

Федеральное государственное учреждение
«ФЕДЕРАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР
«ИНФОРМАТИКА И УПРАВЛЕНИЕ»
Российской академии наук»

На правах рукописи

Киселёв Глеб Андреевич

**Разработка методов, моделей и экспериментальных средств
исследования коалиционного поведения когнитивных агентов**

Специальность: 05.13.01 - Системный анализ, управление и обработка
информации

Диссертация на соискание учёной степени

кандидата технических наук

Научный руководитель:
доктор технических наук
профессор Дивеев А.И.

Москва - 2020

Оглавление

Оглавление	2
Введение	4
1 Обзор предпосылок создания архитектуры агента со знаковой картиной мира	9
1.1 Психологические и нейрофизиологические предпосылки синтеза плана поведения	9
1.1.1 Знаковое представление деятельности агента. Процессы распространения активности	9
1.1.2 Персонифицированность знаний когнитивного агента	12
1.1.3 Модель пространственной логики Поспелова	13
1.1.4 Состояние исследований по пространственному представлению знаний когнитивным агентом	16
1.1.5 Выводы параграфа 1.1	19
1.2 Особенности использования поведенческого планировщика в когнитивных архитектурах	20
1.2.1 Состояние исследований на данный момент	20
1.2.2 Моделирование внимания когнитивного агента	26
1.2.3 Робототехнические реализации когнитивных архитектур	28
1.2.4 Выводы параграфа 1.2	29
1.3 Особенности коалиционного планирования поведения когнитивных агентов	30
1.3.1 Принципы распределения задач в коллективах агентов	31
1.3.2 Способы коммуникации когнитивных агентов	33
1.3.3 Выводы параграфа 1.3	35
1.4 Выводы главы 1	36
2 Планировщики семейства MAP	37
2.1 Формальная модель компонент знака	39
2.2 MAPCORE - Прецедентный и иерархический способ синтеза плана поведения	40
2.2.1 Процесс пополнения картины мира агента	40
2.2.2 Алгоритм синтеза плана поведения	43
2.2.3 Алгоритм создания прецедента планирования	49
2.2.4 Выводы параграфа 2.1	50
2.3 MAPSPATIAL - Синтез пространственного плана поведения	52
2.3.1 Моделирование фокуса внимания и описание карты местности	52
2.3.2 Процедуры уточнения и абстрагирования внимания агента	55
2.3.3 Алгоритм поиска пространственных подпланов	57

2.3.4	Выводы параграфа 2.2	59
2.4	MAPMULTI - Синтез многоагентного плана поведения.....	60
2.4.1	Динамическое распределение ролей в коалиции. Аукцион планов.....	60
2.4.2	Синтез многоагентного пространственного плана поведения	63
2.4.3	Выводы параграфа 2.3	64
2.5	Выводы главы 2	65
3	Библиотека map-core.....	66
3.1	Структура библиотеки	66
3.1.1	Обзор модулей строения семиотической сети	66
3.1.2	Обзор модулей планирования.....	68
3.2	Структура библиотеки map-multi.....	75
3.3	Структура библиотеки map-spatial.....	77
3.4	Экспериментальные исследования	81
3.4.1	Классические, иерархические и пространственные эксперименты	81
3.4.2	Эксперименты по адаптации МАР к робототехнической платформе	84
3.5	Выводы главы 3	89
4	Робототехническая реализация.....	90
4.1	Общее описание системы	90
4.2	Перемещение робототехнической платформы.....	94
4.3	Вывод главы 4	97
	Заключение	98
	Список литературы	100
	Приложение А. Когнитивные архитектуры.....	109

Введение

Актуальность темы исследования

Способность синтезировать комплексный план поведения является важной задачей при построении системы управления автономным интеллектуальным агентом, который действует в виртуальной или реальной среде. Целенаправленная деятельность агента базируется на синтезированном плане и является продуктом цикла принятия решений, который требует учета механизмов восприятия и анализа окружающей среды, целеполагания, фокусировки внимания и обучения. Рассмотрение сложных сред и взаимодействие агента с различными объектами в условиях непредсказуемой динамики требуют учета в процессе синтеза управления агентом возможности как иерархической формулировки задачи, так и классических для искусственного интеллекта описания терминальных состояний. Процесс синтеза управления агентом на основе плана осуществляется на этапе предварительной подготовки агента и происходит до реального взаимодействия агента и среды. Тем не менее, для сокращения времени на синтез управления осуществляется применение механизмов ускорения составления плана, его анализа и корректировки по результатам выполнения действий. Предложенный в диссертации подход решает задачу ускорения составления плана с помощью множественного использования прецедентных знаний о возможных способах достижения целей.

В последнее время становится актуальной задача повышения автономности робототехнических устройств и их групп. Для этого требуется соответствующая адаптации имеющихся оптимальных алгоритмов к условиям реальной среды. Адаптация позволяет выходить за рамки симуляционных экспериментов, но требует дополнительных знаний, используемых агентом. Для синтеза целенаправленной деятельности в группе агент должен иметь представление о сенсорной составляющей описания рассматриваемого объекта, знания о его использовании группой агентов, иметь сценарий будущих действий и возможность уточнения этого сценария для рассматриваемой ситуации. Для представления знаний агента в настоящей работе используется оригинальный знаковый подход, в рамках которого все вышеописанные виды знаний инкапсулированы в единую психологически и биологически правдоподобную структуру. Психологическое и биологическое правдоподобие рассматриваемой модели позволяет адаптировать механизмы восприятия и анализа окружающей среды человеком для робототехнической платформы и построить систему реакций на внешние факторы. Анализ и оптимизация этой системы является комплексной задачей и рассматривается в настоящей диссертации.

Для формирования психологически правдоподобного представления знаний агента об условиях взаимодействия в реальной среде было разработано знаковое представление пространственной логики Д.А. Поспелова. Пространственные логики позволяют конструировать представление знаний, с помощью которого удается учитывать местоположение агента и объектов, рассчитывать траекторию до них по имеющейся карте местности, планировать и оптимизировать деятельность, исходя из знаний об условиях, в которых действует агент. Также, знаковое представление псевдофизической логики позволяет формировать систематизированное описание имеющихся знаний агента на языке, схожем с естественным.

Решаемые современными когнитивными архитектурами комплексные задачи требуют наличия групп или коалиции агентов, деятельность которых взаимосвязана. В работе были использованы современные механизмы создания рефлексивного представления знаний агента в коллективе, основывающиеся на исследованиях А. Н. Леонтьева, А. Фодора, Д.Н. Узгадзе, А. Г. Асмолова, Э. А. Вачнадзе, В. О. Корепанова, Д. А. Новикова. Для описания деятельности агента в условиях реальной среды, его представление было дополнено биологически правдоподобной моделью внимания, которая основана на исследованиях об анализе информации на основе саккадических движений глаз человека. Использование модели внимания является актуальной задачей для повышения автономности когнитивной архитектуры и частично реализовано в когнитивных архитектурах Lida, Clarion, Carina, но имеющиеся реализации не предоставляли возможность описывать пространственные отношения между агентом и окружающими его сущностями.

Рефлексивное восприятие состояния когнитивного агента и состояния агентов в коалиции позволило создать протокол коммуникаций, согласующиеся с современными исследованиями по наличию личностной информации агента, которая не может быть передана другим участникам группы. Система взаимодействий агентов основана на использовании прецедентных абстрактных знаний относительно возможностей других агентов коалиции, получаемых на основе анализа успешности выполнения плана агентами группы. Рассматриваемая система позволила создать оригинальный способ динамического распределения ролей агентов и построить современный гибридный подход выбора субоптимального многоагентного плана поведения.

Предмет исследования – методы планирования поведения робототехнического агента с учетом неопределенности и динамики внешней среды и наличия других участников группового поведения.

Целью исследования является разработка методов, алгоритмов и экспериментальных средств иерархического психологически правдоподобного планирования поведения в коллективе агентов с накоплением базы прецедентов выполнения планов во внешней среде.

Поставленные задачи:

1. Исследовать психологически и биологически правдоподобные способы иерархического представления пространственных знаний интеллектуальных агентов.
2. Разработать иерархический алгоритм планирования поведения агента с использованием представления знаний, поддерживающего работу в группе агентов (знаковой картины мира).
3. Разработать алгоритм планирования поведения когнитивного агента, учитывающий пространственную составляющую знаний.
4. Исследовать протоколы коммуникаций агентов в многоагентных системах и разработать психологически правдоподобный многоагентный алгоритм планирования поведения.
5. Построить робототехническую реализацию предложенных алгоритмов и моделей с целью реализации возможностей осуществления целенаправленной деятельности робототехнического агента.

Научная новизна и результаты, выносимые на защиту

1. Впервые разработана и адаптирована для робототехнических устройств психологически правдоподобная модель рефлексивного представления знаний агента.
2. Интегрирована псевдофизическая логика представления окружающего пространства в алгоритм планирования поведения агента.
3. Разработаны основные принципы иерархического представления фокуса внимания агента.
4. Разработана модель динамического назначений ролей на основе рефлексивного представления знаний в коалиции когнитивных агентов.
5. Предложена экспериментальная программная реализация системы управления робототехнической платформой на базе знакового способа представления знаний.

Практическая значимость

Применение описанных в диссертации комплексных подходов позволяет создавать иерархические системы управления робототехническими платформами, повышает их автономность и расширяет список возможных решаемых задач. Экспериментальная программная среда по работе с когнитивными агентами со знаковой картиной мира

предоставляет возможность создания механизмов проверки биологически и психологически правдоподобных гипотез группового поведения.

Методы исследования

Теоретические результаты работы получены с использованием методов системного анализа, теории графов, теории алгоритмов, методов планирования поведения.

Достоверность результатов подтверждена результатами вычислительных экспериментов.

Апробация результатов исследования

Основные результаты докладывались на: Всероссийской Научной Конференции Молодых Учёных с Международным Участием (Тверь, 2016 г); Четвартом Всероссийском научно-практическом семинаре «Беспилотные транспортные средства с элементами искусственного интеллекта» (Казань, 2017 г.); IV Всероссийской научно-технической конференции «Интеллектуальные системы, управление и мехатроника» (Севастополь, 2018 г); Национальной конференции по искусственному интеллекту (Вороново, 2018 г. ; Ульяновск, 2019 г.); XII мультиконференции по проблемам управления (Геленджик, 2019 г.); 2-5 Международных Конференциях по Интерактивной Коллаборативной Робототехнике (Хэт菲尔д, 2017; Лейпциг, 2018; Стамбул, 2019; Санкт-Петербург, 2020 гг.).

Публикации

Основные результаты по теме диссертации изложены в 12 печатных работах [1-12], 2 из которых изданы в рецензируемых журналах из списка ВАК [1-2], 9 – в материалах всероссийских и международных конференций [3-11].

Объем и структура работы

Диссертация состоит из введения, четырёх глав, заключения и приложения. Полный объём диссертации составляет 117 страниц с 20 рисунками. Список литературы содержит 111 наименований.

Структура диссертации

В первой главе рассмотрены психологические и нейрофизиологические предпосылки синтеза плана поведения, разобраны особенности использования поведенческого планировщика в современных когнитивных архитектурах и приведены основные особенности синтеза

коалиционного плана поведения. Представлено знаковая реализация псевдофизической логики Д.А. Поспелова, разобран психологически правдоподобный способ формирования рефлексивного представления когнитивного агента.

Во второй главе представлены основные особенности алгоритма планирования поведения, в число которых входит синтез иерархического, прецедентного, пространственного и коалиционного планов. Приводятся разработанные процедуры пространственного уточнения и абстрагирования. Разобран алгоритм аукциона абстрактных планов поведения.

В третьей главе описаны основные особенности создания библиотек семейства тар. Приведен обзор основных модулей библиотек и диаграммы классов. Представлены библиотеки, реализующие расширение функционала алгоритма синтеза плана, посредством реализации пространственного и многоагентного планирования. Приведена реализация алгоритмов обучения с подкреплением, позволяющих синтезировать элементарные действия для проверки поведения интеллектуального агента.

В четвертой главе рассмотрена модульная архитектура робототехнической платформы, воплощающей деятельность когнитивного агента в условиях реальной среды. Приведено взаимодействие модуля планирования поведения и модулей управления актуаторами агента, а также пополнение знаний с помощью модулей обработки сенсорных данных робототехнической платформы. Представлено описание протокола робототехнической коммуникации в условиях решения комплексных задач коалицией агентов. Приведены результаты робототехнических экспериментов.

В приложение включена сравнительная таблица современных когнитивных архитектур (приложение А).

В заключении приводятся основные результаты, полученные в работе.

Глава 1

1 Обзор предпосылок создания архитектуры агента со знаковой картиной мира

1.1 Психологические и нейрофизиологические предпосылки синтеза плана поведения

1.1.1 Знаковое представление деятельности агента. Процессы распространения активности.

Семиотическое описание как внутреннего представления внешней среды, так собственных действий агента основано на психологически правдоподобном знаковом представлении знаний [13-14]. Описание представления знаний у человека приведено в работах А. Н. Леонтьева [15], исходя из которых, знания об явлении являются производной четырех основных типов данных - образом явления, значением явления, смыслом явления для субъекта деятельности и именем явления. Каждый тип данных соответствует одноименной компоненте знака, и вышеописанную последовательность можно представить в виде кортежа компонент $\langle p, m, a, n \rangle$ соответственно.

Образ явления представлен результатом работы субъекта деятельности по распознаванию явления и систематизированному описанию его составляющих. Примером образной компоненты знака в робототехнике служит описание внешнего вида окружающих объектов, распознанных роботом посредством работы алгоритмов SLAM (одновременная локализация и построение карты), а также объединение описаний этих объектов в представление текущего фокуса

внимания. Значение явления представляется общепринятым в коллективе агентов в рамках культурно-исторического подхода описанием явления. Общепринятое описание предоставляет возможность коллективу агентов одинаково воспринимать информацию о возможном использовании явления. Примером такого использования значений в робототехнике служит передача задачи или сценария деятельности другому агенту и единообразное восприятие их. Смыслы явления синтезируются в процессе взаимодействия субъекта деятельности с явлением и являются конкретизацией значений явления. Примером синтеза смысла явления в робототехнике является конкретизация сценария деятельности агента относительно объекта в его фокусе внимания. Этот вид конкретизации предусматривает уточнение общей информации касательно класса таких объектов информацией по конкретному объекту и вырабатыванием сценария деятельности. Компонента имени выполняет функцию связывания компонент знака в единую структуру и позволяет составлять сообщения в рамках протокола коммуникации агентов.

Компоненты знака представлены специальной структурой – каузальной матрицей [14], которая является структурированным набором ссылок на другие знаки, элементарные признаки и реактивные действия. Все элементы каузальной матрицы связаны со структурой представленного знаком явления. В качестве примера каузальной матрицы рассмотрим матрицу на сети образов робота, описывающую автомобиль (рис. 1). В столбце 1 приведена ссылка на результат процесса распознавания роботом переднего бампера автомобиля, в столбце 2 ссылка на результат процесса распознавания передних колес, далее задних колес в столбце 3 и зеркала в столбце 4.



$$Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{array}{l} \text{Передний бампер} \\ \text{Переднее колесо} \\ \text{Заднее колесо} \\ \text{Зеркало} \end{array}$$

Рисунок 1. Пример каузальной матрицы.

В простейшем случае, каузальную матрицу можно описать кортежем $z = \langle e_1, e_2, \dots, e_t \rangle$ длины t событий e_i . Каждое из событий e_i на сети образов описывает результат распознавания элемента в момент времени t_i или ссылку на его элементарный признак, на сетях значений и смыслов ссылку на другой знак или реактивное действие. Например, действие по поднятию блока на сети смыслов абстрактного стационарного робота, имеющего манипулятор с 4 степенями

свободы, будет иметь на сети смыслов ссылку на список под действий по изменению состояния всех 4 сервоприводов и функцию их реализации. Если рассматривать робота с другим манипулятором, то вышеописанное действие будет иметь иную реализацию на сети личностных смыслов, но оба робота будут иметь одинаковую структуру матрицы на сети значений, что позволит им действовать кооперативно. В рамках реализации описания объектов внешней среды и состояния агента с использованием каузальной структуры компонент знака были использованы матрицы двух видов – объектные и процедурные. Объектная матрица отличается от процедурной отсутствием эффектов активации матрицы и служит для описания статического состояния системы, а процедурные матрицы используются для описания существующих в представлении агента процессов.

Описанная выше структура компонент знака позволяет связывать компоненты и формировать сематические сети W_p, W_m, W_a . Элементы сетей имеют соответствующие типы отношений, выраженные на сети p отношением часть-целое, на сети m – объект-роль, а на сети a – коалиция-участник коалиции. Между узлами каждой сети установлено взаимно-однозначное соответствие, которое описывается функциями связывания по сетям в рамках одного знака: $\psi_p^m, \psi_m^a, \psi_a^p$ и обратными им функциями. Функции связывания ставят в соответствие каузальной матрице каждой сети соответствующую матрицу из другой сети, либо генерируют её по требованию при отсутствии.

На каждой семантической сети $W_x \in \{W_p, W_m, W_a\}$ существует соответствующая итерационная функция распространения активности вверх $\varphi_x \uparrow$ и вниз $\varphi_x \downarrow$, которые позволяет активировать требуемые множества каузальных матриц на заданную глубину. Глубина распространения активности может быть задана пользователем или получена циклом рассуждений в рамках конкретной задачи. Примером реализации процессов распространения активности в алгоритме планирования служит процесс активации матриц на сети смыслов, входящих в описание отношения «блок а, на блоке б». При использовании функции $\varphi_a \downarrow$ на один шаг активируются матрицы знаков «блок а» и «блок б», а при использовании функции $\varphi_a \uparrow$ на один шаг, в простейшем случае, активируются матрицы знаков всех действий, в описание которых входит это отношение.

В большинстве случаев, объекты и процессы описываются трехмерными каузальными матрицами, каждый слой которых представляет либо уникальный прецедент распознавания, либо реактивную процедуру выполнения действия. Кортеж из 5 элементов $\langle W_p, W_m, W_a, R^n, \Theta \rangle$ называют картиной мира когнитивного агента, где $R^n = \langle R^m, R^a, R^p \rangle$ – отношения на компонентах

знака, а Θ – операции на множестве знаков. Отношения на компонентах знака позволяют реализовать процессы распространения активности, описанные выше.

1.1.2 Персонифицированность знаний когнитивного агента.

Персонифицированность знаний когнитивных агентов широко используется в когнитивных архитектурах и позволяет осуществлять функции рефлексии и синтезировать план поведения, основываясь на личных знаниях агента об окружающей среде и возможностях других агентов. Собственное самосознание, оценка своих возможностей и возможностей других агентов формируют личность когнитивного агента. Как постулировал А.Н. Леонтьев – личность формируют онтогенетически сложившиеся свойства и, чем более выраженными становятся различия врожденных и прижизненно приобретаемых свойств, тем более индивидуальной становится личность. Из этого можно сделать вывод, что на формирование личностного смысла по большей степени влияют синтезированные в процессе деятельности знания агента и его умения. А.Н. Леонтьев утверждал, что персонализм постулирует существование некоторого начала, образующего ядро личности, вокруг которого формируется жизненный опыт агента и его отношения к окружающей действительности.

Аллан Фодор – в монографии «Язык мысли» [16] опирается на идеи функционального материализма и постулирует, что ментальные представления актуализируются и изменяются индивидом в зависимости от приобретаемого им опыта, но основы познания остаются врожденными. В описании когнитивных процессов и ментального представления большая роль отдана оценке собственного мировоззрения агента.

Также, персонифицированность знаний была рассмотрена в работе Д. Н. Узнадзе [17] в контексте создания установок, описанных посредством субъективного восприятия потребности и возможности её удовлетворения субъектом. Сами установки были описаны, как бессознательные явления, но они порождали активность агента и могут быть отнесены к синтезу агентом личностного смысла. А. Г. Асмолов [18] постулирует, что при изучении сознательной деятельности можно выделить 4 уровня установочной регуляции: уровни смысловой, целевой и операционной установок и уровень психофизических механизмов. Эти уровни также характеризуются формами выражения личностных смыслов агента касаемо его деятельности. В исследованиях Э. А. Вачнадзе [19] описана образная компонента невербальных смысловых установок агента и обращено внимание на связь образной сферы и внутреннего мира личности.

Вышеописанные исследования способствовали созданию знакового представления субъекта деятельности. Каждый агент с ЗКМ воспринимает свои действия в качестве доступных ему из опыта механизмов влияния на окружающую среду. Считается, что базовые (врожденные,

полученные в процессе культурно-исторического подхода) знания об абстрактном представлении доступных коалиции видов деятельности имеются у каждого агента и описаны на семантической сети значений. Эти знания не являются атрибутом самой личности и не имеют неразрывной связи с ней. Понятие личности с ЗКМ представлено знаком «Я», а все остальные агенты описаны абстракцией «Они» (см. рис. 2).

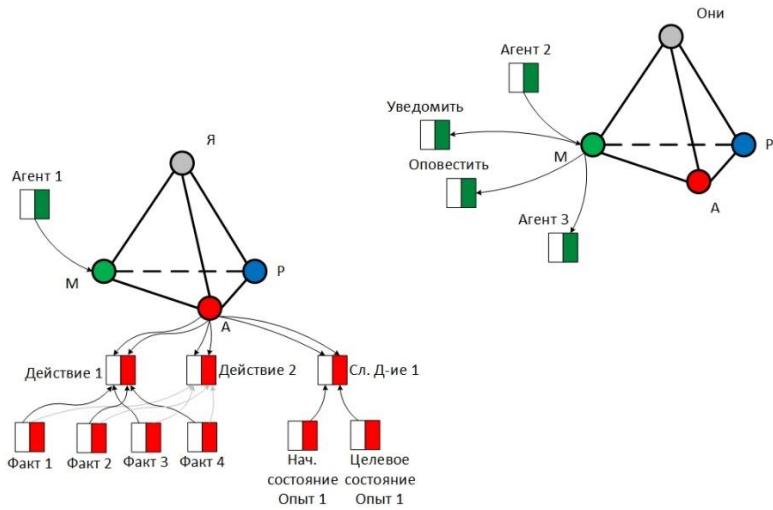


Рисунок 2. Знаковое представление агента и знаний об имеющейся коалиции.

Личностные, опытные знания агента о своих возможностях представлены на сети смыслов и характеризуют как элементарные действия (поднять ложку, пнуть мяч), так и комплексные сценарии действий, которые ранее способствовали достижению желаемого состояния окружающей среды (заварить кофе, доехать из комнаты А в комнату Б). Уточненные, комплексные сценарии могут быть заново «развернуты» в последовательность элементарных действий при необходимости, а также позволяют совершать выбор агентов для кооперирования (см. параграф 1.3). На сети значений присутствует описание родовой или видовой принадлежности, типа агента, который получен в процессе культурно-исторического подхода.

Знания о коалиции описаны знаком «Они». На сети значений знака представлены ссылки на знаки других агентов коалиции, каждый из этих знаков опосредует знания об умениях агентов, доступные агенту «Я». Также, знак «Они» описывает доступные агенту способы связи с другими агентами.

1.1.3 Модель пространственной логики Поспелова.

Нечеткое представление пространственных ориентиров когнитивным агентом связано с исследованиями Д.А. Поспелова [20] и А. Н. Леонтьева [15]. В модели деятельности А.Н.

Леонтьева нижним уровнем является ограничение деятельности агента псевдофизическими логиками, схема которых сложилась в процессе культурно-исторического развития агента. Д.А. Поспелов описал топологическую шкалу квантификаторов пространственных отношений и представил правила перехода между квантификаторами, в зависимости от размеров и местоположения объектов.

В монографии «Ситуационное управление» [20] была выдвинута гипотеза для оценки расстояний между двумя объектами, с помощью размещения между ними третьего объекта и оценки его величины. (см. таб.1). Было подмечено, что оба объекта имели одинаковый размер и находились на одной прямой.

Размер	Расстояние
Нулевой	Вплотную
Очень маленький	Очень близко
Маленький	Близко
Средний	Не далеко, не близко
Большой	Далеко
Очень большой	Очень далеко
Очень-очень большой	Очень-очень далеко

Таблица 1. Соотношение расстояния между объектами и размером объекта, помещенного между ними.

Помимо квантификаторов расстояний между объектами одинакового размера была рассмотрена гипотеза поиска квантификатора, оценивающего расстояние между объектами разных размеров с правилом приведения объектов к одинаковой величине. Был рассмотрен случай выбора квантификатора расстояния относительно объекта с известным квантификатором до объекта, удаленного от промежуточного объекта. Квантификаторы расстояний между промежуточным объектом и начальным, а также, между промежуточным и конечным были известны. Правила соотношения расстояний между объектами a_1, a_2 и a_3 одинакового размера можно описать по формуле: $(a_1 R_i a_2)(a_2 R_i a_3) \Rightarrow (a_1 R_x a_3)$, тогда нахождение R_x описано в таблице 2.

