places. It determines the colour set, i.e., the type of all the tokens that can be put in the place.

WP describes the individual cell, and the format is a record containing *se*-field for the identity of the UAV, *re*-field for the identity of the orbit, and a *no*-field for a cell number. Variable $\{se = x, re = v, no = k\}$ is a set according to the colour tokens that are associated to the cell place.

Control places {IdleOr1, IdleOr2} belong to the supervisor which ensures if it is possible, the orbit s is assigned to one aerial vehicle by adding or removing tokens from these places. Initially, the number of tokens in each control place equals 1, and it has green colour indicating the availability of the orbit.

A transition firing may indicate that some operation is performed. Due to the weak firing rule of PNs, we must deal with the synchronization issues for modelling multiple UAVs. In UAV simulations, an UAV is not supposed to perform t_3 before UAV has finished operation t_4 . This approach ensures the synchronization among UAVs to transit from one task stage to another (e.g., before UAVs start a new sweeping search stage, all UAVs have aligned up).

Next, let us look at the transition move 1, which has a guard: $[s = z, x = \text{orbit}(1)]$. The purpose of a guard is to define an additional constraint which must be fulfilled for a transition to be enabled. In this case, the guard tells us that only the tokens representing vehicles assigned orbit – orbit (1) can move to the place 1 (because the guard for all bindings evaluates to false and thus prevents enabling).

To realize the effects of UAV behaviours on the simulation environment, we allow for the association of procedures with arcs in task net and its subnets. Therefore, firing transition move 1 in this net can result in the update of information of availability of the first orbit according to the implementation of and the procedures associated with the arc from IdleOr1 to the transitions. This approach greatly facilitates rapid prototyping of new UAV simulations.

The main difference between timed and untimed CPN models is that the tokens in a timed CPN model—in addition to the token colour—can carry a second value called a time stamp. In general, a time stamp can be a non-negative integer or real. In the current CPN model, only non-negative integers are supported. The time stamp tells us the time at which the token is ready to be used, i.e., the time at which it can be removed from the place by an occurring transition. The time stamp of tokens are written after the @ symbol.

In reality, any resource requires non-zero times for performing jobs. In our CPN, time inscriptions are attached to the transitions. These times represent the time units needed for an UAV to pass through a single cell. This will allow us to evaluate how efficiently a system performs its operations. For example, in Fig. 2, there is attached a time delay inscription @+2 to the transitions: t_4 , t_5 , t_6 , t_7 , t_8 , t_{10} , and t_{11} . It is now also necessary to calculate the time stamps to be given to the output tokens. The time stamp given to the output tokens is the value of the global clock plus 2 units. Intuitively, this represents the fact that an UAV can leave the cell 2 time units after the end of the previous operation. The tokens on a place will carry a time stamp if the colour set of the place is timed.

