

АНАЛИТИКО-ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ КООРДИНАЦИИ ГРУППОВОГО ПОВЕДЕНИЯ АГЕНТОВ

Д.А. Мурашов (Санкт-Петербург)

Введение

Современный подход к имитационному моделированию (симуляции) сложных мультиагентных систем использует концепцию «систем, основанных на поведении», или Behavior-Based Systems (BBS) [1]. В основу BBS-подхода положена идея, что рациональность коллективного поведения в мультиагентных системах является эмерджентным свойством, возникающим как результат взаимодействия агентов друг с другом, со средой, и как результат обобщения опыта этих взаимодействий. Т.е. агенты в рамках концепции BBS имеют память и умеют «мыслить». Такой подход позволяет описывать широкий спектр мультиагентных систем, вплоть до сложных объектов произвольной природы.

Ряд современных агентно-ориентированных средств автоматизации моделирования (АОСАМ), к примеру, учитывает взаимодействия агентов со средой через реализацию механизма обратной связи [2]. Однако для моделирования более интересных составляющих BBS-систем, таких как обобщение опыта или меж-агентное взаимодействие, у большинства из АОСАМ отсутствуют необходимые компоненты.

Один из видов систем, охватываемых концепцией BBS, который большинство современных АОСАМ пока не могут моделировать – системы с координатором. Под координатором понимается агент, задающий стратегические цели для других агентов. Такого рода модели описывают сложные системы, например, социальные, где осуществляется управление большой группой. Агенты в таких системах имеют значительную свободу выбора действий. Они учитывают глобально заданные задачи, но всегда действуют исходя из собственных знаний, опыта и локальной оценки текущей ситуации. Нередко в реальном мире действия отдельного агента противоречат глобально заданной политике. Такая гибкость – важная особенность подобных систем.

В настоящей статье для симуляции подобных систем предлагается практический подход к групповой координации агентов, использующий динамическое изменение весов во взвешенном графе иерархии предпочтений, и алгоритм АНР [3]. Предложенный механизм может быть использован как часть фреймворка или компонент (плагин) АОСАМ.

Сама по себе идея использования алгоритма АНР для моделирования выбора альтернатив агентами не нова [4], [5], как и динамическое изменение весов для изменения поведения агентов [6], [7]. Однако, во-первых, описываемый в данной статье подход интерпретирует иерархию предпочтений как механизм групповой координации. Иерархия предпочтений позволяет описать трансляцию стратегических целей, сделанных на верхнем уровне иерархии, в принятие решений отдельными агентами на локальных уровнях. Во-вторых, в рамках предлагаемого подхода используется разделение графа иерархии на глобальную и локальную часть, каждая из которых изменяется параллельно разными классами агентов (координаторы и исполнители). У каждого из исполнителей существует экземпляр глобальной части, дополненной индивидуальной для каждого исполнителя локальной частью. Координатор может влиять на исполнителя, изменяя глобальную, но не локальную часть графа. Совокупно эти два положения составляют новизну. В дополнение, ни одно из АОСАМ пока не реализует похожего подхода.

Статья имеет следующую структуру. В п. 1 раскрывается текущее положение дел в области АОСАМ, в п. 2 приводится описание предлагаемого подхода, в п. 3

описывается пример того, как предлагаемому подходу можно дать содержательную интерпретацию, и в п. 4 приводятся рекомендации по улучшению предлагаемого подхода.

1. Современные средства автоматизации моделирования сложных объектов произвольной природы на базе агентного подхода

АОСАМ сложных объектов (СЛО) произвольной природы можно разделить на два класса: основанные на программировании (большинство разработчиков предпочитает называть их фреймворками) и основанные на конфигурировании. Часто последние – это «статические» программные среды. Они имеют набор готовых компонентов, графический интерфейс для работы с ними и иногда могут быть расширены дописыванием компонентов на каком-нибудь языке программирования, хотя чаще всего используется изменение параметров готовых компонентов.

