

# Sensor Robot Planning in Incomplete Environment

Shan Zhong<sup>1</sup>, Zhihua Yin<sup>2</sup>, Xudong Yin<sup>1</sup>, Yufeng Yao<sup>1</sup>

<sup>1</sup>Computer Science and Engineering College, Changshu Institute of Technology, Changshu, China; <sup>2</sup>School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang, China.  
Email: sunshine620@csitg.edu.cn

Received February 15<sup>th</sup>, 2011; revised March 5<sup>th</sup>, 2011; accepted March 8<sup>th</sup>, 2011.

## ABSTRACT

*Aiming at the former formalized methods of robot planning should give the environment state, can not obtain the new knowledge of the environment. In order to improve the reason ability for obtaining new knowledge of the environment state, the actions in the process of planning such as external action and sensing action are formalized. A formalized reasoning method—CPNI (Colored Petri Net for Planning in incomplete environment) based on two kinds of actions is proposed, and the reasoning rule as Fluent Calculus in incomplete environment is applied. Robot planning experiment is modeled and simulated by using the tool CPNTools and the result shows the state knowledge of the door and the action sequence to reach the goal can be generated automatically in the CPNI net system.*

**Keywords:** Sensor, Robot Planning, Colored Petri Net, Action Sequence

## 1. Introduction

Robot planning [1] is the behavior planning process to reach the goal, it mainly includes two problems: generate the action sequence to reach the goal, and obtain the dynamic knowledge in the process of reasoning.

Petri net [2-3] is a formalized description tool and suitable model for modeling the system characterized by synchronism, dynamics, concurrency. Silva [4] converted the planning graph to the acyclic Petri net to plan, but can not represent the incomplete state knowledge. Vittorio [5] introduced a method for robot planning based on Petri net, formally describing actions and the relations between the actions, but it lacks the formal description for the state including the environment state and robot state, can not generate automatically the action sequence to the goal, and also can not represent the incomplete knowledge.

The above work can not realize the robot planning, namely, automatically generate the action sequence to the goal and obtain the knowledge in the process of the reasoning, finally realizing robot planning in incomplete environment.

In this paper, the sensor is introduced into the robot, converting the process of sensing the state to the sensing action of the according CPNI net system, through the sensing action, the robot can get the new knowledge of the world state, realizing the representation of the dynamic and unknown environment and then reasoning in an

incomplete environment. The Fluent Calculus [6-7], sensing action and the external action are introduced to the generating algorithm of the CPNI net system, the generated CPNI net system can not only realize robot planning and also the dynamic knowledge of the robot reasoning.

## 2. Reasoning Rule and the Example

### 2.1. Fluent Calculus

As for the Fluent Calculus, all changes to the world can be the result of named actions. A possible state history is a sequence of actions to reach the goal called situation. Fluent is for representation of the atomic properties of the physical world. The function State(s) denotes the state in situation s, which links the two key notions named state and situation.

Precondition axioms are used to formally specify the circumstances under which an action is possible in a state in situation s, denoted by predicate Poss (a, state(s)) showed in Equation (1), which means at the state state(s), the action A(x) can be executed. The  $\Pi_A(\bar{x}, z)$  as a pure state axiom about z is the conjunction of the conditions under which the action can be happened.

$$Poss(A(\bar{x})) \equiv \Pi_A(\bar{x}, z) \quad (1)$$

State update axioms defines the effects of an action a as the difference between the state prior to the action,

State(s), and the successor State(Do(a, s)), showed in Equation (2), in which  $\Delta(\bar{x}, z)$  is the pure state axiom representing the current state,  $v^+$  (the positive effect) represents the adding states, while  $v^-$  (the negative effect) represents the removed states, and  $state(s)$  represents the unchanged states and the unknown states.

$$\begin{aligned} & Poss(A(\bar{x}), s) \wedge \Delta(\bar{x}, State(s)) \\ \supset & State(Do(A(\bar{x}), s)) = State(s) \circ v^+ - v^- \end{aligned} \quad (2)$$

## 2.2. Example for Fluent Calculus

The robot example is showed as **Figure 1**. The round represents the robot. There are four rooms such as R501, R502, R503, R504, and alley. Every neighbor room is connected by a door, and the door also connects the neighbor room and alley, for example, D12 connects room R501 and room R502, and DA2 connects Alley and room R502.

