

ОБЗОР БИБЛИОТЕК ДЛЯ ПОСТРОЕНИЯ МУЛЬТИАГЕНТНЫХ СИСТЕМ В СРЕДЕ PYTHON

А.М. Журавлев

zhuravlev.am@rea.ru

SPIN-код: 9327-0412

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Выполнен обзор существующих библиотек, на базе которых возможно построение сложных мультиагентных систем. При выборе лучшей библиотеки учитывают ее работоспособность, поддержку со стороны разработчиков, простоту ее использования и установки в наиболее распространенной операционной системе Windows, т. е. осуществляют качественный анализ библиотек по основным характеристикам с точки зрения удобства конечного пользователя. В статье приведены примеры простейших скриптов для проверки работоспособности каждой из библиотек, проанализированы особенности библиотек и некоторые ошибки, которые могут возникать при их установке. Актуальность статьи объясняется все возрастающим интересом к искусственному интеллекту в целом и мультиагентным системам и их разработке в частности.

Ключевые слова

Мультиагентные системы, искусственный интеллект, фреймворк, Python, терминал, Windows, FIPA (Foundation for Intelligent Physical Agents — фонд интеллектуальных физических агентов), BDI (belief, desire, and intention — модель убеждений, желаний и намерений)

Поступила в редакцию 10.09.2018

© МГТУ им. Н.Э. Баумана, 2018

Введение. С момента возникновения первых электронных вычислительных систем их развитие характеризовалось переходом на все более высокие уровни абстракции. Так, изначально работа с компьютерами была неразрывно связана с самой архитектурой системы. Со временем мы пришли к тому, что совершенно различные работы выполняются на совершенно разных компьютерах, с разными архитектурами и мощностями. В настоящее время наблюдается бурный рост интереса к различным системам искусственного интеллекта (ИИ). В связи с этим меняются и сами принципы человеко-машинного взаимодействия. С определенной долей уверенности можно утверждать, что происходит переход на более высокий уровень абстракции — взаимодействие человека и ИИ. Одна из разновидностей систем ИИ, о которой в дальнейшем пойдет речь, — мультиагентные системы [1].

Мультиагентная система (МАС) — это система, образованная несколькими взаимодействующими интеллектуальными агентами. Интеллектуальные агенты, в свою очередь, — это агенты (чаще всего под агентом подразумевается некая компьютерная программа), которым присущи следующие признаки:

- целеустремленность;
- обучаемость;

- социальность;
- независимость.

Мультиагентные системы могут быть использованы для решения таких проблем, которые сложно или невозможно решить с помощью одного агента или монолитной системы [2].

Теория мультиагентных систем зародилась еще в 40-х годах прошлого века и с тех пор нашла широкое применение в различных отраслях человеческой жизни, в том числе:

- игровое моделирование;
- моделирование экономических систем;
- роботостроение.

В данной статье рассмотрены различные мультиагентные библиотеки Python с целью выбора среди них наилучшей для дальнейшего использования в игровой среде и применения алгоритма обучения с подкреплением.

Для рассмотрения были выбраны следующие готовые решения:

- Spade;
- LOMAP;
- PADE;
- Aiomas;
- Spyse.

Каждая из библиотек оценивается по следующим критериям:

- простота установки;
- понятность;
- работа в системе Windows;
- наличие технической документации;
- наличие начальных примеров;
- наличие форумов с поддержкой;
- отличительные особенности.

Оценка осуществляется по 10-балльной шкале, где 1 — это очень плохо, 10 — очень хорошо. Также немаловажным критерием является совместимость библиотеки со спецификациями FIPA (Foundation for Intelligent Physical Agents — фонд интеллектуальных физических агентов) [3].

Библиотека SPADE. Spade (Smart Python Multi — Agent Development Environment) [4] — мультиагентная и структурная платформа, базирующаяся на технологии XMPP/Jabber и написанная на языке Python. Технология предлагает множество инструментов для упрощения конструирования МАС, вроде коммуникационных каналов, архитектур типа «клиент — сервер» и др. На рис. 1–3 представлены главная страница документации Spade, одна из глав документации и начальный пример библиотеки соответственно.