Перечисленные два класса отличаются трудозатратами на программирование, требуемыми для создания модели, и как следствие, гибкостью. Системы Any Logic [2] и FlexSim [8], например, ограничивают набор действий для каждого агента. Они расширяются за счет подключения новых подсистем, как правило, использующих компоненты, ориентированные на узкую предметную область. Программные библиотеки, такие как JADE (Java) [9], Aivika (Haskell) [10], SObjectizer (C++) [11], дают очень большую свободу взамен сложности: они очень хорошо решают задачу взаимодействия между агентами, описывают смену состояний, но остальные части создаваемой модели приходится программировать на языке программирования общего назначения. Отдельно выделяются средства автоматизации моделирования, занимающие промежуточное положение относительно двух классов, такие как NetLogo или Gama. Они имеют средства визуализации и специальные языки программирования, которые частично – декларативно, частично – императивно описывают аспекты многоагентной среды (пространственные, временные), взаимодействие между агентами, и могут быть расширены использованием языка(ов) программирования общего назначения.

В основном АОСАМ реализуют парадигмы конечных автоматов, CSP [12], и иногда обеспечивают доступ агентов к информации об окружающей их среде (присутствие других агентов, пространственная информация). Средства координации ими не реализуются по умолчанию; в том случае, если АОСАМ реализуют средства изменения поведения агента, это делается локально, за счет описания действий агента на основе деревьев альтернативных решений [2], без координирующей единицы.

2. Моделирование изменения предпочтений агентов

В системах с координацией взаимодействие агентов, входящих в состав группы, осуществляется через формирование иерархии отношений предпочтения командования: агент-координатор определяет глобальные цели, а исполнители их реализуют локально. В настоящей статье для моделирования разделения по уровням иерархии и моделирования процесса принятия решения предлагается подход на основе алгоритма аналитического иерархического процесса (Analytic Hierarchy Process, АНП) [3].

В алгоритме АНП используется граф иерархии предпочтений (см. рис. 1). Предварительно введем ряд формальных обозначений, связанных с графом.

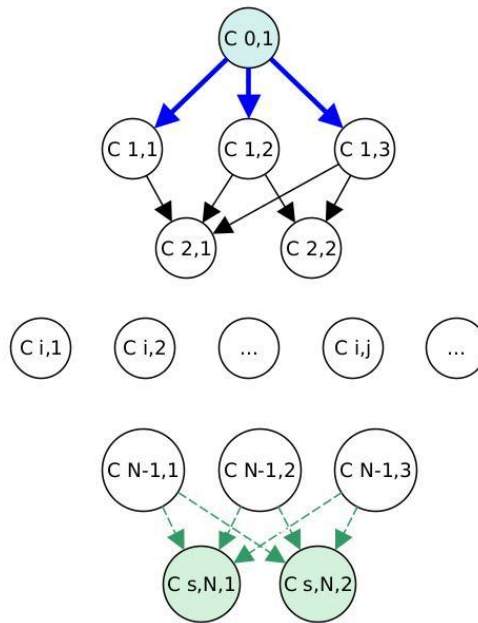


Рис. 1. Граф предпочтений

(жирными стрелками показаны значения предпочтений, динамически задаваемые координатором глобально, тонкими пунктирными – значения предпочтений, динамически устанавливаемые агентом локально перед принятием решения)

$C_{i,j}$ – вершины графа предпочтений, i – уровень, j – индекс вершины на уровне j .
 Две нижних вершины $C_{k,N,1}, C_{k,N,2}$ – альтернативы, возможные сценарии [13] для агента k , один из которых ему предстоит выбрать.

В алгоритме АНР используется попарное контекстное сравнение вершин одного уровня агентом, принимающим решение, для ранжирования альтернатив с учетом текущей ситуации в некотором контексте или с некоторой позиции. Для интерпретации предлагаемого описания на естественном языке будем говорить, что если вершины уровня X сравниваются попарно относительно некоторой вершины O уровня $Y < X$, то производится сравнение *альтернатив* уровня X *в контексте* O или *относительно вершины* O .