The actions which the robot can do are as follows: go (door), enter(room) and open(door), the precondition axiom and the state update axiom are as follows:

1) The action go(x) means robot goes to the door x. The constraint is the door x' which the robot currently at is different from the goal door x, the constraint can be defined in the guard function of the transition. The precondition axiom and the state update axioms will be described showed in Equations (3) and (4):

$$Poss(go(x), z) \triangleq In(r) \wedge At(x') \wedge C(x1, r, \_) \wedge C(x, r, \_) \quad (3)$$

The Equation (3) represents the preconditions of the action go(x),  $In(r) \wedge At(x')$  denotes the robot is at the door x' of the room r,  $c(x1, r, \_)$  represents the door x1 connects the room x1 and  $\_$ .

$$\begin{aligned} & Poss(go(x), s) \wedge Holds(In(r) \wedge At(x'), s) \\ \supset & State(Do(go(x), s)) \triangleq In(r) \wedge At(x') \\ & \wedge C(x1, r, \_) \wedge C(x, r, \_) \circ At(x) - At(x') \end{aligned} \quad (4)$$

The Equation (4) represents the successor state of action

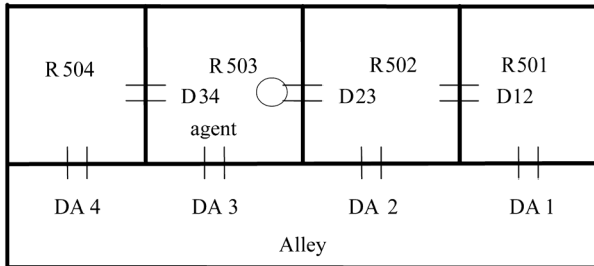


Figure 1. Robot planning in the incomplete scene.

go(x), the predicate  $c(x', r, \_)$  represents the unchanged state, the adding state  $v^+$  is represented by the predicate  $At(x)$  The removed state  $v^-$  is represented by the predicate  $At(x')$ , and it means the robot is not at the door x' of the room r.

2) The action enter(r) represents the robot enter the room r, the constraint is the current room r' which agent in is different from the goal room r. The precondition axiom and the state update axioms are showed as Equations (5) and (6):

$$Poss(enter(r), z) \triangleq In(r') \wedge At(x) \wedge C(x, r, r') \quad (5)$$

The Equation (5) represents the precondition of the action enter(r), namely, the robot is at door x in the room r' and the door is closed.

$$\begin{aligned} & Poss(enter(r), s) \wedge Holds(In(r') \wedge At(x), s) \\ \supset & State(Do(enter(r), s)) \triangleq In(r') \wedge At(x) \\ & \wedge C(x, r, r') \circ \neg Closed(x) - Closed(x) \end{aligned} \quad (6)$$

The successor state of the action enter(r) showed in Equation (6) are as follows: The adding state  $v^+$  is  $\neg closed(x)$ , the deleted state is  $closed(x)$ .

3) The action open(x) represents opening a door. The precondition axiom and the state update axiom are showed as Equations (7) and (8), respectively.

$$Poss(Open(x), z) \triangleq closed(x) \wedge key(x) \quad (7)$$

The Equation (7) represents the preconditions of the action: the predicate closed(x) represents the door x is closed and the robot has the key to the door x.

$$\begin{aligned} & Poss(open(x), s) \wedge Holds(Closed(x), s) \\ \supset & State(Do(open(x), s)) \triangleq Closed(x) \wedge Key(x) \circ \\ & \neg Closed(x) - Closed(x) \end{aligned} \quad (8)$$

The Equation (8) represents the state update axiom of action open(x), the adding state  $v^+$  is  $\neg closed(x)$ , the removed state  $v^-$  is  $closed(x)$ .

4) The action *check\_true(x)* and *check\_false(x)* represents sensing the state of the door. The precondition axiom and the state update axiom are showed as Equations (9) and (10), respectively.

$$\begin{aligned} & Poss(check\_true(x) \vee check\_false(x), z) \\ \triangleq & \neg(closed(-x) \wedge closed(x)) \end{aligned} \quad (9)$$

The Equation (9) shows the precondition of the action *check\_true(x)* and *check\_false(x)* is that robot do not know the state of door x.

$$\begin{aligned}
 & Poss(enter(r),s) \\
 & \supset State(Do(check\_true(x) \vee check\_false(x),s)) \\
 & \triangleq sense(\neg Closed(x)) \\
 & = true \wedge \neg Closed(x) \wedge sense(\neg Closed(x)) \\
 & = false \wedge Closed(x)
 \end{aligned} \tag{10}$$

The Equation (10) denotes shows the successor state of action enter(r), if the robot senses the door is closed, then the adding state  $v^+$  is  $closed(x)$ , else the adding state is  $\neg closed(x)$ .

