

doi: 10.36724/2409-5419-2020-12-3-47-53

# ПРИМЕНЕНИЕ ВЕРОЯТНОСТНЫХ И ВРЕМЕННЫХ АВТОМАТОВ В ПРОГРАММАХ УПРАВЛЕНИЯ МНОГОАГЕНТНЫХ СИСТЕМ

**БЕЛЯЕВ**

**Сергей Алексеевич**

## АННОТАЦИЯ

Рассмотрены классические модели временного и вероятностного автомата, построенного с использованием теории цепей Маркова – система «автомат – случайная среда», описываемая цепью Маркова. Неопределенность вероятностного автомата может определяться разными причинами, в том числе – недетерминированностью правил изменения состояний моделируемой системы, что является одним из важнейших препятствий при моделировании многоагентных систем. Разработанная модель предназначена для программы управления многоагентных систем, учитывает стохастическое поведение окружающей среды при применении в условиях многоагентных систем. Представлены ключевые отличия классической стационарной случайной среды и многоагентных систем, в том числе рациональность поведения агентов и необходимость кооперации или противодействия. В модели не предусмотрены обучаемые алгоритмы, их применение требует большой выборки для проведения обучения, не во всех мультиагентных системах достаточно времени для набора необходимой статистики и адаптации поведения. Описаны подходы к построению архитектуры программы управления агента в многоагентной среде на основе теорий временных и вероятностных автоматов, предложен формат хранения моделей в файловых хранилищах. Предложенная архитектура учитывает в том числе возможность изменения состояния временных автоматов на основании состояния вероятностных автоматов и наоборот. Описан обобщенный алгоритм функционирования программы управления. Предложенный алгоритм позволяет реализовывать программы управления многоагентных систем в случае конечного количества состояний агента и использует гибкие механизмы управления, взятые из модели временного автомата. Приведены основные результаты и предложения по дальнейшим исследованиям, в том числе по построению иерархических моделей.

## Сведения об авторе:

к.т.н., доцент, доцент Санкт-Петербургского государственного электротехнического университета «ЛЭТИ», г. Санкт-Петербург, Россия, bserge@bk.ru

**КЛЮЧЕВЫЕ СЛОВА:** временной автомат; вероятностный автомат; случайная среда; программа управления; многоагентная система; архитектура; алгоритм.

**Для цитирования:** Беляев С.А. Применение вероятностных и временных автоматов в программах управления многоагентных систем // Наукомкие технологии в космических исследованиях Земли. 2020. Т. 12. № 3. С. 47-53. doi: 10.36724/2409-5419-2020-12-3-47-53

## Введение

Теория вероятностных автоматов разрабатывается многие десятилетия [1–2] и имеет множество применений в технических устройствах, работающих в стационарных случайных средах [3]. Параллельно с ней разрабатывается теория временных автоматов [4–5], которая позволяет учитывать изменение времени при вычислении переходов автомата из одного состояния в другое. Оба подхода активно используются при разработке многоагентных систем, имеют свои преимущества и недостатки. Существуют и другие способы описания поведения агентов, например, онтологические подходы [6], но они требуют более сложных описаний и в данной работе не рассматриваются.

## Временной и вероятностный автоматы

Временной автомат — это конечный автомат [7], расширенный набором вещественных таймеров (часов). При выполнении временного автомата значения всех таймеров увеличиваются с одинаковой скоростью. При выполнении переходов значения таймеров могут сравниваться с целочисленными константами, используемыми в качестве инвариантов (условий перехода), и, соответственно, разрешать или запрещать переходы. Значения таймеров могут сбрасываться.