Оценка альтернатив производится с использованием матриц попарного контекстного сравнения вида:

$$C_i^{(C_{k,vert_k})} = \left(\frac{r^{(C_{k,vert_k})}_{C_{i,a}}}{r^{(C_{k,vert_k})}_{C_{i,b}}} \right)_{\{(a,b)\} = Ind_i \times Ind_i} \quad (1)$$

Здесь a, b – индексы вершин на уровне i , Ind_i – индексное множество вершин уровня i , $C_{k,vert_k}$ – контекст, в рамках которого задается сравнение, $r^{(C_{k,vert_k})}_{C_{i,a}}$ – вес (не обязательно вес дуги), соответствующий оценке значимости вершины $C_{i,a}$ в контексте $C_{k,vert_k}$.

Матрица (1) может быть преобразована в вектор нормированных весов, и обратно. Обозначим $C_i^{r^{(C_{k,vert_k})}}$ как вектор нормированных весов:

$$C_i^{(C_k, Vert_k)} = \left(C_{i, Vert_i}^{(C_k, Vert_k)} \right)_{Vert_i \in Ind_i}^T, \quad \sum_j C_{i,j}^{(C_k, Vert_k)} = 1. \quad (2)$$

Алгоритм АНР позволяет получить вектор нормированных весов $C_i^{(C_k, Vert_k)}$ из матриц $\{C_i^{(C_m, Vert_m)}\}_{Vert_m \in Ind_m}$ и $C_m^{(C_k, Vert_k)}$. Говоря иначе, АНР позволяет оценить альтернативы $\{C_{i, Vert_i}\}_{Vert_i \in Ind_i}$ в контексте $C_{k, Vert_k}$ через промежуточные контексты $\{C_{m, Vert_m}\}_{Vert_m \in Ind_m}$:

$$C_i^{(k, Vert_k)} = \mathbf{АНР} \left(\{C_i^{(m, Vert_m)}\}_{Vert_m \in Ind_m}, C_m^{(k, Vert_k)} \right) \quad (3)$$

Введенные обозначения позволяют теперь сформулировать формально алгоритм, моделирующий процесс координации группового поведения. Исходные данные для симуляции следующие. *Ориентированный взвешенный граф* предпочтений до уровня N , задаваемый как тройка из вершин, дуг и весов:

$$Pref = \left(\{C_{i, Vert_i}\}_i, \{(C_{i-1, Vert_{i-1}}, C_{i, Vert_i})\}_{i>1}, \{C_{i, Vert_i}^{(C_{i-1, Vert_{i-1}})}\}_i \right), i \in \overline{0, N-1}, Vert_x \in Ind_x$$

Этот граф – общий для всех агентов. Не обязательно, чтобы для каждой пары вершин смежных уровней существовала дуга, если это не дуги между уровнями 0 и 1. На уровнях $[1, N-1]$ граф составляется экспертами с учетом особенностей предметной области, для которой разрабатывается симуляция. Предполагается, что промежуточные уровни неизменны на протяжении симуляции. Для уровней 0 и 1 в начале симуляции веса $C_1^{(C_0, Vert_0)}$ зададим равными друг другу ($C_{1, Vert_1}^{(C_0, Vert_0)} = \frac{1}{|Ind_1|}$).

Вводятся два класса агентов: *координатор* и *исполнители*. Предполагается, что координатор – это наиболее осведомленная единица (агент). На протяжении симуляции координатор получает данные о состоянии агентов и среды от исполнителей. Координатор собирает и анализирует поступившие данные с целью уточнения дальнейшей стратегии группы. Уточнение стратегии достигается изменением весов $C_{1, Vert_1}^{(C_0, Vert_0)}$ верхних уровней графа $Pref$, видного всем агентам.