### 3. CPNI Net System

#### 3.1. The Introduction for CPNI Net

**Definition 1** A CPNI net system is a 7-tuple system  $\Sigma = (P, T_v; F, C, I^-, I^+, M_0)$ , which is a timed colored Petri net system has the following characteristics:

- 1) The place  $p_i \in P$  represents the states place and the auxiliary place, the state place includes both the state of robot and the environment state, the auxiliary place is playing an auxiliary role in the robot planning such as the sensor place, the action sequence place and the goal place.
- 2) The transition  $t \in T$  represents the actions which the robot has the ability to do. The actions include external action and sensing action.
- 3)  $M_0$  is the initial state represents the initial state of the agent and the environment.

#### 3.2. The Representation of External Action

**Definition 2** The external action in the robot planning can be represented as **Figure 2**. The place  $p_{i1}, p_{i2}, \dots, p_{ik}$  denotes the removed states,  $p_{o1}, p_{o2}, \dots, p_{om}$  denotes the adding states,  $p_{io1}, p_{io2}, \dots, p_{io}$  denotes the states represented by the same place including the unchanged states (the place and the token are the same before and after action happened) and the changed states (the place is the same and the token is changed).

#### 3.3. The Representation and the Sensing Action

**Definition 3** The internal action namely the sensing action is showed as **Figure 3**, the place  $P\_Sense$  denotes the existing of the sensor, according to the different sensors, there are different sensing actions. If there are tokens in place  $P\_Sense$ , then the robot can do the sensing action, the transition  $check\_true(x)$  represent when the robot sensing the state is true, then a token is flowed to the place  $Check\_state$ , while the transition  $check\_false(x)$  represents when the robot sense the state is holding, then a negative token is flowed to the place  $Check\_state$ .

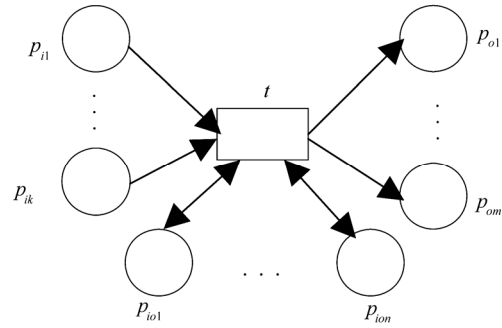


Figure 2. Representation of external action.

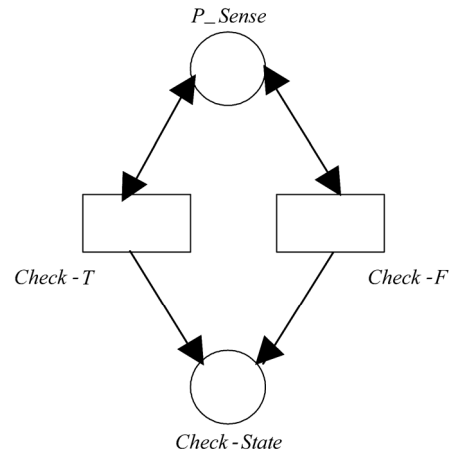


Figure 3. Representation of sensing action.

### 4. The Constructing Algorithm for CPNI Net System

The input of algorithm: the precondition axiom and the successor state axiom, the initial state and the goal state.

The output of algorithm: the robot planning CPNI net system.

**Step 1** Define  $P, T, F, C, I^-, I^+, M_0$  as set,  $P$  is the collection of places,  $p_{ActionSequences} \in P$  stores action sequence,  $p_{Goal} \in P$  stores the goal of agent,  $T$  is the collection of timed transition,  $F$  is the collection of arc,  $I^-$  and  $I^+$  are the arc set,  $M_0$  is the initial marking converting from initial state, for the convenient, the subset  $F1$  of  $F$  is defined.

Define  $i$  as a counter, the initial value of  $i$  is 1

$$P = P \cup p_{ActionSequences} \cup p_{Goal};$$

**Step 2** Loop the implementation of the following part until all the preconditions and successor states of the action have already been constructed, and when the  $i$ -th action is executing,

$$T = T \cup \{t_i\}; //Action is represented by transition$$

$$F1 = F \cup \left\{ (t_i, p_{ActionSequences}), (p_{ActionSequences}, t_i), (t_i, p_{Goal}), (p_{Goal}, t_i) \right\};$$

if  $t_i$  is the external action, then  
 if  $p_{ik}$  is the removed state, then

