# ДИСКРЕТНО-СОБЫТИЙНАЯ МОДЕЛЬ ГРУППОВОГО УПРАВЛЕНИЯ МОБИЛЬНЫМИ РОБОТАМИ

**Потехин А.И.**

*Институт проблем управления им. В.А. Трапезникова РАН*

branishtov @mail.ru, an_pot@mail.ru

*Аннотация. Разрабатывается дискретно-событийная система группового управления автономными мобильными роботами. Исходная система управления роботом задана множеством компонент нижнего уровня (движение к цели, обход препятствий и т. д.). Координация и синхронизация компонент осуществляется с помощью связей соответствующих конечных автоматов. Каждой компоненте находится множество компонент - постусловий и множество компонент – предусловий, что изображается в виде фрагментов сети Петри. Соединение фрагментов образует сеть Петри, которая является дискретно-событийной системой моделирования и управления верхнего уровня.*

Ключевые слова: автономный робот, дискретно-событийная модель группового управления верхнего уровня, сеть Петри.

## Введение

Исследование методов и стратегий группового движения автономных мобильных роботов в загруженных препятствиями пространствах рассматривалось во многих статьях [1]. При этом система управления мобильным роботом, вне зависимости от его функционального назначения, рассматривалась состоящей из отдельных взаимодействующих компонент, содержащих описание процедур непрерывных процессов (например, обход препятствия, движение к цели и т. д.). [2]. Каждая компонента такой системы (системы нижнего уровня) содержит формальную процедуру общения с внешней средой робота (например, в виде программы или алгоритма). Очень важно: функционирование каждой компоненты можно точно описать независимо от других компонент, за исключением взаимодействия компонент друг с другом. Поэтому, в первую очередь должна быть решена задача их взаимодействия (координация и синхронизация) в зависимости от функционального назначения робота (робот – лидер или робот – последователь). Робот – лидер или робот – последователь отличаются лишь составом компонент и назначением. Поведение каждой компоненты определяется через возможные состояния компоненты и переходы между ними. Для этого строится конечный автомат, управляющий поведением компоненты. В простейшем случае поведение компоненты представляется конечным автоматом с двумя состояниями компоненты: рабочее и нерабочее. Однако, в этом случае в сложной сети автоматов могут возникать ложные циклы, ловушки, клубы [3]. Поэтому первой задачей данного исследования была разработка типового автомата, управляющего поведением широкого класса компонент, входящих в систему управления роботом нижнего уровня. Для каждой компонент определяются компоненты – последователи и компоненты – предшественники. Компоненты – последователи относительно данной компоненты это те, процедуры которых начинают функционировать только после окончания функционирования процедуры данной компоненты. Компоненты – предшественники это те, окончание процедуры которых являются причиной начала функционирования данной компоненты. Компоненты – последователи и компоненты – предшественники для каждой компоненте определяются исходя из роли робота, заданного поведения робота в различных ситуациях. Такое взаимодействие компонент хорошо представляется сетью Петри: каждой компоненте соответствует позиция сети Петри и соответствующие ей позиции – постусловия и позиции – предусловия [4]. Таким образом, связь каждой компоненты системы робота с компонентами – последователями и предшественниками можно изобразить в виде соответствующих фрагментов сети Петри. Объединение фрагментов образует управляющую и при необходимости моделирующую сеть Петри робота. По сети Петри строятся связи между управляющими автоматами компонент системы нижнего уровня.

Таким образом, построение дискретно-событийной системы группового управления автономными мобильными роботами состоит в построении управляющей сети Петри робота – лидера и управляющей сети Петри роботов – последователей.

## 1 Управляющий автомат

Построим конечный автомат, управляющий поведением широкого класса компонент системы управления роботом нижнего уровня. Исходим из того, что каждая компонента может находиться в одном из следующих состояний: начальном (процедура не активна), рабочем (осуществляется процесс обработки входной информации и выдача управляющих воздействий на исполнительные механизмы робота), в состоянии окончания работы процедуры (например, цель достигнута, обход препятствия закончен).

На рисунке 1 изображен граф переходов автомата, содержащий состояния и переходы некой компоненты, обозначим ее как *A*. Состояние $a_1$ – начальное состояние компоненты. Состояние $a_2$ – рабочее состояние компоненты, в этом состоянии выполняется соответствующая процедура (например, обход препятствия, движение к цели и т. д.). Состояние $a_3$ – состояние окончания работы процедуры. Возможно, что процедура имеет два и более различных состояния окончания работы. Так на рисунке 1 показаны два состояния окончания работы - $a_3^1$ , $a_3^2$ .