Существует множество различных МАС, но именно Spade основана на технологии XMPP [5].

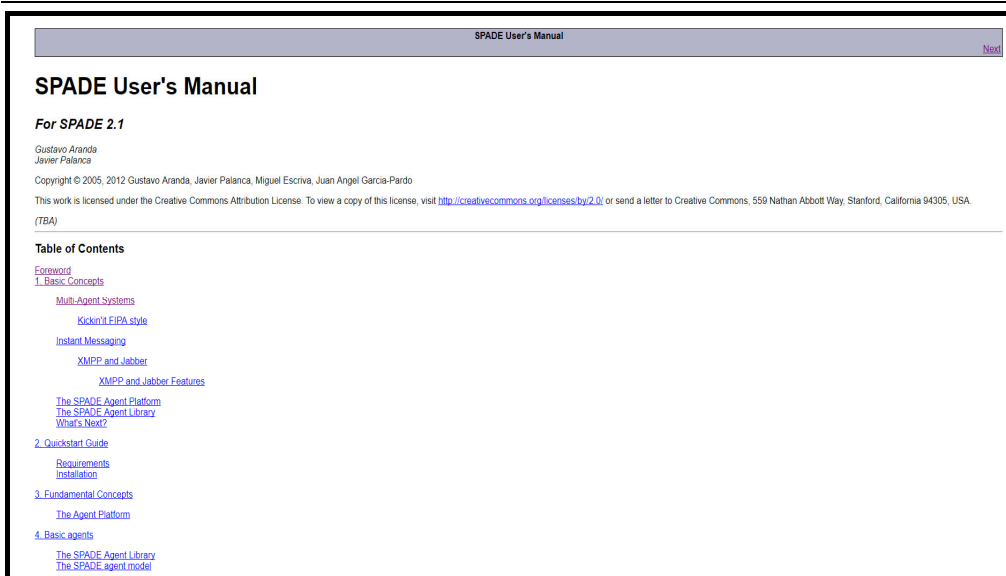


Рис. 1. Главная страница документации в библиотеке SPADE

Перейдем непосредственно к установке библиотеки. На странице пакета предлагается скачивание архива для установки. Также возможна установка с помощью менеджера пакетов Python под названием pip.

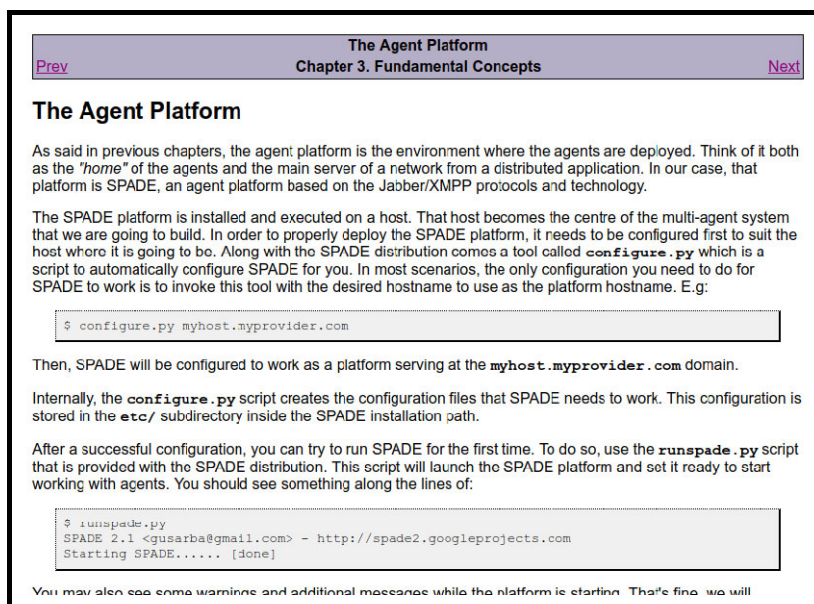


Рис. 1. Пример главы документации библиотеки Spade

При попытке установить пакет получаем ошибку. После поиска информации по этому вопросу оказывается, что данная библиотека работает под управлением системы Python 2. По данному критерию выставим 5 баллов, поскольку работа под Python 2 все же возможна.