С математической точки зрения временной автомат [4–5] — это система  $T = (L, l_0, A, C, P, E)$ , в которой  $L$  — конечное множество состояний,  $l_0$  — начальное состояние ( $l_0 \subseteq L$ ),  $A$  — конечное множество действий (алфавит автомата),  $C$  — конечное множество таймеров,  $P: L \rightarrow TC$  — множество допустимых состояний с учётом значения всех таймеров и соответствующих инвариантов,  $E: L \times TC \times A \times R(C) \times L$  — множество переходов. Каждый переход представляет собой кортеж  $\langle l_1, g, a, r, l_2 \rangle$  — переход из состояния  $l_1$  в состояние  $l_2$  с выполнением действия  $a$ , предусловием  $g$  и множеством сбрасываемых часов  $r$ . Для описания состояния автомата и возможных переходов необходимо не только знать текущее состояние, но и значения таймеров, оно описывается кортежем  $\langle l, v \rangle$ , где  $l$  — состояние,  $v: C \rightarrow R+$  — вещественные значения таймеров. В начальном состоянии  $\langle l_0, v_0 \rangle: v_0(c) = 0$  для всех таймеров  $C$ . Шаг вычисления временного автомата состоит в продвижении времени ( $\langle l, v \rangle \rightarrow \langle l, v+d \rangle$ ) и выполнении перехода ( $\langle l, v \rangle \rightarrow a \langle l', v' \rangle$ ). При продвижении времени не выполняется никакое действие, а значение всех таймеров увеличивается на заданную константу. Для выполнения перехода проверяются инварианты, выбирается один из возможных переходов, выполняется переход в новое состояние, при необходимости сбрасываются указанные таймеры.

Вероятностный автомат — это конечный автомат, функционирующий в дискретном времени [1], описываемый системой  $A = (X, Y, S, M, a_0)$ , в которой  $X$  — конечное множество входных символов,  $Y$  — конечное множество

выходных символов,  $S$  — конечное или счётное множество состояний,  $a_0$  — начальное состояние автомата,  $M = \langle \mu(a', y/a, x) \rangle$  означает условную вероятность перехода автомата из состояния  $a$  при входном сигнале  $y$  в состояние  $a'$  при выходном сигнале  $x$  [2]. Сумма условных вероятностей по всем переходам должна равняться единице. Все возможные переходы вероятностного автомата в случае постоянных условных вероятностей представляют в виде конечномерных или счетномерных матриц  $\{\mu_{ij} = \mu(a_j, y/a_i, x)\}$ . Для сравнения, в случае детерминированного автомата каждая строка матрицы условных вероятностей содержит одну единицу, остальные элементы равны нулю. Вероятностный автомат называется марковским автоматом (с детерминированной функцией выходов), если условное вероятностное распределение  $\mu(y/a, x) = \sum_{s \in S} \mu(s, y/a, x)$  принимает только значения 0 или 1, т.е. следующее состояние вероятностного автомата и выходной символ взаимно независимы, условное распределение последующего состояния цепи Маркова зависит от текущего состояния и не зависит от предыдущих состояний.

В [1] проводятся исследования и сравнения различных вариантов построения вероятностных автоматов, исходя из ограничения, что реакции среды на выходные символы автомата равны +1 (выигрыш) или -1 (проигрыш). Данный подход ограничивает применение описанных моделей в стационарных и переключаемых случайных средах. В случае стационарной случайной среды может быть заранее вычислено математическое ожидание выигрыша и проигрыша на одно состояние вперёд. Переключаемые случайные среды предусматривают переключение стационарных случайных сред в соответствии с цепью Маркова.

При рассмотрении многоагентных систем [8–9] они также могут быть представлены как случайные среды, в которых неопределённость связана с недетерминированностью правил изменения состояний моделируемой системы [10–11]. Однако при этом не учитывается рациональность поведения агентов, что может приводить к неожиданным закономерностям, связанным с целями, преследуемыми агентами, в том числе не всегда есть возможность вычисления выигрыша или проигрыша в каждый момент времени. Соответственно, вероятностные автоматы не применяются при проектировании агентов, которые должны уметь формировать коалиции [12].

## Модель программы управления

Программа управления агентом в многоагентных системах должна учитывать с одной стороны случайный характер поведения окружающей среды, с другой стороны — закономерности, возникающие в зависимости от целей, преследуемых рациональными агентами. Каждый рациональный агент стремится максимизировать значение своей функции полезности с учётом ограничений на доступные вычислительные ресурсы, доступную память и на время



принятия решений. При проектировании многоагентных систем, если не решается задача формирования коалиций, то для упрощения часто поведение агентов объединяется с поведением окружающей среды. В итоге рассматривается агент и окружающая среда как антагонист [13]. При этом поведение всех агентов усредняется и существенно усложняется решение целевой задачи за счёт невозможности учёта целевых функций участвующих агентов. Как оправдание этому может служить невозможность в общем случае предсказать поведение всех агентов в многоагентной системе, тем не менее некоторые нечёткие или вероятностные оценки поведения агентов могут быть даны. Анализ возможного поведения агентов независимо от окружающей среды с одной стороны упрощает описание самой среды, с другой стороны повышает точность принятия решений программой управления.