R	R+1	R-1
Вплотную	Очень близко	Вплотную
Очень близко	Близко	Вплотную

Близко	Не далеко, не близко	Очень близко
Не далеко, не близко	Далеко	Близко
Далеко	Очень далеко	Не далеко, не близко
Очень далеко	Очень-очень далеко	Далеко
Очень-очень далеко	Очень-очень далеко	Очень далеко

Таблица 2. Топологическая шкала квантификаторов расстояний до удаленного объекта.

Основываясь на вышеперечисленных отношениях, в статьях [4][10] была описана деятельность агента, основанная на предположении о существовании эгоцентрических координат, которые условно можно разделить на 2 пространства – ситуация, в которой агент находится и карта, местоположения объектов и субъектов в которой выражены нечетко (рис. 3).

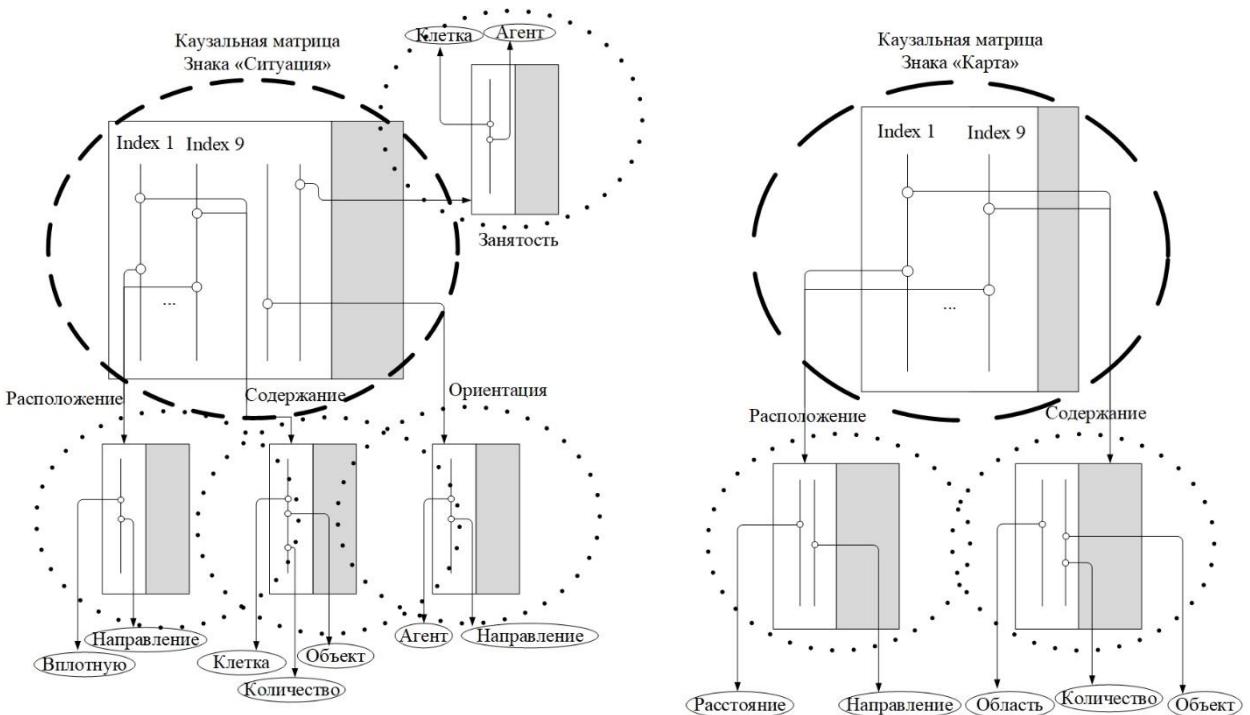


Рисунок 3. Каузальные матрицы знаков «Ситуация» и «Карта».

Знаковое пространственное представление ситуации состоит из описания фокуса внимания агента, разделенного на 9 клеток. В пользу такого разделения можно привести работы [21-23], в которых описан процесс формирования фокуса внимания человека посредством саккад (подробнее в разделе 1.2.2). Все окружающие агента клетки в каждый момент времени описаны соответствующими каузальными матрицами знаков «расположение» и «содержание». Каузальная матрица знака «расположение» описывает расположение клетки и направление до нее, каузальная матрица знака «содержание» поясняет количество объектов внутри соответствующей клетки и их тип. Направления представлены знаками отношений в двумерном

пространстве «слева», «справа», «сверху», «снизу», а также промежуточными «справа-сверху» и т. д. По топологической шкале Д.А. Поспелова, отношения между клетками выражены знаком квантификатора «Вплотную». Центральная клетка, в начальный момент времени, содержит агента и представлена каузальными матрицами знаков «ориентация» и «занятность». Перемещение агента происходит из центральной клетки в любую пустую клетку его фокуса внимания, после этого фокус вычисляется заново.

Карта местности в картине мира агента представлена 9 областями, координаты которых статичны, а содержание может меняться. Каузальные матрицы соответствующего знака карты описывают местоположение каждой из областей относительно текущего местоположения агента и информацию, доступную агенту об объектах, содержащихся в этих областях. Карта содержит более полную информацию об окружающих объектах среды и активация новых каузальных матриц с её описанием является более трудоемкой процедурой, относительно пересоздания фокуса внимания. Каузальные матрицы карты активируются в момент пересечения агентом границы области. Отношения расстояния до областей представлены по направлениям и описаны знаками квантификаторов расстояний:

- «Содержит» – центральная клетка с агентом находится в области;
- «Вплотную» - область содержит некоторые клетки фокуса внимания;
- «Близко» - область не содержит клетки фокуса внимания, но граничит с ними;
- «Далеко» - область граничит с областью, которая описана квантификатором «Близко» от агента.

Количество знакового представления квантификаторов может быть расширено при необходимости.

1.1.4 Состояние исследований по пространственному представлению знаний когнитивным агентом

Использование пространственного представления знаний агента об окружающей среде обусловлено биологически или психологически правдоподобной архитектурой агента. Реактивные агенты, как правило, ограничены представлением знаний в виде клеточных графов [24-26], которые позволяют синтезировать траекторию перемещения агента на плоскости, но не имеют привязки к интеллектуальной целенаправленной деятельности. Планирование траектории представляет поиск пути на графе, в котором вершины являются положениями агентов в пространстве, ребра элементарными траекториями, а веса ребер длинами соответствующих элементарных траекторий. Агенты, пользующиеся алгоритмами такого вида, могут

синтезировать траекторию с приоритетами и задержками, синтезировать планы как без учета перемещений других агентов, так и учитывая их перемещения, выбирать из всего множества планов тот, который подходит для всех агентов, устранивая конфликты между ними. Одним из способов интеграции планирования траектории в деятельность когнитивного агента является использование вышеописанных методов, в качестве эвристики выбора направления перемещения [11] (см рис. 4). Такой подход позволяет агенту синтезировать план действий, оперируя как действиями по перемещению объектов, осуществлением коммуникации с другими агентами, так и оптимально выбирать траекторию собственных перемещений.

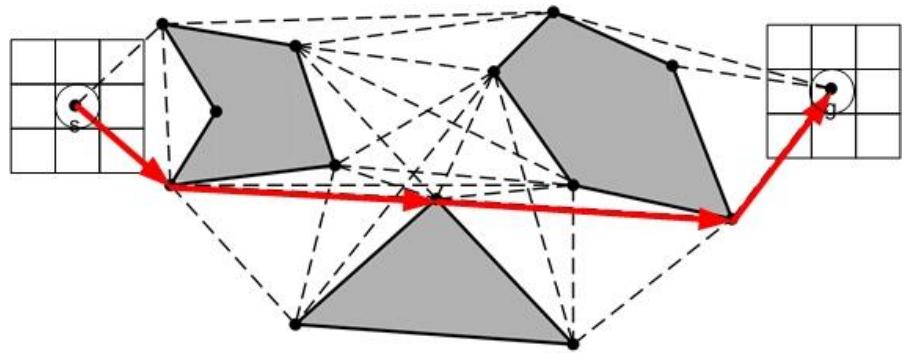


Рисунок 4. Эвристическое использование алгоритма поиска траектории перемещения при планировании поведения агента.

Биологически правдоподобные архитектуры синтезируют план действий и перемещений агента, основываясь на имеющейся информации о строении мозга млекопитающих. В Швейцарии с 2013 года функционирует проект The Human Brain Project [28], в рамках которого, ученые из 135 институтов систематизируют исследования развития и функционирования человеческого мозга. Эти данные могут помочь построить правдоподобную модель мозга, механизмы которого, обеспечивают быстрое вычисление траектории перемещения агента и позволяют планировать деятельность. Планирование перемещений агента тесно связано с возможностью распознавания окружающей среды и своевременной обработки полученных данных. В работе [29] описана модель, полученная на основе исследований активности мозга крыс [29-31]. Модель описывает иерархическую среду, представленную картами различных масштабов. Планирование перемещения агента происходит с учетом всех возможных траекторий достижения цели, что требует большого количества ресурсов для расчета возможных исходов деятельности при учёте динамики изменения среды. Эта проблема частично была устранена в [32-33], что привело к созданию системы RatSLAM, которая позволила агенту перемещаться на большие расстояния в реальной местности, а также системам локализации и картирования [34-35]. Основными недостатками работы являются перегруженность знаниями об удаленных объектах, отсутствие протоколов коммуникаций, позволяющих передавать знания другим

агентам и отсутствие унифицированного подхода аппроксимации знаний. Эти недостатки не позволяют сравнивать подход реализации агента [33] с подходами, представленными когнитивными архитектурами.

В рамках психологически правдоподобного подхода к вопросу планирования поведения агента рассматриваются задачи, связанные с неполнотой и неточностью описания окружающей среды. Для решения поставленных задач используется широкий спектр способов представления знаний агентом, многие из которых позволяют аппроксимировать знания об окружающей среде до необходимого для планирования поведения уровня. В работах [36-37] описан подход, в рамках которого представление пространственных знаний агентом осуществляется с помощью набора наиболее вероятных действий в текущем положении. Основываясь на [37], была создана когнитивная архитектура SemaFORR. В архитектуре приведены 3 вида геометрических абстракций – трассы, области и конвейеры. Выбор маршрута перемещения основан на деятельности 3 уровней эвристических «советчиков»: уровень реактивной деятельности, уровень планирования перемещений и уровень разрешения конфликтов (см. подробнее подпараграф 1.2.1).

Вероятностные подходы представления геометрических характеристик среды представлены в реализациях архитектуры LIDA [38] и в работе [39]. Основные достижения, описанные в обоих работах, основываются на байесовском представлении геометрических знаний о среде роботом и вероятностной оценке корректности данных, полученных с сенсоров. Алгоритмы позволяют синтезировать суждения робота об успешности взаимодействия с объектами и точности перемещения по спланированной траектории. Сравнение когнитивных архитектур представлено в параграфе 1.2.

Последнее время были предприняты попытки выстроить взаимодействие между алгоритмами планирования поведения и алгоритмами обучения с подкреплением [40, 4]. Характерной особенностью объединения является возможность синтезировать символный план поведения агента и расширить каждое из абстрактных действий списком поддействий, который реализует последовательность моторных операций. В данном контексте проявляется отсутствие явного представления в картине мира агента пространственных знаний, которые описываются последовательностью простейших шагов, синтезированных алгоритмами обучения с подкреплением. Такой механизм характерен для создания множества списков поддействий, реализующих поведение манипулятора робота, но и возможен в рамках реализации перемещения колесной базы.

1.1.5 Выводы параграфа 1.1

На основании адаптации психологически правдоподобного знакового представления окружающей среды когнитивного агента для робототехнического устройства были сформулированы следующие основные принципы синтеза плана поведения агента:

1. Агент должен иметь пространственное представление окружающей среды. Псевдофизическая логика представления позволяет разделять окружающее пространство на 2 подпространства – динамически формируемый фокус внимания агента и карту статичной (в простейшем случае) величины. Отношения местоположения объектов и субъектов деятельности агента, относительно его собственного местоположения позволяют сформировать психологически правдоподобное суждение «Требуемый предмет находится далеко от меня. Мне необходимо к нему приблизиться».

2. Знания агента персонифицированы и описывают субъективную картину мира агента. Опыт агента является его личностной характеристикой и позволяет синтезировать план поведения в окружающей среде. Знаковое представление других агентов коалиции позволяет реализовать рефлексивные рассуждения 1 и 2 рода.

1.2 Особенности использования поведенческого планировщика в когнитивных архитектурах

План поведения агента с когнитивной архитектурой является продуктом цикла принятия решений. В большинстве известных когнитивных архитектур цикл принятия решений занимает ключевую позицию и характеризует цель создания архитектуры. Комплексность цикла принятия решений позволяет синтезировать план поведения в рамках неоднозначности представления окружающей среды, производить рассуждения и оперировать опытом агента. Наиболее развитые архитектуры способны использовать цикл принятия решений, совмещенный с протоколами коммуникаций агентов и синтезировать многоагентные планы поведения. В новейших архитектурах, адаптированных под робототехнические устройства, возникла потребность в создании дополнительных уровней восприятия среды и взаимодействия с ней [2]. Уровни реактивного и тактического взаимодействия со средой позволяют агенту дополнять синтезированный план поведения и осуществлять перепланирование при неуспешном исполнении плана, а также пополнять картину мира агента данными о среде.

1.2.1 Состояние исследований на данный момент

В таблице 3 (см. приложение 1) рассмотрены 10 актуальных на момент создания диссертации когнитивных архитектур [41-50]. Для каждой из архитектур были выделены 11 признаков, а также приведены дополнительные данные о разработчике архитектуры, ссылка на полное описание архитектуры, перечислены найденные автором эксперименты на основе архитектуры и приведена ссылка на реализацию.

Описания структур когнитивных архитектур можно условно разделить на две части – иерархические и модульные. Иерархические архитектуры [41-44, 48] как правило, выстроены относительно фаз понимания и реакции на данные, полученные с датчиков. Иерархия позволяет воспринимать данные, выделять необходимую циклу принятия решений информацию и пополнять собственные знания, с помощью рассуждений, относительно полученной информации. Архитектуры с иерархической структурой совместимы с робототехническими устройствами и позволяют находить решение робототехнических задач, но их адаптация к задачам других типов предполагает разделение иерархии на составляющие. Модульные архитектуры [45,46,50] слабо адаптированы для решения комплексных проблем робототехники. Их распределённая структура позволяет создать аналог иерархии и решать задачи такого типа, но в то же время, целью создания этих архитектур являются симуляция когнитивных не роботизированных агентов, создание интеллектуальных ассистентов.

Как описано в [51] большинство способов представления знаний можно свести к:

1. Продукционным моделям [43-45];
2. Семантическим сетям [41, 42, 48, 50];
3. Фреймам [47];
4. Формальным логическим моделям [46,49].

Продукционные модели представляют знания в виде правил «Если условие, то действие».

В общем виде продукция описывается формулой: $N = \langle A, U, C, I, R \rangle$, где N – имя продукции, в простейшем виде обозначающее порядковый номер продукции, A – область применения продукции, требуемое для ограничения поиска необходимой продукции по базе знаний. U – условие применимости продукции, которое требуется для активации следствия продукции. C – ядро продукции. I – следствие (постусловия) продукции и R – дополнительная информация по продукции. Экспертные системы [52], основанные на продукционной модели представления знаний включают базу знаний (продукций), рабочую память и решатель. Правила в базе знаний состоят из части условий (антecedента) и части эффектов (консеквента). Рабочая память состоит из наборов пар атрибут-значение, истинность которых устанавливается с помощью применения правил из базы знаний. В процессе роста базы знаний появляются частично противоречащие друг другу правила, что приводит к ухудшению скорости вычисления требуемого результата. Существует два вида логического вывода в системах с продукционными моделями – прямой и обратный. Прямой вывод заключается в циклическом поиске решения от исходных фактов к заключениям. Первым шагом прямого вывода является активация всех продукции, чьи условия истины в настоящий момент времени, после этого в рабочую память вносятся полученные заключения, которые дополняют имеющиеся условия на следующем шаге итерации. Цикл завершается при активации целевого условия. При обратном выводе происходит цикл поиска фактов, подтверждающих целевую гипотезу, после этого, эти факты назначаются промежуточными гипотезами и цикл рекурсивно повторяется до момента активации уже известных фактов.

Семантические сети представляют знания в виде графа, вершинами которого являются сущности, а ребра (дуги) отношения между ними. Сущности описывают объекты и субъекты деятельности, а также их абстрактные представления. Отношения между сущностями зависят от области знаний, описываемой сематической сетью, базовыми отношениями являются:

- Структурные («часть-целое»);
- Таксономические («класс-подкласс»);
- Временные («раньше-позже»);

- Логические («и, не, или»);
- Каузальные (причина - следствие).

Также описываются функциональные качества сущности, которые могут быть представлены в зависимости от реализации сети как в виде вершин, так и в виде дуг. Для поиска ответа на запрос в базе знаний, представленной семантической сетью, используют механизмы поиска и активации подграфа, содержащего сущности запроса. При росте базы знаний, также, как и в продукционных моделях происходит снижение скорости обработки запросов. Одним из способов решения этой проблемы является разделение общей сети на связанные подсети, каждая из которых будет описывать отдельный вид отношений. Семантические сети наиболее психологически правдоподобно формируют представление когнитивного агента о сущности и возможностях взаимодействия с ней.

Фреймовые модели служат для описания данных в формальном виде. Под фреймом представляют структуру $F = \langle N, D, C, P \rangle$, где N – имя фрейма, D – множество слотов с фактами, описывающими фрейм, C – множество слотов, ответственных за связь с другими фреймами, P – множество слотов, описывающих действия, связанные с фреймом. Слоты могут быть как терминальными значениями, так и фреймами другого уровня иерархии. Принято разделять множество фреймов на фреймы-образцы и фреймы-экземпляры. Первые служат для описания сущностей в базе знаний, вторые формируются в процессе распознавания поступающих данных. Также, в некоторых моделях, используются фреймы-структуры, фреймы-роли, фреймы-сценарии и фреймы-ситуации.

Формальные модели представления знаний требуют наличие словаря используемых понятий и синтаксических правил, связывающих эти понятия. Понятия могут быть истинными и ложными. Всегда истинные понятия называются аксиомами, основываясь на которых осуществляется механизм логического вывода, позволяющий синтезировать новый набор истинных понятий. Систему, основанную на формальном способе представления знаний, называют формальной системой. В общем виде, формальная система представляется в виде: $F = \langle A, V, W, R \rangle$, где A – словарь понятий, V – множество синтаксических правил, W – множество аксиом, а R – множество правил логического вывода. Примером формального способа представления знаний является предикативный способ. Основным недостатком формальных моделей является сложность пополнения знаний и расширение спектра применения модели.

Для большинства рассмотренных архитектур способ представления знаний может быть классифицирован неоднозначно, но приведенные выше данные подтверждают, что семантические (или семиотические в расширенном виде) сети являются наиболее актуальным на

данный момент способом представления знаний. Этот вывод подтверждается наличием развитых средств агентной коммуникации в рассматриваемых архитектурах и возможностью семиотического представления естественного языка [53]. Но на момент создания настоящей работы взаимодополняющие структуры семантического представления знаний и нейросетевых алгоритмов не позволяют генерировать сложный, связный и логичный текст, а склонны синтезировать бинарные ответы на поставленные перед агентом вопросы.

Процесс планирования и принятия решений в каждой из архитектур представлен иерархически. Иерархии позволяют распараллелить многоуровневые процессы обработки данных, а в архитектурах [41, 42, 49, 50] в когнитивный цикл дополнительно внесена возможность отслеживания изменений окружающей среды на действия агента. Перепланирование, как способ реакции на изменения среды, характерно для робототехнических архитектур настоящего времени и предоставляет возможность динамически изменять план агента. Динамическая вариативность деятельности характерна и для обучающихся агентов. Так как большинство реализаций процесса планирования поведения не являются алгоритмами, синтезирующими план в реальном времени, то реакция когнитивного агента на изменение среды может быть существенно замедленна. Но агент, деятельность которого основана на обработке опытных знаний, способен в более сжатые сроки адаптироваться к изменениям окружающей среды, применив прецедент деятельности в схожей ситуации и синтезировав реакционное поведение.

Обучение когнитивных агентов реализуется посредством решения двух условных задач – задачи накопления декларативных и процедурных знаний высокого уровня абстракции и задачи уточнения абстрактных действий до атомарного уровня с помощью алгоритмов обучения с подкреплением. В первом случае, агент сохраняет прецедент успешного выполнения плана и способен адаптировать имеющийся прецедент в будущих взаимодействиях со средой. Также, функционал некоторых когнитивных агентов позволяет им рассуждать относительно условий выполнимости и невыполнимости плана, выявлять надежность агентов-кооператоров, абстрагировать имеющийся план до уровня обобщенного сценария, который может послужить примером деятельности для других агентов. Уточнение абстрактных действий с помощью обучения с подкреплением реализовано в архитектурах [41, 43, 44] и является биологически правдоподобным способом. Этот вид обучения позволяет адаптировать спланированные действия к условиям окружающей среды аналогично тому, как это делает человек. В качестве примеров решаемых задач можно привести развитие архитектуры STRL [6] и подход [54]. В первой статье агент синтезирует план поведения по перемещению в робототехническом полигоне и взаимодействию с блоками в домене «Мир Блоков», далее алгоритмы обучения и

глубокого обучения с подкреплением синтезируют последовательность элементарных движений робототехнической платформы и манипулятора для их исполнения. Синтезированная последовательность под действий позволяет синхронизировать состояние робототехнической платформы с последовательностью спланированных состояний и произвести корректировку в процессе деятельности, а не после её завершения. Переиспользование последовательностей атомарных под действий осуществляется посредством их объединения в опции [55]. В подходе [56] авторы рассмотрели более тесное взаимодействие алгоритма символического планирования и алгоритма обучения с подкреплением, но эксперименты были ограничены синтетическими средами и не предусматривали робототехническую постановку. При наличии множества опций, реализующих действия агента, возникает возможность выбора и адаптации реализации действия без дополнительных итераций планирования. В качестве примера можно привести абстрактное действие – «взять объект X», который может находиться как на столе, так и на полке шкафа, на полу или внутри сумки и требовать значительное время для синтеза конкретных реализаций работы сервоприводов манипулятора агента.

Возможность когнитивного агента ориентироваться на местности и пополнять собственную картину мира обусловлена его деятельностью по распознаванию окружающей среды, основополагающую роль в которой занимают алгоритмы локализации и построения карты местности – SLAM. Алгоритмы SLAM позволяют учитывать взаимодействие проприоцептивных (колесные энкодеры, IMU,...) и экстероцептивных (камеры, лидары,...) датчиков в рамках отслеживания одометрии робототехнической платформы, на основе которой происходит локализация. Основными требованиями для системы SLAM являются:

- Надежные одометрия и локализация, обуславливающие замыкание петли распознавания карты и предотвращающие ложные интерпретации положения агента при работе в динамической среде;
- Скорость распознавания и построения карты. В динамической системе у агента может не быть достаточного количества времени для повторного распознавания карты;
- Способность сопоставления и «склеивания» карт.

Актуальные в настоящее время подходы SLAM позволяют робототехническим платформам выстраивать внутреннее представление окружающей среды, основываясь на малом числе датчиков, например, на единственной камере, которая измеряет расстояние до объектов с помощью построения карты глубин [56]. Использование небольшого количества датчиков негативно сказывается на скорости и точности построения карты и особенно ярко проявляется в случае малых мобильных платформ и летательных аппаратов. Подходы SLAM внутри

небольших помещений без окон и открытых дверей показывают удовлетворительные результаты и позволяют точно находить местоположение агента. В рамках решения проблемы локализации при наличии неидеальных условий для SLAM (в помещении присутствуют окна, двери, ходы) актуальным подходом является совмещение SLAM с алгоритмами обучения с подкреплением [57, 58]. Одной из наиболее известных сред для обучения агентов распознаванию карты является проект Habitat [59]. Подход SLAM с единственной камерой и использование алгоритмов обучения с подкреплением для улучшения качества локализации и распознавания дополнили когнитивную составляющую архитектуры STRL.

Как было описано ранее, из-за неточности распознавания окружающей среды, большинство когнитивных архитектур имеют вероятностное представление выполнения действий, в архитектуре [42] присутствует байесовский способ прогнозирования местоположения агента после выполнения действия, а архитектура [47] имеет возможность оценивать вероятность выполнения действия посредством оценки выполнимости действия несколькими эвристическими советчиками. Неопределенность данных обуславливает наличие фаз понимания в архитектурах и использование набора допущений, уточнение которых выполняется иерархией уровней взаимодействия со средой. Интеграция алгоритмов локализации и картирования когнитивных архитектур наиболее часто происходит посредством реализации SLAM в ROS (Robot Operating System) [60]. Наиболее известными реализациями являются ORBSLAM2 [61] и RTAB-MAP [62].