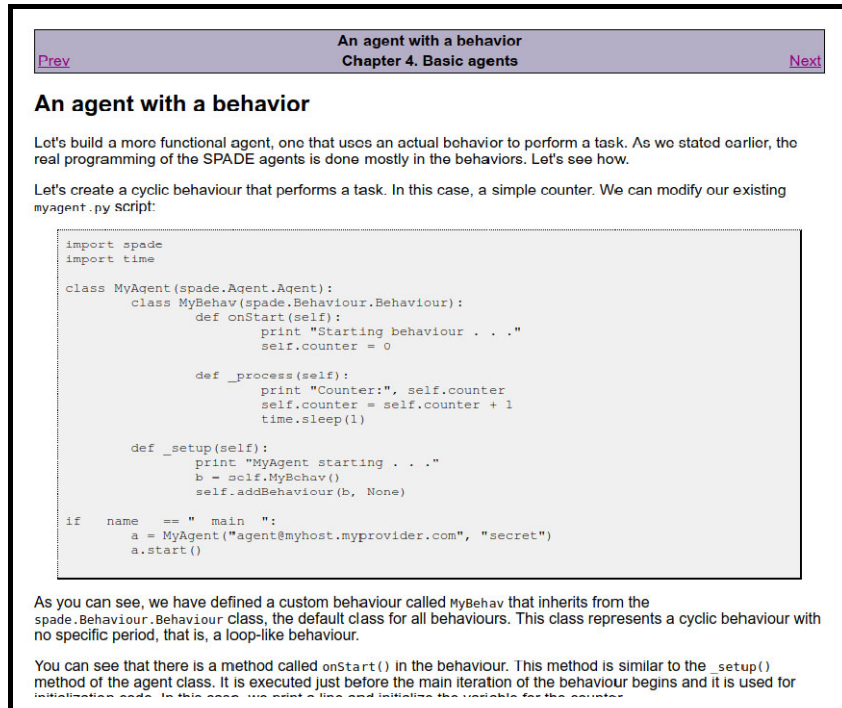


Рис. 3. Один из начальных примеров библиотеки Spade

Под управлением Python 2 библиотека была успешно установлена посредством pip, однако потребовалась частичная дополнительная конфигурация посредством запуска одного файла, поэтому выставим ей 9 баллов по данному критерию.

В документации подробно рассмотрены все аспекты создания агентов и их работы. По критерию понятности поставим 10 баллов.

Установка в среде Windows 10 прошла без каких-либо проблем, также были найдены инструкции по установке библиотеки для версий Windows 8.1 и 7 на странице библиотеки сервера GitHub. Таким образом, по критерию работы в Windows библиотека получает 10 баллов.

На рис. 3 изображен начальный пример обширной документации Spade. Описаны все возможности библиотеки и частично даже теория мультиагентных систем. Поставим 10 баллов за техническую документацию.

Приведено несколько начальных примеров агентов: обычный агент, агент с некоторым заданным поведением, агент, посылающий сообщения и др. За начальные примеры библиотека получает 9 баллов.

Страница проекта Spade существует на сервере GitHub, где разработчики регулярно отвечают на вопросы и освещают обновления платформы. Поставим библиотеке 10 баллов за поддержку.

Помимо этого библиотекой Spade поддерживается BDI-модель агентов, также вся библиотека является FIPA-совместимой, что суммарно обеспечивает ей 8 баллов по шкале «отличительные особенности».

Библиотека LOMAP. LOMAP (LTL Optimal Multi-Agent Planner) [6] — пакет для автоматического планирования оптимальных путей для мультиагентных систем. Включает в себя следующие алгоритмы:

- Policy synthesis (синтез политик);
- Multi-agent optimal run (оптимальный мультиагентный проход);
- Robust multi-agent optimal run (устойчивый оптимальный мультиагентный проход);
- Incremental policy synthesis (инкрементный синтез политики);
- Optimal run (оптимальный проход).

Данные алгоритмы являются отличительной особенностью данной библиотеки, что дает ей 4 балла по данному критерию.

При попытке установить библиотеку также выводилось сообщение об ошибке. Данная библиотека не предназначена для работы в Python 3. Поставим 5 баллов за поддержку Python 2.

