

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления
Кафедра Интеллектуальных информационных технологий

ОТЧЁТ
по ознакомительной практике

Выполнил:

Е. С. Мухомедзянов

Студент группы
421701

Проверил:

Н. В. Малиновская

Минск 2025

СОДЕРЖАНИЕ

Введение	3
1 Формализация и сравнительный анализ языков представления информации в интеллектуальных системах: AgentSpeak(Jason) и GOAL	4
2 Формализация и сравнительный анализ инструментов для разработки интеллектуальных систем: Jason и AutoGen	11
Заключение	15
Список использованных источников	17

ВВЕДЕНИЕ

Цель:

Провести сравнительный анализ современных технологий разработки интеллектуальных систем, закрепить навыки формализации информации, а также приобрести и развить навыки исследования, анализа и сравнения технологических решений.

Задачи:

1. Изучить языки агентно-ориентированные языки программирования (AgentSpeak(Jason) и GOAL) как инструменты формализации декларативных знаний, провести анализ их особенностей, выявить сходства и различия между ними.
2. Рассмотреть современные инструменты для работы с инструментами для разработки многоагентных систем (AgentSpeak(Jason) и GOAL), выявить их функциональные возможности и подходы к представлению знаний, сходства и отличительные особенности.
3. Формализовать полученные результаты в виде SCn-кода.
4. Оформить библиографические источники к полученным результатам по ГОСТ 7.1-2003.

1 ФОРМАЛИЗАЦИЯ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЯЗЫКОВ ПРЕДСТАВЛЕНИЯ ИНФОРМАЦИИ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ: AGENTSPEAK(JASON) И GOAL

AgentSpeak(Jason)

- :=** [Язык агентно-ориентированного программирования, разработанный Ананд С. Рао, Жоме Ф. Хюбнер и Рафаэл Х. Бордини]
- :=** [AgentSpeak(L)]
- :=** [AgentSpeak (Jason)]
- ∈** *агентно-ориентированный язык*
- ⇒** *разработчик**:
 - Ананд С. Рао
 - Жоме Ф. Хюбнер
 - Рафаэл Х. Бордини
- ⇒** *год создания**:
1996
- ⇒** *основная парадигма**:
декларативное и агентно-ориентированное программирование, BDI-модель (Belief–Desire–Intention)
- ⇒** *описание**:
[AgentSpeak — это агентно-ориентированный язык программирования, основанный на парадигме рациональных агентов BDI (Belief-Desire-Intention). Он позволяет разрабатывать интеллектуальных агентов, поведение которых определяется знаниями об окружающем мире (убеждения), целями (желания) и планами их достижения (намерения).]
- ⇒** *описание**:
[Jason — это интерпретатор и расширение языка AgentSpeak, реализованный на Java. Он предоставляет среду разработки и выполнения для создания многоагентных систем, ориентированных на автономное поведение, реактивность и проактивность агентов.]
- ⇒** *назначение**:
[AgentSpeak(Jason) активно используется в исследованиях в области искусственного интеллекта и для моделирования систем, где необходима автономия, адаптивность и взаимодействие между множеством интеллектуальных агентов.]
- ⇒** *основные конструкции**:
 - { • [верование(belief)]
 - [цель(goal)]
 - [событие(event)]
 - [план(plan)]
 - [действие (action)]
 - }
- ⇒** *среды разработки**:
 - { • [Jason IDE]
 - :=** [Графическая интегрированная среда разработки на базе Eclipse, предназначенная для создания, редактирования, исполнения и отладки программ на AgentSpeak(L)]
 - ⇒** *примечание**:

[Обеспечивает удобный интерфейс для работы с агентами, включает подсветку синтаксиса, шаблоны кода, средства запуска и визуализации состояний агентов. Позволяет напрямую интегрировать Java-код, отлаживать логику BDI и наблюдать за перцепциями и планами в режиме исполнения.]

- [Jason CLI (Command-Line Interface)]

⇒ *примечание**:

[Подходит для серверного исполнения, автоматического тестирования MAS-программ и встраивания Jason-агентов в более крупные системы. Позволяет быстро запускать .mas2j-проекты и получать вывод из среды агента.]