Модель программы управления на основе временных автоматов с использованием вероятностных автоматов для предсказания поведения агентов в многоагентной системе выглядит следующим образом:  $U=(L, I_0, A, C, P, E, Y, M)$ , в ней  $L$  — конечное множество состояний,  $L=(L_{in}, L_{out})$  — внутреннее состояние и состояние окружающей среды,  $I_0$  — начальное состояние ( $I_0 \subseteq L$ ).  $A$  — конечное множество действий (алфавит автомата),  $A = \{a_{in}, a_{out}\}$  — свои действия, действия остальных агентов.  $C$  — конечное множество таймеров,  $P: L \rightarrow TC$  — множество допустимых состояний с учётом значения всех таймеров и соответствующих инвариантов,  $E: L \times TC \times M \times A \times R(C) \times L$  — множество переходов,  $M = \langle \mu(I'_{out}, I_{out}, I_{in}, a_{out}) \rangle$  означает условную вероятность перехода автомата из состояния  $(I_{out}, I_{in})$  при выполнении действия  $a_{out}$  в состояние  $I'_{out}$ .

Состояния программы управления условно разделены на два вида: внутренние состояния, описывающие состояния агента в процессе управления, и внешние состояния, описывающие состояния окружающей среды (остальных агентов). Аналогичным образом разделены действия — внешние и внутренние. При этом используются все преимущества модели временного автомата, но дополнительно, при переходе из состояния в состояние учитывается условная вероятность перехода агентов окружающей среды в то или иное состояние.

Предлагаемая модель программы управления в отличие от временного автомата позволяет отдельно обрабатывать изменения состояния окружающей среды, что позволяет структурировать автомат. При этом поведение окружающей среды может быть, как абсолютно непредсказуемым, когда переходы в любые состояния равновероятны, так и полностью детерминированным. При анализе многоагентных систем предполагается, что все агенты ведут себя рационально, исходя из этого можно оценить вероятности перехода из состояния в состояние многоагентной системы в зависимости от целей, которых стремятся достичь агенты.

При переходе агента из состояния в состояние учитывается не только его внутреннее состояние  $L_{in}$ , настройки таймеров  $C$  и допустимые переходы  $P$ , но и состояние, в котором находится окружающая среда  $L_{out}$  и информация о  $M$ , вероятностях перехода в следующие состояния в зависимости от действий агента. Формирование реалистичных оценок вероятностей перехода окружающей среды из состояния в состояние невозможно без априорных знаний о законах поведения данной среды, иначе все варианты перехода будут равновероятны, а это равносильно отсутствию информации. При проектировании программы управления в многоагентной системе предполагается, что разработчик при описании состояний окружающей среды знает основные цели агентов и возможные методы достижения этих целей и, соответственно, имеет возможность оценить вероятности перехода окружающей среды из одного состояния в другое. С одной стороны, это накладывает ограничения на разработчика модели, с другой стороны, предположим, что разрабатывается временной автомат, предназначенный для управления в многоагентной системе. Можно выделить два принципиально разных подхода такой разработки: без учёта состояния окружающей среды и с учётом её состояния. В первом случае агент сможет выполнить целевую задачу только при отсутствии какого-либо противодействия и отсутствии погрешностей в управлении. Во втором случае в модель необходимо заложить априорные знания об окружающей среде. В ситуации многоагентной системы часто не представляется возможным заранее предсказать поведение агентов, поэтому в случае реализации программы управления с использованием временного автомата поведение агента будет исключительно реактивным, а при использовании предложенной модели появляется возможность с некоторой вероятностью предугадывать поведение остальных агентов и учитывать это при переходе из одного состояния в другое. В данном случае можно говорить об успешном объединении двух моделей — вероятностного и временного автомата, при принятии решения используются как данные таймеров и условий переходов, которые используются для изменения состояния агента без внешнего воздействия, так и вероятностные данные о поведении окружающей среды.