Перед началом симуляции каждому агенту-исполнителю назначается набор сценариев, количество которых равно $|Ind_N|$. Агент постоянно выполняет цикл «выбор сценария – исполнения сценария», параллельно передавая данные о наблюдаемой среде, окружающих его агентах и собственном состоянии координатору. Каждому сценарию устанавливается соответствие с вершиной уровня N в графе предпочтений. Таким образом, агент расширяет граф $Pref$, добавляя к нему уровень N , видный только ему. Обозначим это дополнение на уровнях $N-1, N$ как граф Alt . Веса $\{C_{s, N, Vert_N}^{(C_{N-1, Vert_{N-1}})}\}$ в начале симуляции будем считать равными друг другу.

$$Alt = \left(\{C_{s, N, Vert_N}\} \cup \{C_{N-1, Vert_{N-1}}\}, \{(C_{N-1, Vert_{N-1}}, C_{s, N, Vert_N})\}, \{C_{s, N, Vert_N}^{(C_{N-1, Vert_{N-1}})}\} \right)$$

Процедура инициализации симуляции выглядит так, как показано в листинге 1.

Листинг 1

инициализироватьСимуляцию($Pref$, $nAgents$):

$coordinator \leftarrow$ создатьКоординатора($Pref$)

$agents \leftarrow \emptyset$

$s \leftarrow nAgents$

while $s > 0$:

$Alt \leftarrow$ создатьГрафАльтернатив(s)

$agents \leftarrow agents \cup$ создатьИсполнителя($Pref$, Alt)

запуститьАгента(s)

$s \leftarrow s - 1$

запуститьАгента($coordinator$)

while симуляцияПродолжается:

continue

Работа координатора выглядит следующим образом (листинг 2). На протяжении симуляции агент-координатор оценивает ситуацию и обновляет веса $C_{1,Ind_1}^{(C_{0,0})}$ в графе $Pref$ на соответственно его глобальной оценке ситуации. Так же обновляются «копии» $Pref$, находящиеся у исполнителей. Так координатор влияет на решения агентов.

Листинг 2

1. $Pref$

2.

3. **while** симуляцияПродолжается:

4. $situation \leftarrow$ ждатьНовыхДанных()

5. $C_1^{(C_{0,0})} \leftarrow$ обновитьСтратегию($situation$)

На протяжении симуляции агент-исполнитель s выполняет предписанные ему сценарии [14], выбирая на каждой итерации наиболее подходящий (листинг 3). На каждой итерации агент оценивает окружение. Он попарно сравнивает назначенный ему набор возможных действий (граф Alt), составляя матрицы $\left\{ C_{s,N}^{(C_{N-1,Vert_{N-1}})} \right\}_{Vert_{N-1}}$ вида (1) для всех контекстов уровня $N - 1$, потом при помощи алгоритма АНР выбирает наилучшее решение (сценарий) и применяет его. Важно, что в отличие от графа $Pref$, изменения в котором «подхватываются» всеми агентами, у каждого агента-исполнителя имеется отдельный экземпляр Alt , который не затрагивает других агентов.

$Pref$
 Alt

$$\text{оп} \left(\left\{ C_N^{(C_1, vert_1)} \right\}_{Vert_1 \in Ind_1}, 1 \right):$$

$$\left\{ C_N^{(C_1, vert_1)} \right\}_{Vert_1} \leftarrow \left\{ \text{transform} \left(C_N^{(C_1, vert_1)} \right) \right\}_{Vert_1}$$

if $l = 0$:

$$\text{return АНР} \left(\left\{ C_N^{(C_1, vert_1)} \right\}_{Vert_1 \in Ind_1}, C_{N-1}^{(C_{0,0})} \right)$$

else:

$$\text{return оц}\left\{\text{АНП}\left(\left\{C_N^{(C_1, \text{vert}_1)}\right\}_{\text{vert}_1}, C_1^{(C_{1-1}, \text{vert}_{1-1})}\right)\right\}_{\text{vert}_{1-1}}, l-1)$$