Установить данный пакет можно как через архив со страницы библиотеки, так и через pip. Поставим 10 баллов за простоту установки.

Не удалось найти информацию о работе LOMAP под управлением системы Windows, однако успешно удалось его запустить под управлением Windows 10. Поставим 6 баллов за поддержку Windows.

Разработчики лишь кратко описывают методы библиотеки и работу с ней, ссылаясь на простоту алгоритмов. Это увеличивает сложность понимания библиотеки — 4 балла за понятность.

Главная страница документации данной библиотеки и советы по работе с ней показаны на рис. 4 и 5 соответственно.

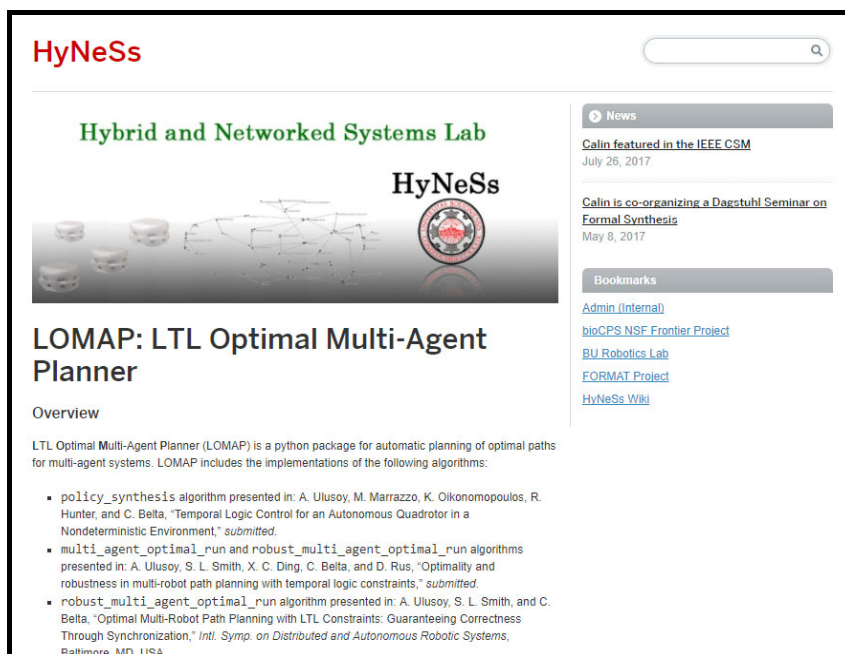


Рис. 4. Главная страница документации по LOMAP

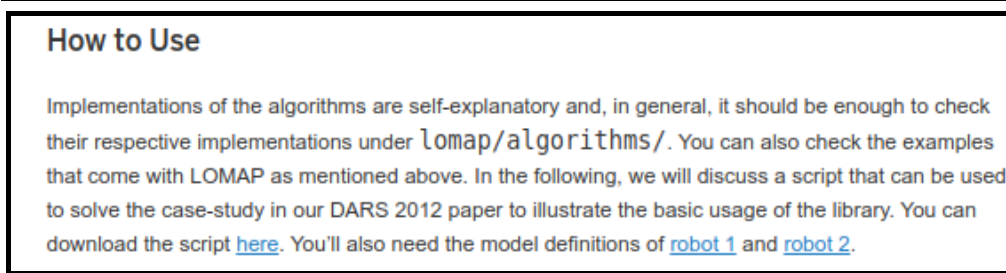


Рис. 5. Советы по работе библиотеки из документации Lommap

Также у библиотеки фактически нет никакой документации, лишь объяснение некоторых базовых методов, в результате чего она получает 3 балла за наличие документации.

Начальные примеры лежат в папках, создающихся при установке библиотеки. Поставим 8 баллов за начальные примеры.

На странице проекта чрезвычайно мало каких-либо ссылок на авторов библиотеки. Поставим 2 балла за поддержку библиотеки.

Библиотека PADE. PADE (Python Agent Development framework) [7] — фреймворк для разработки, запуска и управления мультиагентными системами в распределенных компьютерных средах. PADE полностью написана на Python и использует библиотеки из проекта Twisted [8] для сообщения узлов сети. На рис. 6 представлена главная страница документации данной библиотеки.