Описанная модель не включает в себя обучаемые алгоритмы, их применение требует большой выборки для проведения обучения [14], не во всех мультиагентных системах достаточно времени для набора необходимой статистики и адаптации поведения. Вопрос расширения модели программы управления в многоагентных системах и включения в неё обучаемых алгоритмов требует дополнительных исследований.

### **Разработка архитектуры программы управления**

Разработка архитектуры программы управления на основании предложенной модели может использовать

в качестве основы архитектуру программы на основе конечного автомата, описанную в [15]. В ней выделены три ключевых компонента: менеджер состояний, менеджер событий и набор моделей. Набор моделей — описание конечных автоматов, менеджер событий обрабатывает события, получаемые из окружающей среды, менеджер состояний обеспечивает переход из одного состояния в другое.

Модель программы управления предполагает два вида состояний — внутренние и внешние и соответствующие переходы между ними, сопровождающиеся описанием действия агента. В случае вероятностных переходов дополнительно указываются вероятности перехода из одного состояния в другое, в остальных случаях могут изменяться состояния таймеров (увеличиваться, сбрасываться), могут использоваться ограничения на переходы в зависимости от значений таймеров.

Менеджер событий обрабатывает как внутренние события, связанные с изменением значений таймеров и выполнением условий переходов, так и внешние события из окружающей среды. События используются для инициации процесса изменения состояния модели.

Менеджер состояний увеличивает значения таймеров на соответствующую константу, затем проверяет условия перехода из состояния в состояние, выполняет необходимые действия и соответствующие переходы. В случае стационарной случайной среды матрица переходов вероятностных состояний остается неизменной, иначе по заданным правилам вычисляются новые значения матрицы переходов. Данный подход позволяет учитывать не только изменение состояний окружающей среды, но и изменять вероятности окружающей среды в зависимости от действий программы управления агентом.

DTD-схема модели временного автомата может выглядеть следующим образом.

```
<!DOCTYPE timeModel [  
    <! ELEMENT timeModel (state, action, timers)>  
    <! ATTLIST timeModel manager CDATA "time">  
    <! ELEMENT state EMPTY>  
    <! ATTLIST state id ID #REQUIRED parent IDREF  
#REQUIRED inFunc CDATA #REQUIRED outFunc CDATA  
#REQUIRED >  
    <! ELEMENT action EMPTY>  
    <! ATTLIST action from IDREF #REQUIRED to IDREF  
#REQUIRED condition CDATA #REQUIRED function  
CDATA #IMPLIED timersReset CDATA #IMPLIED >  
    <! ELEMENT timers (timer)>
```

<! ELEMENT timer EMPTY>

```
<! ATTLIST timer name CDATA #REQUIRED value  
CDATA "0">
```

]>

В описанной схеме для каждого состояния заданы входные и выходные действия. Для каждого действия определены условия перехода, задана исполняемая функция и перечислены таймеры, которые должны сбрасываться при выполнении данного перехода. Условия могут включать сложные математические выражения, в том числе данные таймеров и информацию о состояниях и вероятных переходах вероятностного автомата.

DTD-схема модели вероятностного автомата может выглядеть следующим образом.

<! DOCTYPE probModel [

```
<! ELEMENT probModel (state, edge)>
```

```
<! ATTLIST probModel manager CDATA "probability">
```

<! ELEMENT state EMPTY>

```
<! ATTLIST state id ID #REQUIRED parent IDREF  
#REQUIRED inFunc CDATA #REQUIRED outFunc CDATA  
#REQUIRED >
```

<! ELEMENT edge EMPTY>

```
<! ATTLIST edge from IDREF #REQUIRED to IDREF  
#REQUIRED condition CDATA #IMPLIED function CDATA  
#IMPLIED value CDATA #IMPLIED >
```

]>

В описанной схеме задаётся матрица переходов между состояниями, если значение перехода не указано, считается, что данный переход невозможен. Для оценки возможности выполнения перехода может потребоваться проверка условий, связанных с состоянием и действиями агента. При выполнении перехода может вызываться функция, которая пересчитывает состояние модели с учётом действий агента.

Обобщённый алгоритм функционирования программы управления.

Шаг 1. Загрузка из файлового хранилища описаний временного и вероятностного автомата.