- [Jason + Moise+]

⇒ *примечание**:

[Позволяет агентам действовать в соответствии с формально определённой организационной структурой: ролями, обязанностями, схемами взаимодействия. Используется для моделирования командной работы, иерархий и договорённостей между агентами.]

- [Jason Web Integration / Jason-as-a-Service]

⇒ *примечание**:

[Основано на создании REST-интерфейсов и запуске Jason-агентов как компонентов серверной логики. Используется в проектах по построению диалоговых агентов или облачных MAS.]

}

⇒ *стратегия вывода**:

BDI-цикл рассуждения (выбор плана по событию)

⇒ *поддержка**:

- [асинхронная коммуникация речевыми актами]
- [параллельное выполнение агентов с независимыми циклами принятия решений]
- [обработка событий и целей с приоритетами и условиями активации]
- [интеграция с Java]
- [работа в распределённой среде]

⇒ *применение**:

- *симуляцию поведения групп агентов (например, роботов)*
- *разработку интеллектуальных помощников*
- *обучение и моделирование в области ИИ*
- *соревнования Multi-Agent Programming Contest*

⇒ *примеры использования**:

- {• [Проект JADE and Jason Integration (Франция, INRIA)]
- [RoboCup Rescue Simulation League (Международные соревнования по спасательным роботам)]

⇒ *примечание**:

[Команды агентов (например, пожарные, медики, полицейские) были реализованы с помощью Jason и взаимодействовали в реальном времени с симулятором, принимая решения на основе перцепций (входных данных) и заранее определённых планов (например, «если здание горит и доступно, отправить агента тушить пожар»)].

- [Исследование адаптивных умных домов (США, Университет Джорджии)]
- [Учебные проекты “Охотники за пищей”, “Wumpus World”]

}

- ⇒ *реализации**:
- {• *Jason*
 - *Python AgentSpeak*
- ⇒ *описание**:
- [Неофициальная реализация AgentSpeak(L) на языке Python, ориентированная на образовательные и экспериментальные цели. Позволяет писать BDI-агентов с использованием синтаксиса, вдохновлённого Jason, но в среде Python.]
- *AgentSpeak(XL), 2APL*
- ⇒ *описание**:
- [Расширенная версия AgentSpeak, предложенная в рамках проекта 3APL/2APL, с дополнительными конструкциями для обработки планов и интеграции с внешними действиями. Обеспечивает более гибкое управление планами и возможностями агента.]
- }
- ⇒ *особенности**:
- {• [простота логического описания поведения агентов]
- := [AgentSpeak использует декларативный подход для задания целей и поведения, приближённый к естественному языку рассуждений]
- [гибкость и расширяемость]
- := [Jason поддерживает интеграцию с Java, что позволяет подключать собственные функции, библиотеки и компоненты среды]
- [высокая пригодность для обучения и исследований]
- := [Jason активно используется в университетах и научных проектах благодаря открытости, документации и поддержке стандартов BDI]
- [открытый исходный код]
- := [Jason распространяется по лицензии GPL и может быть модифицирован под конкретные задачи]
- [интеграция с организационными и средовыми расширениями]
- := [Поддержка Moise+ и CArtAgO позволяет моделировать сложные организационные и физические аспекты многоагентных систем]
- [мощные средства отладки и визуализации]
- := [Средства визуального анализа BDI-состояний агентов в Jason IDE упрощают тестирование и верификацию поведения]
- }
- ⇒ *ограничения**:
- {• [отсутствие встроенного графического интерфейса среды]
- := [Jason не включает визуальной среды моделирования агентов и окружения по умолчанию]
- ⇒ *уточнение**:
- [Для моделирования среды и взаимодействия с физическим или виртуальным миром требуется подключение внешних библиотек, таких как CArtAgO.]
- [ограниченная масштабируемость]
- := [При большом количестве агентов или сложных коммуникационных паттернах производительность может снижаться]
- ⇒ *уточнение**:
- [Jason не оптимизирован для высоконагруженных распределённых систем, и при моделировании сотен агентов требуется дополнительная оптимизация.]