while симуляцияПродолжается:

$$\left\{C_{s,N}^{(C_{N-1}, \text{vert}_{N-1})}\right\}_{\text{vert}_{N-1}} \leftarrow \{\text{assessAlternatives}(\text{Vert}_{N-1})\}_{\text{vert}_{N-1} \in \text{Ind}_{N-1}}$$

$$C_N^{(C_{0,0})} \leftarrow \text{оц}\left\{\text{АНП}\left(\left\{C_{s,N}^{(C_{N-1}, \text{vert}_{N-1})}\right\}_{\text{vert}_{N-1}}, C_{N-1}^{(C_{N-2}, \text{vert}_{N-2})}\right)\right\}_{\text{vert}_{N-2}}, N - 2)$$

выполнять($\text{argmax}_i (C_{N,i}^{(C_{0,0})})$)

Листинг 1 – работа исполнителя.

Наличие каналов связи между агентами и координатором и процедур обработки этих данных, результатом которых являются значения весов $\left\{C_{s,N}^{(C_{N-1}, \text{vert}_{N-1})}\right\}_{\text{vert}_{N-1}}$ и $C_{1, \text{vert}_1}^{(C_{0, \text{vert}_0})}$, подразумевается процедурами «ждатьНовыхДанных», «определитьСтратегию» и «assessAlternatives», однако их вид и состав определяется индивидуально для каждой реализации. Это точки расширения АОСАМ, реализующего подход.

3. Пример содержательного описания иерархии предпочтений для агентов в среде с противодействием

Рассмотрим возможную задачу симуляции роя в среде с противодействием. В рое имеется два класса агентов: *агенты-атакующие* и *агенты-работники*. У роя есть конкурирующие рои. Перед каждым роем стоит задача поиска и сбора ресурсов, защиты накопленных ресурсов, и атаки на конкурирующий рой с целью отнять ресурсы. Рис. 2 изображает вариант иерархии предпочтений для описанной ситуации.