Шаг 2. Проверка состояния окружающей среды.

Шаг 3. Определение внешнего состояния.

Шаг 4. Вычисление вероятностей перехода окружающей среды в новое состояние с учётом текущего состояния агента.

Шаг 5. Вычисление значений таймеров.



Шаг 6. Проверка условий изменения внутреннего состояния на основании полученного состояния окружающей среды и значений таймеров.

Шаг 7. Уточнение вероятностей перехода окружающей среды в новое состояние с учётом возможного изменения состояния агента.

Шаг 8. Изменение внутреннего состояния, сброс таймеров.

Шаг 9. Переход к шагу 2.

Описанный алгоритм позволяет реализовывать программы управления для многоагентных систем на основе моделей временных автоматов с использованием таймеров и внутренних событий изменения состояний, а также позволяет структурировать данные об окружающей среде и функционирующих в ней агентах с помощью вероятностного автомата. Предложенный алгоритм будет успешно функционировать при конечном количестве состояний агентов, реализация для случая счтного количества состояний требует дополнительных исследований. Алгоритм предполагает, что в описаниях временного и вероятностного автоматов заложены априорные знания об эффективном поведении агента в окружающей среде и о возможных реакциях среды в зависимости от её состояния и действий агента. Сравнение с концептуальной моделью подсистемы оценки обстановки интеллектуального агента реального времени [16] показывает, что объём знаний, используемый при описании временного и вероятностного автомата, отличается не существенно, но запись получается более компактной, исполнение более простым. Как следствие предложенная программа управления предлагает упрощённые решения.

## Заключение

Рассмотрена возможность объединения подходов по построению программы управления в многоагентной системе одновременно на основе временного и вероятностного автоматов, описана соответствующая математическая модель, архитектура для программной реализации и обобщённый алгоритм. В предложенном решении реализовано согласованное использование результатов работы обоих видов автоматов.

За рамками настоящего исследования остались использование в модели обучаемых алгоритмов, случай счтного количества состояний вероятностного автомата, возможность применения иерархических моделей. Предложенное решение предполагает рациональное поведение агентов и использует априорные знания о целях, преследуемых агентами, и вероятным действиям, которые они предпримут для достижения данных целей.

## Литература

1. *Варшавский В.И.* Коллективное поведение автоматов. М.: Наука, 1973, 408 с.

2. *Бухарев Р.Г.* Основы теории вероятностных автоматов. М.: Наука, 1985, 288 с.

3. *Varshavsky V.I., Marakhovsky V.B.* Asynchronous control device design by net model behavior simulation // Proceedings of the 17th International Conference on Application and Theory of Petri Nets 1996 (Osaka, Japan, June 24–28, 1996). Springer Berlin Heidelberg, 1996. LNCS, Vol. 1091. Pp. 497–515.

4. *Alur R., Dill D.L.* A Theory of Timed Automata // Theoretical Computer Science. 1994. No. 126. Pp. 183–235.

5. *Lasota S., Walukiewicz I.* Alternating Timed Automata // ACM Transactions on Computational Logic. 2008. No. 9 (2). Pp. 1–26.

6. *Лебедев С.В., Пантелеев М.Г.* Онтолого-ориентированное проектирование подсистемы оценки обстановки автономного робота // Робототехника и техническая кибернетика. 2015. № 4(9). С. 65–69.

7. *Osipov V., Vodyaho A., Zhukova N.* About one approach to multilevel behavioral program synthesis for television devices // International journal of computers and communications. 2017. No. 11. Pp. 17–25.

8. *Shoham Y., Leyton-Brown K.* Multiagent Systems. Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press. 2009. 532 p.

9. *Vidal J.* Fundamentals of Multiagent Systems. 2007. URL: <https://github.com/josemvidal/FMAS/blob/master/mas-20070824.pdf> (дата обращения 12.09.2019)

10. *Беляев С.А., Матросов В.В.* Опыт создания среды имитационного моделирования // I-methods. 2018. Т. 10. № 3. С. 14–22.

11. *Постников Е.В., Беляев С.А.* Применение системы имитационного моделирования при решении задач информационно-аналитического обеспечения для подготовки специалистов ГИКЦ // Материалы XXV международной научно-методической конференции «Современное образование: содержание, технологии, качество» (Санкт-Петербург, 23 апреля 2019 г.). СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2019. С. 286–288.