ROS является распространенной робототехнической операционной системой, позволяющий создавать отдельные модули (узлы), объединять их в пакеты и стеки, реализующие комплексный уровень иерархии когнитивной архитектуры. Все узлы ROS имеют различные способы связи, в число которых входят сервисы, топики и тд. Поддерживается визуализация робототехнических платформ в Gazebo, V-Rep и других средах. Подробнее о ROS и о роли в структуре архитектуры STRL в главе 4. Внутреннее представление окружающей среды индивидуально для когнитивных агентов и является одним из элементов персонификации агента.

Процесс персонификации когнитивных агентов происходит посредством субъективизации декларативных и процедурных знаний и присутствует в явном виде в архитектурах [41- 44, 50]. Наиболее ярко он выражен в архитектурах STRL и CARINA, в которых представлена само-модель агента, которая реализует функции авторефлексии, а также информационной и стратегической рефлексии. Персонификация знаний характерна для систем с выраженным мета-уровнем процесса принятия решений, наличие которого обуславливает комплексность процесса осознавания окружающего пространства и поиска места агента в нем. Процесс целенаправленной предметной деятельности тесно связан с процессом моделирования фокуса внимания агента.

Процесс планирования поведения когнитивных архитектур позволяет синтезировать взаимодействия с объектами и субъектами, входящими в фокус внимания агента, поэтому моделирование фокуса внимания является частью процесса принятия решений.

Процесс моделирования внимания позволяет осуществить комбинаторную оптимизацию алгоритмов планирования и обучения, посредством выявления наиболее важных объектов среды. Фокус внимания порождает процесс осознавания агентом присутствия сущности в непосредственной близости и возможность с ней взаимодействовать. Психологические и биологические предпосылки к созданию фокуса внимания рассмотрены в подпараграфе 1.2.2. В архитектуре [44], понятие фокуса внимания объединяет подсистемы мотивации, внимания и осознавания, в архитектурах [42, 43, 45, 50] фокус внимания является неотъемлемой частью цикла принятия решений и явно не описан. В работе [47] присутствуют эвристические советчики первого уровня, характеризующие поведение агента в рассматриваемой ситуации. В STRL произошло явное выделение иерархического фокуса внимания агента, позволившее характеризовать доступные и недоступные ему действия. Целенаправленная деятельность в фокусе внимания агента может включать как атомарные осознаваемые агентом действия, так и комплексные, реализация которых является последовательностью выполнения множества реактивных операций. Реактивное поведение характерно для роевых структур, реализующих взаимодействие роботов слабой вычислительной мощности. Одним из примеров интегрирования когнитивной архитектуры и роя реактивных агентов является взаимодействие когнитивной архитектуры [48] и системы SCARAB (Situative Cognitive Architecture for Reasoning About Behavior).

1.2.2 Моделирование внимания когнитивного агента

Актуальной проблемой в когнитивной робототехнике является процесс выделения сущностей из потока сенсорных данных и их обработка посредством поиска возможных взаимодействий с ними. Нейрофизиологические и биологические исследования обосновывают взаимосвязь процессов формирования фокуса внимания и саккадических движений глаз субъекта, а также анатомически и функционально правдоподобные сведения о воздействии корковых структур головного мозга в этих процессах [21-23]. Процесс моделирования внимания неотрывно связан с процессом локализации и распознавания объекта. Также, как было замечено ранее, процесс формирования фокуса внимания является неотъемлемой частью когнитивной деятельности агента по установлению связи между циклом принятия решений, в котором используется абстрактное представление сущностей окружающей среды, и премоторной подготовкой движения. Количество входящих в фокус точек внимания является обсуждаемым

вопросом, так как процесс деятельности агента непрерывен, и каждая распознанная сущность последовательно обрабатывается агентом и формирует возможную моторную реакцию. В настоящей работе процесс формирования фокуса внимания дискретизирован по времени и участвует в итеративном синтезе поведения агента.

Когнитивные агенты настоящего времени используют принципы машинного обучения для формирования наборов элементарных внешних признаков распознаваемых сущностей, соотнесение которых с имеющимися прецедентами распознавания позволяют категорировать эти объекты. Результатом процедуры распознавания объекта является его образ, характеризующий набор высокоуровневых признаков, важных для агента. Объект распознается не только основываясь на сформированном образном представлении агента об объекте, но и посредством поиска общих, концептуальных знаний культурной среды агента касающегося рассматриваемого объекта. При наличии ограниченного числа внутренних представлений рассматриваемой сущности, но при отсутствии описательной речевой информации об объекте, агент не может оперировать представлением объекта в цикле принятия решений, так как в картине мира агента не будет произведено определение роли сущности в целенаправленной деятельности. Связывание образа сущности с имеющимся описанием происходит посредством реализации упомянутой в подпараграфе 1.1.1 функции связывания Ψ_p^m , а активация доступных агенту действий с распознанной сущностью функцией Ψ_m^a связывания значения сущности и её смысла в целенаправленной предметной деятельности агента.

Фокус внимания агента позволяет найти пересечение множества всех доступных агенту образов, знак которых присутствует в его картине мира, и множества образов всех сущностей и явлений, находящихся в моторной близости от агента. Также, немаловажным элементом фокуса внимания является собственное представление местоположения агента в имеющемся окружении. Представление сущности «Я» по отношению к остальным объектам и субъектам занимает основополагающую роль в процессе формирования «эгоцентрированного» мировоззрения [63]. Как постулируют авторы в [64] на перцептивное восприятие окружающей среды значительное влияние оказывают социальные установки (аттитюды) агента, позволяющие регулировать скорость реакции на распознанные сущности. В контексте ЗКМ агента, социальные установки предназначены для персонификации реализации последовательности функций $\Psi_p^m, \Psi_m^a, \Psi_a^p$, абстрактно описывающих обработку распознанной сущности когнитивным агентом, и синтез смысла ситуации, в которой агент находится.

1.2.3 Робототехнические реализации когнитивных архитектур

Адаптация когнитивной архитектуры к робототехническому устройству производится посредством создания дополнительных модулей, регламентирующих поведение робототехнической платформы согласно принципам, синтезированным архитектурой. Наиболее часто реализуемое поведение роботов представлено действиями по перемещению платформ и манипулированию объектами окружающей среды. Моделирование последовательности действий и их ограничений производится когнитивной архитектурой, которая реализует решение абстрактных задач, связанных с видами итеративной деятельности человека. В качестве примеров деятельности можно привести оказание вспомогательных функций при одевании одежды, выполняемые платформой PR2 [65], новизна работы заключается в использовании взаимосвязи модуля принятия решений и 2 рекуррентных нейросетей для предсказания силы нажатия актуатором агента и динамической регулировки нажатия. Также, актуальна деятельность платформы Tug [66], выполняющий функции логистического распределения лекарств в больничных учреждениях.

В архитектуре LIDA была предпринята попытка разработки и реализации байесовского психологически правдоподобного представления пространственной логики на платформе PR2 [67]. Целью работы являлось создание множественных «когнитивных карт» робота, конкретизирующих навигационные процессы. В процессе реализации поставленной задачи произошло смешивание навигации условного «тактического» уровня архитектуры, и замена алгоритмов локализации на байесовские модели. Этот метод упрощает иерархию распознавания и обработки данных о среде, снижая качество локализации агента. В рамках архитектуры STRL, уровни тактического и стратегического планирования совместно повышают автономность агента, реализуя биологически правдоподобную множественную проверку местоположения робототехнической платформы.

Деятельность по адаптации архитектуры SOAR к робототехническим платформам [68] и реализация возможностей по управлению платформой с помощью голосовых команд [69] привело к созданию обучающихся робототехнических когнитивных агентов (Rosie), одним из вариантов воплощения которых является робот Fetch [70]. Примерами решаемых задач такого робота являются: построение башни из блоков, поиск и доставка предметов, и др. Характерной чертой агентов Rosie является возможность выстраивания комплексной иерархии распознавания среды и поиск в пространстве прецедентов решения задачи. Знания, синтезируемые рассуждениями о среде, способствуют агенту находить взаимосвязь между прецедентами решений и быстрее адаптировать их к новым задачам. На стратегическом уровне архитектуры STRL опыт агента также сохраняется и обрабатывается, а поиск в пространстве планов основан

на рассуждениях о подходящем прецеденте планирования. Также используется опыт по частичному решению поставленной задачи.

Архитектуры CAMAL [48] и HiPOP [49] созданы для решения коллаборативных задач по построению карты местности и взаимодействию с объектами окружающей среды. Архитектура CAMAL реализует BDI (убеждения, желания, намерения) нотацию по установлению протокола коммуникации между агентами и организует взаимодействие роботов AmigoBots, Pioneer 3-DX и ActivityBots. Как было уточнено ранее, реактивное поведение роботов со слабой вычислительной мощностью синтезируется подсистемой SCARAB. В архитектуре HiPOP используются роботы Ressac, Effibot, LAAS-CNRS. Роботы моделировались в среде MORSE [71], связь между роботами была реализована с помощью ROS. Реализация действий на роботе происходила с помощью связки ROS+Orocoss [72]+Sockets. В реальной среде было проведено 8 экспериментов, подтвердивших необходимую комплексность обработки факторов внешней среды.

1.2.4 Выводы параграфа 1.2

Когнитивные архитектуры позволяют повысить уровень автономности интеллектуальных агентов и способствуют реализации комплексной деятельности, элементами которой является многоуровневое иерархическое распознавание окружающей среды, синтез плана поведения, создание протоколов коммуникаций, моделирование внимания агента, оценка реакции окружающей среды на действия агента и механизмы динамической корректировки деятельности. В связи с этим, современная робототехническая архитектура должна базироваться на следующих свойствах:

1. Комплексной иерархией получения знаний в процессе обработки данных с датчиков. Иерархия STRL позволяет считывать данные с камер, лидаров, УЗ и других доступных датчиков, что позволяет строить карту местности и производить локализацию агента алгоритмами SLAM, адаптировать построенную карту к знаниям агента о среде и синтезировать комплексный план поведения агента.

2. Фокусом внимания. Биологически и психологически правдоподобный способ моделирования фокуса внимания агента позволяет уменьшить комбинаторную сложность задачи, предоставляя STRL-агенту преимущество по сравнению с агентами, воплощающими другие архитектуры.

3. Развитым протоколом коммуникаций. Многоагентная составляющая позволяет не только строить карты большей точности, но и выполнять комплексные задачи. Описание возможных протоколов коммуникаций произведено в параграфе 1.3.

1.3 Особенности коалиционного планирования поведения когнитивных агентов

Исследование процесса многоагентного подхода планирования поведения является актуальной задачей, область применения решений которой выходит за рамки робототехники и распространяется на логистические, инженерные и строительные процессы. Спектр многоагентных подходов можно условно разделить на две основные группы – группа подходов централизованного планирования и группа распределенного планирования.

Для централизованных подходов характерна высокая скорость вычисления общего плана поведения группы агентов, хорошая связность планов, отсутствие большинства ошибок, возникающих при пересечении планов и распределении действий агентов по временным интервалам. Централизованные способы планирования позволяют снизить уровень нагрузки на коммуникационные сети агентов, выстроить четкую межагентную иерархию, в рамках которой каждый из агентов-исполнителей будет выполнять поставленную перед ним задачу, не рассуждая относительно возможностей других агентов. Централизованный подход предоставляет возможность создавать гетерогенные коалиции агентов, в рамках которых способны осуществлять целенаправленную взаимодополняющую деятельность как когнитивные, так и реактивные агенты. Негативной составляющей централизованного подхода является полная доступность всех знаний о возможностях агентов агенту-координатору и комплексность построения плана при наличии множественных целей агентов. При потере связи с агентом-координатором возникает необходимость выбора нового агента-координатора, с учётом всех выполненных изменений в текущем плане поведения. Процесс восстановления потерянной связи с координирующим агентом возникает с большой периодичностью при робототехнических испытаниях и, в системах с четкой иерархией, требует исполнения соответствующего протокола после каждого элементарного действия робототехнического агента.

Большая автономность агентов достигается с помощью использования распределённых подходов к синтезу плана поведения. Большинство распределённых подходов можно характеризовать следующей последовательностью действий:

1. Каждый из агентов синтезирует план по достижению собственного целевого положения;
2. Запускается протокол координации планов, в рамках которого устраняются временные и ресурсные разногласия между агентами; (опционально)
3. Конечный план распределяется между агентами для исполнения; (опционально)

4. Каждый локальный конфликт решается посредством соответствующих протоколов межагентных коммуникаций.

Описанная группа подходов позволяет реализовать достижение комплексного набора целей, снижает зависимость агентов от центрального, координирующего агента (некоторые подходы позволяют не использовать агента-координатора, а все межагентные конфликты решаются локально). Распределённый способ планирования требует более автономных агентов, каждый из которых должен выполнять комплексные действия, из-за этого возникает проблема использования агентов с реактивным поведением. Комплексность задачи управления группой агентов с распределённым подходом заключается в необходимости наличия сложных протоколов локальной межагентной коммуникации и механизмов перепланирования. Механизмы перепланирования действуются при отсутствии возможности выполнить ранее синтезированный план из-за использования необходимых ресурсов другими агентами. Распределённый способ синтеза плана позволяет не разглашать возможности агентов и синтезировать субоптимальный и менее зависимый от постоянной доступности межагентной коммуникации план. Межагентная коммуникация в распределённой системе может основываться на сетях доверия [73-75], при этом агенты будут искать проверенные способы исполнения плана и выстраивать коалиции по достижению поставленных задач (подробнее в 1.3.1).

На стратегическом уровне архитектуры STRL используется гибридный подход синтеза плана поведения группой агентов, который предоставляет возможность использовать сильные стороны как централизованного планирования, так и распределённого, для синтеза комплексного субоптимального плана поведения. Распределённая составляющая подхода заключается в отсутствии знаний о подробностях пространственного и моторного выполнения действий агентами у агента-координатора, которым может стать любой из агентов, в зависимости от его знаний о настоящей задаче. Агенты самостоятельно синтезируют подпланы, характеризующие собственное локальное поведение, публично описывая абстрактное представление собственной деятельности (подробнее в 1.3.2).

1.3.1 Принципы распределения задач в коллективах агентов

Основными причинами кооперирования агентов являются: отсутствие достаточного количества ресурсов для самостоятельного достижения целевой ситуации, наличие общих целей и функциональное дополнение собственных возможностей. В рамках организации структуры кооперации используется набор правил [76, 77], аналогией которому в человеческом коллективе является локальное уточнение культурно-исторического подхода при организации деятельности.

Под правилами рассматривается коллективная ответственность за исполнение взятых группой обязательств, ограничивание участниками кооперации возможных способов достижения целей, вовлеченность участников коллектива в процесс достижения поставленных целей [78-79]. Для современных планировщиков поведения и координации исполнения плана характерно иерархическое представление действий агентов и возможность возврата на уровень абстрактного описания действия без подтвержденного факта его реализации [80]. Каждый из агентов устанавливает локальные взаимодействия по решению промежуточных задач согласно своим представлениям о вовлеченности других агентов коллектива в свою стратегию достижения цели и о взаимозаменяемости агентов. Пример возможных локальных реактивных взаимодействий для кооперирующих агентов представлен в модели К. Рейнольдса [81].

В группах со множеством агентов возникает избыточность возможных действий и наличие рефлексивной оценки [82-83] деятельности других агентов позволяет формировать локальные коалиции. Структура локальной коалиции может иметь иерархический характер, но общая структура группы быть линейной. При робототехнической постановке задачи, наличие избыточности действий позволяет использовать фактор локального местоположения агентов, выбирая наиболее физически близкого к локальной цели, и оценивать достижимость поставленной цели исходя из состояния агента. Состояние в робототехнических задачах характеризует задействование агента в неоконченном выполнении других задач, уровень зарядки аккумулятора, отсутствие сломанных деталей.

Процесс организации [84, 85] многоагентной целенаправленной деятельности [86, 87] формирует проблему распределения ролей агентов [88, 89, 1] в коллективе. В классических многоагентных системах определение роли агента в коллективе основывается на рефлексивной оценке его деятельности [90]. Как было описано ранее, одним из способов оценки возможности кооперирования является построение графов репутации [91], наиболее подробно представленных в когнитивных архитектурах, основанных на BDI [73] способе представления знаний. Помимо прецедентной оценки деятельности агента в работах [92, 93] представлены протоколы запросов и разведывательный протокол. Разведывательный протокол позволяет опрашивать участников кооперации об известных фактах и оценивать правдивость опрошенного агента. После завершения процедуры оценивания правдивости участников кооперации, агент использует протокол запросов и опрашивает мнение участников, подтвердивших свою правдивость. Возможность правдиво отвечать на поставленные вопросы формируется реализацией стратегии агента по нераспространению собственных (внутренних) знаний и рефлексивному исследованию доступных знаний других агентов [94, 95].

Для выбора общего плана действий при локальных коммуникациях агентов или при централизованном планировании поведения используется ряд методов, наиболее известными среди которых являются организация аукционов планов [96], использование контрактных сетей [97] или моделей социальных соглашений [98]. Эти методы описывают правила установления межагентной коммуникации и входят в стандартизованный перечень средств взаимодействия агентов [99].

1.3.2 Способы коммуникации когнитивных агентов

Психологически правдоподобные способы коммуникации агентов основываются на передаче сценариев желаемого поведения, которые могут быть как вырождены в предикативно-описательную структуру необходимых терминальный состояний, так и скомпонованы в последовательность операций с абстрактным описанием требуемых действий [100]. Вырожденные сценарии представляют описание начального и целевого состояния агента, дополненные требуемыми условиями. В робототехнической постановке, под условиями выполнения вырожденного сценария рассматриваются адаптация ситуаций и исполнение требуемого действия в процессе достижения целевой ситуации. Невырожденные сценарии обосновывают личностные знания агента о последовательности действий для достижения целевого состояния. Невырожденные сценарии могут являться как абстракцией опыта построения плана в настоящих условиях, так и быть получены из внешних источников. Абстракция невырожденного сценария использует объектно-ролевую схему, в рамках которой происходит замена конкретных сущностей на абстракции, используемые в языке формирования межагентных сообщений (объект – инструментив, местоположение – локатив и т.д.). При адаптации сценария для задачи происходит замена семантических ролей, используется механизм абстракции сущности в рамках имеющихся классов. Пример формирования сценария:

Опыт: «Передвигаться прямо агенту 1. Повернуть агенту 1 налево. Подъехать ближе к столу агенту 1. Поднять блок «А» агенту 1.»

Вырожденный сценарий: «Начальное состояние (стол далеко, вокруг агента ничего нет). Конечное состояние (стол близко, стол перед агентом, блок в манипуляторе). Действие «поднять».

Невырожденный сценарий: «Передвигаться прямо агенту. Повернуть агенту налево. Подъехать ближе к столу агенту. Поднять блок.»

Невырожденные сценарии позволяют быстрее синтезировать конкретизированный план поведения, уточняя на каждой итерации планирования имеющийся абстрактный прецедент осуществления деятельности. Вырожденные сценарии требуют от агента рассуждений в процессе синтеза плана поведения, но позволяют не распространять внутренние характеристики агента и опубликовывать только абстрактное представление о имеющихся возможностях.

Для описания различий в характеристиках агентов коалиции, на стратегическом уровне архитектуры STRL используются ограничения при абстрактном синтезе группового плана поведения. Ограничения описаны на уровне формулирования задачи на стандартном для планирования языке разметки PDDL 3 [101] в поле «:constraints». После считывания задачи и пополнения картины мира агентов происходит процесс выбора агента-координатора и синтез общего субоптимального плана поведения (подробнее в параграфе 2.3). Агент-координатор формирует вырожденные сценарии пространственного поведения для каждого из входящих в план агентов и рассыпает их. Вырожденность сценариев заключается в предположении агента-координатора о начальном и целевом местоположении агента, которое описывается фокусом внимания и требуемом действием. Основная сложность подхода в правильном предположении о целевом размере и наполнении фокуса внимания. Так как агент-исполнитель будет синтезировать план на основе не разглашаемых, личных данных, то его маршрут не обязательно будет совпадать с представлениями о требуемом маршруте агента-координатора, поэтому при создании невырожденного сценария агент-координатор не требует полного совпадения предполагаемого им состояния и целевого состояния агента-исполнителя. Действие, которое должен выполнить агент-исполнитель выражено абстрактно и описывает общую схему по взаимодействию с конкретной сущностью. Моторное уточнение действия происходит при локальном планировании агентом-исполнителем.

Агенты синтезируют пространственные подпланы или адаптируют имеющиеся прецеденты, после чего каждый агент рапортует о завершении самостоятельного планирования. Общий план действий рассыпается каждому из агентов, которые заменяют абстрактные личные действия на синтезированные подпланы и сохраняют план в качестве прецедентов планирования. Имеющийся успешный опыт достижения целевого состояния используется для рефлексивного представления о возможностях других агентов. В робототехнической постановке, агент-координатор уточняет пространственные детали полигона, и агенты синтезируют план с актуальными для имеющейся задачи ограничениями.

Синтез и использование абстрактных сценариев для формирования межагентный сообщений является актуальной задачей, деятельность по реализации которого ведётся в

развивающихся когнитивных архитектурах. Межагентные коммуникации представлены в архитектурах [43-45, 47, 48, 49]. В большинстве случаев его реализация шаблонизирована и не позволяет человеку получить семантически-связное предложение о дальнейшей робототехнической деятельности. Наиболее подходящий для робототехнических задач протокол коммуникаций в архитектуре HiPOP позволяет использовать сценарии восстановления плана после непредвиденных нарушений, запускает протокол перепланирования и синхронизации агентов.

1.3.3 Выводы параграфа 1.3

Синтез многоагентного плана поведения является комплексной задачей, требующий наличия средств межагентной коммуникации, выбора стратегии установления взаимодействий в группе, наличия средств по неразглашению личных данных агентов. Основными отличиями процесса построения многоагентного плана поведения в когнитивной архитектуре STRL от способов, примененных в других архитектурах, являются:

1. Динамический выбор ролевого состава коалиции, на основе прецедентных данных и рефлексивных рассуждений о возможностях агентов;
2. Использование личностных знаний агентов о реализации требуемой абстрактной деятельности;
3. Сценарный подход к синтезу коммуникационных сообщений агентами коалиции.

1.4 Выводы главы 1

Создание когнитивной архитектуры, способной обеспечивать деятельность агента в условиях реальной среды является комплексной задачей, для реализации которой необходимо учитывать множество описанных в главе факторов. Современные подходы используют методы, позволяющие синтезировать требуемое для выполнения узкоспециализированных задач поведение. Отсутствие качественного обобщения и улучшения этих методов не позволяет робототехническим агентам обеспечивать достаточный функционал для осуществления деятельности в процессе решения многоагентных иерархически-сложных задач. В главе обосновывается выбор знаковой картины мира в качестве основы для создания когнитивной архитектуры, в которой используются ранее не реализованные подходы:

1. Знания агента персонифицированы и реализован протокол динамического распределения ролей на основе имеющихся знаний о коалиции;
2. Используется иерархия представления пространственных знаний, позволяющая синтезировать комплексный и адаптированный для выполнения в реальной среде план поведения агента;
3. Знания агента пополняемы в процессе деятельности и представлен сценарный метод описания комплексных знаний, позволяющий реализовать решение многоагентных иерархических задач.

Глава 2

2 Планировщики семейства МАР

В главе рассмотрены алгоритмы синтеза классического, иерархического, пространственного и многоагентного планов поведения. Автором диссертации было усовершенствованы алгоритмы распознавания, означивания, планирования и сохранения опыта классического планирования и реализованы остальные представленные алгоритмы. На момент начала диссертационной деятельности, классический алгоритм планирования использовал большинство знаковых процедур и был способен распознавать и означивать базовые задачи домена «Мир блоков». Автором был введен субъектный способ представления знаний агентом, усовершенствован алгоритм пополнения знаний, с помощью реализации возможности пополнять имеющиеся прецедентные знания интерпретацией данных новых задач. Были обновлены механизмы означивания предикатов задачи, возможных действий и ограничений. Расширение механизмов означивания позволило интерпретировать большее количество типов задач и ограничить перебор возможных действий агента, с помощью процедуры предзаполнения матриц действий ссылками на знаки уточненных ролей объектов. Также, автором было переработана структура представления ситуаций планирования и был осуществлен перенос ситуаций на сеть образов.

Классический алгоритм планирования был дополнен функциями предактивации возможных прецедентных действий, что позволило снизить сложность алгоритма в n раз, где n – количество знаков в картине мира агента на каждом шаге планировщика. Были разработаны алгоритмы распознавания и означивания иерархических задач, а классический планировщик дополнен функционалом поиска доступных иерархических планов. Функционал структурных методов синтеза следующей ситуации на каждом из рекурсивных шагов планировщика был расширен и представлен в ООП виде.

Автором были внесены незначительные изменения в алгоритм сохранения классического плана поведения, в число которых входит ООП представление задачи планирования и реструктуризация механизмов оптимизации картины мира агента. Также была создана библиотека map-core и произошло разделение знакового функционала и алгоритма планирования, как одной из возможных функций агента с ЗКМ.