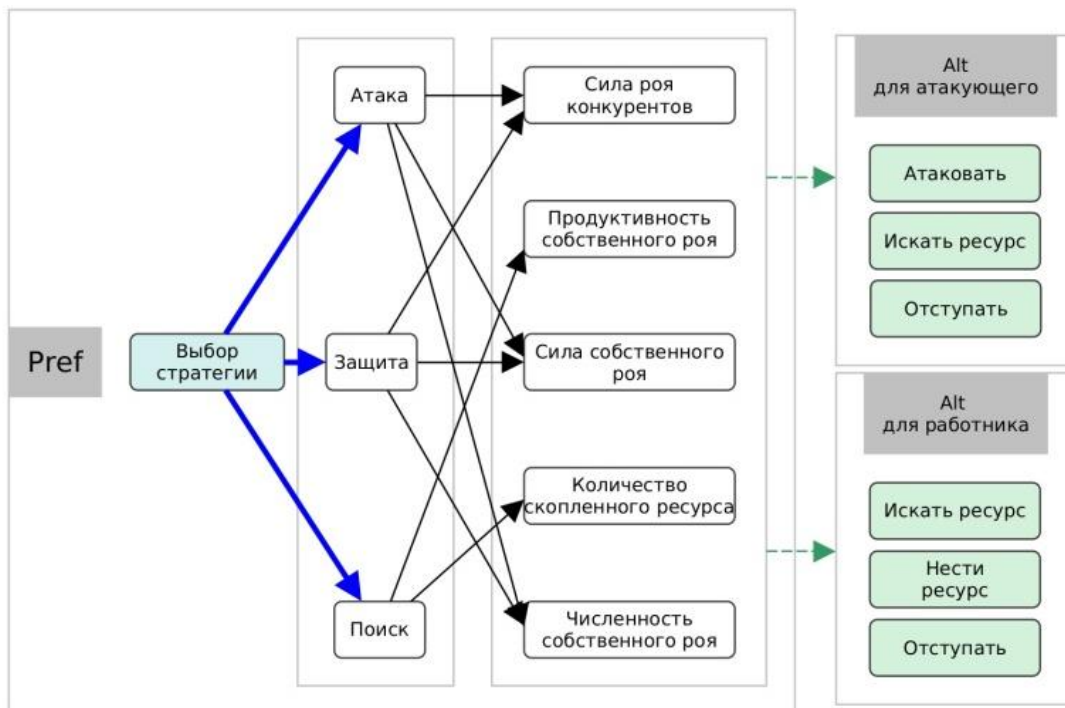


Рис. 2. Иерархия предпочтений для предлагаемого примера

Симуляция для такой задачи может выглядеть следующим образом. Координатор занимается анализом поступающей к нему информации: подсчитывает агентов, ресурсы, отслеживает динамику, выполняет статистический анализ и пр. Затем на основе этой информации координатор выставляет приоритеты для стратегий «Поиск», «Защита» и «Атака» в графе *Pref*. К примеру, в рое осталось мало атакующих агентов. Тогда логично установить стратегию «Защита», соответствующую осторожному поведению агента. Изменения, сделанные координатором в общем графе *Pref*, влияют на всех агентов-исполнителей.

Каждый агент-исполнитель (работник или атакующий) имеет возможность попарно взвешивать все назначенные ему альтернативы проколов в контекстах «Сила роя конкурентов», «Продуктивность собственного роя», и далее по рисунку, основываясь на своем опыте взаимодействия со средой и другими агентами и на основе информации, полученной от других агентов и извлеченной из опыта. Опыт взаимодействия, к примеру, может представляться в виде статистики наблюдаемых поражений, или статистики эффективности выполнения задачи в зависимости от состояния среды (сила ветра и пр.) – в зависимости от конкретной задачи, стоящей перед создаваемой симуляционной моделью. Предпочтения, выставленные агентом для себя в графе *Alt*, не влияют на предпочтения других агентов.

Координатор не вмешивается в индивидуальный процесс принятия решений и в общение между агентами, что позволяет одновременно и координировать рой, и не ограничивать агентов. Действия агента могут обоснованно противоречить заданной «политике» в определенных ситуациях: например, если агент класса «атакующий» заметил слабого противника без защиты. В таком случае, даже если бы координатор выставил приоритет стратегии «Защита», возможна ситуация, когда индекс $i = \operatorname{argmax}_i (C_{N,i}^{(C_{0,0})})$, полученный по результатам оценивания (листинг 3), соответствует сценарию «Атаковать», т.е. агент стал действовать в противоречие глобальной стратегии. Это не только полезно с точки зрения координации, но и реалистично с позиции моделирования реальных процессов.

4. Проблемы предлагаемого подхода и направления дальнейшего развития

Контекстная попарная оценка для предлагаемой структуры на каждом из уровней представляет большую сложность. Упрощение процедуры контекстной оценки – одно из направлений дальнейшего развития.

Возможно, предложенный подход может быть применен не только для симуляции, но и для реальной групповой координации. Исследование этого вопроса – другое возможное направление.

Предполагается, что веса для промежуточных уровней задаются экспертом в момент создания симуляционной модели. Возможно, этот процесс можно упростить или автоматизировать применением данных реальных систем. Возможно, задача попарного сравнения может быть решена применением статистических методов. Это – один из вариантов дальнейшего развития подхода, который решит проблему сложности контекстной интерпретации. Также предполагается, что промежуточные веса статичны. Это может быть непрактично для описания динамической системы.

Структура иерархии предпочтений алгоритма АНР напоминает структуру нейронной сети. Возможно, здесь могут быть применимы методы обучения нейронных сетей.