12. *Bouzouita K., Chaari W.L., Tagina M.* Assessing Organizational Effectiveness of Cooperative Agents // Procedia Computer Science. 2017. No. 112. Pp. 917–926.

13. *Рассея С., Норвиг П.* Искусственный интеллект. Современный подход: пер с англ. М.: Вильямс. 2018. 1408 с.

14. *Коробов Д.А., Беляев С.А.* Современные подходы к обучению интеллектуальных агентов в среде Atari // Программные продукты и системы. 2018. Т. 31 № 2. С. 284–290.

15. *Smolyakov I.Y., Belyaev S.A.* Design of the Software Architecture for Starcraft Video Game on the Basis of Finite State Machines // 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)(Saint Petersburg and Moscow, Russia, 28–31 Jan. 2019). IEEE, 2019. Pp. 356–359.

16. *Лебедев С.В., Пантелеев М.Г.* Концептуальная модель подсистемы оценки обстановки интеллектуального агента реального времени // Известия СПбГЭТУ «ЛЭТИ». 2015. № 7. С. 41–46.

## APPLICATION OF PROBABILISTIC AND TIME AUTOMATA IN CONTROL PROGRAMS OF MULTI-AGENT SYSTEMS

SERGEY A. BELYAEV,

St-Petersburg, Russia, bserge@bk.ru

**KEYWORDS:** time automata; probabilistic automata; random environment; control program; multi-agent system; architecture; algorithm.

### ABSTRACT

Classical models of the time and probabilistic automaton constructed using the theory of Markov chains - the system "automaton-random environment" described by the Markov chain are considered. The uncertainty of the probabilistic automaton can be determined by various reasons, including the nondeterminism of the rules of state change of the simulated system, which is one of the most important obstacles in the modeling of multi-agent systems. The developed model is intended for the control program of multi-agent systems, takes into account the stochastic behavior of the environment when used in multi-agent systems. The key differences between the classical stationary random environment and multi-agent systems are presented, including the rationality of agent behavior and the need for cooperation or counteraction. The model does not provide trainable algorithms, their application requires a large sample for training, not all multi-agent systems have enough time to set the necessary statistics and behavior adaptation. Approaches to building the architecture of the agent control program in a multi-agent environment based on the theories of time and probabilistic automata are described, a format for storing models in file storage is proposed. The proposed architecture takes into account the possibility of changing the state of time automata based on the state of probabilistic automata and vice versa. The generalized algorithm of the control program functioning is described. The proposed algorithm makes it possible to implement control programs for multi-agent systems in the case of a finite number of agent states and uses flexible control mechanisms taken from the time machine model. The main results and proposals for further research, including the construction of hierarchical models are presented.