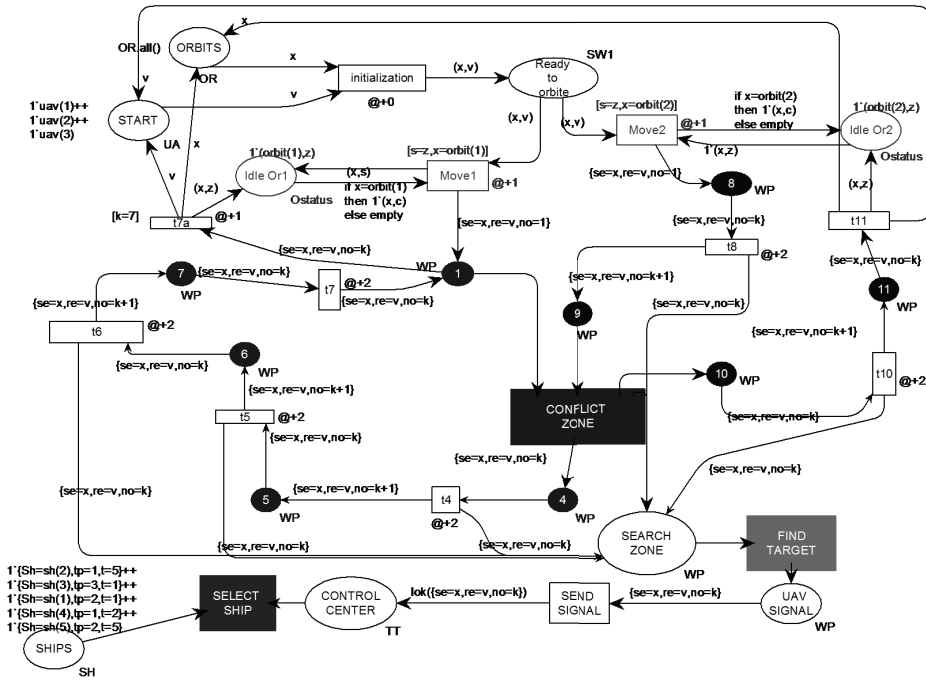


Fig. 2. CPN model of UAV system layout

Source: own elaboration.

4. Case study I: conflict resolution

The transition *Conflict_Zone* is the substitution transition. In contrast to the ordinary transitions, substitution transitions represent modules – substitution subnets composed of set of places and transitions. This transition represents the conflict zone and Fig. 3 shows its layout. When transition t_2 or t_3 is fired, a token is moved into place 2. It means that an UAV transits to cell 2. Conflicts arise when the UAVs from both orbits s try to occupy the same shared cell $\{2\}$ at the same time. In this situation, the transitions t_1 and t_3 are in conflict (both transitions are enabled at the same time). A conflict-free supervisor enables only one orbit using transition priorities. In this case, high-priority transition t_2 is used to model the priority of the first orbit. For transitions t_2 and t_3 , the time inscriptions use user-defined function (*if $s = z$ then 2 else 4*). It is used to avoid the detected collision in cell 2. It means, for example, that if cell 2 is occupied by a UAV from orbit 1, the velocity of UAV from orbit 2 is decreased, and transition t_3 is enabled after 4 time units. The places denoted as $S1$ and $S2$ are added to places 2 and 3. The colours of tokens in these places correspond to green and red traffic lights, and they denote the availability of conflict places 2 and 3.

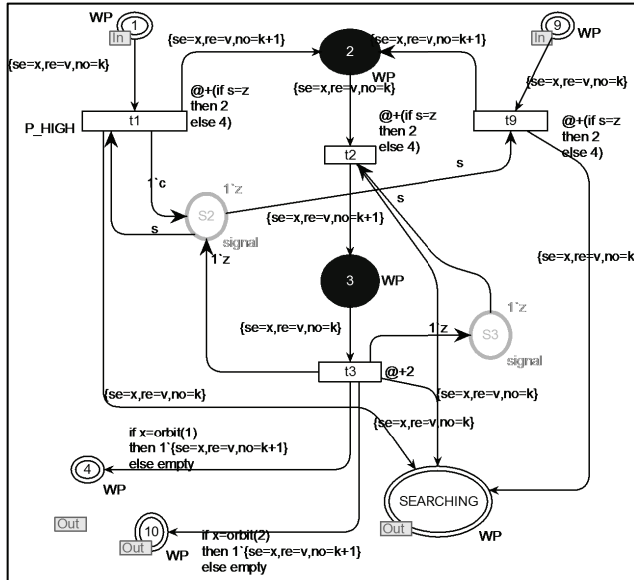


Fig. 3. CPN subnet of UAV conflict zone

Source: own elaboration.

5. Case study II: striking targets and strategy of selecting ships

While a UAV is sweeping across the search area, it may discover targets. In that situation (Fig. 4), an UAV sends signal to the control centre to help an object in need (transition *Find_target* is enabled when the guard is satisfied).

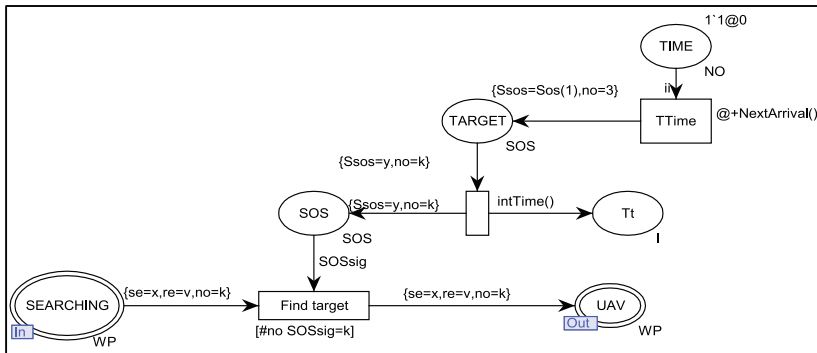


Fig. 4. CPN subnet of UAV conflict zone

Source: own elaboration.