- [отсутствие встроенного обучения]
:= [Jason не поддерживает механизмы машинного обучения или адаптации поведения агентов]
⇒ *уточнение**:
[Изменение поведения возможно только путём жёстко заданных правил и перепланирования вручную, без самонастройки.]
 - [сложности отладки в распределённой среде]
:= [Трассировка и анализ поведения агентов в распределённых MAS затруднены при большом объёме взаимодействий]
⇒ *уточнение**:
[Хотя Jason IDE предоставляет Mind Inspector, для распределённых сред требуется внешний мониторинг и логгирование.]
 - [отсутствие поддержки динамически изменяемых планов]
:= [Агенты не могут на лету модифицировать свои планы или генерировать новые без перепрограммирования]
⇒ *уточнение**:
[Планы задаются статически в исходном коде и не могут быть автоматически синтезированы в ходе исполнения.]
- ⇒ *библиографические источники**:
- { • [Rao A.S. AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language. MAAMAW'96.]
 - [Hubner J., Bordini R. Jason: A Java-based Interpreter for an Extended Version of AgentSpeak. 2005.]
 - [Bordini R., Hubner J., Wooldridge M. Programming Multi-Agent Systems in AgentSpeak using Jason. Wiley, 2007.]
- }

GOAL

- := [Агентно-ориентированный язык программирования для создания рациональных BDI-агентов, разработанный в Университете Утрехта (University of Utrecht), Нидерланды.]
- := [GOAL (Goal-Oriented Agent Language)]
- ∈ *агентно-ориентированный язык*
- ⇒ *разработчик**:
- *Кожиман Субраманян (Koen Hindriks)*
 - *Вирт Вербург (Virgil Verbrugge)*
- ⇒ *год создания**:
2001
- ⇒ *основная парадигма**:
декларативное и агентно-ориентированное программирование, BDI-модель (Belief–Desire–Intention)
- ⇒ *описание**:
[GOAL — это специализированный язык программирования интеллектуальных агентов, основанный на BDI-модели, где агенты управляются своими убеждениями (Beliefs), целями (Goals) и действиями. Язык предоставляет декларативные средства для задания логики агентов и взаимодействия между ними.]
- ⇒ *описание**:
[В GOAL акцент сделан на чистое разделение между знаниями и целями агента, а также на формальное определение условий активации планов. GOAL обеспечивает

- формальную верифицируемость поведения агентов, что делает его особенно подходящим для разработки надёжных автономных систем.]
- ⇒ *назначение**:
- [GOAL предназначен для создания интеллектуальных агентов, способных действовать автономно в динамической среде, принимать решения на основе текущих убеждений и целей, а также взаимодействовать с другими агентами в многоагентных системах.]
- ⇒ *основные конструкции**:
- {• [belief base (база знаний)]
 - [goal base (база целей)]
 - [conditional actions (условные действия)]
 - [event module (обработка событий)]
 - [action specifications (декларации действий)]
 - }
- ⇒ *среды разработки**:
- {• [GOAL IDE]
 - := [Интегрированная среда разработки, реализованная на Java, включает редактор, отладчик и симулятор поведения агентов]
 - ⇒ *примечание**:
 - [Позволяет программировать агентов, выполнять тестовые сценарии, визуализировать состояния когнитивной архитектуры и пошагово отслеживать принятие решений.]
 - [GOAL CLI]
 - := [Командная среда для запуска и тестирования GOAL-программ без графического интерфейса]
 - ⇒ *примечание**:
 - [Подходит для пакетной обработки, CI-тестов и интеграции с другими системами.]
 - }
- ⇒ *стратегия вывода**:
- {• [событийно-управляемая активация планов (реактивный BDI-подход)]
 - [оценка условий активации планов и выбор действий]
 - }
- ⇒ *поддержка**:
- [асинхронная коммуникация между агентами]
 - [работа с логическими базами знаний и целями]
 - [поддержка многозадачности на уровне агентов]
 - [возможность формальной верификации поведения агентов]
- ⇒ *применение**:
- *моделирование рационального поведения автономных агентов*
 - *многоагентные симуляции и распределённые системы*
 - *интеллектуальные роботы и виртуальные помощники*
 - *автоматизация принятия решений*
 - *игры и моделирование социальных сценариев*
- ⇒ *примеры использования**:
- {• [образовательные симуляции: курсы и лабораторные работы в Нидерландах и других странах по теме агентных систем]
 - [многоагентные модели поведения в кризисных ситуациях и социальных симуляциях]
 -