Были разработаны описанные в главе алгоритмы многоагентного распознавания и означивания задачи, дополнена структура поиска возможных планов поведения в многоагентной

постановке. Усовершенствован алгоритм сохранения опыта в многоагентном случае с помощью выявления и сохранения подпланов. Разработан алгоритм пространственного означивания, планирования и сохранения опыта. Алгоритмы были представлены в виде библиотек map-multi и map-spatial. Также, были разработаны робототехнические решения, описание которых присутствует в главе 4. Функционал взаимодействия алгоритмов планирования семейства MAP представлен на рисунке 5.

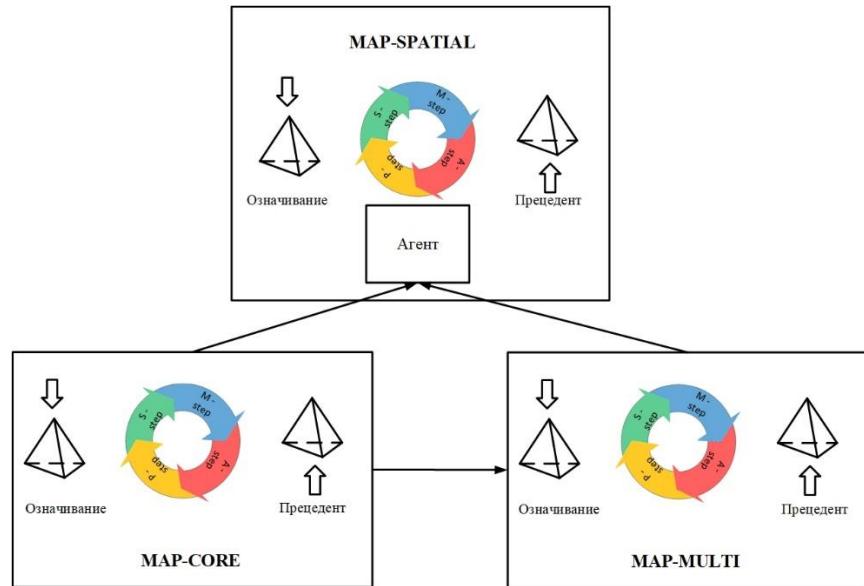


Рисунок 5. Схема взаимодействия алгоритмов семейства MAP.

2.1 Формальная модель компонент знака

Еще раз отметим, что в общем случае, картина мира агента с ЗКМ представлена семиотической сетью, базовым элементом которой является знак. Знаки обозначают объекты среды, участников коалиции, доступные агентам действия, отношения между объектами, пространственные и количественные абстракции. Каждый из знаков имеет каузальные матрицы на сетях значений (m), образов (p) и смыслов (a), а также уникальное имя знака (n). На сети m представлена ролевая связь между знаками, примером служит отношение «блок а» - «блок» или отношение «самолет 1» - «самолет». На этой сети, у знаков действий присутствуют не конкретизированные матрицы, обозначающие абстрактное описание действия. На сети p присутствуют отношения «часть – целое», примером которого в простейшем случае является представление «блок а» - «ситуация 1». В большинстве случаев, описание сущностей задачи присутствует в контексте представления отношения в ситуации. Например, ссылка на знак «блок а» будет сначала присутствовать в каузальной матрице знака отношения «блок а на блоке б» или «блок а справа», а уже ссылка на знак отношения будет содержаться в каузальной матрице ситуации. На сети a присутствует уточненное описание действий и активированные матрицы знаков, в множестве которых происходит поиск решения текущего шага итерации планирования. Сеть a представляет аналогию рабочей памяти, в которой происходят основные действия по достижению целевого состояния. Компоненты знаков образуют одноименные семантические сети W_m, W_p, W_a , где между узлами сетей установлено взаимно-однозначное соответствие, позволяющее задавать функции связывания между ними. Кортеж из 5 элементов $\langle W_p, W_m, W_a, R^n, \Theta \rangle$ называют картиной мира когнитивного агента, где $R^n = \langle R^m, R^a, R^p \rangle$ - отношения на компонентах знака, а Θ – операции на множестве знаков.

Алгоритмы, представленные далее, позволяют формировать и пополнять семиотическую сеть агента, а также производить соответствующие операции, с помощью которых агент достигает целевое состояние.

2.2 MAPCORE - Прецедентный и иерархический способ синтеза плана поведения

2.2.1 Процесс пополнения картины мира агента

Пополнение картины мира агента происходит с помощью анализа поставленной проблемы и её классификации. Имеющиеся знания агента обновляются согласно логике проблемы и формируется задача, описывающая представление агента о сложившейся проблеме. Также, формирование задачи может происходить не только с помощью описания проблемы, но и посредством обработки результатов алгоритма целеполагания. Например, агент может выявить задачи более низкого уровня (для обнаружения объекта в слабоосвещенной комнате необходимо включить свет или раздвинуть шторы) и синтезировать подплан для их решения. Подцели также пополняют картину мира агента и активируют знания с ними связанные. Классическая постановка задачи планирования осуществляется с помощью передачи агенту сообщения в шаблонизированной форме, частным случаем которой является формулирование проблемы на языке разметки PDDL (см. 3.1.2). Передача проблемы на этом языке осуществляется с помощью двух файлов – файла задачи и файла домена. Файл домена является абстрагированным описанием среды, в которой необходимо искать способы решения задачи и ограничивает агента в пространствах действий и ролей объектов. Этот метод является классическим для областей искусственного интеллекта, связанных с процессом принятия решений, поэтому в настоящей работе будет кратко изложена синтаксическая составляющая языка PDDL и механизмы его распознавания в главе 3.

Как было описано выше, формирование задачи планирования происходит с помощью данных, полученной из описания проблемы. Полученные данные агент адаптирует к собственной картине мира с помощью алгоритма означивания. Этот алгоритм позволяет находить связанные с описанием сущностей задачи закономерности, позволяющие классифицировать объекты и субъекты среды, исходя из опыта взаимодействий с ними. Алгоритм означивания состоит из 13 шагов (см. алг. 1):

1. **For** n **in** N : # Актуализация объектов, субъектов и предикатов картины мира агента согласно данным задачи
2. **If** $n \notin S$ **then**
3. $S \cup \{s(n)\}$
4. **Else:**
5. $z_n^m = \psi_m^m(s_n)$
6. **For** act **in** A : # Актуализация действий
7. **If** $act \notin S$ **then**
8. $S \cup \{s(act)\}$

9. $z_{act}^m = \psi_m^m(s_n)$
10. $z_{act}^a = \psi_m^a(z_{act}^m)$
11. **Else:**
12. $z_{act}^m = \psi_m^a(s_n, N)$
13. $z_{sit-start}^p, z_{sit-goal}^p = \psi_n^p(N)$ # Создание терминальных состояний

Алгоритм 1. Процесс означивания проблемы планирования.

Означивание начинается с актуализации знаний о сущностях задачи, их типах, и предикатах, которые описывают сущности. Сущностями являются как объекты, так и субъекты задачи, места действий, описания их характеристик и т.д. На первом шаге происходит итерация по N – множеству имен сущностей. Если сущность не представлена в картине мира агента (в множестве знаков S), то на 3 шаге создается знак этой сущности $s(n)$ и активируются отношения, представленные в описании задачи относительно неё. Если сущность представлена в картине мира агента, то на шаге 5 создаётся или активируется каузальная матрица сущности на сети значений z_n^m , формирующая внутреннее представление агента о связи сущности с целями задачи. Так как в задачу планирования входят предикативные описания отношений, то для каждого из предикатов, на шаге 5, создается матрица на сети значений, связывающая предикат с ролевым составом объектов, которые он характеризует. Далее в алгоритме, на шагах 6-11 представлен механизм означивания действий задачи (действия представлены множеством A , которое пополняется в процессе распознавания домена). Основным отличием означивания действий от означивания остальных сущностей является создание или активация матриц z_{act}^m не только на сети значений функцией $\psi_m^m(s_n)$, но и на сети смыслов (матрицы z_{act}^a) функцией копирования матрицы $\psi_m^a(z_{act}^m)$ на шагах 9-10. Активация на сети смыслов требуется для характеристики способностей планирующего агента. Основным отличием каузальных матриц действий на сетях смыслов и значений является фактор конкретизации. Каузальные матрицы действий на сети значений более абстрактны и регламентируют ролевой состав объектов, а каузальные матрицы на сети смыслов характеризуют конкретизированные действия с одной сущностью. Если в картине мира присутствует знак действия, то он обновляется и создается матрица на сети значений z_{act}^m процедурой $\psi_m^a(s_n, N)$. На шаге 13, агент формирует описание начальной и целевой ситуаций, формирует их знаки и создает каузальные матрицы на сети образов $z_{sit-start}^p, z_{sit-goal}^p$, выделяя требуемую для планирования информацию функцией $\psi_n^p(N)$. Сложность актуализации знаний заключается в переборе всех сущностей задачи и действий, которые могут быть применены с этими сущностями и оценивается как $O(n)$, где n – число рассматриваемых сущностей. Заранее известное ролевое представление сущностей ограничивает

перебор возможных вариантов действий с сущностями при предактивации матриц на сети смыслов и оценивается как $O(m)$, где m – число действий задачи. Поэтому, сложность всего алгоритма оценивается, как $O(m \times n)$.

Приведем пример означивания задачи домена «Мир Блоков». Предположим, что ЗКМ агента пуста и необходимо означить задачу полностью. Тогда на первом шаге, агент выбирает все объекты и сформирует отношения тип (роль) – объект (а, б, в, г – блоки; большой, маленький – размер) и тип – подтип (объект – блок; объект – размер). Далее создаются знаки «Я» и знак «Ситуация». Для каждого из объектов и субъекта создается соответствующий знак и добавляется матрица на сети значений. Если субъект – агент, то в матрицу значений агента будет добавлена ссылка на знак «Я». Далее происходит создание знаков и матриц значений ролей объектов и добавление в них ссылок на знаки объектов. Аналогичная процедура происходит и для предикатов. Создаются знаки и матрицы предикатов «На», «На столе», «Пустой», «Манипулятор пуст» и «Манипулятор задействован». Матрицы на сети значений знаков предикатов дополняются ссылками на соответствующие типы объектов. Далее, следует процесс создания знаков и матриц действий «Поднять», «Положить», «Состыковать» и «Расстыковать». Каждая матрица на сети значений действий пополняется ссылками на соответствующие структуре действия знаки предикатов. Далее, матрицы копируются на сеть смыслов и происходит проверка ограничений задачи. Если присутствуют ограничения, то ссылки на роли заменяются ссылками на конкретизированные объекты. Например, агент не может взаимодействовать с большими блоками, но может с маленькими и средними. Тогда в матрицах всех действий агента будет ссылка на маленькие и средние блоки, вместо ссылки на знак роли «блок». Алгоритм означивания завершается созданием знаков и матриц на сети образов начальной и конечной ситуаций. Примером начальной ситуации служит нахождение всех блоков на полу, и пустой манипулятор, а примером конечной ситуации башня из блоков. Ссылки на знаки ситуаций добавляются в каузальные матрицы знака «Ситуация».

Помимо синтеза классического плана поведения, агент с ЗКМ способен синтезировать иерархический план поведения, основываясь на иерархической формулировке задачи планирования, которая представлена на языке разметки HDDL. Основным отличием от классических задач является отсутствие описания целевого состояния агента, которое заменено на последовательность абстрактных действий («принеси кофе», «доставь кубик А в комнату 2» и т.д.). Описание абстрактных действий является частным случаем формулирования сценария деятельности и требует списка возможных поддействий различного уровня иерархии. Представление абстрактных действий, согласно правилам языка разметки, осуществляется в методах задачи. Актуализация знакового представления задачи происходит так же, как и в

классическом случае, за исключением замены формирования каузальной матрицы целевой ситуации на формирование знаков всех методов и их конкретизацию на сети смыслов. Так как в описании методов отсутствует конкретизированная, заранее заданная последовательность действий, а присутствует только абстрактное описание возможных действий без уточнения их последовательности, то сложность алгоритма означивания иерархической задачи возрастает до $O(n \times m^2)$.

Для иерархической постановки задачи добавляется означивание подзадач и методов. Примером подзадачи является сущность «Прибыть в», решением которой является метод с единственным действием «Ехать из комнаты X в комнату Y». Для каждой из подзадач создается знак подзадачи и матрица на сети значений, в эту матрицу добавляется ссылка на знак действия. Если подзадача иерархически сложная, то в матрицу значений подзадачи добавляются ссылки на соответствующие матрицы вложенных подзадач. Примером служит матрица знака «Доставить груз», расширенного домена «Мир Блоков», в рамках которого присутствуют комнаты, в которых находятся грузы. Комнаты соединены коридорами, по которым агент способен добираться из 1 комнаты в другую. Каузальная матрица будет включать ссылки на знаки «Приехать в комнату 1», «Забрать груз», «Приехать в комнату 2», «Выгрузить груз». Далее следует означивание начальной ситуации и создание матрицы смыслов и знака целевого метода. Этот метод описывает последовательность действий, которые должны быть выполнены. Примером служит «Доставить груз «а» в комнату 2; доставить груз «б» в комнату 1». Для пополнения матрицы смыслов происходит поиск требующихся методов («Доставить»), копирование их матриц значений на сеть смыслов и заполнение ролей на требуемые объекты. Таким образом, если распространить активность от матрицы метода «Доставить» вниз по сети значений, то становится очевидным, что были заменены матрицы «Комната ?pB» на «Комната 2», а «Комната ?pД» на «Комната 1», а также матрицы блоков и агента.

2.2.2 Алгоритм синтеза плана поведения

Синтез плана поведения осуществляется итерационной процедурой синтеза и актуализации следующей ситуации и проверкой на активацию матриц знаков, ссылки на которые формируют описание целевой ситуации. На начальном этапе актуализируются все прецеденты планирования, после этого агент совершает 4 шага – S, M, A, P. (см. рис. 6.). Также, как и алгоритм означивания, синтез плана осуществляется как для иерархической постановки задачи, так и для классической. Для иерархической задачи агент находит уточнение каждого из абстрактных действий последовательно, согласно правилам очередности задач, представленном в описании задачи планирования. Основным отличием синтеза иерархического плана является наличие

дополнительного ограничения на выбор действий при уточнении каждой из абстрактных задач. Например, для задачи доставки объектов ограничения будут регламентировать деятельность агента только с целевыми объектами, уменьшая комбинаторную сложность шага выбора следующего действия. (см. алг. 2). Алгоритм планирования может синтезировать план как от начальной ситуации к целевой, так и от целевой к начальной, посредством поиска действий, эффекты которых привели к активации целевой ситуации.

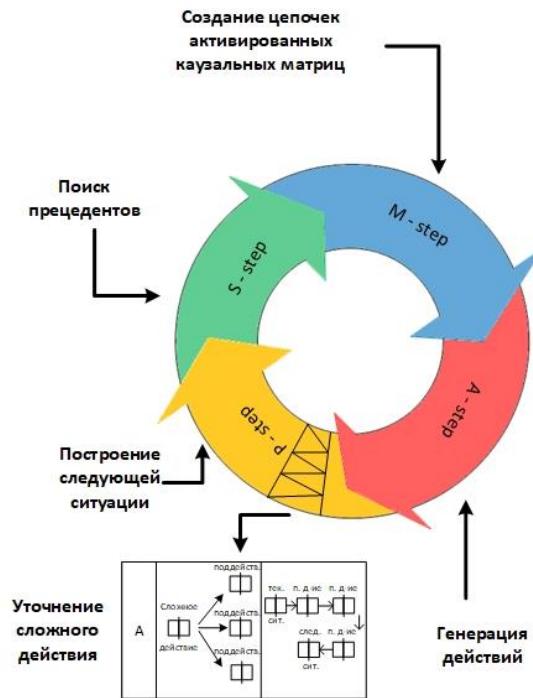


Рисунок 6. Процесс классического планирования.

1. **Function MAP_SEARCH** ($z_{sit-cur}^p, z_{sit-goal}^p, plan, i$):
2. **If** $i > i_{max}$: **Return** \emptyset # Проверка итерации
3. $z_{sit-cur}^a, z_{sit-goal}^a = \psi_p^a(z_{sit-cur}^p, z_{sit-goal}^p)$ # Активация ситуаций
4. **For** $\forall s : s \in Pr$: # Активация прецедентов
5. **If** $\forall z_s^a : z_s^a \geq z_{sit-cur}^a$ **then** $A_{pr} \cup \varphi_a \uparrow ((s(z_s^a), 1))$
6. $C_p \cup \varphi_p \downarrow ((z_{sit-cur}^p), 4)$ # Активация матриц знаков объектов
7. **For** $\forall c_p : c_p \in C_p$: # Активация матриц действий на m
8. $F_m \cup \varphi_m \uparrow (s(z_c^p), 3), z_c^p = c_p[-1]$
9. **For** $\forall f : f \in F_m$: # Активация матриц объектов на m
10. $C_m \cup \varphi_m \downarrow ((z_f^m), 6)$
11. **For** $\forall c_m : c_m \in C_m, \forall c_a : c_a \in C_a$: # Ограничение на действия с объектами ситуации

12. **If** $s(c_m[-1]) = s(c_a[-1])$ **then** $C_{mr} \cup c_m$
13. $A_{com} \cup \psi_m^a(C_{mr}, f)$ # Синтез всех действий
14. **For** $\forall z_{act}^a : z_{act}^a \in A_{com}$: # Отбор применимых действий
15. **If** $\Delta(I^c(z_{act}^a), z_{sit-cur}^a) = 0$ **then** $A_{suit} \cup \{z_{act}^a\}$
16. $A_{suit} \cup A_{pr}$ # Дополнение применимых действий прецедентами
17. **For** $\forall z_{suit}^a : z_{suit}^a \in A_{suit}$: # Эвристический отбор действий
18. **If** $\Delta(z(s(\psi_a^a(z_{suit}^a), \psi_a^a(z_{sit-cur}^a)))_{sit-next}^a, z_{sit-goal}^a) > \Delta(\psi_a^a(A_{suit} \setminus \{z_{suit}^a\}))$ **then** $A_{cand} \cup \{z_{suit}^a\}$
19. **For** $\forall z_{cand}^a : z_{cand}^a \in A_{cand}$:
20. $[z_{sub}^a] = \psi_p^a(z_{cand}^a)$ # Поиск поддействий
21. $z_{sit-next}^p = I^e(z_{cand}^a) \cup I^c(z_{sit-cur}^a) \setminus I^c(z_{cand}^a)$ # Формирование следующей ситуации
22. $\{plan\} \cup \{z_{cand}^a, z_{sit-cur}^a\}$ # Пополнение плана
23. **If** $z_{sit-goal}^p \in z_{sit-next}^p$ **then** # Если достигнута конечная ситуация
24. $\{F_{plans}\} \cup \{plan\}$ # Добавляем план в множество возможных
25. **Else**: # Иначе вызов следующего шага рекурсии
26. $Plans := MAP_SEARCH(z_{sit-next}^p, z_{sit-goal}^p, plan, i+1)$

Алгоритм 2. Синтез плана поведения МАР агентом.

Синтез плана начинается с проверки настоящей итерации планирования на достижение максимальной длины плана $i > i_{\max}$. Если максимальная длина превышена, то план не может быть синтезирован без разделения настоящей задачи на подзадачи. Далее, на шаге 3 происходит активация каузальных матриц на сети смыслов начальной и целевой ситуаций $z_{sit-cur}^a, z_{sit-goal}^a$, схожих по структуре каузальным матрицам ситуаций на сети образов, функцией $\psi_p^a(z_{sit-cur}^p, z_{sit-goal}^p)$ перехода между сетями образов и смыслов. На шагах 4 и 5 происходит проверка каузальных матриц на сети смыслов всех прецедентов (множество Pr) планирования на содержание ссылок на знак настоящей ситуации $z_{sit-cur}^a$. Если прецедент был найден функцией распространения активности вверх от знака ситуации по сети смыслов $\varphi_a \uparrow ((s(z_s^a), 1)$, то множество активных в настоящей ситуации прецедентов планирования A_{pr} пополняется. Далее, на шаге 6 происходит поиск множества цепочек матриц C_p , полученных функцией распространения активности вниз $\varphi_p \downarrow ((z_{sit-cur}^p), 4)$ по сети образов от матрицы текущей

ситуации. На шагах 7-8 функцией распространения активности вверх по сети значений $\varphi_m \uparrow (s(z_c^p), 3)$ и выбором матрицы $z_c^p = c_p[-1]$ отбираются роли объектов настоящей ситуации и формируется F_m - множество матриц не полностью означенных действий на сети значений. Шаги 9-10 в процессе распространения активности вниз по сети значений $\varphi_m \downarrow ((z_f^m), 6)$ формируют C_m - множество цепочек матриц, полученных при распространении активности вниз по сети значений от матриц действий из F_m . Они требуются для формирования C_{mr} - множества цепочек матриц объектов, входящих в настоящую ситуацию, действия с которыми возможны для агента планирования. Далее, с помощью процедуры $\psi_m^a(C_{mr}, f)$ связывания матриц действий на сетях значений и личностных смыслов происходит формирование и активация возможных действий A_{com} . Это множество ограничивается доступными действиями, при решении иерархической задачи. Из множества A_{com} , на шагах 14 - 15 отбираются действия A_{suit} , которые в условиях $I^c(z_{act}^a)$ активируют матрицы знаков, входящие в описание матрицы смыслов $z_{sit-cur}^a$ настоящей ситуации (или в эффектах, при обратном планировании). На шаге 16 происходит объединение множества найденных действий A_{suit} и множества прецедентов планирования A_{pr} и на шаге 18 происходит эвристический формирование множества действий A_{cand} , которые наиболее быстро приведут к активации матрицы смыслов целевой ситуации. Каждое из действий в A_{suit} применяется к настоящей ситуации, и матрица новой ситуации $z(s(\psi_a^a(z_{suit}^a), \psi_a^a(z_{sit-cur}^a)))_{sit-next}^a$ сравнивается с матрицей целевой ситуации. Для каждого из отобранных действий на шагах 19-22 формируется новая ситуация, далее, прецедентные действия уточняются до списка элементарных под действий $[z_{sub}^a]$ на шаге 20 функцией $\psi_p^a(z_{cand}^a)$. На 23 шаге формируется следующая ситуация $z_{sit-next}^p$ с помощью пополнения условий каузальной матрицы настоящей ситуации эффектами выбранного действия $I^e(z_{cand}^a)$ и удалением условий действия $I^c(z_{cand}^a)$. Выбранное действие и ситуация пополняют текущий план $\{plan\}$. Если матрица образов следующей ситуации $z_{sit-next}^p$ содержит ссылки на знаки, входящие в описание целевой ситуации, либо метод иерархической постановки задачи целиком уточнен, то происходит сохранение готового плана поведения в множество $\{F_{plans}\}$ и продолжается поиск других доступных планов. Если матрица целевой ситуации не была активирована, или иерархический метод уточнен не полностью, на шаге 26 происходит рекурсивный вызов рассмотренной процедуры. Отдельно необходимо упомянуть про процедуры синтеза A_{com} , эвристического отбора A_{cand} и поиска под действий z_{sub}^a .

Процедура синтеза действий A_{com} начинается с получения всех возможных действий агентом посредством распространения активности вверх от знака агента по сети смыслов. Далее, для каждого из действий происходит проверка на отсутствие не конкретизированных матриц ролей объектов, которые могут присутствовать при наличии информации об ограниченности возможностей агента взаимодействовать с определёнными типами объектов. Так как осуществляется итерация по всем возможным действиям сложность этой процедуры $O(m)$, где m – количество имеющихся действий. Например, если агент может взаимодействовать только с большими объектами, во всех его действиях, в алгоритме означивания, будет заменена ссылка на знак роли *размер* на ссылку на знак роли *большой*, но все возможные матрицы действий для каждого большого объекта созданы не будут. При нахождении полностью заполненной матрицы, она добавляется в множество A_{com} , а алгоритм продолжит работу. Далее, для снижения комбинаторной сложности, производится удаление матриц всех действий, не подходящих под шаблон описания матрицы действия. Такие матрицы синтезируются в процессе адаптации опыта задачи для настоящей ситуации. Эта процедура позволяет оставлять число полностью синтезированных действий в размерности m и не рассматривать дополнительные ошибочные действия. После этого производится уточнение всех возможных знаков ролей, входящих в отобранные действия, знаками имеющихся объектов задачи. Этот процесс увеличивает сложность процедуры формирования A_{com} до $O(m \times n)$, так как для каждого действия должны быть проверены все знаки ролей и получены все объекты, их уточняющие. Процедура синтеза новых действий является наиболее алгоритмически сложной в МАР-алгоритме и характеризует всю сложность алгоритма как $O(m \times n)$. Упрощение процедуры происходит при использовании прецедентных знаний агента. В этом случае, на каждом из шагов планирования, в картине агента имеется полный список уже уточненных действий и сложность уменьшается до $O(m)$.