$$P = P \cup \{ p_{ActionSequences}, p_{Goal}, p_{ik} \};$$

$$F = F1 \cup \{ (p_{ik}, t_i) \};$$

else if  $p_{ik}$  is the unchanged state then

$$P = P \cup \{ p_{ActionSequences}, p_{Goal}, p_{ik} \};$$

$$F = F1 \cup \{ (t_i, p_{ik}), (p_{ik}, t_i) \};$$

else if  $p_{ik}$  is the adding state then

$$P = P \cup \{ p_{ActionSequences}, p_{Goal}, p_{ik} \};$$

$$F = F1 \cup \{ (p_{ik}, t_i) \};$$

end if

else if  $t_i$  is a sensing action then

$$P = P \cup \{ p_{ActionSequences}, p_{Goal}, p_{sense} \};$$

$$F = F1 \cup \{ (p_{sense}, Check\_F), (p_{sense}, Check\_T), (Check\_F, p_{sense}), (Check\_T, p_{sense}), (Check\_T, Check\_State), (Check\_F, Check\_State) \};$$

end if

Step 3  $i = i + 1$ , if all the actions have been iterated for once, then go to the Step 4, else go the Step 2

Step 4 After the unchanged state, the adding state and the removed state of all the actions are constructed, according to the initial states of system, the token is added to the relative place then the initial marking  $M_0$  is obtained. In the following part, the CPNI model of the entire system will be constructed in the example.

### 5. Simulation Experiment

The initial state for the robot is showed as **Figure 1**, the robot is at the door "D23" of the room "R503" and has the key of door "DA3". The doors "DA3" and "D23" are closed. The initial state are as follows:  $M_0 = \{In("R503") At("D23"), Key\{"DA3"\}, Closed\{"D23","DA3"\}, C\{"DA1","R501","Alley"\}, ("DA2","R502","Alley"), ("D23","R502",$

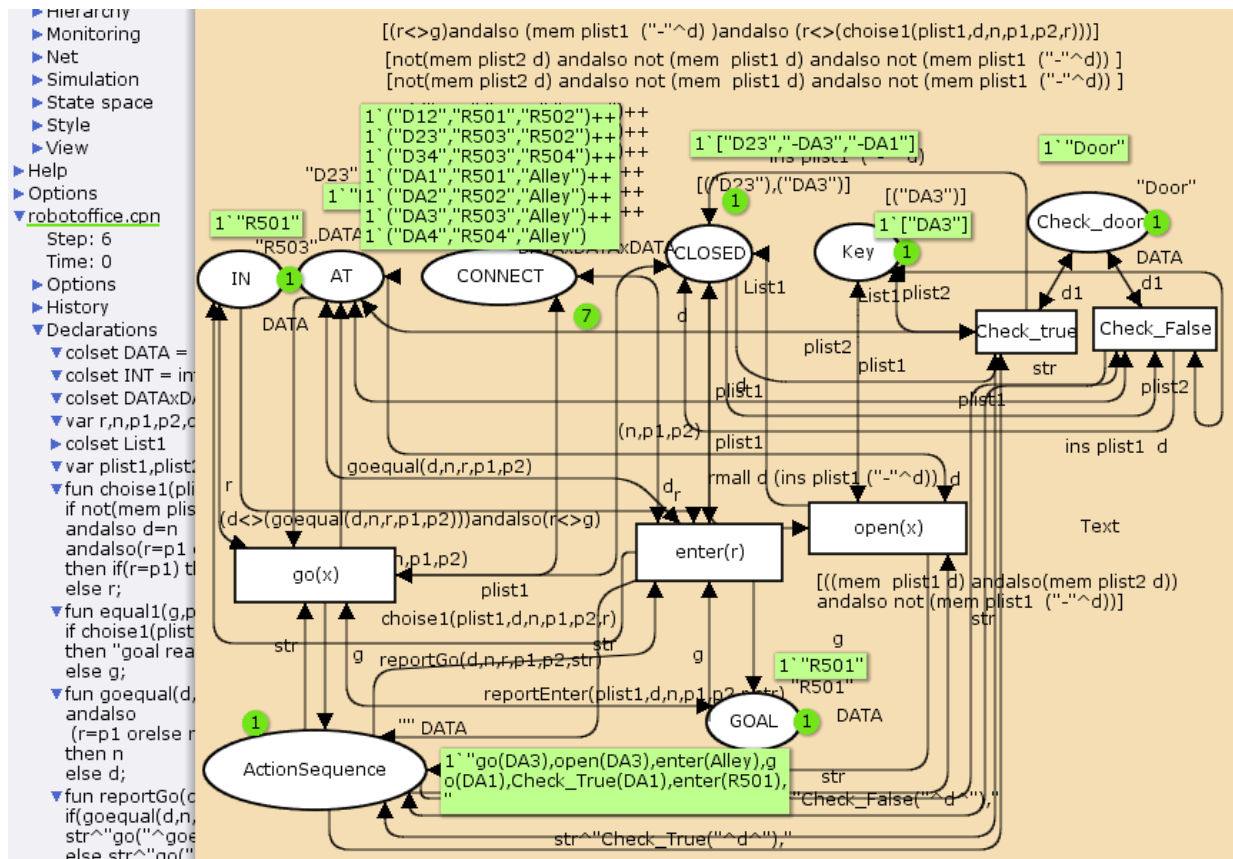


Figure 4. Simulation figure for the office scene.