- [проекты по автономной навигации роботов на соревнованиях и в университетских лабораториях]
 - [интеллектуальные помощники для планирования задач и поведения в виртуальных средах]
 - [исследования в области автоматизации юридических рассуждений и логики обязательств]
- }
- ⇒ *реализации**:
 - {• *GOAL Agent Programming Platform*
 - *Eclipse Plugin for GOAL*
 - *интеграция с среды Environments и MAS-tools*
- }
- ⇒ *особенности**:
 - {• [логическая строгость и формальная семантика]
 - [возможность верификации программ агентов]
 - [модульность и чистое разделение когнитивных компонентов]
 - [поддержка декларативных и реактивных моделей поведения]
 - [интеграция с Java и Prolog]
 - [использование в академических и исследовательских целях]
- }
- ⇒ *ограничения**:
 - {• [менее развитая экосистема по сравнению с более популярными языками, как Prolog или Python]
 - [высокий порог входа для новичков без подготовки в логике и агентных системах]
 - [отсутствие поддержки динамически изменяемых структур данных]
 - [ограниченная поддержка промышленных задач и интеграций]
- }
- ⇒ *библиографические источники**:
 - {• [Hindriks K.V., de Boer F.S., van der Hoek W., Meyer J.-J.Ch. Agent Programming in GOAL: A Logic-Based Approach. AAMAS 2001. URL: <https://homepages.cwi.nl/~koen/PUBLICATIONS/aamas2001.pdf>]
 - [GOAL Programming Platform. URL: <https://goalapl.atlassian.net/>]
 - [Koen V. Hindriks. Programming Rational Agents using GOAL. Lecture Notes, TU Delft, 2023. URL: <https://www.cs.tudelft.nl/~koen/>]
 - [Agent-Based Modelling and Simulation with GOAL: Applications and Techniques. Springer, 2021.]
- }

Сравнение языков логического программирования *AgentSpeak(Jason)* и *GOAL*

- = {• *AgentSpeak(Jason)*
- *GOAL*
- }
- ⇒ *сходства**:
 - {• [Оба языка реализуют декларативный и агентно-ориентированный подход на основе BDI-модели (Belief–Desire–Intention)]
 - [Позволяют описывать поведение интеллектуальных агентов с помощью убеждений, целей и планов]
 - [Используют события и механизмы выбора планов для реагирования на изменения окружающей среды]

- [Предназначены для построения автономных, реактивных и проактивных агентов]
- [Поддерживают разработку многоагентных систем, взаимодействующих между собой]
- [Обладают формальной семантикой и используются в академических исследованиях и обучении]

}

⇒

различия:*

- {• [GOAL чётко разделяет убеждения и цели в отдельных базах, в то время как в Jason они более гибко определяются через события и планы]
- [GOAL акцентирует внимание на верифицируемости и логической строгости, Jason — на гибкости и расширяемости среды исполнения]
- [GOAL имеет собственную среду исполнения и инструменты отладки, Jason реализован как интерпретатор на Java и легко интегрируется с Java-приложениями]
- [GOAL использует декларативный стиль с логическими правилами для активации действий, в Jason планы записываются в императивной форме с условиями активации]
- [Jason поддерживает широкие возможности коммуникации через речевые акты, в GOAL коммуникация возможна, но реализуется менее подробно]
- [Jason активно применяется в международных соревнованиях (например, RoboCup), в то время как GOAL чаще используется в университетских исследованиях и обучении]

}

⇒

рекомендации по выбору:*

- {• [GOAL рекомендуется использовать в проектах, где важна формальная верификация поведения агентов, логическая строгость и моделирование когнитивных процессов]
- [GOAL хорошо подходит для обучения логике и основам BDI-подхода в университетских курсах]
- [AgentSpeak (Jason) целесообразно выбирать для создания гибких, расширяемых многоагентных систем с акцентом на взаимодействие, интеграцию с Java и участие в симуляциях или соревнованиях]
- [Jason удобен для прототипирования и разработки систем, где требуется динамическое поведение агентов и сложные речевые взаимодействия]

}

2 ФОРМАЛИЗАЦИЯ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ ИНСТРУМЕНТОВ ДЛЯ РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ: JASON И AUTOGEN