### REFERENCES

- Varshavsky V.I. *Kollektivnoe povedenie avtomatov* [Collective behavior of automata]. Moscow: Nauka, 1973, 408 p. (In Rus)
- Bukharev R.G. *Osnovy teorii veroyatnostnyh avtomatov* [Fundamentals of the theory of probabilistic automata]. Moscow: Nauka, 1985, 288 p. (In Rus)
- Varshavsky V.I., Marakhovsky V.B. Asynchronous control device design by net model behavior simulation. *Proceedings of the 17th International Conference on Application and Theory of Petri Nets* 1996, Osaka, Japan, June 24–28, 1996. Springer Berlin Heidelberg, 1996. LNCS, Vol. 1091. Pp. 497-515.
- Alur R., Dill D.L. A Theory of Timed Automata. *Theoretical Computer Science*. 1994. No. 126. Pp. 183-235.
- Lasota S., Walukiewicz I. Alternating Timed Automata. *ACM Transactions on Computational Logic*. 2008. No. 9 (2). Pp. 1-26
- Lebedev S.V., Panteleev M.G. Ontology-oriented design of the subsystem of environmental assessment Autonomous robot. *Robotics and technical cybernetics*. 2015. No. 4(9). Pp. 65-69. (In Rus)
- Osipov V., Vodyaho A., Zhukova N. About one approach to multi-level behavioral program synthesis for television devices. *International journal of computers and communications*. 2017. No. 11. Pp. 17-25.
- Shoham Y., Leyton-Brown K. *Multiagent Systems. Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press. 2009. 532 p.
- Vidal J. *Fundamentals of Multiagent Systems*. 2007. URL: <https://github.com/josemvidal/FMAS/blob/master/mas-20070824.pdf> (date of access 12.09.2019)
- Belyaev S.A., Matrosov V.V. The experience of creating a simulation environment. *I-methods*. 2018. Vol. 10. No. 3. Pp. 14-22. (In Rus)
- Postnikov E.V., Beljaev S.A. *Primenenie sistemy imitacionnogo modelirovaniya pri reshenii zadach informacionno-analiticheskogo obespechenija dlja podgotovki specialistov GIKC* [Application of the simulation system in solving problems of information and analytical support for the training of GICC specialists]. *Materialy XXV mezdunarodnoj nauchno-metodicheskoj konferencii "Sovremennoe obrazovanie: soderzhanie, tehnologii, kachestvo"* [Proc. of the XXV international scientific and methodological conference "Modern education: content, technology, quality", St.-Petersburg, on April 23, 2019]. St.-Petersburg, 2019. Pp. 286-288. (In Rus)
- Bouzouita K., Chaari W.L., Tagina M. Assessing Organizational Effectiveness of Cooperative Agents. *Procedia Computer Science*. 2017. No. 112. Pp. 917-926.
- Russell S., Norvig P. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2010. 1152 p.
- Korobov D.A., Belyaev S.A. Modern approaches to training intelligent agents in the Atari environment. *Software and systems*. 2018. Vol.31. No. 2. Pp.284-290. (In Rus)

15. Smolyakov I.Y., Belyaev S.A. Design of the Software Architecture for Starcraft Video Game on the Basis of Finite State Machines. *Proc.2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, Saint Petersburg and Moscow, Russia, 28-31 Jan. 2019. IEEE, 2019. Pp. 356-359.
16. Lebedev S.V., Panteleev M.G. Conceptual model of the sub-

system for assessing the situation of an intelligent real-time agent. *Izvbestia SPbETU "LETI"*. 2015. No. 7. Pp.41-46. (In Rus)

**INFORMATION ABOUT AUTHOR:**

Belyaev S.A., PhD, Docent, Lecturer of the St. Petersburg Saint Petersburg Electrotechnical University "LETI".

**For citation:** Belyaev S.A. Application of probabilistic and time automata in control programs of multi-agent systems. *H&ES Research*. 2020. Vol. 12. No. 3. Pp. 47-53. doi: 10.36724/2409-5419-2020-12-3-47-53 (In Rus)



# НАУКА и АСУ – 2020

ВСЕРОССИЙСКАЯ НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ

\*\*\*\*\*

В октябре 2020 года ООО «Институт инноваций и наукоемких технологий» (ООО «Институт «ИНТЕХ») совместно с партнерами проведет Всероссийскую научно-техническую конференцию, посвященную теоретическим и прикладным проблемам развития и совершенствования автоматизированных систем управления специального назначения «НАУКА И АСУ — 2020».

[http://intech-spb.com/conferences/  
konferencia\\_asu\\_vka@mail.ru](http://intech-spb.com/conferences/konferencia_asu_vka@mail.ru)

По итогам конференции отобранные оргкомитетом доклады в виде статей будут опубликованы в журналах из Перечня ВАК, РИНЦ.

Участие в конференции и публикация материалов в сборнике тезисов БЕСПЛАТНО.

Полная информация о конференции, дата, место проведения, требования к материалам докладов будет выложена на сайте конференции <http://intech-spb.com/conferences/>.

Тематика конференции включает работу следующих шести секций:

Состояние и перспективы развития современных автоматизированных систем управления специального назначения.

Математическое, программное и информационно-лингвистическое обеспечение автоматизированных систем управления.

Безопасность в автоматизированных системах управления специального назначения.

Применение современных инфокоммуникационных технологий и средств при разработке, техническом обеспечении и эксплуатации автоматизированных систем управления специального назначения.

Состояние и перспективы развития систем, комплексов и средств радиосвязи специального назначения.

Проблемы развития автоматизированных систем управления технологическим процессом.