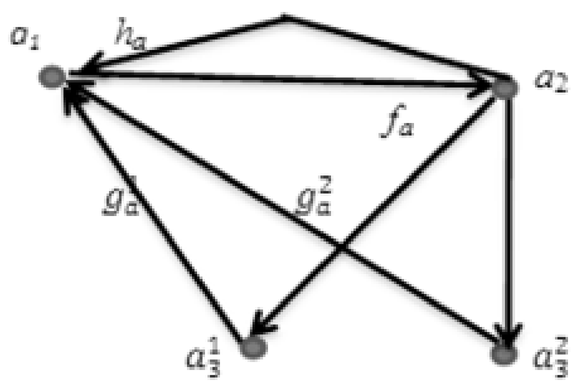



*Рис. 1. Граф переходов автомата, моделирующий поведение компоненты А робота*

Важнейшим элементом автомата является состояние $a_3$, это короткое по времени состояние по сравнению с длительностью работы процедуры в состоянии $a_2$. В теории дискретно-событийных систем (DES) это состояние соответствует неуправляемому ожидаемому событию. Сопоставим состоянию $a_3$ логическую переменную *а.* Для состояния $a_2$ также определим логическую переменную, обозначим ее для простоты как $a_2$. Функция $f_a$ – логическая функция запуска процедуры из начального состояния. Функции $g_a^1$, $g^2_a$ - логические функции перевода компоненты в начальное состояние. Функция $h_a$ - логическая функция перевода компоненты в начальное состояние $a_1$ из состояния $a_2$. Функция $h_a$ принимает единичное значение в случае необходимости прервать работу процедуры, например, при несовместимости с одновременной работой других процедур, или при установке всей системы в исходное состояние.

## 2 Варианты взаимодействия компонент системы нижнего уровня робота

Рассмотрим пример взаимодействия двух компонент *А* и *В*. *Б*удем говорить, что компоненты *А* и *В* содержат соответствующие процедуры *А* и *В*. Пусть выполняется процедура *А,* после окончания ее выполнения запускается процедура компоненты *В*, затем после окончания работы процедуры *В* снова запускается процедура *А*, затем процесс повторяется.

Обозначим этот процесс как последовательность действий в виде линейного алгоритма $A \to B \to A$ [3]. Автомат компоненты *А* имеет состояния $a_1$, $a_2$ , $a_3$., автомат компоненты *В* - состояния $b_1$, $b_2$, $b_3$.. Рассмотрим часть этого процесса, а именно фрагмент $A \to B$. Пусть исходное состояние автоматов будет ($a_2$, $b_1$), что соответствует рабочему состоянию процедуры *А* и начальному состоянию процедуры *В*. После окончания процедуры *А* состояние автоматов будет ($a_3$ , $b_1$), затем после запуска процедуры *В* состояние автоматов будет ($a_3$, $b_2$), наконец, процедура *А* переходит в начальное состояние и состояние автоматов будет ($a_1$ , $b_2$). Таким образом, имеет место последовательность состояний автоматов:

$$(a_2 , b_1), (a_3 , b_1), (a_3, b_2), (a_1, b_2)$$

Из подпоследовательности ($a_3$, $b_1$), ($a_3$, $b_2$) находим функцию $f_b$, из подпоследовательности ($a_3$, $b_2$), ($a_1$, $b_2$) - функцию $g_a$:

$$f_b = a, g_a = b_2.$$

Аналогично, последовательность состояний автоматов фрагмента $B \to A$ имеет вид:

$$(a_1, b_2), (a_1, b_3), (a_2, b_3) , (a_2, b_1),$$

из которой находим функции $f_a$ и $g_в$:

$$f_a = b, g_b= a_2.$$

Процесс получения функций *f, g* автоматов *А* и *В* может быть более простым, если эту последовательность действий изобразить в виде сети Петри (рис. 2).