Рис. 6. Главная страница документации по PADE

Установка, согласно документации, была проведена через pip. Поставим 10 баллов за простоту установки.

Разработчиками подтверждена работа под управлением Windows, на тестовом примере была испытана работа в системе Windows 10. Итого 9 баллов за работу в Windows.

Был написан тестовый скрипт для проверки работоспособности системы:

```
from pade.misc.common import set_ams, start_loop

if __name__ == '__main__':
    set_ams('localhost', 8000)
    start_loop(list(), gui = True)
```

При запуске скрипта была выведена ошибка. В дальнейшем оказалось, что данная библиотека также не предназначена для работы под управлением Python 3.

В документации были хорошо и понятно расписаны возможности системы. Поставим 8 баллов за понятность.

На рис. 7 представлен список документации по этой библиотеке.

В наличии множество различных начальных примеров. Поставим 8 баллов за начальные примеры. Ниже представлен один из них (рис. 8).

Технической документации по библиотеке немного, но это связано это, скорее всего, с легковесностью самой библиотеки. Поставим 7 баллов за наличие технической документации. На главной странице библиотеки дана ссылка на GitHub проекта. Система заслуживает 8 баллов за поддержку проекта. Данная библиотека является FIPA-совместимой, что в результате добавляет ей 3 балла по критерию «отличительные особенности».

Библиотека aiomas. Aiomas [9] — простая в использовании библиотека для каналов типа «запрос—ответ», удаленного вызова процедур (RPC) и мультиагентных систем (МАС). Она написана на Python на основе библиотеки `asyncio`. Библиотека предоставляет три уровня абстракции построенных на основе TCP доменных сокетов:

- каналы «запрос — ответ»;
- вызовы удаленных процедур (RPC);
- агенты и контейнеры.

Как выглядит документация библиотеки, можно увидеть на рис. 9.



Рис. 7. Список документации по библиотеке PADE

Creating my first agent

In order to create your first agent, open your favorite text editor and type (or copy and paste) the following code:

```

from pade.misc.utility import display_message
from pade.misc.common import set_ams, start_loop
from pade.core.agent import Agent
from pade.acl.aid import AID

class AgenteHelloWorld(Agent):
    def __init__(self, aid):
        super(AgenteHelloWorld, self).__init__(aid=aid, debug=False)
        display_message(self.aid.localname, 'Hello World!')

if __name__ == '__main__':

    set_ams('localhost', 8000, debug=False)

    agents = list()

    agente_hello = AgenteHelloWorld(AID(name='agente_hello'))
    agente_hello.ams = {'name': 'localhost', 'port': 8000}
    agents.append(agente_hello)

    start_loop(agents, gui=True)

```

Рис. 8. Усложненный начальный пример библиотеки PADE

Docs » Welcome to aiomas' documentation! [Edit on Bitbucket](#)

Welcome to aiomas' documentation!

PyPI | [Bitbucket](#) | [Mailing list](#) | IRC: #asyncio

aiomas is an easy-to-use library for *request-reply channels*, *remote procedure calls (RPC)* and *multi-agent systems (MAS)*. It's written in pure Python on top of *asyncio*.

The package is released under the MIT license. It requires Python 3.4 and above and runs on Linux, OS X, and Windows.

Below you'll find a list of features. You can also take a look at the [overview section](#) to learn what aiomas is and see some simple examples. If you like this package, go and [install it!](#)

Рис. 9. Главная страница документации по aiomas

На главной странице подтверждена совместимость библиотеки с Windows, также это было проверено на примере Windows 10. Поставим 9 баллов за работу в Windows.

Согласно документации, устанавливаем библиотеку aiomas через pip. В итоге она получает 10 баллов за простоту установки.