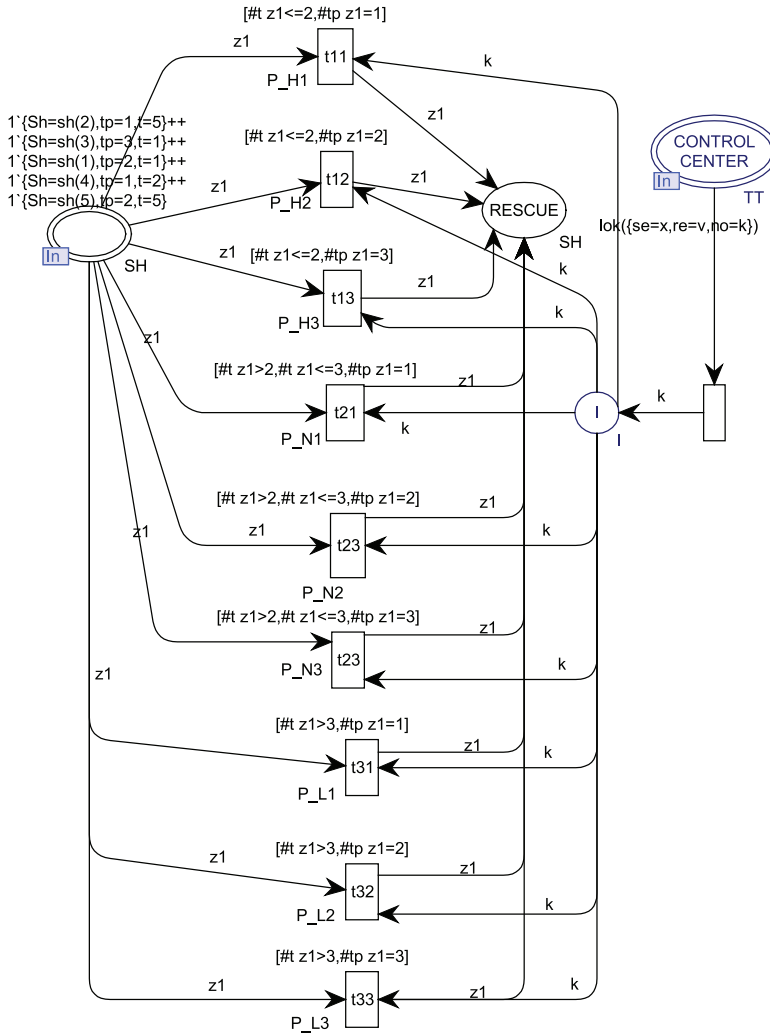


Fig. 5. Implemented control policy to select ships

Source: own elaboration.

Fig. 5 describes the strategy of selecting a suitable ship for saving the crew of the ship in distress. The key is setting the priority which ship will be selected to do the rescue operation. Depending on the rescue mission needs, the most suitable ship can be divided into three groups: group 1 – the most suitable ship, group 2 – suitable ship, and group 3 – the least suitable ship. The priorities should also be set regarding ship’s ETA (estimated time of arrival) to rescue position. Accordingly, the strategy is modelled as transitions t_{11} to t_{33} , where priority is given from P_{H1} to P_{L3} , which are the types of ships with

the highest priority, to P_L1 to P_L3 which are the ships with the lowest priority. If there are more ships with the same priority, then the ship with lower ETA to target is set to be chosen. It is solved by the guard function.

For example, the transitions t_{13} has P_H1 and priority and the guard $[\#t_{z1} \leq 2, \#tp_{z1} = 1]$. It means that this transition is enabled when the vessel is selected which has type 1 and it is far from a target, less than 2 time units.