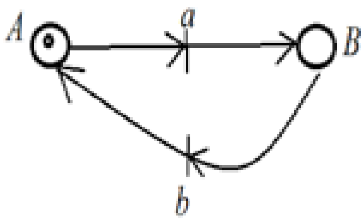



*Рис.2. Маркированная сеть Петри последовательности A→B→A*

Процесс перемещение фишки описывается в терминах сети Петри так: при выполнении предусловия *A* (фишка в позиции *A*) и при $a$ = 1 выполняется переход, и фишка из позиции *A* переходит в позицию - постусловие *B*, получаем функции

$$f_b = a,\ g_a = b_2.$$

Аналогично, при выполнении предусловия *B* (фишка в позиции *B*) и при $b$ = 1 выполняется переход, и фишка из позиции *B* переходит в позицию - постусловие *A*, получаем функции

$$f_a = b,\ g_b = a_2.$$

Рассмотрим более сложный случай последовательности процедур: после окончания процедуры *A* происходит одновременный запуск процедур *B* и *C*, обозначим этот фрагмент как *A* → *B C*. Сеть Петри этого фрагмента показан на рис.3.

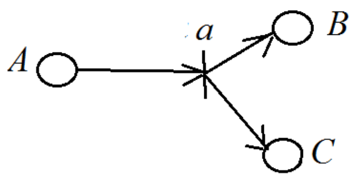



*Рис.3. Сети Петри фрагмента A → B C*

Предполагая, что позиция *A* содержит фишку, строим функции *f*, *g* автоматов *A, B , C*:

$$f_b = a,\ f_c = a,\ g_a = b_2\ c_2$$

Возможна ситуация, когда после окончания процедуры *A* происходит запуск процедуры *B* или *C*, обозначим это как *A* → ( *B* ∨ *C*). Сеть Петри этого фрагмента показан на рис.4.

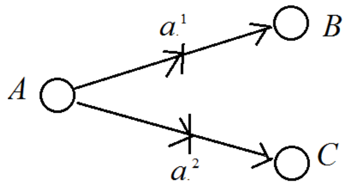



*Рис.4. Сети Петри фрагмента A → ( B ∨ C)*

В этом случае автомат компоненты *A* имеет два состояния окончания - $a^1$ и $a^2$. Соответствующие функции *f*, *g* автоматов компонент *A, B, C* имеют вид:

$$f_b = a^1,\ f_c = a^2,\ g^1{}_a = b_2,\ g^2{}_a = c_2$$

Общий случай: *A*→ (*B* C ∨ Д E):

$$f_b = f_c = a^1,\ f_д = f_e = a^2,\ g^1{}_a = b_2\ c_2,\ g^2{}_a = д_2\ e_2$$

Рассмотренные примеры соответствуют фрагментам взаимодействия компоненты *A* с другими компонентами, при этом процедура *A* рассматривалась как процедура - предусловие.

Рассмотрим фрагменты, в которых процедура *A* является процедурой - постусловием. На рисунке 5 показана сеть Петри фрагмента ( *B C* → *A*), который означает окончание работы обеих процедур B и C запускает процедуру *A*, и фрагмента ( *B* ∨ *C* → *A*), который соответствует окончанию работы процедур *B* или *C* и запуск процедуры *A*.

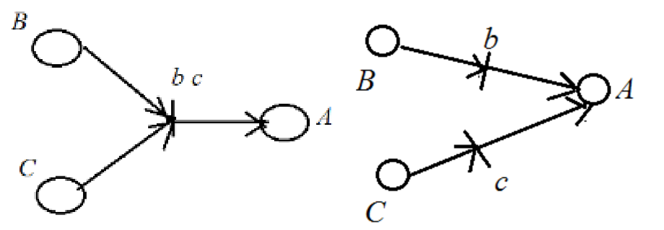



*Рис.5. Сеть Петри фрагментов $B\ C \rightarrow A$ и $(B \vee C) \rightarrow A$*

Из фрагмента $B\ C \rightarrow A$ находим: $f_a = b\ c$, $g_b = a_2$, $g_c = a_2$.

Из фрагмента $(B \vee C) \rightarrow A$ находим: $f_a = b \vee c$, $g_b = a_2$, $g_c = a_2$.

## 3 Дискретно-событийная система управления верхнего уровня робота - лидера

Множество компонент робота – лидера взяты из [2]:

1. Планирование подцели - $A$
2. Движение к цели - $B$
3. Поиск препятствий - $C$
4. Обход сложного препятствия - $D$
5. Обход препятствия типа «стена» - $E$

Как отмечалось выше, компоненты – последователи и компоненты – предшественники для каждой компоненте системы робота определяются исходя из роли робота и предполагаемого его поведения в различных ситуациях. В нашем случае возможные фрагменты взаимодействия компонент робота – лидера представлены в таблице 1.