Эвристический поиск, используемый в настоящем алгоритме, осуществляется посредством синтеза знака следующей ситуации и оценивания похожести его матрицы образов на образ целевой ситуации в классической интерпретации или оценкой заполненности матрицы рассматриваемого метода в иерархической постановке. Следующая ситуация формируется аналогично шагу 21 алгоритма 2. После оценивания каждого из действий выбираются те действия, которые имеют наибольшее число схожих элементов (в классическом случае это заполненные матрицы знаков отношений) и пополняют A_{cand} .

Процедура поиска поддействий сложного прецедентного действия z_{sub}^a происходит рекурсивно и схожа по структуре алгоритму 2. Исключениями являются отсутствие S и M шагов, на A-шаге используются имеющиеся действия, среди которых происходит поиск тех, которые

применимы в настоящей ситуации. На Р-шаге происходит проверка новой ситуации и сравнение её с имеющимися прецедентными ситуациями. Если новая ситуация не входит во множество прецедентных, то ветка поиска подплана заканчивается и алгоритм возвращается к предыдущему шагу ветвления, полученному, посредством множественности применимых действий в каждой ситуации. Сложность этого алгоритма $O(m \times i)$, где m – возможные поддействия сложного действия, а i – их число. Так как в большинстве задач число i существенно меньше числа возможных действий при формировании A_{com} , то эта процедура выполняется быстрее, чем процедура синтеза новых действий.

Рассмотрим пример поиска плана для домена «Мир блоков». После означивания домена и задачи было сформировано 19 новых знаков, среди которых 4 блока, 4 действия, предикаты и роли. На шаге 3 алгоритма 2 матрицы начальной и конечной ситуаций активируются и скопируются на сеть смыслов, после этого произойдет поиск прецедентов планирования. В простейшем случае, множество прецедентов пусто. Далее активируются цепочки матриц, оканчивающиеся матрицами знаков объектов и субъекта (блок «а», «б», «в», «г», агента «Я»). Всего будет найдено 11 цепочек, знаки последних элементов которых потребуются для активации каузальных матриц на сети значений каждого из действий. Будет активировано 4 матрицы действий «Поднять», «Положить», «Состыковать» и «Расстыковать». После этого, для каждого из действий произойдет процесс распространения активности вниз по сети значений и отбор из всего множества цепочек матриц тех, чей последний элемент будет входить в цепочки, полученные из целевой ситуации. Например, в случае действия «Поднять» будут отобраны 20 цепочек «поднять – блок?x – блок – имя блока», где имена блоков будут «а», «б», «в», «г». Далее будет сформировано 4 возможных действия, обозначающие подъем каждого из блоков. На шаге 14-15 будут отобраны применимые действия, которые при прямом планировании будут действиями «Поднять» для каждого из 4 блоков (все блоки одинаковы и лежат на полу, на блоках ничего не находится) или 1 действие «Состыковать» при обратном планировании (существует башня из блоков, нужно синтезировать план по её постройке). В случае иерархического планирования, множество допустимых действий ограничивается действиями рассматриваемого метода. Далее, будет эвристически отобрано действие «Поднять б» при необходимости башни «а-б-в-г» или действие «Состыковать в г» при обратном планировании. При наличии прецедента будет отобрана матрица «Построить а-б-в-г». При отборе действий иерархического планирования, матрицы одного и того же действия будут отбираться до момента активации требуемых условий. При перемещении в необходимую комнату, все действия кроме перемещения в связанную коридором комнату будут отбрасываться. Пусть комнаты связаны по принципу 1-2-3, агент находится в комнате 1, по условиям подзадачи ему нужно в комнату 3.

Тогда все действия кроме перемещения из комнаты 1 в 2 будут отброшены на первой итерации, а на второй будет активировано действие перемещения из 2 в 3.

Следующим шагом является построение новой ситуации для каждого из отобранных действий. В случае прямого планирования, матрица новой ситуации будет содержать неизменную информацию касательно блоков «а», «в» и «г», но добавится ссылка на знак «Манипулятор задействован блок б», а ссылки на «Манипулятор пуст» и знаки состояния блока «б» будут убраны. Далее, при использовании прецедента, будет совершен поиск поддействий, ссылки на которые будут добавлены в план (поднять б, состыковать б а, поднять в, состыковать в б, поднять г, состыковать г в). Последним шагом является проверка на достижение целевой ситуации, при которой итеративно будут сравниваться все события матрицы целевой ситуации и события матрицы следующей ситуации. В иерархическом случае будет произведена проверка на активацию всех требуемых поддействий. Если условие не пройдено, то цикл повторяется.

2.2.3 Алгоритм создания прецедента планирования

После завершения алгоритма планирования происходит отбор наиболее приемлемого по длине плана и создание прецедента планирования. Создание прецедента планирования является одной из наиболее важных составляющих процесса принятия решений. Как было описано ранее, прецеденты планирования ускоряют процесс поиска плана поведения и позволяют сравнивать варианты выполнения плана, выбирая наиболее приемлемый исходя из набора факторов описания ситуации. Процесс создания прецедента планирования описан в алгоритме 3.

1. **For** $\forall st \in \{plan\}$:# Отбор ситуаций и действий плана
2. $A \cup st_{act}, Sit \cup st_{sit}$
3. **For** $\forall s \in S$: # Оптимизация ЗКМ
4. **If** $z_s^a \in A$ **then**
5. **Del** $z_s^a \notin A$
6. **Elif** $z_s^a \in Sit$ **then**
7. **Del** $z_s^{ap} \notin Sit$
8. **Else:** Del s
9. $s_{plan} = S(plan_{name})$ # Создание знака плана
10. $z_{plan}^a = \psi^a(s_{plan})$ # Создание матрицы смыслов знака плана
11. $z_{plan}^a = I^c(z_{plan}^a) \cup \{z_{sit-start}^a\}, I^e(z_{plan}^a) \cup \{z_{sit-goal}^a\}$ # Пополнение матрицы смыслов ссылками на начальную и конечную ситуацию

12. $z_{plan}^p = \psi_p^p(s_{plan})$ # Создание образа плана
13. **For** $\forall z_{act}^a \in A$: # Пополнение образа ссылками на действия плана
14. $I^c(z_{plan}^p) \cup \{\psi_a^p(z_{act}^a)\}$
15. $Agent.save(S)$

Алгоритм 3. Создание прецедента планирования МАР агентом.

Процесс создания прецедента начинается с выделения матриц действий (A - множество действий плана) и ситуаций (Sit - множество всех ситуаций плана). Далее, на шагах 3-8 происходит оптимизация картины мира агента, с помощью итерации по всему множеству знаков S и удалению каузальных матриц на сетях смыслов и образов у действий и ситуаций z_s^{ap} , не входящих в план. Знаки ситуаций, которые вели в тупиковые ветки планирования или в планы не оптимальной длины удаляются. На шагах 9-11 происходит создания знака прецедента s_{plan} с помощью процедуры $S(plan_{name})$ и формирование его каузальной матрицы смыслов z_{plan}^a функцией $\psi_a^a(s_{plan})$, в условия $I^c(z_{plan}^a)$ которой входят ссылка на знак начальной ситуации $\{z_{sit-start}^a\}$, а в эффекты $I^e(z_{plan}^a)$ – конечной $\{z_{sit-goal}^a\}$. На шагах 12-14 происходит создание матрицы прецедента на сети образов z_{plan}^p и наполнение условий этой матрицы $I^c(z_{plan}^p)$ ссылками на знаки с пустыми каузальными матрицами образов действий, входящих в план функцией $\{\psi_a^p(z_{act}^a)\}$. Процесс создания прецедента планирования заканчивается шагом 15 – сохранением оптимизированной картины мира агента. Сложность алгоритма сохранения прецедента заключается в переборе всех имеющихся знаков, а также в удалении неиспользованных каузальных матриц и знаков. Оценкой сложность является $O(m \times n)$, где m – количество действий плана, а n – количество знаков в картине мира агента.

При рассмотрении задачи домена «Мир Блоков», в каузальную матрицу образов прецедента будут добавлены ссылки на знаки действий, позволившие активировать матрицу целевой ситуации. В случае построения башни а-б-в-г, в первое событие будет добавлена матрица знака «поднять б», а в последнее «состыковать г в». Матрица смыслов будет содержать в условии ссылку на знак начальной ситуации (а, б, в, г на столе и манипулятор пуст), а в эффектах будет ссылка на знак целевой ситуации «а на столе, б на а, в на б, г на в, манипулятор пуст, блок г пуст）.

2.2.4 Выводы параграфа 2.1

В параграфе рассмотрен основной алгоритм синтеза плана поведения агента со знаковой картиной мира. Были уточнены данные о ранее упомянутых процедурах со знаками и приведен

пример работы алгоритма планирования. Детально рассмотрен алгоритм создания прецедента планирования, дальнейшие улучшения которого представлены в следующих параграфах. Основными отличиями рассмотренного алгоритма планирования от алгоритмов, используемых в других архитектурах, является:

1. Использование психологически и биологически правдоподобного представления знаний;
2. Семиотическое ограничение сущностей из картины мира агента условиями рассматриваемой задачи;
3. Унифицированность алгоритма синтеза плана, поддерживающий как классический, так и иерархический способ формулирования задачи.

2.3 MAPSPATIAL - Синтез пространственного плана поведения

2.3.1 Моделирование фокуса внимания и описание карты местности

Процесс синтеза плана поведения агента в условиях наличия двумерной окружающей среды отличается от классического. Анализ проблемы планирования дополняется актуализацией знаний агента о среде, в число которых входит информация об удаленности объектов и субъектов от агента, а также наличие препятствий на пути агента. Как было описано в подпараграфе 1.1.3, картина мира агента пополняется знаками карты местности и фокуса внимания. Для описания пространственных отношений используются знаковое представление псевдофизической логики Д.А. Поспелова. В отличии от классической задачи планирования, пространственная задача формулируется как с помощью языка разметки PDDL, на котором описаны абстрактные возможности агента с их ограничениями, так и с помощью синтаксиса формата JSON с пространственным описанием проблемы. Файл пространственного описания содержит перечень точных начальных и целевых координат объектов и субъектов планирования, а также препятствий и условий, адаптирующих классические представление отношений между сущностями. Процесс пространственного означивания представлен в алг 4.

1. **For** n **in** N : # Означивание объектов, субъектов и предикатов
2. **If** $n \notin S$ **then**
3. $S \cup \{s(n)\}$
4. **Else:**
5. $z_n^m = \psi_m^m(s_n)$
6. **For** w **in** W : # Означивание неперемещаемых препятствий
7. **If** $w \notin S$ **then**
8. $z_w^p = \psi_p^p(s_n, w)$
9. $S \cup \{s(w)\}$
10. **For** act **in** A : # Означивание действий
11. **If** $act \notin S$ **then**
12. $S \cup \{s(act)\}$
13. $z_{act}^m, z_{act}^a = \psi_m^m(s_n), \psi_m^a(s_n, N)$
14. **Else:**
15. $z_{act}^a = \psi_m^a(s_n, N)$
16. $r, c = markup(coords, init)$ # Расчёт координат клеток и областей начальной ситуации и карты
17. $z_{sit-start}^p, z_{map-start}^p = \psi_n^p(N, r, c)$ # Создание соответствующих знаков и матриц
18. $r, c = markup(coords, goal)$ # Расчёт координат клеток и областей целевой ситуации и карты
19. $z_{sit-goal}^p, z_{map-goal}^p = \psi_n^p(N, r, c)$ # Создание соответствующих знаков и матриц

Алгоритм 4. Означивание пространственной картины мира.

Шаги 1-5 алгоритма 4 выполняют те же функции, что и в алгоритме классического означивания, за исключением добавочных типов сущностей, характеризующих пространственные составляющие алгоритма. Дополнением множества отношениями задачи являются: учет направления агента, содержание объектов в клетках фокуса внимания, пространственные отношения, местоположение, которое характеризует одновременно и расстояние до объекта, и направление в его сторону, занятность агентом какой-либо из клеток. Множество всех означаемых сущностей обозначено N , множество всех знаков картины мира агента S , z_n^m - матрица знака с именем n на сети значений, полученная с помощью процедуры означивания $\psi_m^m(s_n)$. Далее следует означивание неперемещаемых препятствий на карте из множества W , которые могут быть выражены стенами, столами и т.д. Их местоположения описываются на шаге 8 в виде списка координат, который в простейшем случае характеризует координаты левого верхнего и правого нижнего углов минимального прямоугольника, в который может быть вписан неперемещаемый объект. Ссылка на этот список содержится в каузальной матрице образов знака препятствия z_w^p , которая синтезируется функцией $\psi_p^p(s_n, w)$ и дополняет знания агента о настоящей задаче. Далее, каждое из действий из множества A означивается аналогично классической задаче. Элементарные действия могут быть активированы только в фокусе внимания агента, например перемещение осуществляется из одной клетки фокуса в другую, а оперирование с объектами возможно, только при наличии объекта в соседней от агента клетке и совпадению направления агента направлению из клетки с агентом до соответствующей клетки с объектом. Поэтому потребовалось дополнение каузальных матриц действий ссылками на знаки пространственных предикатов, описывающие состояние агента. Каузальные матрицы z_{act}^m, z_{act}^a формируются соответствующими процедурами $\psi_m^m(s_n), \psi_m^a(s_n, N)$. На шаге 16 происходит знаковое уточнение функцией $markup(coords, init)$ координат карты и формируются знаки областей. Области также содержат каузальные матрицы на сети образов с ссылками на описание их координат r . Также, на шаге 16 формируется внезнаковое описание координат клеток фокуса внимания с начальной ситуацией, отношения между клетками и областями и описание объектов, находящихся в соответствующих клетках. Координаты объектов и субъектов описываются в JSON файле с помощью координат центра сущности и радиуса минимальной окружности, которая может быть описана около неё. Далее, на шаге 17 происходит активация начальной ситуации и карты $z_{sit-start}^p, z_{map-start}^p$ с помощью функции $\psi_n^p(N, r, c)$, описывающей механизм создания фокуса внимания агента. Матрица на сети смыслов фокуса внимания имеет ссылки на знаки клеток, координаты которых описаны в соответствующих каузальных матрицах

образов клетки. Фокус внимания агента, в простейшем случае, имеет 2 матрицы на сети образов, первая, также, как и в классическом случае, описывает местоположение объектов в каждой из клеток и состояние агента, а вторая описывает координаты клеток фокуса внимания. На этом шаге создается знак карты местности, каузальные матрицы на сети образов которого, составлены аналогично каузальным матрицам ситуации и описывают наполнение и местоположение областей относительно агента. На шагах 18 и 19 происходят аналогичные процедуры для предположительного целевого состояния $z_{sit-goal}^p, z_{map-goal}^p$. Сложность алгоритма пространственного означивания сохраняется $O(n)$, но число сущностей n возрастает.

В контексте решения задачи расширенного пространственного домена «Мир блоков» (см рис. 7), увеличивается список сложных ролей более высокого уровня абстракции. Например, базовыми ролями являются «блок», «препятствие», «стол», «стена», «ничего», «агент», а ролями более высокого уровня являются роли «начальное направление», «конечное направление», «вид стола», «регион x, y, z» и т.д. Для простых ролей создаются знаки и матрицы на сети значений, а для сложных в каузальных матрицах добавляется ссылка на более конкретную роль. Далее происходит создание знака и матриц на сети образов для сущности «Стена». Каузальные матрицы содержат ссылку на список координат левого верхнего угла и правого нижнего. Например, знак «Стена», в простейшем случае, будет иметь 4 каузальных матрицы образов, фиксирующие координаты стен вокруг карты задачи. Процесс означивания предикатов и действий происходит структурно однотипно с классическим способом, в контексте рассматриваемого домена добавляются предикаты расстояния «Включает», «Содержит», «Близко», «Далеко», «Вплотную», «Ориентация», «Местоположение» и т.д. Для различных предикатов может быть добавлено разное количество матриц на сети значений. Например, для знака «Далеко» добавляется 2 каузальных матрицы, структурно характеризующие различные способы задания предиката. Для задачи «Мир блоков» было сформировано 15 предикатов и 6 действий. Далее происходит расчёт координат клеток и областей и создание соответствующих знаков и матриц на сетях образов и значений. Матрица на сети образов содержит ссылку на координаты клетки или области. После этого формируются знаки начальных и целевых ситуаций (фокусов внимания агентов) и карт и для каждой добавляются 2 матрицы на сети образов. Например, в первой матрице целевой ситуации задачи представлено 18 событий о местонахождении объектов вокруг агента, а во второй матрице 9 событий, содержащих ссылки на знаки клеток с соответствующими координатами.

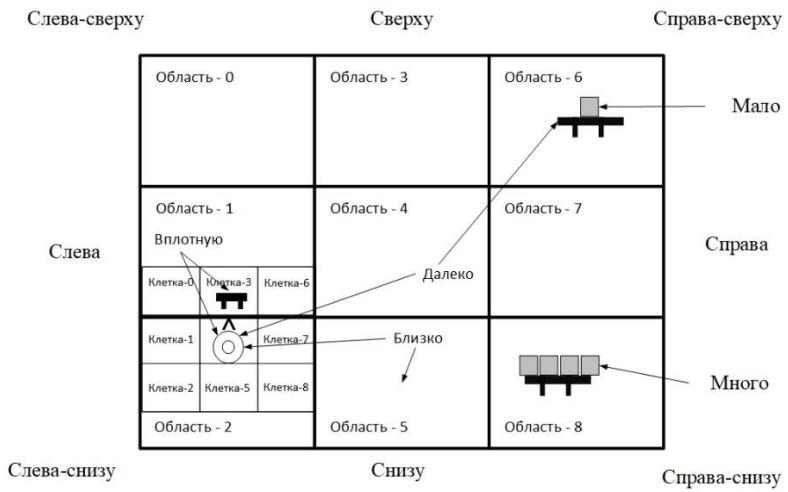


Рисунок 7. Фокус внимания агента и представление карты для расширенного домена «Мир блоков»

После процесса означивания происходит вызов процедуры классического планирования, целью которой является синтез последовательности абстрактных действий агента, с учетом ограничений на его деятельность. Далее, для каждого из абстрактных действий плана агент создает начальные и конечные пространственные ситуации и карты. Так как предсказать точное целевое пространственное состояние агента в условиях наличия препятствий затруднительно без синтеза плана поведения, состояния предсказываются эвристически. Эвристика заключается в том, что создается ситуация, в которой агент наиболее близок к начальному состоянию, а целевой объект помещен в наиболее дальнюю клетку. Направление агента устанавливается согласно направлению к целевой клетке. Если такое состояние создать невозможно, по причине наличия препятствий с требуемой стороны, то создается случайное состояние, в котором объект находится в соседней от агента клетке.

2.3.2 Процедуры уточнения и абстрагирования внимания агента

Для синтеза плана поведения в условиях среды с препятствиями был создан механизм уточнения и абстрагирования фокуса внимания. Как было описано ранее, уточнение позволяет уменьшить размер фокуса, разделив центральную клетку агента на 9 новых клеток, при условии, что не достигнут минимальный размер клетки, а процедура абстрагирования объединяет пустые клетки вокруг агента в одну и создает клетки большего размера. Процедуры являются обратными и на знаковом уровне выражены с помощью знаков метадействий «Абстрагироваться» и «Уточнить». Процесс уточнения приведен в алгоритме 5.

1. **Function** Clarify_search($ag, z_{cur-sit}^p, z_{goal-sit}^p, plan, i$):
2. **Function** Devide_sit($z_{cur-sit}^p$): # Функция создания новых матриц ситуации и карты.

3. $v = \varphi_p \downarrow ((z_{sit-cur}^a), 1)$ # Вычисление текущих координат
4. $nv = v / 3$ # Вычисление нового размера клетки
5. $rl = locater(s_r, s_o, s_w)$ # Вычисление новых отношений с областями
6. $cl = cell_creater(nv, rl, s_o, s_w)$ # Вычисление новых координат
7. $ag_st = state(s_{ag}, s_{dir}, S, z_{sit-cur}^p)$ # Формирование состояния агента
8. **Return** $z_{sit}^p, z_{map}^p = \psi_n^p(ag_st, rl, cl)$ # Создание знаков и кауз. Матриц на p
9. **If** $s_{abstr} \notin plan[-1]$: # Если последнее действие не *абстрагироваться* (или *уточнить* в случае абстрагирования)
10. $z_{sit-new}^p, z_{map-new}^p = Devide_sit(z_{cur-sit}^p)$ # Синтезируем новые матрицы ситуации и карты
11. **If** $cl_lv > goal_cl_lv$: # Если уровень уточнения больше уровня уточнения целевой ситуации
12. $z_{sit-goal-new}^p, z_{map-goal-new}^p = Devide_sit(z_{goal-sit}^p)$ # Уточняем целевую ситуацию
13. $z_{clarify}^a = I^c(z_{clarify}^a) \cup \{\psi_p^a(z_{cur-sit}^p)\}, I^e(z_{clarify}^a) \cup \{\psi_p^a(z_{sit-new}^p)\}$ # Создаем кауз. Матрицу действия *уточнить* (или *абстрагироваться* в случае абстрагирования)
14. $\{plan\} \cup \{z_{clarify}^a, z_{sit-cur}^a\}$ # Добавляем матрицу в план
15. $plan = MAP_SEARCH(z_{sit-new}^p, z_{map-new}^p, z_{sit-goal-new}^p, z_{map-goal-new}^p, plan, i)$ # Продолжаем поиск
16. **Return** $plan, i$

Алгоритм 5. Уточнение пространственной ситуации.

Уточнение начинается проверкой фактора отсутствия действия абстрагирования в качестве последнего действия плана s_{abstr} на шаге 9. Далее, вызывается функция уточнения для настоящей ситуации и формирование матриц $z_{sit-new}^p, z_{map-new}^p$. На шаге 3 происходит получение размера клетки старой ситуации v с помощью процедуры распространения активности вниз $\varphi_p \downarrow ((z_{sit-cur}^a), 1)$ по сети образов от каузальной матрицы настоящей ситуации, а на шаге 4 вычисление нового размера клетки nv . На шаге 5 происходит уточнение местоположения областей, объектов и препятствий и вычисление местоположения новых областей относительно новых координат фокуса внимания rl внезнаковой функцией $locater(s_r, s_o, s_w)$. На шаге 6 уточняется внезнаковое представление о местоположении клеток и их наполнение объектами cl функцией $cell_creater(nv, rl, s_o, s_w)$, и на шаге 7 происходит уточнение состояния агента ag_st

. Далее, на шаге 8 создаются знаки новой ситуации и карты и возвращаются их каузальные матрицы образов z_{sit}^p, z_{map}^p . На шаге 11 происходит проверка достижения уровня уточнения целевой ситуации $goal_cl_lv$, и если он меньше, чем уровень уточнения новой ситуации cl_lv , то целевая ситуация также уточняется и формируются матрицы $z_{sit-goal-new}^p, z_{map-goal-new}^p$. Необходимость уточнения целевой ситуации обосновывается добавлением в план действий одного уровня размерности клеток, с учетом изменения этой размерности только действиями уточнения и абстрагирования. На шаге 13 происходит создание матрицы на сети смыслов знака уточнения, с помощью добавления в условия матрицы $I^c(z_{clarify}^a)$ ссылки на знак, неуточненной ситуации $\{\psi_p^a(z_{cur-sit}^p)\}$, а в эффекты $I^e(z_{clarify}^a)$, ссылки на знак уточненной ситуации $\{\psi_p^a(z_{sit-new}^p)\}$. После этого, действие уточнения добавляется в план и возобновляется процедура синтеза плана поведения.

2.3.3 Алгоритм поиска пространственных подпланов

Процедура синтеза пространственных подпланов является производной от базовой процедуры классического планировщика. Также, в этой процедуре используются ограничения на допустимые действия, как при иерархическом планировании. Вначале, происходит синтез классического абстрактного плана, действия которого уточняются процедурой пространственного планирования. Каждое из действий классического планировщика характеризует набор требуемых действий пространственного планировщика. Например, в последовательность требуемых действий попадает действие поднятия объекта, тогда список допустимых пространственных действий агента будет состоять из действий по перемещению, повороту и подъему объектов.

Характерной особенностью процесса синтеза пространственного плана является наличие возможности уточнения ситуации при $A_{cand} = \emptyset$ или при активации не связанного с перемещением действия при условии, что уровень уточнения настоящей ситуации меньше, чем целевой уровень уточнения. В первом случае, агент характеризует рассматриваемую ситуацию, как безвыходную на текущем уровне уточнения, а детализация пространства позволяет находить проходы между неперемещаемыми препятствиями и достигать цели. Во втором случае, уровень уточнения целевой ситуации характеризует необходимое расстояние от агента до целевого объекта, при котором агент способен с ним успешно взаимодействовать. После уточнения, агент синтезирует подплан большего уровня детализации окружающего пространства и подъезжает к цели на требуемое расстояние. Второй характерной особенностью является наличие

возможности абстрагировать фокус внимания, уменьшив детализацию окружающего пространства и ускорив перемещение к целевому местоположению.