Напишем тестовый скрипт (рис. 10).

```
test.py
1 import asyncio
2 import aiomas
3
4 class Callee(aiomas.Agent):
5     @aiomas.expose
6     def spam(self, times):
7         return 'spam' * times
8
9 class Caller(aiomas.Agent):
10    async def run(self, callee_addr):
11        print(self, 'connecting to', callee_addr)
12        callee = await self.container.connect(callee_addr)
13        print(self, 'connected to', callee)
14        result = await callee.spam(3)
15        print(self, 'got', result)
16
17 container = aiomas.Container.create(('localhost', 5555))
18 callee = Callee(container)
19 caller = Caller(container)
20 aiomas.run(until=caller.run(callee.addr))
21 container.shutdown()
```

Рис. 10. Тестовый скрипт для библиотеки aiomas

Данная программа выведет результат, показанный на рис. 11.

```
C:\Windows\system32>python D:\DropBox\Scripts\aiomas\test.py
Caller('tcp://localhost:5555/1') connecting to tcp://localhost:5555/0
Caller('tcp://localhost:5555/1') connected to CalleeProxy('tcp://localhost:5555/0')
Caller('tcp://localhost:5555/1') got spamspamspam
```

Рис. 11. Результат работы программы

Таким образом, библиотека полностью работоспособна и может быть использована в дальнейшем.

Библиотека оказалась довольно понятной в силу своей простоты, она получает 8 баллов за понятность.

Объем технической документации невелик, но хорошо детализирован. Поставим 6 баллов за наличие технической документации. Начальные примеры рассмотрены достаточно подробно, поэтому дадим за них 8 баллов. На главной странице библиотеки есть ссылка на bitbucket (аналог GitHub) проекта. Поставим 8 баллов за поддержку продукта.

По сути, данная библиотека является «пустышкой», но поддерживает более низкоуровневые протоколы обмена информацией между агентами. Добавляем 2 балла.

Библиотека spyse. Spyse [10] — фреймворк для разработки и платформа для строительства мультиагентных систем с использованием языка Python.

Данная библиотека недоступна для установки посредством pip. Документации на нее больше нет, поскольку ее разработка была приостановлена ради про-

движения библиотеки *pastiche*, которой пока еще нет в открытом доступе. Таким образом, дальнейшее использование данной библиотеки не предусматривается, все баллы для нее нулевые.

Результаты оценки. Результаты оценки всех рассмотренных библиотек сведены в таблицу.

Критерий	Библиотека				
	Spade	LOMAP	PADE	aiomas	spyse
Поддержка Python 3	5	5	5	10	0
Простота установки	9	10	10	10	0
Понятность	10	4	8	8	0
Работа в Windows	10	6	9	8	0
Наличие технической документации	10	3	7	6	0
Наличие начальных примеров	9	8	8	8	0
Наличие поддержки	10	2	8	8	0
Отличительные особенности	8	4	3	2	0
<i>Итого</i>	71	42	58	60	0

Таким образом, побеждает библиотека Spade с ее 71 баллом. Ее преимуществом также является хорошая совместимость с FIPA- и BDI-моделями агентов. Альтернативной библиотекой для применения в проектах на базе языка Python 3 будет библиотека aiomas.

Заключение. В статье рассмотрены различные мультиагентные библиотеки для разработки соответствующих систем посредством языка Python. По различным критериям, нацеленным на оценку в основном удобства конечного пользователя и работоспособности, были проверены пять библиотек: SPADE, LOMAP, PADE, aiomas, spyse.

По результатам данного исследования лучшей по указанным выше критериям оказалась библиотека SPADE. Она получила наивысшие баллы практически по всем пунктам, за исключением поддержки Python 3 и простоты установки. По этим двум критериям лучшие оценки у библиотеки aiomas. Последняя оказалась на втором месте в данном исследовании, хотя по некоторым пунктам она получила не самые высокие баллы.

Библиотека PADE также является неплохой альтернативой первым двум библиотекам.

Данные библиотеки могут быть в дальнейшем использованы для разработки систем искусственного интеллекта на основе теории мультиагентных систем.

Литература

- [1] Peng P., Wen Yi., Yang Ya., Yuan Q., Tang Zh., Long H., Wang J. Multiagent bidirectionally-coordinated nets: emergence of human-level coordination in learning to play StarCraft combat games. URL: <https://arxiv.org/abs/1703.10069> (дата обращения 25.07.2018).