Jason

- \equiv [Инструмент и платформа для программирования многоагентных систем на базе языка AgentSpeak]
- \equiv [Платформа для разработки BDI-агентов (belief-desire-intention)]
- \in *агентно-ориентированное программное обеспечение*
- \Rightarrow *разработчик**:
 - *Jomi F. Hübner*
 - *Rafael H. Bordini*
- \Rightarrow *год создания**:
2005
- \Rightarrow *текущая стабильная версия**:
[2.5.1]
- \Rightarrow *дата выпуска текущей версии**:
[30 января 2024]
- \Rightarrow *лицензия**:
GNU GPL v3
- \Rightarrow *язык разработки**:
Java
- \Rightarrow *назначение**:
[Разработка и моделирование BDI-многоагентных систем с автономным поведением на основе логических правил и целей]
- \Rightarrow *варианты исполнения**:
 - { • *Jason IDE*
 - \equiv [Встроенная среда разработки на базе Eclipse]
 - \Rightarrow *примечание**:
[Поддержка редактирования кода на AgentSpeak, управление проектами, отладка и визуализация среды]
 - *Jason CLI*
 - \equiv [Командная оболочка для запуска MAS-программ Jason]
 - \Rightarrow *примечание**:
[Позволяет исполнять агенты без GUI, подходит для серверного или тестового запуска]
- }
- \Rightarrow *функциональные возможности**:
 - { • [интерпретация планов BDI]
 - [обработка отказов и метаданные планов]
 - [интеграция с Java-библиотеками и пользовательскими внутренними действиями]
 - [визуальные инструменты отладки («mind inspector»)]
 - }
- \Rightarrow *поддерживаемые языки**:
 - { • *AgentSpeak(L)*
 - *FIPA-ACL (через JADE/SACI)*
 - }

- ⇒ *экосистема расширений**:
 - { • *Moise+*
 - *Cartago*
 - *Jason IDE*
 - *FIPA Messaging*
 - }
- ⇒ *преимущества**:
 - { • [простота логического описания поведения агентов]
 - [гибкость и расширяемость за счёт Java-интеграции]
 - [хорошо подходит для академических и исследовательских проектов]
 - [открытый исходный код и активное сообщество]
 - [интеграция с JADE, OpenMAS и другими средами]
 - }
- ⇒ *недостатки**:
 - { • [относительно низкий уровень абстракции среды исполнения]
 - [ограниченные графические средства моделирования и визуализации]
 - [отсутствие встроенных средств машинного обучения или генерации поведения]
 - }
- ⇒ *библиографические источники**:
 - { • [Bordini R.H., Hübner J.F., Wooldridge M. Programming Multi-Agent Systems in AgentSpeak using Jason. Wiley, 2007.]
 - [Jason [Электронный ресурс] // Официальный сайт. URL: <https://jason.sourceforge.net> (дата обращения: 12.05.2025).]
 - [Bordini R.H., et al. Multi-Agent Programming: Languages, Tools and Applications. Springer, 2009.]
 - }

AutoGen

- := [Платформа для создания многоагентных систем, основанных на LLM и управляемых автогенерацией действий и ролей]
- ∈ *средство для создания LLM-агентов*
- ∈ *агентно-ориентированное программное обеспечение*
- ⇒ *разработчик**:
Microsoft Research
- ⇒ *год создания**:
2023
- ⇒ *лицензия**:
MIT
- ⇒ *текущая стабильная версия**:
0.2.5
- ⇒ *дата выпуска текущей версии**:
12 марта 2025
- ⇒ *язык разработки**:
Python
- ⇒ *назначение**:
[Оркестровка агентов с LLM-ядром для автоматизации задач, требующих координации, общения и ролевой специализации]
- ⇒ *архитектурные особенности**:
[AutoGen построен на взаимодействии между агентами, каждый из которых пред-