Пусть после запуска системы управления робота - лидера начинается работа компоненты $A$ (планирование подцели), остальные компоненты находятся в начальном состоянии.

**Фрагмент № 1.** Компонента $A$ находится в рабочем состоянии (планирование подцели). После определения текущей подцели ($a$ = 1) запускаются компоненты $B$ (движение к цели) и $C$ (поиск препятствий). В таблице 1 этот фрагмент представлен строкой 1. Функции $f$, $g$ автоматов $A$, $B$, $C$ следуют из рис.3.

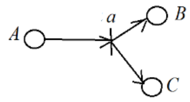
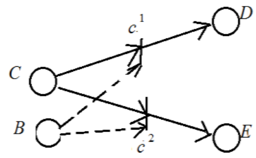
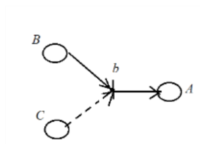
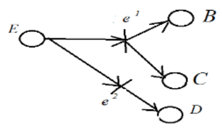
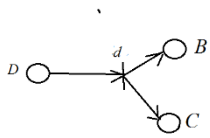
**Фрагмент № 2**, в таблице 1 - строка 2. Компонента $C$ находится в рабочем состоянии (поиск препятствий). В результате поиска препятствий может быть обнаружено либо «сложное препятствие» либо препятствие типа «стена». В первом случае автомат процедуры $C$ окажется в состоянии $c^1_3$ ($c^1$ = 1), во втором случае - $c^2_3$ ($c^2$ = 1). В первом случае будет запуск процедуры $D$ (обход сложного препятствия), во втором случае - запуск процедуры $E$ (обход препятствия типа «стена»). Таким образом, имеем фрагмент $C \rightarrow D \vee E$ (смотри рис.4). В тоже время очевидно, что в обоих случаях необходимо остановить выполнение процедуры $B$, так как процедура $B$ (движение к цели) не совместима с процедурами $D$ и $E$. Этот факт изобразим как $C \rightarrow D$ (B) $\vee$ $E$ (B). В сети Петри этот фрагмент изображен так: позиция $B$ соединена с переходами пунктирными стрелками. Как было отмечено выше, в случае необходимости прерывать работу процедуры, например, при несовместимости с одновременной работой других процедур, надо перевести управляющий автомат этой процедуры в начальное состояние с помощью функции $h$. В нашем случае для перевода автомата процедуры $B$ в начальное состояние из состояния $b_2$ строим функцию $h_b = d_2 \vee e_2$. Функции $f$, $g$ автоматов $C$, $D$, $E$ следуют из рис.4.

**Фрагмент № 3**, в таблице 1- строка 3. Компонента $B$ находится в рабочем состоянии (движение к цели). После достижения цели ($b$ = 1) запускается компонента $A$ (планирование подцели) и прерывается работа компоненты С (поиск препятствий).

**Фрагмент № 4**, в таблице1- строка 4. Компонента $E$ находится в рабочем состоянии (обход препятствий типа «стена»). Имеет место два окончание работы этой процедуры. После окончании обхода препятствия типа «стена» ($e^1$ = 1) одновременно запускаются компоненты $B$ (движение к цели) и $C$ (поиск препятствий). При обнаружении «сложного препятствия» ($e^2$= 1) запускается компонента $D$ (обход сложного препятствия). Функции $f$, $g$ определяются из рис. 5.

**Фрагмент № 5**, в таблице1 - строка 5. Компонента $D$ находится в рабочем состоянии (обход сложного препятствия). После успешного обхода препятствия ($d$ =1) запускаются компоненты $B$ (движение к цели) и $C$ (поиск препятствий).