После синтеза следующей ситуации и карты, также как и в классическом случае, производится проверка совпадения ссылок на знаки в матрице образов следующей ситуации и ссылок в матрице образов целевой. При проверке не учитывается местоположение объектов в фокусе внимания и направление агента по причине возможной неточности предсказания целевой ситуации. Сравнение основано на проверке размерности клеток, следующей и целевой ситуаций, а также сравнения каузальных матриц знаков отношений с целевым объектом. Каузальные матрицы знака карты позволяют различать идентичные матрицы фокуса внимания и достигать целевой ситуации. Например, при перемещении агента по карте с малым объемом препятствий, могут активироваться ситуации одинакового размера, в которых не будет присутствовать ссылки на знаки объектов.

Процесс создания прецедента пространственного планирования позволяет выделять подпланы действий одинаковой размерности и подпланы, реализующие абстрактные действия (см алг. 6).

8. ...

9. **For** act **in** $plan$:

10. $s_{sp} = S(act)$

11. **For** s_{spact} **in** act :

12. **If** $s_{spact} \in \{s_{clarify}, s_{abstract}\}$: # Если существуют действия абстрагирования и/или уточнения, то формируем подпланы

13. $s_{subplan} = getsub(act)$

14. $z_{sp}^p, z_{sp}^a = \psi_p^p(s_{sp}), \psi_a^a(s_{sp})$ # Формируем матрицы подпланов аналогично матрицам плана

15. $z_{sp}^p = I^c(z_{sp}^p) \cup I^p(\psi_p^p(s_{subplan})) \cup I^p(\psi_p^p(s_{spact}))$

16. ...

Алгоритм 6. Создание пространственного прецедента.

Первые 8 шагов алгоритма описаны в алгоритме 3. Их функционал заключается в оптимизации картины мира агента и удалению ситуаций, не приводящих к целевому состоянию. На шаге 9 происходит итерация по всем абстрактным действиям act плана поведения $plan$ и для каждого из действий создается знак подплана s_{sp} на шаге 10. Далее происходит итерация по

пространственному подплану абстрактного действия и поиск действий абстрагирования и уточнения фокуса внимания (шаг 12) s_{spact} . При обнаружении этих метадействий, на шаге 13 происходит создание знака подплана $s_{subplan}$, процедурой $getsub(act)$ структура которого совпадает со структурой знака прецедента планирования (см алг 3). Каузальная матрица на сети образов знака подплана $s_{subplan}$ содержит ссылки на знаки действий, которые синтезировал агент не изменяя размер фокуса внимания, а матрица на сети смыслов содержит ссылки на знаки начальной и конечной ситуаций одинаковой размерности. На шагах 14 и 15 происходит создание матриц подплана z_{sp}^p, z_{sp}^a абстрактного действия s_{sp} схожей конструкции. В условия матрицы образов $I^c(z_{sp}^p)$ добавляются ссылки на знаки поддействий $I^p(\psi_p^p(s_{spact}))$ и вложенных подпланов $s_{subplan}$. Далее, процесс возвращается к схожей конструкции с процессом алгоритма 3, за исключением добавления в матрицу образов знака плана ссылок на знаки подпланов s_{sp} .

2.3.4 Выводы параграфа 2.2

В параграфе рассмотрены основные особенности разработанных алгоритмов для взаимодействия агента и двумерной среды. Представлен алгоритм формирования фокуса внимания агента и изложен способ синтеза плана поведения. Основными особенностями рассмотренной структуры являются:

1. Возможность психологически и биологически правдоподобно синтезировать план поведения, адаптированный к условиям реальной среды;
2. Комплексность иерархии формирования цепочки подпланов каждого рассмотренного уровня уточнения среды. Использование рассмотренного процесса позволяет накапливать знания о возможностях агента взаимодействовать со средой и увеличивает шанс наличия соответствующего прецедента в базе знаний агента при дальнейших итерациях планирования.

2.4 МАРМУЛТИ - Синтез многоагентного плана поведения

2.4.1 Динамическое распределение ролей в коалиции. Аукцион планов.

Рассмотренный в настоящей работе многоагентный способ синтеза плана поведения основывается на взаимодействии агентов, имеющих различные функциональные особенности. Задачи, решаемые гетерогенными коалициями агентов, требуют взаимодействия агентов для достижения целевого состояния. Также, как и алгоритм, рассмотренный в 2.1, многоагентный алгоритм позволяет синтезировать планы поведения как для классических, так и для иерархических задач. Список возможностей агентов описывается расширенным полем ограничений, при распознавании которого определяются все возможные отношения, в которых могут состоять агенты и объекты среды. Возможности агентов могут пересекаться, что приводит к созданию множества возможных планов поведения коалиции.

Для создания уникальной картины мира каждого из агентов, их деятельность распределяется по независимым процессам. Координация деятельности производится наиболее опытным агентом, чей опыт заключается в наиболее полных знаниях о способах осуществления действий с объектами рассматриваемой задачи. Механизм распределения агентов представлен в алгоритме 7.

1. **For** ag **in** $agents$:
2. $proc = process(ag)$ # Выделение процессов для агентов
3. $exp = proc(ag, T)$ # Поиск самого опытного агента
4. $major = \max(exp(agents))$
5. **For** ag **in** $agents$: # Информируем всех агентов о самом опытном
6. $proc(major)$
7. **Function** $agent_activation(T, ag)$: # Функционал агента
8. $exp = GROUND(ag, T)$ # Процедура означивания
9. $Plans = MAP_SEARCH(exp)$ # Процедура синтеза планов
10. $Plan = \text{best}(plans)$ # Отбор субоптимального для агента плана
11. **If** ag **is** $major$:
12. $Plans = \text{get_plans}(agents)$ # Формирование множества всех планов
13. $Strategy = \text{DecisionStrategies}(plans)$ # Получение стратегии отбора планов
14. $Solution = \text{Strategy}(plans)$ # Применение стратегии
15. $\text{Send}(solution)$ # Рассылка агентам конечного плана
16. **Else**:

17. Send(plan) # Отсылка наиболее опытному агенту субоптимального плана
18. Solution = Get() # Получение конечного плана
19. Save_solution(solution) # Процедура создания прецедента

Алгоритм 7. Распределение задач между агентами

На шаге 2 происходит создание нового процесса *proc* для каждого из планирующих агентов *ag* и межпроцессного способа связи. Далее, на шаге 3 ожидается ответ агента о количестве новых знаков *exp*, созданных при означивании поставленной задачи. На шаге 4 вычисляется наиболее опытный агент *major* и на шаге 6 эта информация рассыпается всем агентам. Каждый из агентов на шаге 8 означивает задачу и рассчитывает количество новых знаков, которые пополнили его картину мира. На шаге 9 происходит вызов процедуры синтеза возможных планов поведения *MAP_SEARCH(exp)*. Далее, на шаге 10 агент выбирает субоптимальный план *best(plans)*, основываясь на критериях наименьшей длины плана, наибольшего количества идущих подряд действий одного агента (из рассуждений о снижении нагрузки на межагентную коммуникацию) и наличия ранее спланированных схожих действий у агента, за которым будет закреплено действие плана. Выбор агента по схожим действиям происходит с помощью перебора прецедентов планирования в картине мира агента, и оценки предписанных рассматриваемому агенту действий. После выбора субоптимального плана, агент отсылает текстовое сообщение агенту координатору, который выбирает стратегию *Strategy* оценивания плана на шаге 13. На момент создания работы, наиболее успешной стратегией выбора плана, в условиях решения робототехнических задач, является стратегия аукциона планов, которую агент осуществляет на шаге 14. Аукцион, в простейшем случае, осуществляется с помощью замены имени знака «Я» агента в плане на его представление у агента-координатора и голосование агента за собственный план. План, который наибольшее число агентов выбрало субоптимальным рассыпается агентам на шаге 15. Далее, на шаге 18 агенты принимают план и сохраняют его в качестве прецедента планирования на шаге 19.

Означивание задачи, рассмотренное в шаге 8 алгоритма 7 расширяет функционал алгоритма 1 наличием знаков «Я» и «Они». (см. рис. 3, глава 1). Как было описано в подпараграфе 1.1.3, знания агента о возможностях других агентов абстрактны и ограничиваются классическим представлением действий. Каждый из агентов, на шаге 9 алгоритма 1 создаёт все возможные матрицы действий, доступных агенту, а в многоагентном случае создаются также действия, доступные другим агентам. Активация этих действий производится посредством активации знаков агентов, входящих в коалицию настоящей задачи и распространением активности по сети смыслов вверх от знаков агентов. Также, на этапе означивания происходит активация действий

«Оповестить» и «Уведомить», матрицы которых на сети значений связаны отношением класс-подкласс с матрицами знака действия «Согласовать». Знак действия «Оповестить» активируется при наличии более двух агентов коалиции, а знак «Уведомить» требуется для создания парной межагентной коммуникации. Каузальные матрицы на сети значений знаков «Оповестить» и «Уведомить», имеют ссылку на функцию, отвечающую за преобразование знакового представления плана действий в текстовое сообщение, которое может быть передано с помощью межпроцессного способ связи. Текстовое представление плана действий основано на шаблонизированном подходе генерации предложения, в состав которого входит представление агента, который высылает ответ, тип сообщения и представление имеющейся проблемы.

Процесс синтеза многоагентного плана в шаге 9 алгоритма 7 основан на рекурсивном подходе, описанном в алгоритме 2. Основным отличием является дополнение способа создания A_{com} на шаге 13. Возможные действия синтезируются для каждого из агентов в отдельности и заполняются незаполненные на шаге уточнения ограничений роли объектов. Результатом работы алгоритма планирования является множество планов различной длины, в которые входят все способы совмещения возможных действий по достижению целевого состояния и агентов, которые могут выполнять эти действия.

В контексте домена «Мир блоков», действия агента ограничиваются возможным размером блока. На этапе означивания, также, как и в одноагентном случае, создаются знаки действий и их матрицы на сетях значений и смыслов. Далее, матрицы смыслов уточняются, согласно правилам задачи. Например, у знака действия «Поднять» будет 2 каузальных матрицы на сети смыслов, в первой будет ссылка на знак блока «б», знак «большой» и на агента «агент 1», а во второй ссылка на знак блока «в», знак «маленький» и на агента «агент 2». При синтезе плана будет сформирован план «поднять б агент 1», «состыковать б а агент 1», «поднять в агент 2», «состыковать в б агент 2». Трудность заключается в том, что способности агентов могут пересекаться и, при наличии агента 1, способного поднимать как маленькие, так и большие блоки и такого же агента 2 будет сформировано 2 различных плана, среди которых потребуется совершить выбор. При усложнении задачи до 4 блоков, 3 типов блоков и 3 агентов, способных оперировать с 2 различными типами блоков каждый, оптимальных по длине планов будет 6, а не оптимальных по длине еще больше. Для отбора субоптимального для агента плана потребуется выбрать наименьший план по длине, план с наибольшей последовательностью действий любого агента, план в котором он сам участвует и тот, в котором все агенты, которые оперируют с блоками уже оперировали с ними в опыте самого агента.

2.4.2 Синтез многоагентного пространственного плана поведения

Механизм синтеза многоагентного пространственного плана поведения основан на взаимодействии многоагентного алгоритма планирования и пространственного. Также, как и при синтезе многоагентного плана поведения механизм распределения ролей формирует процессы, в которых агенты пополняют имеющуюся картину мира и синтезируют план поведения. Распределение задач между агентами основано на плане, который синтезируется под управлением агента-координатора агентами участниками на основе ограничений, описанных в формулировке классических многоагентных задач. После формирования классического плана, агент координатор осуществляет синтез начальной и конечной пространственных ситуаций, также, как и в одноагентном случае поступает агент самостоятельно. После этого следует этап передачи ситуаций агентам-исполнителям, которые синтезируют все возможные планы поведения по достижению целевых ситуаций. Ограничения возможных действий сохраняются.

В контексте расширенного домена «Мир блоков», агент-координатор формирует список вырожденных сценариев, например: (поднять б, агент 1) – пространственная каузальная матрица начальной ситуации (фокус внимания агента в начальный момент времени), пространственная каузальная матрица целевой ситуации (фокус внимания агента, в котором он держит блок), 2 словаря описания ситуаций, (состыковать б а, агент 1) – описание и т.д. Каждый из сценариев передается агенту-исполнителю, в первом случае, ограничивая его деятельность действиями «Поднять», «Повернуть», «Переместиться». Агент-исполнитель возвращает список под действий плана «Переместиться», «Повернуть», «Переместиться», ..., «Поднять». Далее происходит адаптация конечной ситуации, в которой находится агент (фокус внимания с поднятым блоком и соответствующим окружением) и предполагаемой ситуации агентом-координатором и итеративно запускается поиск плана для следующего вырожденного сценария.

После завершения процессов синтеза каждого из подпланов, агенты-исполнители формируют сообщение о завершении деятельности и уведомляют об этом агента-координатора. Агент координатор ожидает окончания планирования всех агентов-исполнителей и информирует их о завершении планирования. На этапе создания пространственного прецедента каждый из агентов заполняет матрицу образов знаков собственных подпланов, оставляя матрицы образов знаков подпланов других агентов не заполненными. Например, в контексте задачи «Мир блоков», для подпланов других агентов сформируется знак «Подплан-номер», в каузальную матрицу образов которого не будут входить ссылки на под действия агента.

2.4.3 Выводы параграфа 2.3

Агенты с ЗКМ способны синтезировать как самостоятельные, так и многоагентные планы поведения. Синхронизация деятельности агентов осуществляется с помощью распределения действий на основе рефлексивных рассуждений агентов о прецедентах планирования и выборе агентов-исполнителей из имеющейся коалиции гетерогенных агентов. Основным функционалом описанного алгоритма планирования является:

1. Возможность синтезировать план поведения как для классических задач достижения целевой ситуации, так и для иерархических;
2. Способность создавать и использовать прецеденты планирования для ускорения поиска решения проблемы;
3. Наличие механизмов использования пространственных знаний и в процессе синтеза плана поведения;
4. Допустимость многоагентной постановки задачи планирования, в рамках которой ни один из агентов самостоятельно не способен достичь целевой ситуации;
5. Наличие средств поиска консенсуса в процессе межагентной коммуникации;
6. Неразглашение личностных знаний агентов о способах уточнения абстрактного плана поведения.

2.5 Выводы главы 2

В главе рассмотрены основные алгоритмы создания и обновления знаковой картины мира агентов, описание классического, иерархического, многоагентного и пространственного способов построения плана, алгоритм сохранения прецедента планирования, уточнения пространственных координат агентов, динамического распределения ролей и аукциона планов. Описанные алгоритмы позволяют синтезировать комплексный иерархический план поведения агентов в коалиции и имеют функционал, значительно расширяющий возможности рассматриваемой когнитивной архитектуры по сравнению с её аналогами.

Глава 3

3 Библиотека map-core

3.1 Структура библиотеки

Библиотека map-core является основной реализацией функций пополнения и актуализации знаний агента с ЗКМ. Также, в библиотеке приведен пример реализации функции планирования поведения агента. Структура библиотеки разделена на две основные части: «swm» и «planner».

3.1.1 Обзор модулей строения семиотической сети

В «swm» представлены механизмы пополнения и актуализации настоящего состояния ЗКМ. В этой части библиотеки присутствует описание основных компонент знака, приведены основополагающие труды по ЗКМ и рассмотрен основной функционал агента. Функционал агента представлен классами агента и менеджера, который инициализирует агента в модуле «agent.py». В модуле «sign_task.py» представлен базовый функционал формирования задачи агенту, который заключается в реализации класса «Task» и его методе оптимизации и сохранения знаний агента, а также функции загрузки ЗКМ. Функция загрузки ЗКМ реализована отдельно от класса задачи, так как её вызов возможен для пополнения знаний для уже имеющегося объекта задачи.

В модуле «semnet.py» реализованы основные классы, описывающие составляющие ЗКМ и методы взаимодействия с ними. В их число входит класс «Sign», реализующий знаковое представление сущности, в число методов которого входят проверка знака на представление абстрактной сущности методом «is_abstract», методам добавления каузальных матриц по сетям значений, образов и смыслов «add_*» (* - название сети)(алг. 1, шаг 9), добавления связи знака со знаком, описывающим более абстрактную сущность по каждой из сетей «add_out_*», методы удаления каузальных матриц «remove_*» (алг. 3 шаг 5), переименования знака «rename» (шаг 2 и шаг 17 алг. 3), распространения активности и получения ссылок на знаки, входящие в требуемую каузальную матрицу знака «spread_up_activity_act», «spread_up_activity_obj» (hddl означивание и т.д.), «spread_down_activity_obj», «spread_up_act_one», «spread_up_activity_slice», «spread_up_activity_motor» (шаги 6 и 10 алг.2 и т.д.).

Класс «CausalMatrix» представляет методы работы с каузальными матрицами, в число которых входят методы сравнения матриц «__sub__», «__contains__» добавления события к матрице «add_event», получения события «get_event», добавление ссылки на знак, моторную

реализацию действия или список координат объекта «add_feature» (шаги 1-5 алг. 1 и т.д.), проверки на заполненность «is_empty», каузальность «is_causal», ограничений включения ссылок на знаки другой матрицы «includes», копирование матрицы «сору», расширение её функционала «expand» (шаг 13 алг. 2 и т.д.), замена ссылки на знак ссылкой на другой знак «replace», проверки на включение ссылок на знаки для каузальных матриц ситуаций «resonate», получение знаков матрицы «get_signs» и процессы распространения активности «spread_down_activity», «spread_down_htn_activity_act», «spread_down_activity_view» (означивание hddl, шаг 8 алг. 2 и т.д.).

Класс «Event» реализует функционал работы с событиями, методы которого позволяют сравнивать два события «__eq__», проверять включение ссылок на требуемые знаки «__contains__», добавления ссылки на одновременно активированные каузальные матрицы знаков «add_coincident»(шаги 6-11 алг. 1), проверять совпадение ссылок на знаки в двух разных событиях «resonate», и в событиях, относящихся к прецедентам действий «exp_resonate»(шаг 5 алг.2). Также, разработан метод копирования событий «сору», расширения описания события «expand», замены ссылки на один знак ссылкой на другой «replace» и получения знаков, описывающих событие «get_signs»(шаг 13 алг. 2 и т.д.).

Классы «Connector», «Actuator» и «View» имеют схожий функционал и описывают создание ссылки на знак, моторную реализацию действия и описание внешних характеристик сущности соответственно. Основной их задачей является осуществление возможности получения целевого объекта. Диаграмма представленных классов файла «semnet.py» представлена на рисунке 8.

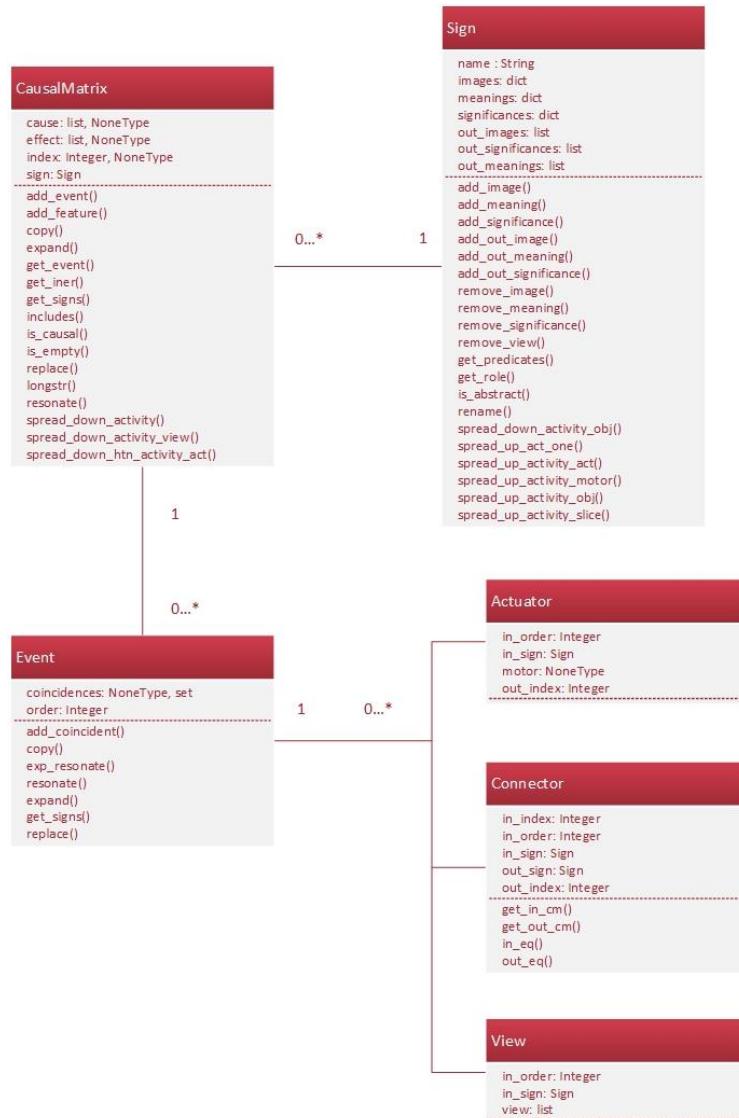


Рисунок 8. Диаграмма классов семиотической сети.

3.1.2 Обзор модулей планирования

В части библиотеки, представляющей планирование поведения, модулем, с которого начинается синтез решения задачи является «*mappanner.py*». В нем описан процесс поиска файла задачи и домена, а также вызов алгоритмов распознавания задачи. После распознавания задачи происходит активация агента планирования, чей функционал представлен в модуле «*planning_agent.py*», механизмы означивания в модулях «*pddl_grounding.py*» и «*hddl_grounding.py*», задача планирования в модуле «*planning_task.py*», механизм распознавания задачи планирования в модулях «*pddl_parser.py*» и «*hddl_parser.py*», примеры задач планирования в папки «*benchmarks*» и алгоритм поиска в модуле «*map_search.py*».

Агент планирования представлен классом «*PlanningAgent*», который наследует класс базового агента с ЗКМ и расширяет его функционал. Также, как и в базовой версии, агент

запускается специальным менеджером агента, который создает отдельный процесс для поиска решения задачи. Процесс поиска начинается с получения задачи методом «*get_task*», который вызывает метод родительского класса поиска прецедента планирования, распознает и означивает задачу. Далее, вызывается описанный в 2.1 алгоритм поиска планов, после завершения которого вызывается метод сортировки планов и прецедент сохраняется методом класса задачи.

Язык разметки PDDL был разработан группой Дрю Макдермота в 1998 году для проведения международного соревнования планировщиков IPC [103]. На момент создания диссертации было анонсировано 5 официальных версий языка. В первом обновлении под номером 1.2 было введено понятие домена и задачи планирования, далее, в 2.1 введена возможность использовать числовые данные, плановые метрики. В обновлении 2.2 были добавлены производные предикаты, позволившие моделировать зависимость фактов друг от друга и добавлена возможность описывать несвязанные с планом события. В обновлении 3.0 добавлены мягкие и жесткие ограничения на терминальные состояния, через которые должен пройти агент в процессе выполнения плана. В последнем, 3.1 обновлении была улучшена выразительность языка, в частности улучшен синтаксис описаний ограничений на деятельность агентов.

Распознавание классической pddl постановки осуществляется с помощью создания классов домена и задачи планирования. Задача планирования содержит объектно-ориентированную реализацию начального, целевого состояний и поля ограничений на деятельность агента. Описание домена планирования реализуется посредством классового представления имени задачи

(define (domain domNAME)

представления требований, регламентирующих версию языка разметки

(:requirements :strips :typing)

список типов задачи

(:types agent, object, block)

предикатов с обозначением типов переменных в контексте предиката

(:predicates (predicate1 ?x – type1 ?y – type1)

(predicate2 ?x – type2)

(predicateN ?x – typeN))

и действий, в описание которых входит список параметров, условий и эффектов действия

(:action actName

:parameters (?x – type1)

:precondition (and (predicate1 ?x ?y) (predicate2 ?x) (predicateN ?x))

:effect

```

(and (not (predicate1 ?x ?y))
     (not (predicate2 ?x))
     (not (predicateN ?x)))
     (predicate3 ?x ?y)))

```

Описание задачи планирования состоит из имени задачи, ссылки на имя домена, к которому относится задача

(define (problem probName) (:domain domNAME)

поля описания объектов и их типов

(:objects

a - type1

...

a1 - agent

a2 - agent)

начального состояния, представленного списком предикатов, истинных в начальный момент времени

(:init

(predicate1 a1)

...

(predicateN w b)

)

конечного состояния, представленного списком предикатов, истинных в целевой момент времени

(:goal

(and

(predicate5 b a)

...

(predicate1 a2)

))

и поля ограничений, которое описывает предикаты, истинные только для подгруппы агентов

(:constraints

(and

(and (always (forall (?x - type1)

(implies (predicate6 z ?x)(predicate7 a1 ?x))))

)

(and (always (forall (?x - type2)

(implies (predicate6 w ?x)(predicate7 a2 ?x))))

))))

Используемый в настоящей работе распознаватель pddl является дополненной версией распознавателя проекта Pyperplan [102]. Основным механизмом распознавания является рекурсивный обход всех символов задачи и создание графа полей по заранее заданной логике формирования задачи. Далее, для каждого поля выбирается «формула», являющаяся представлением структуры будущего ООП описания.