“R503”), (“D34”, “R503”, “R504”), (“DA2”, “Alley”, “R502”), (“DA3”, “Alley”, “R503”), (“DA4”, “Alley”, “R504”) } }.

The transition  $go(x)$ ,  $enter(r)$  and  $open(x)$  represents the action of robot such as  $go(x)$ ,  $enter(r)$  and  $open(x)$ , respectively. The internal action needs two transitions to denote, the transition  $check\_true(x)$  denotes the executing action when the sensing state is true, while the  $check\_false(x)$  denotes the executing action when the sensing state is false.

The goal for robot is in room “R501”, so the four places to represent the goal marking is:  $M_g = In(“R501”), D, D, D \}$ , in which D is any value.

Using the constructing algorithm of CPNI net system for the office instance, the simulation result is showed in **Figure 4** using CPNTools [8].

The simulation result is shown in **Figure 4**, the CPNI net system is at the end state, the token in the place In is “R501”, the token in the place AT is “DA1”, and the robot has realized the goal.

The action sequence showed in place *Action-Sequence* is showed as: 1“go(DA3),check\_true(DA3),enter(Alley),go(DA2),check\_true(DA2),enter(R502),go(D12),check\_false(D12),go(DA2),enter(Alley),go(DA1),open(DA1),enter(R501)”,

From the action sequence place we can conclude: when the sensor robot go to the unknown state door such as “DA3”, the robot will use the sensor to sense the door state, and get the state of the door is not closed, so the state of door “DA3” is updated, and a token “-DA3” is flowed to the place Closed. When the robot go to the door D12, then the sensing action  $check\_false(D12)$  happened, so the robot can not go to the goal room through the door, therefore, the robot re-plan to the door D12, and finally searched a path from the initial state to the goal state.

## 6. Conclusions

This paper does some tries in applying sensor in the robot planning and resolves the robot planning in incomplete environment. Compared to the former work in the

literature [3-5,9], our paper does some tries to represent robot planning in incomplete environment, which is realized by divided the actions to two sorts, one is external action which the robot used to change itself and environment state, and the other is the sensing action which the robot used to sense the unknown environment state. The next work is to use hierarchical Petri net to represent robot planning to solve the state explosion problem.

## REFERENCES

- [1] R. Reiter, “Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems,” MIT Press, London, 2001.
- [2] C. Y. Yuan, “The Principle and Application of Petri Net,” Electronic Industry Press, Beijing, 2005
- [3] P. F. Palamara, V. A. Ziparo, L. Iocchi, *et al.*, “A Robotic Soccer Passing Task Using Petri Net Plans,” *Proceedings of 7th International Conference on Autonomous Agents and Multiagent Systems*, Estoril, 2008, pp. 1711-1712.
- [4] F. Silva, M. Castilho and L. Kunzle, “Petriplan: A New Algorithm for Plan Generation,” *Proceedings of IBERAMIA/SBIA*, Spinger-Verlag, Brazil, 2000, pp. 86-95.
- [5] V. A. Ziparo and L. Iocchi. “Petri Net Plans,” *Fourth International Workshop on Modelling of Objects, Components, and Agents*, Roma, 2006, pp. 267-290.
- [6] Y. Jin and M. Thielscher, “Iterated Belief Revision, Revised,” *Artif Intell*, Vol. 171, No. 1, 2007, pp. 1-18. doi:10.1016/j.artint.2006.11.002
- [7] S. Sardina, G. De Giacomo, Y. Les-Perance, *et al.*, “On the Semantics of Deliberation in Indigolog-from Theory to Implementation,” *Annals of Mathematics and Artificial Intelligence*, Vol. 41, No. 2-4, 2004, pp. 259-299.
- [8] CPNTools, “A Computer Tool for Colored Petri Nets,” 2008. <http://www.daimi.au.dk/CPNTools/>
- [9] Y. S. Liu, S. Zhong and Y. Z. Zhan, “A Model for Representing Reasoning about Actions Based on Colored Petri Net,” *Journal of Jiangsu University*, Natural Science Edition, Vol. 31, No. 3, 2010, pp. 335-338.