Summary

We have presented a model for multi-UAV simulation using high level CPNs. PNs provide a rigid formalism of visualization for designing and tuning control structures of UAV simulations. The conflict resolution problem for multiple UAVs in a common airspace is studied. The proposed method avoids the conflicts by changing velocities and maintaining the space orbits. To avoid conflicts and apply the proposed method, the authors suggest adding control places (supervisors) and transition guards which disable firing of particular transitions and limit the number of UAVs in critical zones.

The redesign of the proposed UAV simulations indicates that the PN approach facilitates rapid prototyping of simulations.

References

- Alejo, D., Diaz-Banez, J.M., Cobano, J.A., P´erez-Lantero, P., Ollero, A. (2013). The velocity assignment problem for conflict resolution with multiple aerial vehicles sharing airspace. *Journal of Intelligent & Robotic Systems*, 69 (1), 331–346.
- Capar, R. (1989). *Traganje i spašavanje ljudi na moru*. Rijeka: Fakultet za pomorstvo i saobraćaj.
- IAMSAR Manual (2010). *Mobile Facilities*. Vol. 3. London–Montreal: IMO Publishing.
- Jensen, K. (1992). *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, vol. 2., Heidelberg: Springer.
- Kezić, D., Gudelj, A. (2010). Design of River System Deadlock Avoidance Supervisor By Using Petri Net. *PROMET*, 22 (3), 215–221.
- Maza, I., Caballero, F., Capitan, J., Martinez-de Dios, J., Ollero, A. (2011). A distributed architecture for a robotic platform with aerial sensor transportation and self-deployment capabilities. *Journal of Field Robotics*, 28 (3), 303–328.
- MERSAR Manual (1993). London: IMO Publishing.
- Murata, T. (1989). Petri Nets: Properties, Analysis, and Applications. *Proceedings of the IEEE*, 77 (4), 541–580.

Ribarić, S., Šnajder, J. (2005). Mapping Petri Net-Based Temporal Knowledge Representation Scheme into CP-Net Model. *Proceedings of the 28th International Convention: MIPRO 2005 Opatija*, 134–139.

Skrzypietz, T. (2012). *Unmanned Aircraft Systems for Civilian Missions*. Potsdam, Deutschland: Brandenburg Institute for Society and Security.

MODEL PLANOWANIA ORAZ KONTROLI DZIAŁAŃ POSZUKIWAWCZYCH I RATOWNICZYCH NA MORZU

Słowa kluczowe: UAV, sieci Petriego, unikanie kolizji

Streszczenie: Podczas akcji poszukiwawczych i ratowniczych na morzu coraz częściej wykorzystuje się bezzałogowe pojazdy powietrzne (UAV). Jednostki te wyposażone są w funkcje decyzyjne, które pozwalają im dotrzeć do podmiotu, który znajduje się w niebezpieczeństwie na morzu. UAV są uruchamiane w ciągu kilku minut. Akcja ratownicza rozpoczyna się po określeniu dokładnej lokalizacji uszkodzonej jednostki. Na podstawie zebranych informacji o położeniu statków, ich zwrotności i prędkości, centrum kontroli określa, który statek ma być zaangażowany w akcji ratowniczej. Ten bardzo autonomiczny system można określić jako dyskretny układ zdarzeń. Niektóre stany takich systemów, jak kolizje, są niepożądane.

W artykule przedstawiono implementację przepływu informacji do nadzoru, kontroli i monitorowania zachowania jednostek UAV podczas akcji wyszukiwania, której celem jest uniknięcie kolizji i właściwa komunikacja z pokładowym podsystemem obliczeniowym. Do określania różnych faz realizacji misji wykorzystano algorytmy planowania i sieci Petriego. Gdy UAV wykrywa uszkodzoną jednostkę, wywoływane są alternatywne zakodowane reakcje i centrum sterowania rozpoczyna wdrożenie planu ratunkowego.

Cite as: Medić, D., Gudelj, A., Krčum M. (2016). A formal model for planning and controlling search and rescue actions at sea. *Ekonomiczne Problemy Usług*, 124, 27–37. DOI: 10.18276/epu.2016.124-03.