Язык разметки HDDL является версией PDDL для описания иерархических задач. Он позволяет описывать подзадачи, решение которых требуется для достижения целевых параметров и предлагает различные методы для синтеза подплана, описывающего решение подзадачи. Распознавание иерархической hddl задачи осуществляется с помощью авторской реализации распознавателя.

В домене, также, как и в классическом случае, представлены поля имени, требований, типов и предикатов. Также, рассмотрены поля подзадач (:task), в описания которых входят параметры подзадачи, условия и эффекты

```
(:task tNameN
  :parameters (?ag-agent ?x-param_1 ?y-param_2 ?z-param_3)
  :precondition()
  :effect())
```

поля методов (:method), в описание которых входит представление параметров метода (в простейшем случае совпадает с параметрами решаемой подзадачи) поле имени решаемой задачи и поле вложенных подзадач (:subtask). При наличии множества подзадач используется поле (:ordering), описывающее предполагаемую последовательность решения подзадач. В большинстве случаев, иерархия выстраивается с помощью вложенности методов.

```
(:method methNameN
  :parameters (?x-param_1 ?y-param_2 ?z-param_3 ?ag-agent)
  :task (tName1 ?ag ?x ?y ?z)
  :subtasks (and
    (task0 (tName1 ?ag ?x ?y))
    (task1 (tName2 ?ag ?y ?z)))
  )
  :ordering (and
    (task0 < task1)))
```

Простейшие действия описываются с помощью соответствующих полей (:action) и по структуре

схожи с действиями классического pddl описания

```
(:action actName
  :parameters (?ag -parameter_1
               ?x -parameter_2
               ?y -parameter_3)
  :precondition
  (and
    (predicate_1 ?x ?y)
    (predicate_2 ?ag)
    )
  :effect
  (and
    (not (predicate_1 ?x ?y))
    (not (predicate_2 ?y))
    (predicate_3 ?ag)
    (predicate_4 ?x ?y)
  )))

```

Задача планирования также, как и в pddl случае состоит из имени, ссылки на домен, перечня объектов и описания начального состояния. Поле целевого состояния заменено на поле (:htn), описывающего подзадачи, решение которых требуется для решения общей задачи.

```
(:htn
  :parameters ()
  :subtasks (and
    (task0 (tName1 ag1 param_1 param_2 param_3))
    (task1 (tName1 ag1 param_1 param_3 param_4))
    )
  :ordering (and
    (task0 < task1)
  )
)
```

Распознаватель иерархической задачи представлен классом «HTNParser», методы которого осуществляют поиск по заранее заданным ключам в файлах задачи и домена. Ключи выделяются с помощью представления файлов задачи и домена в виде стека. Для каждого из ключей вызывается функция компоновки из модуля «branch_parser.py», которая осуществляет поиск по

структуре элемента с помощью регулярных выражений, выявление структурных элементов и их компоновка происходит с помощью ООП подхода. Как в случае классической постановки задачи, так и в иерархическом случае, результатом работы распознавателя является объект класса проблемы планирования.

Означивание классической постановки задачи реализовано функцией «*ground*». Все элементы задачи означиваются в соответствующих функциях, начинающихся с «*signify_**». Для получения прецедентных знаний используется функция «*_update_exp_signs*»(шаг 2 алг.1). В процессе означивания создаются объекты класса «*Sign*» и дополняются каузальными матрицами, с помощью метода «*add_**», где * - имя требуемой сети. Знаки начальной и конечной ситуаций формируются с помощью функции «*_define_situation*»(шаг 12 алг.1). Означивание иерархической задачи происходит подобным образом, знаки элементов задачи синтезируются функциями «*_ground_**», где * - предикаты, действия, методы, предикаты состояний, подзадачи. Результатом алгоритмов означивания является объект класса задачи планирования.

Модуль поиска плана поведения содержит класс «*MapSearch*», методами которого являются метод поиска плана «*search_plan*» и предактивации прецедентов планирования «*precedent_activation*», который вызывает метод рекурсивной итерации для активации целевой ситуации «*_map_iteration*». В методе «*precedent_activation*» вызывается метод «*hierarch_acts*», выполняющий поиск всех предактивированных действий. Метод «*_map_iteration*» вызывает функционал методов «*precedent_search*»(шаги 4-5 алг.2), который осуществляет поиск активных в настоящей ситуации прецедентов планирования, а метод «*_generate_meanings*»(шаг 13 алг. 2) - синтез всех действий. Для синтеза всех возможных действий используется функция «*mix_pairs*», отвечающая за создание пар всех возможных перестановок объектов и ролей в действии. Она выделена в отдельную функцию и не является методом класса «*MapSearch*», так как используется не только в процессе синтеза планов поведения, но и в процессе означивания. В методе «*applicable_search*» осуществляется сортировка и выбор применимых в настоящей ситуации действий, и вызывается метод «*_check_activity*» (шаг 15 алг. 2), осуществляющий сравнение условий (или эффектов при обратном планировании) действия и условий ситуации. Метод «*_meta_check_activity*» (шаг 18 алг. 2) осуществляет эвристический отбор для классической постановки задачи, метод «*_meta_check_htn*» - эвристический отбор для иерархической постановки задачи, метод «*_time_shift_forward*» (шаг 21 алг. 2) создает следующую ситуацию, а метод «*hierarchical_exp_search*» (шаг 20 алг. 2) производит поиск под действий прецедентного действия. Диаграмма основных классов планировщика представлена на рисунке 9.



Рисунок 9. Диаграмма классов алгоритма планирования.

3.2 Структура библиотеки map-multi

Библиотека является реализацией дополнения базовых возможностей агента планирования с ЗКМ, приведенного в библиотеке map-core. Реализовано разделение агентов по функциональным возможностям, создан алгоритм агентной коммуникации. В описании библиотеки присутствуют дополненные и новые классы.

Модуль описания агента планирования «planning_agent.py» пополнился менеджером агентов с расширенным функционалом. Класс «Manager» имеет метод «manage_agents», который осуществляет создание процессов для каждого из агентов планирования, собирает обратную связь от процессов, вычисляет наиболее опытного агента и рассыпает его имя всем агентам. Также, он поддерживает агентный протокол коммуникаций и осуществляет рассылку агентных сообщений. Связь с агентами осуществляется с помощью механизма межпроцессных сообщений (Pipe). Функция «agent_activation» (шаг 1 алг. 7) осуществляет инициализацию класса агента, запускает процессы означивания и поиска плана. Класс агента «MAgent» наследует класс «PlanningAgent», обновленными методами являются метод получения задачи планирования (вызов алгоритмов означивания), метод поиска решений и метод сохранения прецедента планирования. Новым методом является метод создания текстового представления решения. Текстовое представление требуется для передачи сообщений через pipe. Реализован класс «DecisionStrategies», в котором основной функционал представляет методы нахождения консенсусов между агентами. В настоящей реализации представлен метод «auction», осуществляющий аукцион планов.

Расширение поля «:constraints» файлов pddl и hddl задач потребовало адаптации модулей распознавания и модулей означивания. Модули распознавания дополнились блоками работы с ограничениями, а модули означивания реализацией знака «Они» и расширением функций означивания действий. Для pddl постановки были реализованы функции «simple», «specialized» и «nonspecialized» обработки ограничений, которые описывают задачи без ограничений в «simple», с специализированными ограничениями по возможностям классов агентов и по личным агентным возможностям в «specialized», а также специализированные ограничения только по агентным возможностям в «nonspecialized». Первая функция позволяет означивать задачи гомогенный коалиций, вторая предназначена для означивания задач схожих, с доменом «Логистика», в которых агенты имеют общие ограничения по классу (например, грузовики передвигаются только по городу, а самолеты между городами, но не могут выезжать за пределы аэропортов) и по возможностям (какие-то грузовики могут перевозить большие грузы, какие-то маленькие). Третья функция описывает означивание ограничений для задач с разделением

агентов по возможностям. Примером является домен «МирБлоков», в котором агенты перемещают грузы различных масштабов. Для hddl задач была реализована функция «preactivate_actions_nonspecial», которая осуществляет предзаполнение каузальных матриц действий для каждого из агентов. В этой функции, ссылки на знаки ролей заменяются на ссылки на знаки конкретных объектов, с которыми агенты могут взаимодействовать.

В алгоритме поиска плана действий основные изменения были совершены в методе «_generate_meanings». Итеративное заполнение ролей в матрицах действий осуществляется для каждого агента раздельно и происходит проверка на предзаполненность ролей объекта. Проверка позволяет избегать заполнения нескольких ролей в процессе синтеза действий ссылкой на знак одного объекта.

3.3 Структура библиотеки map-spatial

Библиотека осуществляет взаимосвязь планирования поведения агента в двумерном пространстве и классического планирования. Модули библиотеки реализуют распознавание пространственной задачи, алгоритм пополнения знаний агента, вызов библиотек map-core и map-multi для создания субоптимального абстрактного плана действий и расширение каждого из абстрактных действий пространственной реализацией.

Процесс синтеза плана поведения начинается с вызова метода «`_parse_spatial`» объекта класса «`MapPlanner`», который наследует функционал класса базового планирования «`MPcore`». Распознавание задачи планирования и формирование объекта класса проблемы происходит в модуле «`spatial_parser.py`». В модуле «`json_grounding.py`» происходит означивание проблемы планирования, основными функциями являются функции означивания предикатов «`_ground_predicates`» (шаги 1-5 алг. 4), функции означивания действий «`_ground_actions`» (шаг 10 алг. 4), формирования координат клеток и областей функцией «`sign_markup`» (шаги 16, 18 алг. 4), формирования описания состояния агента функцией «`state_prediction`», формирования ситуаций «`define_situation`» (шаги 17, 19 алг. 4) и формирования каузальных матриц со списком координат клеток и областей функцией «`state_fixation`», формирования карты «`define_map`» (шаги 17, 19 алг. 4). Так как эти функции требуются не только в алгоритме означивания, но и при любом формировании пространственных ситуаций, они описаны в модуле «`utils.py`». Также, этот модуль содержит второстепенные функции, необходимые для осуществления синтеза пространственного плана поведения и создания семиотической сети, в число которых входят функция сравнения двух пространственных ситуаций «`sit_similarity`», сравнения пространственных действий «`act_similarity`», поиска требуемых знаков в списке событий «`search_cm`» и «`prm_parser`», получения целевых координат агента, исходя из направления до объекта и его координат «`get_goal_coords_by_block`», вычисления координат клеток и областей «`locater`», прогнозирования координат клеток и областей «`size_founder`», отношений между клеткой и областями «`belonging`», обновленного механизма синтеза знаков коммуникаций «`signify_connection`», принадлежности клетки какой-либо области «`adjoints`», создания описания состояния клеток «`cell_creator`» (шаг 6 алг. 5 и т.д.), сравнения пространственных каузальных матриц «`compare`» и «`resonated`», получения требуемых каузальных матриц «`get_attributes`», определения местоположения областей «`get_reg_location`», формирование описания областей «`get_struct`», обновления каузальной матрицы ситуации «`update_situation`», создания структуры подпланов одинакового уровня уточнения функцией «`tree_refinement`» и т.д.

Также, как и в предыдущих версиях, модуль «planning_agent.py» содержит описание класса агента «SpAgent», класса агентного менеджера и функцию «agent_activation». Получение описания объекта из действий классического представления осуществляется функцией «get_conditions». Функционал менеджера расширен механизмом рассылки агентам информации о завершении деятельности агентов по синтезу подпланов для каждого из абстрактных действий. Функционал агента дополнен методами «get_scenario», «get_spatial_sit_blocks», «copy_action», «load_subtask», «change_start», «get_spatial_finish_blocks» и методами вычисления требуемых координат агента для взаимодействия с объектом с каждой из сторон.

Метод «get_scenario» осуществляет вызов библиотек непространственного планирования, далее, абстрактный план передается вложенной функции «get_subplans» для выявления последовательности действий каждого из агентов. После этого, для каждого из действий формируется начальная и конечная пространственные ситуации методом «get_spatial_sit_blocks» и добавляются в вырожденный сценарий деятельности.

После синтеза сценария, для каждого из действий в методе «agent_activation» происходит вызов метода «change_start», который корректирует начальную ситуацию сценария, с помощью информации из целевой ситуации предыдущего пространственного подплана. Далее, агент синтезирует план методом «search_solution», основной функционал которого заключается в изменении параметров задачи планирования после синтеза пространственного подплана. После этого формируется сообщение о завершении планирования и пересыпается агенту-координатору. После получения сообщения от агента-координатора об окончании планирования всеми агентами, вызывается метод «save_signs» класса «SpTask». Методы класса задачи планирования «SpTask» позволяют создавать новую ситуацию по описанию «create_situation», выделять подпланы в методе «create_subplan» (шаг 13 алг. 6), сохранять план в методе «save_plan».

Модуль синтеза пространственных планов состоит из описания класса «SpSearch», который наследует функционал базового класса «MapSearch» и имеет рекурсивный метод поиска плана «_map_sp_iteration», активации карты и ситуации «state_activation», иерархического пространственного поиска подплана пространственного прецедента «hierarchical_exp_sp_search», проверки применимости действия при планировании от цели к настоящей ситуации «_check_activity_backward_spat», от настоящей ситуации к цели «_check_activity», вызов которых присутствуют в «applicable_sp_search», создания новой начальной ситуации методом «combine_situation», абстрагирования ситуации «abstract_search», который вызывает метод «abstract», уточнения ситуации «clarify_search», который вызывает метод «devide_situation», обновленной генерации действий «_generate_meanings», эвристической проверки действия в пространстве «_sp_check_activity», функций сравнения клеток

«cell_closer», «linear_cell» и «get_stright», проверки размерности клеток в прецеденте планирования «scale_history_situation», активации ситуации после добавления в план прецедентного действия «history_action», активации ситуации после синтеза нового действия «new_action», активации ситуации после включения в план знака подплана «sub_action», адаптирования прецедентной ситуации «adapt_exp_situation», синтез следующей ситуации «_state_prediction» и «_step_generation», формирование следующей ситуации без сетевых ограничений «_time_shift_spat», изменения карты «change_map», методы вызова эвристики тактического уровня алгоритма A* «_get_tactical» и «ASearch», метод вызова жадной эвристики «greedy_search». Диаграмма основных классов библиотеки приведена на рисунке 10.

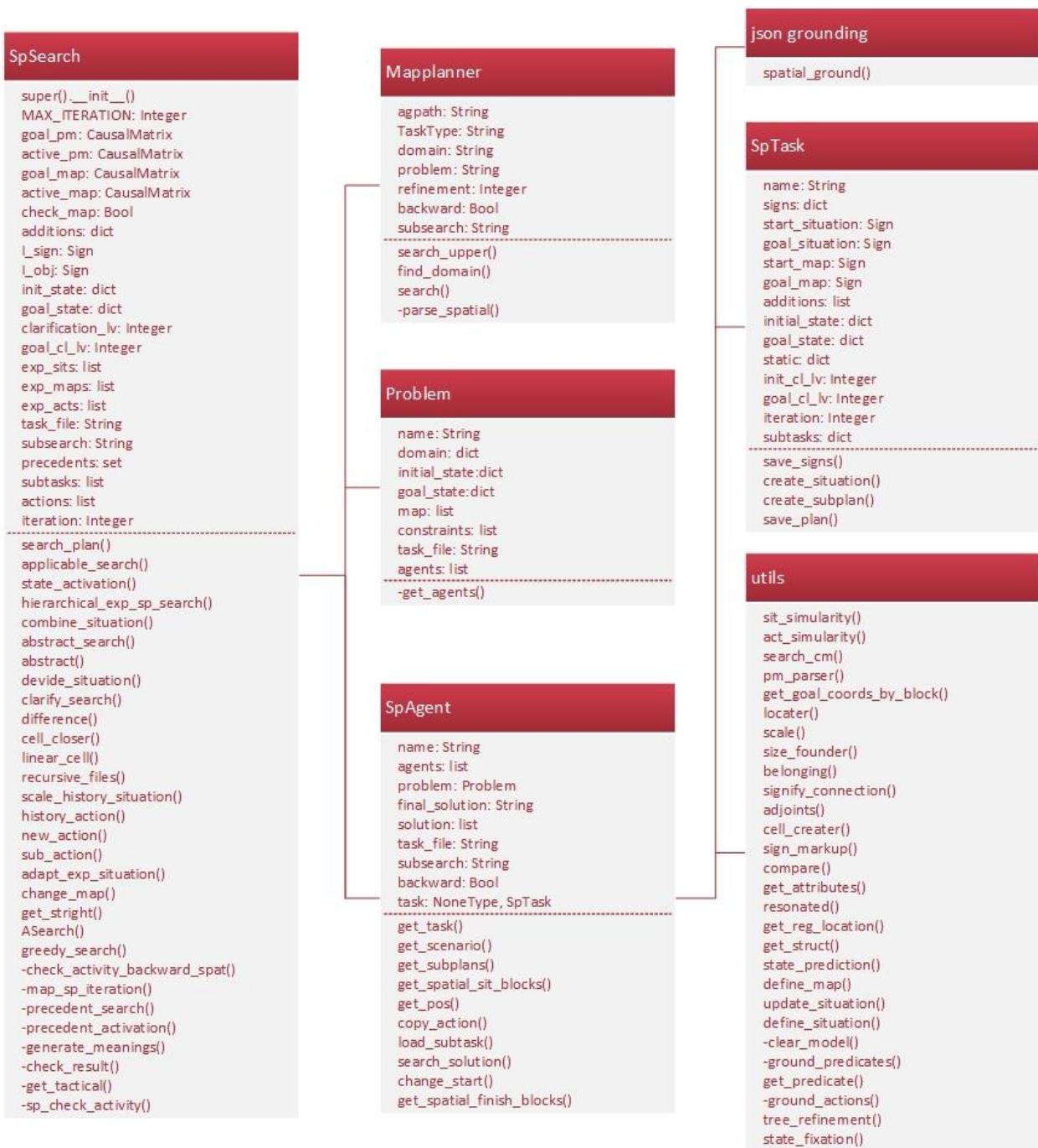


Рисунок 10. Диаграмма классов основных библиотеки map-spatial.

3.4 Экспериментальные исследования

3.4.1 Классические, иерархические и пространственные эксперименты

В процессе создания диссертации было создано несколько версий рассмотренных библиотек. В этом параграфе будут изложены основные эксперименты, которые с ними производились.

Одна из версий библиотеки map-multi была создана для решения задач, связанных с доменами «Мир блоков» и «Логистика». Домен «Мир блоков» достаточно подробно описан ранее, поэтому приведем описание поведения агентов во втором домене. Перемещение грузов осуществлялось в грузовиках по требуемым местам в городе (склады, аэропорты), между городами грузы перевозили самолеты. Возможности агентов были ограничены соответствующими ролями, например, самолеты могли перевозить грузы между городами, но не могли перемещаться внутри городов, а грузовики, наоборот. Помимо этого, каждый из типов агентов подразделялся на подтипы – какие-то агенты могли взаимодействовать с тяжелыми грузами, какие-то с легкими и т.д. Далее представлены таблицы 3 и 4, в которых рассмотрены результаты экспериментов в домене «Мир блоков» в таблице 3 и результаты в домене «Логистика» в таблице 4.

	Кол-во агентов	Кол-во блоков	Кол-во типов блоков	Кол-во типов блоков для агента	Исп-ие опыта	Время для синтеза плана	Кол-во памяти для синтеза плана
Задача 1	3	6	3	2	Нет	8950	38,6
Задача 2	3	6	3	2	Да	17	33,3
Задача 3	3	8	3	2	Да	6400	39,9

Таблица 3. Эксперименты в многоагентном домене «Мир блоков».

	Кол-во агентов «грузовик»	Кол-во агентов «самолет»	Кол-во грузов	Кол-во типов грузов	Длина плана	Время для синтеза плана	Кол-во памяти для синтеза плана
Задача 1	2	1	4	2	40	4040	62,2
Задача 2	4	1	4	2	184	16000	180,4
Задача 3	3	0	6	3	40	5020	69,5

Таблица 4. Эксперименты в многоагентном домене «Логистика».

Результаты экспериментов заключаются в подтверждении тезиса о значительном снижении времени, затраченного агентом на синтез плана поведения при использовании прецедентных данных и о возможности адаптации алгоритма для решения задач на мобильных компьютерах

из-за невысоких требований к памяти устройства. Более подробно рассмотренные эксперименты описаны в [1].

При разработке иерархической версии классического планирования были учтены основные возможности современных планировщиков. В [5] представлено сравнение MAP библиотек с другими планировщиками (MIDCA [104], PANDA [105], HEART [106], eBT+SkROS [107]) основные пункты сравнения представлены в таблице 5.

Алгоритм	Поддержка PDDL	Решение задач без вложенных методов	Решение задач с вложенными методами	Поддержка ROS	Решение многоагентных задач	Исп. простр. характеристик	Исп. прецедентов
MIDCA	-	+	-	+	-	+	-
PANDA	+	+	+-	-	-	-	-
HEART	-	+	+	-	-	-	-
eBT+SkROS	+	+	+	+	-	+	-
MAP	+	+	+	+	+	+	+

Таблица 5. Сравнение возможностей современных планировщиков.

Также, на рисунке 11 представлен график зависимости времени от сложности задач при планировании с использованием планировщиков MAP, PANDA и MIDCA. Планировщики для сравнения были выбраны исходя из критериев популярности и возможности унифицированного представления задачи планирования. MAP (бпр) – без наличия прецедентов, MAP (спр) – имеются прецеденты частичного решения задачи.

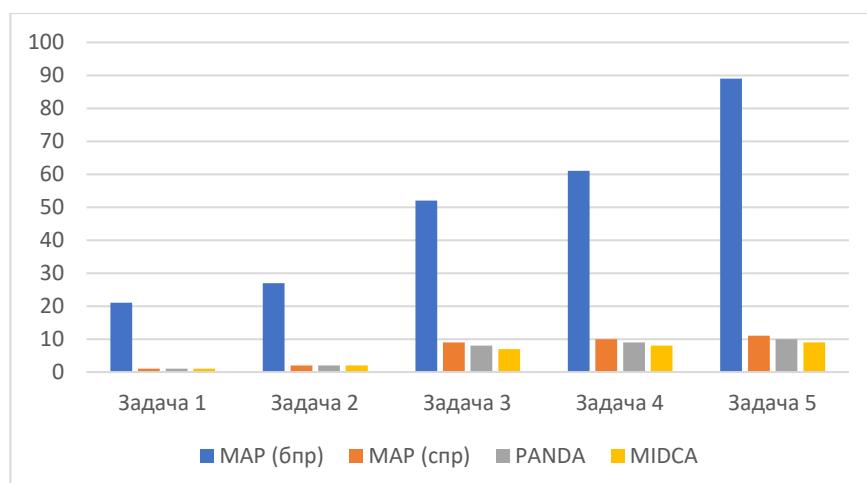


Рисунок 11. Диаграмма зависимости увеличения времени от сложности решения иерархических задач.

Условия рассматриваемых задач перечислены далее:

1. Перемещение из одного конца карты в другой. Объезд препятствия в виде стола.
2. Перемещение из одного конца карты в другой. Объезд множественных препятствий.
3. Выезд из узкого коридора в большую комнату. Наличие препятствий в виде стен.
4. Проезд через узкий коридор и заезд в маленькую комнату. Наличие препятствий в виде множества стен.
5. Перемещение блока с одного стола на другой. Движение с объектом.

Время, затраченное МАР планировщиком больше, чем время, затраченное другими планировщиками. МАР планировщик искал маршрут агента, основываясь на пространственных знаниях об окружающей среде и синтезировал более детализированный план. Например, план остальных планировщиков включал действия наивысшей степени абстракции:

«Проехать из комнаты A в комнату B»; «Объехать стол»; «Проехать коридор».

А план МАР планировщика имел структуру перемещения между промежуточными точками:

«Проехать вперед»; «Уточнить»; «Проехать вперед»; «Повернуть направо»; «Проехать вперед» «Поднять блок B».

При рассмотрении наличия частичного опыта (прецедент полного решения отсутствовал), МАР планировщик синтезировал план за время, сравнимое с планировщиками PANDA и MIDCA и план МАР планировщика может быть исполнен без дальнейшей адаптации, так как каждое из действий имело точные координаты перемещений. Если сравнивать затраты памяти на синтез плана (рис. 12), то планировщик МАР имеет преимущество перед планировщиком PANDA, а также, в МАР поддерживается многоагентная постановка задачи планирования для иерархических задач.