- ставляет собой LLM-инстанс с определённой ролевой функцией. Поддерживаются циклы общения, делегирование, принятие решений и выполнение задач через инструменты.]
- ⇒ варианты исполнения*:
- {• *AutoGen Core*
 - ⇒ *примечание**:
 - [Позволяет моделировать как одиночные агенты, так и команды с контролем ролевого поведения]
 - *AutoGen Studio (в разработке)*
 - := [Визуальная среда проектирования многоагентных сценариев]
- ⇒ функциональные возможности*:
- {• [ролевое программирование LLM-агентов]
 - [создание сценариев кооперации, конкуренции, делегирования]
 - [поддержка инструментов (функций, API), вызываемых агентами]
 - [использование встроенной памяти и истории взаимодействия]
 - [возможность асинхронного исполнения агентов]
 - [интеграция с внешними API и локальными функциями через Python-интерфейсы]
 - [обработка запросов, генерация текстов, планирование действий]
- ⇒ поддерживаемые модели*:
- {• *OpenAI GPT (через API)*
 - *Azure OpenAI*
 - *локальные модели через OpenAI-клонировующие интерфейсы (включая LM Studio)*
- ⇒ преимущества*:
- {• [гибкость построения диалоговых и функциональных цепочек агентов]
 - [высокий уровень абстракции и простота настройки логики через Python]
 - [интеграция с современными LLM для широкого круга задач]
 - [поддержка автономных агентов с планированием и памятью]
 - [активная разработка и поддержка со стороны Microsoft Research]
- ⇒ недостатки*:
- {• [зависимость от внешних LLM-платформ или ресурсов GPU]
 - [относительная новизна и нестабильность некоторых функций]
 - [отсутствие полноценной визуальной среды на момент мая 2025 года]
 - [отсутствие встроенного редактора ролей и сценариев]
- ⇒ библиографические источники*:
- {• [AutoGen [Электронный ресурс] // GitHub repository. URL: <https://github.com/microsoft/autogen> (дата обращения: 12.05.2025).]
 - [Liu B., Wang D., Wang Y. et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. arXiv preprint arXiv:2309.14610, 2023.]
 - [Microsoft Research Blog on AutoGen. URL: <https://www.microsoft.com/en-us/research/blog> (дата обращения: 12.05.2025).]

Сравнение систем Jason и AutoGen

```

=    { •   Jason
      •   AutoGen
    }
⇒   сходства*:
    { •   [Поддерживают создание многоагентных систем]
      •   [Обеспечивают автономное поведение агентов]
      •   [Поддерживают межагентное взаимодействие и передачу сообщений]
      •   [Реализуют программную модель агентов с внутренним состоянием и логикой поведения]
      •   [Позволяют строить сложные сценарии взаимодействия]
      •   [Расширяются за счёт пользовательских функций и внешних библиотек]
    }
⇒   различия*:
    { •   [Jason ориентирован на BDI-модель и логическое программирование, AutoGen — на диалоговые LLM-модели и ролевую координацию]
      •   [Jason использует AgentSpeak и реализован на Java, AutoGen написан на Python и использует современные LLM (например, GPT-4)]
      •   [AutoGen предполагает наличие доступа к языковым моделям через API или локальные модели, Jason работает автономно]
      •   [Jason фокусируется на логике действий, перцепциях и намерениях, AutoGen — на генерации ответов, стратегий и коллаборации между LLM-агентами]
      •   [AutoGen ориентирован на автоматизацию LLM-задач (чтение, планирование, вызовы функций), Jason — на автономных программных агентов в искусственных средах]
      •   [Jason имеет зрелую экосистему и документацию, AutoGen — активно развивающийся, но ещё формирующийся проект]
    }
⇒   рекомендации по выбору*:
    { •   [Jason рекомендуется для учебных, исследовательских и симуляционных задач, связанных с классическими агентными моделями BDI]
      •   [Jason рекомендуется для учебных, исследовательских и симуляционных задач, связанных с классическими агентными моделями BDI]
      •   [AutoGen целесообразен для построения современных LLM-приложений с агентной координацией, включая ассистентов, планировщиков, агентов-аналитиков]
      •   [AutoGen предпочтителен при необходимости в высокоуровневом управлении агентами и взаимодействии с внешними системами через языковые модели]
      •   [Jason лучше подойдёт для реализации агентов в ограниченных средах, AutoGen — для задач, требующих гибкости, творчества и адаптивности через LLM]
    }

```

ЗАКЛЮЧЕНИЕ

В ходе проведённого исследования был выполнен сравнительный анализ современных языков и инструментов, применяемых для проектирования и реализации многоагентных интеллектуальных систем.