- [2] Leike J., Martic M., Krakovna V., Ortega P.A., Everitt T., Lefrancq A., Orseau L., Legg Sh. AI safety gridworlds. URL: <https://arxiv.org/pdf/1711.09883.pdf> (дата обращения 25.07.2018).
- [3] The Foundation for Intelligent Physical Agents. <http://fipa.org/> (дата обращения 25.07.2018).
- [4] Официальный сайт библиотеки SPADE. URL: <https://pythonhosted.org/SPADE/> (дата обращения 25.07.2018).
- [5] Официальный сайт технологии XMPP. URL: <http://xmpp.org/> (дата обращения 30.08.2018).
- [6] Официальный сайт библиотеки LOMAP. URL: <http://sites.bu.edu/hyness/lomap/> (дата обращения 25.07.2018).
- [7] Официальный сайт библиотеки PADE. URL: <http://pade-docs-en.readthedocs.io/en/latest/> (дата обращения 25.07.2018).
- [8] Официальный сайт библиотеки twisted. URL: <http://twistedmatrix.com/> (дата обращения 28.07.2018).
- [9] Официальный сайт библиотеки aiomas. URL: <https://aiomas.readthedocs.io/en/latest/> (дата обращения 28.07.2018).
- [10] Официальный сайт библиотеки spyse. URL: <https://pypi.org/project/spyse/> (дата обращения 28.07.2018).

Журавлев Александр Максимович — студент кафедры «Информационные системы и телекоммуникации», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Алфимцев Александр Николаевич, кандидат технических наук, доцент кафедры «Информационные системы и телекоммуникации», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

REVIEW OF LIBRARIES FOR CONSTRUCTING THE MULTIAGENT SYSTEMS IN THE PYTHON ENVIRONMENT

A.M. Zhuravlev

zhuravlev.am@rea.ru

SPIN-code: 9327-0412

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The article provides a review of the existing libraries based on which it is possible to construct complex multiagent systems. When choosing the best library it is important to consider its operating capacity, developers' support, ease of use and installation in the most popular operational system Windows, i.e. to conduct qualitative analysis of the libraries by the main characteristics concerning the end user's comfort. The article provides examples of the simplest scripts for testing the operating capacity of each library and analyzes the peculiar features of the libraries and some errors that can occur during their installation. The relevance of the article relates to the increasing interest to artificial intelligence on the whole and to multiagent systems and their development in particular.

Keywords

Multiagent systems, artificial intelligence, framework, Python, terminal, Windows, FIPA (Foundation for Intelligent Physical Agents), BDI (belief, desire, and intention)

Received 10.09.2018

© Bauman Moscow State Technical University, 2018

References

- [1] Peng P., Wen Yi., Yang Ya., Yuan Q., Tang Zh., Long H., Wang J. Multiagent bidirectionally-coordinated nets: emergence of human-level coordination in learning to play StarCraft combat games. Available at: <https://arxiv.org/abs/1703.10069> (accessed 25 July 2018).
- [2] Leike J., Martic M., Krakovna V., Ortega P.A., Everitt T., Lefrancq A., Orseau L., Legg Sh. AI safety gridworlds. Available at: <https://arxiv.org/pdf/1711.09883.pdf> (accessed 25 July 2018).
- [3] The Foundation for Intelligent Physical Agents. Available at: <http://fipa.org/> (accessed 25 July 2018).
- [4] SPADE library official website. Available at: <https://pythonhosted.org/SPADE/> (accessed 25 July 2018).
- [5] XMPP technology official website. Available at: <http://xmpp.org/> (accessed 30 August 2018).
- [6] LOMAP library official website. Available at: <http://sites.bu.edu/hyness/lomap/> (accessed 25 July 2018).
- [7] PADE library official website. Available at: <http://pade-docs-en.readthedocs.io/en/latest/> (accessed 25 July 2018).
- [8] twisted library official website. Available at: <http://twistedmatrix.com/> (accessed 28 July 2018).
- [9] aiomas library official website. Available at: <https://aiomas.readthedocs.io/en/latest/> (accessed 28 July 2018).
- [10] spysy library official website. Available at: <https://pypi.org/project/spysy/> (accessed 28 July 2018).

Zhuravlev A.M. — student, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — A.N. Alfimtsev, Cand. Sc. (Eng.), Assoc. Professor, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University, Moscow, Russian Federation.