*Таблица 1. Фрагменты взаимодействия компонент робота – лидера*

| | Фрагмент взаимодействия компонент робота - лидера | Фрагменты сети Петри | Функции управляющих автоматов *f, g, h* |
|---|---|---|---|
| | $A \rightarrow B\ C$ |  | $f_b = a$,<br>$f_c = a$,<br>$g_a = b_2\ c_2$ |
| | $C \rightarrow D\ (B) \vee E\ (B)$ |  | $f_d = c^1$, $f_e = c^2$,<br>$g^1_c = d_2$, $g^2_c = e_2$<br>$h_b = d_2 \vee e_2$ |
| | $B\ (C) \rightarrow A$ |  | $f_a = b$,<br>$g_b = a_2$,<br>$h_c = a_2$. |
| | $E \rightarrow B\ C \vee D$ |  | $f_b = e^1$ ,<br>$f_c = e^1$,<br>$f_d = e^2$<br>$g_e = b_2\ c_2 \vee d_2$ |
| | $D \rightarrow B\ C$ |  | $f_b = d$,<br>$f_c = d$,<br>$g_d = b_2\ c_2$ |

Объединение фрагментов таблицы 1 образует управляющую и при необходимости моделирующую сеть Петри робота - лидера (рис. 6).

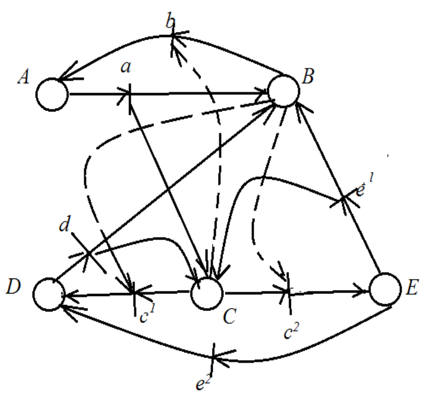



*Рис.6. Управляющая и моделирующая сеть Петри робота - лидера*

Из таблицы 1 логическим сложением по всем фрагментам нетрудно получить функции $f, g, h$ управляющих автоматов:

Автомат A: $f_a = b \vee u$, $g_a = b_2 c_2$

Автомат B: $f_b = a \vee e^1 \vee d$, $g_b = a_2$, $h_b = d_2 \vee e_2$

Автомат C: $f_c = a \vee e^1 \vee d$, $g^1_c = d_2$, $g^2_c = e_2$, $h_c = a_2$.

Автомат D: $f_d = c^1 \vee e^2$, $g_d = b_2 c_2$

Автомат E:, $f_e = c^2$, $g_e = b_2 c_2 \vee d_2$

Примечание. Запуск системы из начального состояния производиться сигналом $u$ (см. функцию $f_a$).

## 4 Дискретно-событийная система управления роботов - последователей

Множество компонент роботов – последователей взяты из [2] :

1. Поиск лидера - *A*,
2. Следование за лидером в группе роботов - *B*,
3. Поиск препятствий - *C*,
4. Обход сложного препятствия - *D*,
5. Обход препятствия типа «стена» - *E*

Нетрудно видеть, что при некоторых допущениях фрагменты взаимодействия компонент роботов – последователей повторяют фрагменты взаимодействия компонент робота - лидера. Поэтому можно считать, что управляющая сеть Петри робота - последователя совпадает с сетью Петри робота - лидера (рис. 6).

## Заключение

Разработана структура типового конечного автомата, управляющего поведением широкого класса компонент, входящих в систему управления робота нижнего уровня.

Определены варианты взаимодействия компонент системы управления роботом.

Для каждого варианта взаимодействия компонент определены правила построения соответствующего фрагмента сети Петри и правила получения функций переходов управляющих автоматов.

Построение дискретно-событийной системы управления верхнего уровня робота – лидера состоит в построении объединенной сети Петри.

Построение дискретно-событийной системы группового управления автономными мобильными роботами состоит в построении управляющей сети Петри робота – лидера и управляющей сети Петри роботов – последователей.

## Литература

1. *Браништов С.А., Харланова П.А., Байбакова О.А.* Проблемы движения мобильных роботов в скоплениях // Труды MLSD2018, Москва, ИПУ РАН, 2018г.
2. Gayan W. Gamage, *George K. I. Mann and Raymond G. Gosine* Discrete Event Systems based Formation Control Framework to Coordinate Multiple Nonholonomic Mobile Robots //The 2009 IEEE/RSJ International Conference onIntelligent Robots and SystemsOctober 11-15, 2009 St. Louis, USA
3. *Закревский А.Д.* и др., Логические основы проектирования дискретных устройств. // Физмат лит, 2007.
4. *Дж. Питерсон*. Теория сетей Петри и моделирование систем. М., Мир, 1984.г. 264 с.