Возможно, задача установления стратегии может быть автоматизирована применением методов автоматического управления. Разработка алгоритма автоматического управления стратегиями (задания значений весов $C_1^{(C_{0,0})}$) – одно из направлений дальнейшего развития.

Исследование других алгоритмов, позволяющих использовать иерархические структуры при моделировании процесса решения. К примеру, граф предпочтений может быть составлен как декомпозиция цели на множество подзадач; в этом случае, вероятно, может быть применен метод дерева целей, предложенный Г. С. Поспеловым [15].

Предложенный подход пока не получил программной реализации. Разработка симуляционной модели с применением предложенного подхода – наиболее важное направление развития.

Выводы

Важный недостаток использования предложенного подхода иерархии предпочтений в задачах, подобных рассмотренной, – в сложности построения и обновления иерархии под конкретную задачу. При создании и при каждом изменении иерархии на уровне $N - 1$ требуется заново определять, каким образом семантика контекста интерпретируется агентом на уровне локального принятия решений. Практически – перепрограммировать процесс рассуждений агентов.

С другой стороны, этот же подход может обеспечить простую (с точки зрения механизма координации) трансляцию стратегических целей в принятие решений отдельными агентами на локальных уровнях. Плюс, он гибок, т.к. не накладывает ограничений на тип агентов внутри класса агентов-исполнителей, на их действия, на процесс попарной оценки сценариев и на взаимодействие агентов между собой и со средой.

Исследования, выполненные по данной тематике, проводились в рамках бюджетной темы № 0073–2019–0004.

Литература

1. **Mataric M.J., Michaud F.** Behavior-based Systems // Handbook of Robotics. Springer, 2008. P. 891-909.
2. AnyLogic. www.anylogic.com (дата обращения: 25.09.2021).
3. **T.L. Saaty.** Decision Making for Leaders: The Analytical Hierarchy Process for Decisions in a Complex World. Belmont, CA: Wadsworth, 1982.
4. **В. Картвелишвили, Э. Лебедюк.** Модель агента и мультиагентного взаимодействия в социальноэкономических системах // Математические и инструментальные методы, 2018. С. 147-165.
5. **M. Drakaki, H. Gören, P. Tzionas.** An Intelligent Multi-agent System Using Fuzzy Analytic Hierarchy Process and Axiomatic Design as a Decision Support Method for Refugee Settlement Siting // Lecture Notes in Business Information Processing, 2018. С. 15-26.
6. **M. Zytniewski, M. Klement, D. Skorupka, S. Stanek, A. Duchaczek.** Application of the AHP method in Modelling the Trust and Reputation of Software Agents // AIP Conference Proceedings, 2016.
7. Brintrup. Behaviour adaptation in the multi-agent, multi-objective and multi-role supply chain // Computers in Industry, 2010. С. 636-645.
8. FlexSim. www.flexsim.com (дата обращения: 25.09.2021).
9. JADE. Java Agent Development Framework. <https://jade.tilab.com/> (дата обращения: 25.09.2021).
10. Aivika. <https://www.aivikasoft.com/ru/products/aivika.html> (дата обращения: 25.09.2021).
11. SObjectizer. <https://github.com/Stiffstream/sobjectizer> (дата обращения: 25.09.2021).
12. C.A.R. Hoare Communicating Sequential Processes. Prentice Hall International. 1985.

13. **Пантелеев М. Г.** Концепция построения интеллектуальных агентов реального времени на основе модели опережающего итеративного планирования // Тринадцатая национальная конференция по искусственному интеллекту с международным участием, 2012. С. 25-33.
14. **В.И. Городецкий, М.Г. Пантелеев.** Сети автономных агентов реального времени в среде с противодействием: особенности и компоненты модели // Материалы конференции «Информационные технологии в управлении», 2021. С. 31-40.
15. **Г.С. Поспелов.** Проблемы программно-целевого планирования и управления. М.: Наука. Главная редакция физико-математической литературы 1.г. С.4.