В первом разделе были подробно рассмотрены агентно-ориентированные языки программирования AgentSpeak (Jason) и GOAL. Оба языка реализуют парадигму BDI (Belief–Desire–Intention), однако различаются в архитектуре управления знаниями и целями, стиле описания поведения агентов и уровне формализации. Jason обеспечивает большую гибкость и интеграцию с Java, что делает его подходящим для симуляций и прототипирования. GOAL, в свою очередь, ориентирован на строгость логической модели и формальную верификацию поведения агентов. Основные конструкции, принципы вывода и области применения каждого языка были формализованы в SCn-коде.

Во втором разделе проанализированы два современных инструмента для разработки многоагентных систем — Jason и AutoGen. Jason представляет собой зрелую платформу с поддержкой BDI-агентов и встроенными средствами визуализации, отладки и взаимодействия с артефактами и организационными структурами. AutoGen, разработанный Microsoft Research, использует мощь языковых моделей (LLM) для координации агентов, специализации ролей и автоматизации сложных задач. Он ориентирован на построение современных LLM-приложений и взаимодействие на естественном языке. Функциональные возможности и архитектурные особенности этих инструментов также были описаны и представлены в SCn-формате.

Проведённый анализ позволил выявить ключевые различия и области применимости каждого из языков и инструментов. Полученные результаты могут быть использованы при проектировании интеллектуальных систем, требующих гибкой координации агентов, адаптивного поведения и поддержки семантической обработки знаний. Также сформулированы рекомендации по выбору технологий в зависимости от задач моделирования, уровня формализации и требований к масштабируемости и интеграции.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Bordini, Rafael H. Programming Multi-Agent Systems in AgentSpeak using Jason / Rafael H. Bordini, Jomi Fred Hübner, Michael Wooldridge. — John Wiley & Sons, 2007.
- [2] Rao, Anand S. Agentspeak(1): Bdi agents speak out in a logical computable language / Anand S. Rao // Proc. of the 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW). — 1996. — P. 42–55.
- [3] Hübner, Jomi F. Jason: A java-based interpreter for an extended version of agentspeak / Jomi F. Hübner, Rafael H. Bordini // Proceedings of AAMAS'05 Workshop on Programming Multi-Agent Systems (ProMAS). — 2005.
- [4] Agent programming in GOAL: A logic-based approach / Koen V. Hindriks [et al.] // Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS). — 2001. — P. 752–759. <https://homepages.cwi.nl/~koen/PUBLICATIONS/aamas2001.pdf>.
- [5] Hindriks, Koen V. Programming rational agents using goal: Lecture notes. — <https://www.cs.tudelft.nl/~koen/>. — 2023. — TU Delft.
- [6] Research, Microsoft. Autogen: Enabling next-gen llm applications via multi-agent conversation. — <https://github.com/microsoft/autogen>. — 2023. — Accessed: 12.05.2025.
- [7] Autogen: Enabling next-gen llm applications via multi-agent conversation / Bolin Liu [et al.] // arXiv preprint arXiv:2309.14610. — 2023.
- [8] Research, Microsoft. Autogen project overview. — <https://www.microsoft.com/en-us/research/blog>. — 2025. — Accessed: 12.05.2025.
- [9] Wooldridge, Michael. Multiagent systems / Michael Wooldridge // The Handbook of Artificial Intelligence. — MIT Press, 2009. — P. 887–926.
- [10] Bordini, Rafael H. Programming multi-agent systems in agentspeak using jason / Rafael H. Bordini, Jomi Fred Hübner, Michael Wooldridge // Computational Logic in Multi-Agent Systems. — 2005. — Vol. 3487. — P. 66–85.
- [11] Wooldridge, Michael. An Introduction to MultiAgent Systems / Michael Wooldridge. — 2nd ed. — John Wiley & Sons, 2009.
- [12] Hindriks, Koen V. Programming rational agents in goal / Koen V. Hindriks // Multi-Agent Programming: Languages, Tools and Applications / ed.: Rafael Bordini [et al.]. — Springer, 2009. — P. 119–157.

[13] Goal types in agent programming / Mehdi Dastani [et al.] // Autonomous Agents and Multi-Agent Systems. — 2006. — Vol. 13, № 3. — P. 271–303.

[14] Practical reasoning for goal agents / Mehdi Dastani [et al.] // Proc. of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS). — 2004. — P. 702–709.