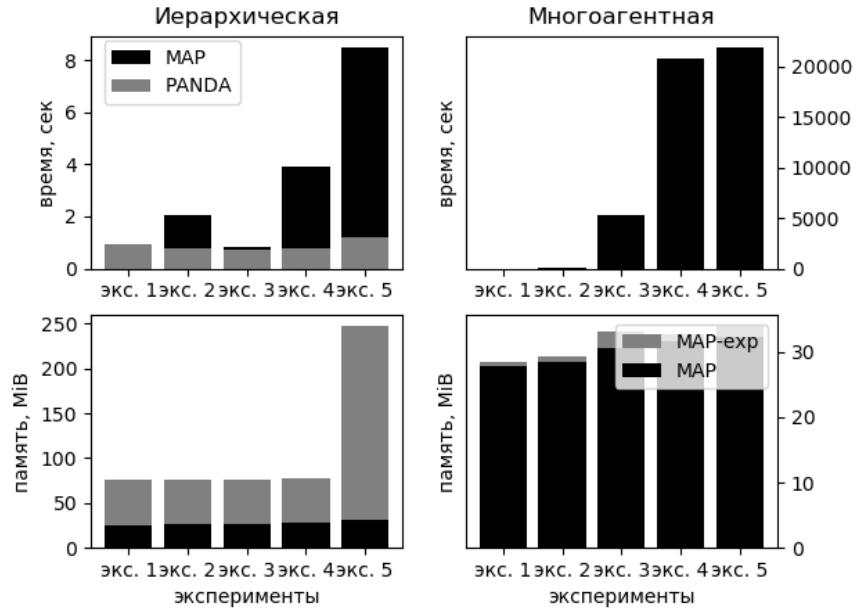


Рисунок 12. Сравнение планировщиков MAP и PANDA в задачах без учета пространственных данных.

3.4.2 Эксперименты по адаптации MAP к робототехнической платформе

В статье [4] описан пример взаимодействия пространственного планировщика map-spatial и on-policy алгоритма обучения с подкреплением. Одним из представленных в статье результатов было создание искусственной среды обучения для манипулятора, а также разработка механизма взаимодействия алгоритма планирования поведения и алгоритма обучения с подкреплением в рамках синтеза управления роботизированной платформой в среде Gazebo. Роботизированная платформа перемещалась по карте без препятствий к столу, на котором находились блоки, далее осуществлялось распознавание блоков с камеры платформы и установление их относительного местоположения. После этого, алгоритм обучения с подкреплением синтезировал управление манипулятором, который взаимодействовал с блоками (см рис. 13) согласно требованиям задачи.

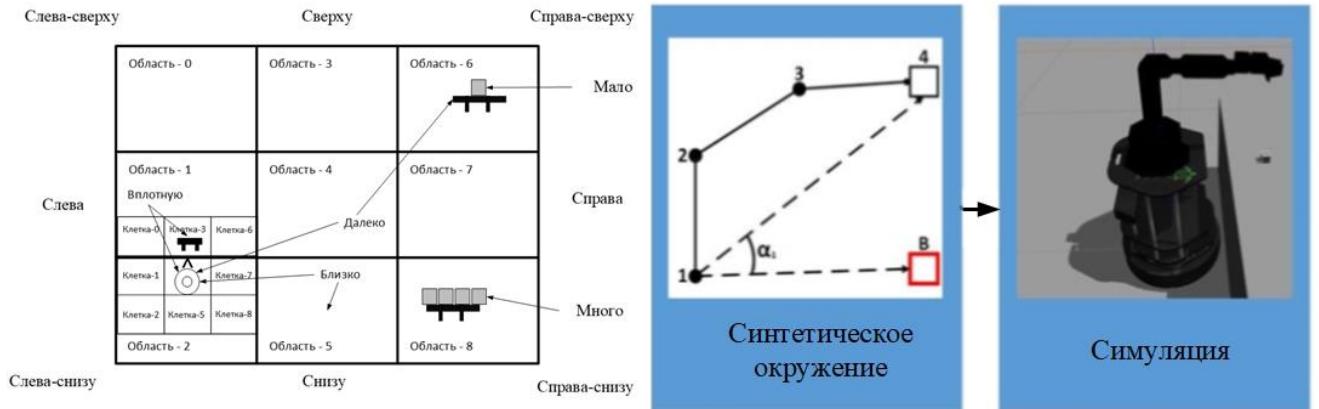


Рисунок 13. Взаимодействие пространственного планирования и алгоритма обучения с подкреплением для управления роботизированной платформой.

На момент создания [4], перемещение агента происходило только с использованием жадной эвристики поиска пути, с помощью которой, планировщик был не способен синтезировать план действий на карте с большим количеством препятствий, а также с узкими и длинными препятствиями. В статье [11] рассмотрен пример улучшения работы алгоритма, с помощью использования A^* эвристики поиска пути. Были решены задачи, отображенные на рисунке 14.

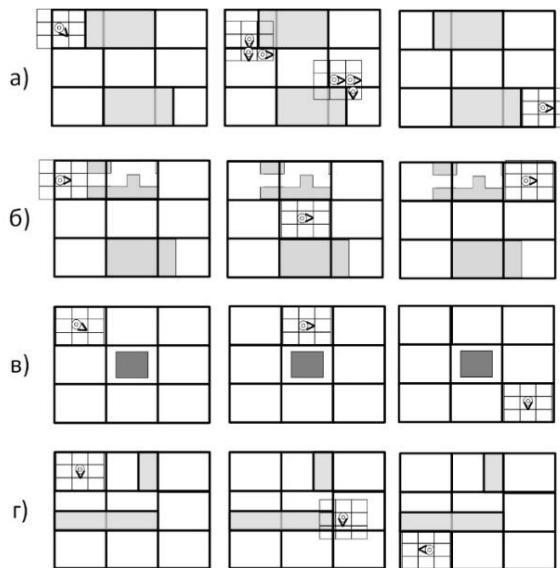


Рисунок 14. Тестирование A^* эвристики.

Также, немаловажным фактором работы с реальной робототехнической платформой является нахождение «гладкой» траектории перемещения платформы, физические данные которой не позволяют совершать резкий поворот на месте, в процессе осуществления перемещения. Усовершенствование алгоритма планирования A^* эвристикой позволило обходить острые углы препятствий без соприкосновений, что способствовало повышению уровня безопасности использования платформы.

Далее был усовершенствован способ взаимодействия агента и окружающих объектов с помощью добавления возможности автоматизации вычисления состояния сервоприводов робототехнической платформы off-policy алгоритмом обучения с подкреплением (Q-обучение [108-110]). В процессе его реализации созданы синтетические среды «Мир блоков» для перемещения платформы и «Манипулятор» для взаимодействия с объектами окружающей среды. Взаимодействие алгоритма планирования поведения и алгоритма обучения осуществлено с помощью создания модулей распознавания, осуществляющих поиск терминальных состояний среды для каждого из синтезированных действий планировщика и их преобразование в состояния среды обучения. Результатом деятельности алгоритма обучения является наполнение нижнего уровня иерархии плана поведения когнитивной архитектуры STRL атомарными действиями (см. рис. 15) и создание оптимальной стратегии синтеза действий.

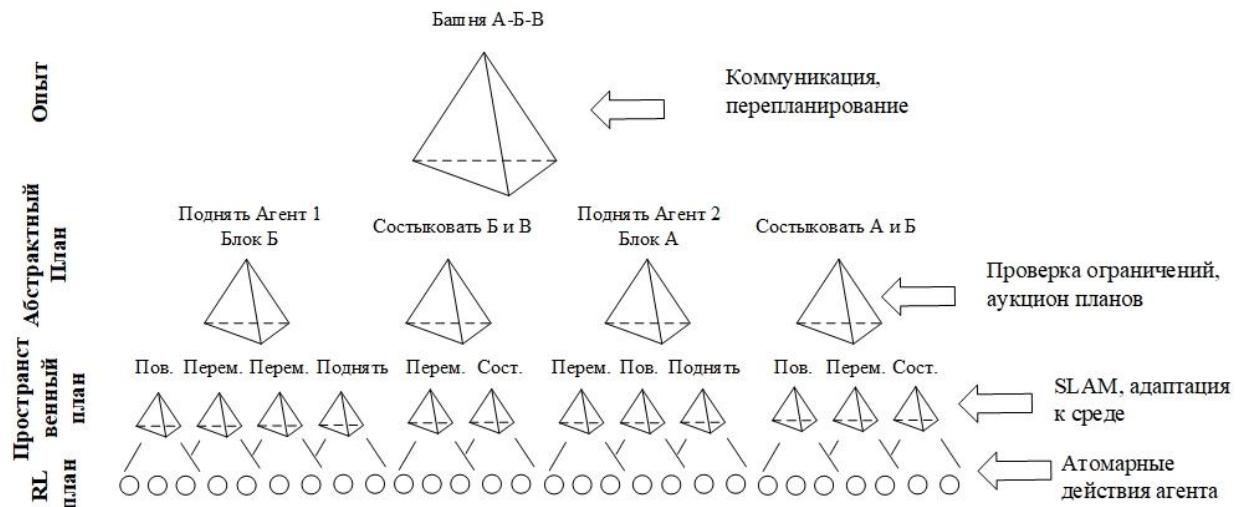


Рисунок 15. Иерархия действий робототехнического агента.

Использованный метод Q-обучения является табличным методом обучения, таблица которого описывает награду за выполнение различных действий в состояниях среды (Q-значения). Целью алгоритма является формирование таблицы, которая будет характеризовать оптимальный выбор действий в каждом состоянии. Q-значения таблицы находятся по формуле: $Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$, α - скорость обучения, а $\max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$ - Q-значение наилучшего действия в момент времени $t+1$. Для описания действий манипулятора, было использовано DQN представление Q-learning алгоритма с параметризацией, в котором таблица значений заменена на нейронную сеть. Выбор действия осуществляется аналогично не параметризованному Q-learning алгоритму, далее происходит сохранение кортежа (семпла) значений (s_t, s_{t+1}, a, r) , из памяти выбираются случайные семплы и вычисляется $\max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$, которое передается в нейронную сеть в качестве эталонного

значения. Алгоритм предобучается на размеченных с помощью планировщика задачах и формирует стратегию, которая используется в тестовых итерациях.

Процесс адаптации спланированных действий выполняется с помощью размечивания каждого из них и выявления подпространств 30x30 клеток, размер которых был получен эмпирическим путем. Процесс обучения агента происходит в среде, характеристики которой задаются динамически для каждого из подпространств и обучение в них распараллелено. Подпространства обеспечили стабильный синтез атомарных действий, в число которых были включены действия по перемещению на 1 клетку в любую сторону, поднятие и состыковка блоков окружения «Мир Блоков».

Среда «Мир Блоков» не входит в перечень доступных для OpenAI gym базовых сред и была создана с помощью обновления среды «Мир Клеток» дополнительными действиями. Среда позволяет оценивать действия агента и делать выводы приблизился ли агент к поставленной цели.

В среде «Манипулятор» положение агента рассматривается агентоцентрировано, каждое из действий поворота агента совершается для взаимодействия с объектом окружающей среды. Количество возможных атомарных действий агента увеличивается действиями изменения состояния сервоприводов.

При инициализации новой задачи, агент получает информацию о положении манипулятора и целевого объекта среды в 3D пространстве. Далее, агент ищет возможные действия на основе функции вознаграждения `tolerance` (`x, bounds, margin, value_at_margin`) из DeepMind Control Suite [111]. Эта функция позволяет агенту получать вознаграждение 1 при вхождении расстояния от актуатора робота до требуемого объекта среды в рамки ограничений. Эмпирическим путем была выявлена проблема синтеза действий с функцией `tolerance`, заключающаяся в неоптимальном выборе последовательности под действий агента, из-за увеличения агентом получаемой награды, с помощью деятельности в пространстве, входящем в ограничения, но не осуществляющей захват целевого объекта.

Были проведены синтетические эксперименты (см. рис. 16). Первые 5 экспериментов проведены для обучения агента и выработке стратегии взаимодействия со средой. Последний, 6 эксперимент, проводился в рамках тестирования полученного результата.

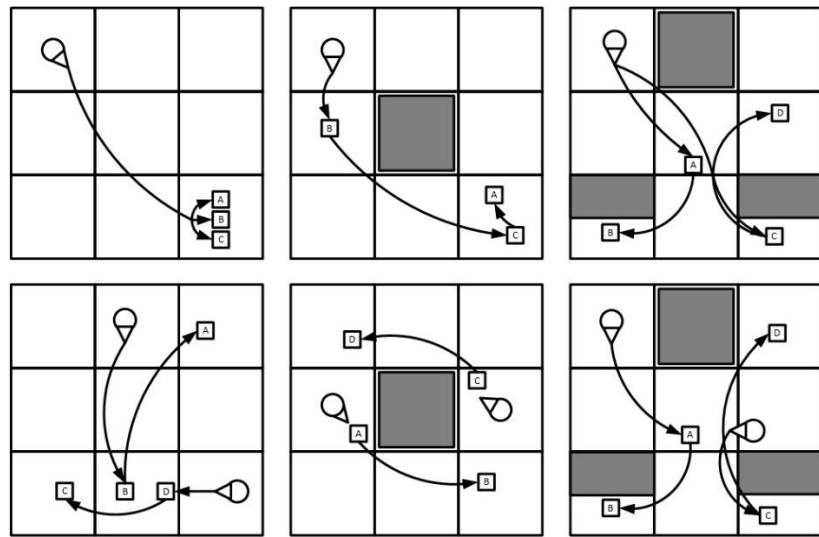


Рисунок 16. Тестирование алгоритма поиска под действий.

Планировщик синтезировал планы поведения, используя агентов без общей памяти. Каждый из агентов был выделен в отдельный процесс, агенты имели общие знания о возможностях других агентов, но не имели детализированного представления пространственных переходов. В процессе уточнения действий агентом, каждое из действий было уточнено самостоятельно, но веса нейросети формировались на основе действий всех агентов. Разделение набора данных для обучения различными агентами отсутствовало, исходя из механической однотипности рассматриваемых задач.

Эксперименты были проведены на ноутбуке с характеристиками: Intel Core i7-8565U CPU 1.8 GHz 199 GHz RAM 16Gb. Оценка результатов проводилась по времени исполнения задачи, количеству шагов планировщика, количеству найденных алгоритмом Q-обучения под действий, количеству под действий манипулятора, найденных DQN. Результаты приведены в таблице 6.

	Одиночная задача 1	Одиночная задача 2	Одиночная задача 2	Многоагентная задача 1	Многоагентная задача 2	Многоагентная задача 3
Время (сек)	22.4	112.5	263.4	283.03	526.5	617.5
Шаги планировщика	6	12	39	11	31	35
Шаги Q-обучения	445	412	640	430	452	608
Шаги DQN	400	900	700	400	900	700

Таблица 6. Результаты экспериментов поиска под действий.

Помимо вышеперечисленных экспериментов, были проведены эксперименты с дополнением планировщика логическим выводом, усовершенствование иерархической структуры и другие эксперименты, более подробно изложенные в [1-12].

3.5 Выводы главы 3

В главе представлена программная реализация описанных ранее алгоритмов в виде библиотек map-core, map-spatial и map-multi. Структура библиотек позволяет синтезировать комплексный иерархический план, который был экспериментально оценен и сравнен с аналогами на каждом из этапов уточнения. Разбиение шагов планировщика на последовательности элементарных действий произведено с помощью алгоритма обучения с подкреплением в разработанных автором средах «Мир блоков» и «Манипулятор». Все библиотеки позволяют синтезировать план как самостоятельно, так и с помощью последовательного вызова более абстрактных планировщиков. Например, при синтезе элементарных действий алгоритмом обучения с подкреплением, происходит вызов библиотеки map-spatial для синтеза пространственного плана, которая в свою очередь вызывает библиотеки map-multi и map-core.

Глава 4

4 Робототехническая реализация

4.1 Общее описание системы

Приведенный выше алгоритм планирования был адаптирован для решения робототехнических задач по перемещению объектов в специализированном полигоне. Для реализации целенаправленного поведения платформы была создана модульная версия архитектуры STRL, которая позволила абстрагировать функционал тактического и реактивного уровней до соответствующих логических модулей (см. рис. 17).

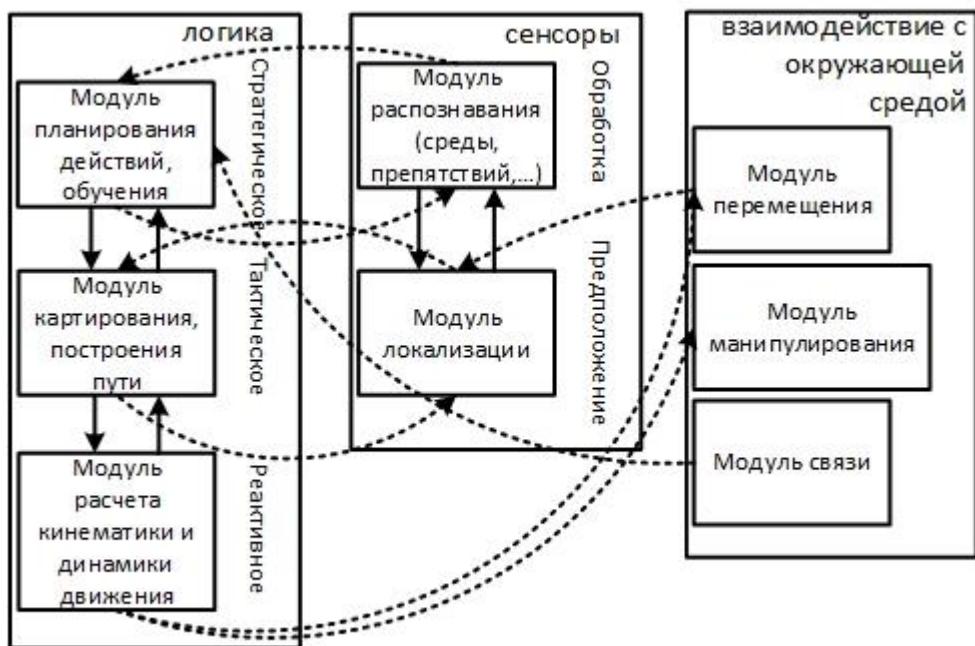


Рисунок 17. Схема робототехнической реализации.

Произведено классическое для робототехнических архитектур разделение уровней обработки информации, которое позволило взаимодействовать со средой и оценивать результаты деятельности. Адаптация архитектуры осуществлена для робототехнической платформы МП-РМ. Во время адаптации выявлен ряд проблем при использовании стандартных модулей и проведена работа по их устранению [2]. Для реализации логических модулей выбрана робототехническая архитектура ROS. Для примера строения логического модуля рассмотрим модуль планирования поведения (см. рис. 18).

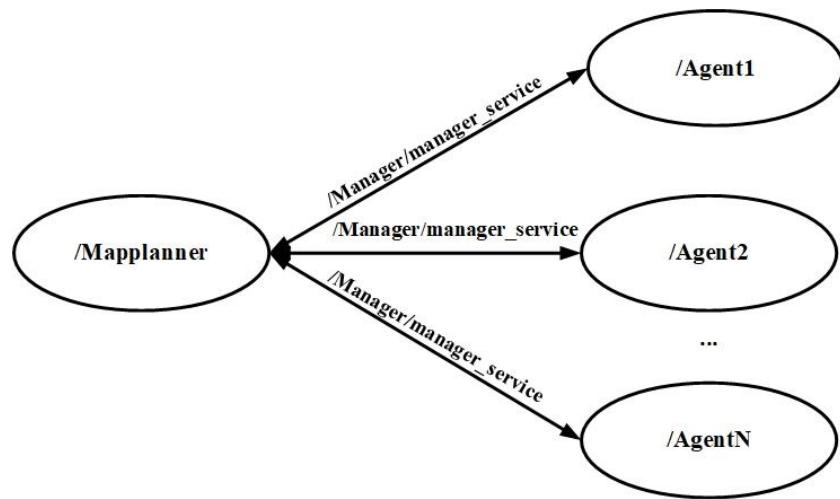


Рисунок 18. Диаграмма ROS-узлов в модуле планирования поведения.

Робототехническая архитектура ROS является набором инструментов, библиотек и правил взаимодействия с робототехническими устройствами. ROS является унифицированным методом создания структуры проекта, связанного с робототехникой и позволяет разрабатывать каждую из частей проекта отдельно, связывая эти части шаблонизированными способами. Архитектура разрабатывается компанией Open Robotics, но в репозиториях, связанных с архитектурой находится большое количество модулей с открытым кодом и сторонних решений. Основными сущностями, связанными с архитектурой, являются:

1. Мастер узел;
2. Узлы;
3. Пакет;
4. Сообщение, топик, издатель, подписчик;
5. Сервис, клиент и сервер сервиса;
6. Действие.

Мастер узел выполняет роль сервера имен, который позволяет связывать между собой другие узлы архитектуры и производить обмен сообщениями между ними. Запуск мастера производится в отдельном от остальной программы процессе с помощью команды *roscore*. После запуска мастера активируется переменная окружения *ROS_MASTER_URI*, содержащая ссылку на подключение к мастеру. Подключение возможно производить на удаленных компьютерах с помощью установления IP адреса мастера и команды:

```
export ROS_MASTER_URI=http://IP-adress:port/
```

Узлы являются исполняемыми программами архитектуры. Каждый из узлов решает узкоспециализированную задачу. Решение комплексных задач происходит с помощью связи

узлов сообщениями. Адреса узлов хранятся в переменной окружения ROS_HOSTNAME. Имена узлов регистрируются в мастер-узле, но сообщения между узлами пересыпаются без его участия. При запущенном мастере, список всех узлов отображается с помощью консольной команды *rosnode list*. Также, активность узла проверится командой *rosnode ping NodeName*. Для запуска узла требуется произвести консольную команду *rosrun package_name node_name*. В контексте настоящей диссертации, каждый из агентов запускается в отдельном узле, что обеспечивает установление надежного протокола связи между ними и подчеркивает инкапсуляцию знаний каждого из агентов.

Пакеты являются более глобальной единицей архитектуры ROS и требуются для решения целого ряда задач. В настоящей диссертации рассмотрен пакет планирования поведения, который реализует весь функционал планирования и позволяет устанавливать его отдельно от остальной системы. Инкапсуляция технических функций в пакеты является распространенной практикой в робототехнике и позволяет находить и устанавливать именно тот пакет, который точно реализует требуемое поведение. Создание пакета происходит с помощью команды

```
catkin_create_pkg <package_name> [depend1] [depend2] [depend3],
```

где зависимостями могут выступать ссылки на библиотеки такие, как *std_msgs*, если предполагается рассылка сообщений между узлами пакета, *rospy* – если разработка узлов будет вестись на языке Python, *roscpp* – если разработка на языке C++ и другие библиотеки, зависимости к которым можно добавить в файле *package.xml* пакета. Для взаимодействия с пакетами, разработчики архитектуры ROS предлагают использовать команду *rospack*. Полный список атрибутов этой команды можно найти на сайте разработчиков [59].

Как упоминалось ранее, узлы архитектуры связываются сообщениями. У каждого сообщения присутствует типизированное описание переменных, где базовыми типами являются типы языка C++. Комплексные типы можно создавать вручную. Сообщения рассыпаются между узлами с помощью топиков, сервисов и действий. Механизм топиков заключается в регистрации имени топика на мастер-узле и публикации сообщений в топик по примеру доски объявлений. Подписчики считывают сообщения из топика и выполняют функции с ними связанные. Каждый из ROS-узлов может иметь несколько процессов издателей и несколько процессов подписчиков. Топики применяются для считывания данных с сенсоров, регламентирования деятельности группы агентов и т.д. Эта способ связи является односторонним, и издатель не нуждается в данных об активности подписчиков. Для работы с топиками разработчиками создана команда *rostopic*, которая принимает в качестве аргументов *list* – для выявления всего списка доступных топиков, *echo* – для просмотра топика, *pub* – для публикации в топик и т.д.

Сервисы и действия являются двунаправленными способами связи. Сервисы позволяют устанавливать синхронное взаимодействие процессов сервера и клиентов, в котором клиенты создают запросы на сервер и получают соответствующие ответы. Также, как и для остальных сущностей ROS архитектуры, для обработки сервисов присутствует команда `rosservice`, которая принимает аргументы (`list`, `type`, `find`, ...). Действия являются механизмом асинхронной двунаправленной связи. Они позволяют создавать задачу для выполнения клиентами длительных функций с фиксацией промежуточных результатов. Действия используются для обработки перемещений робототехнических платформ, при которых серверу важно знать пройденные платформой точки. Структура действий описывается в файлах с расширением «.action» и содержит определение цели действия (аналогична запросу ROS-сервиса), результата действия (аналогичен ответу ROS-сервиса) и промежуточным результатам.

Модуль планирования поведения выполнен в виде ROS-пакета, в котором осуществлена реализация ROS-узлов каждого из агентов планирования и управляющего узла «Mapplanner». Управляющий узел может быть запущен на любом из робототехнических агентов. Его основной функционал позволяет распознавать задачу планирования, выявлять необходимых для синтеза плана агентов и распределять задачи между агентами. Каждый из узлов агентов запускается средствами ROS на выделенных устройствах и ожидает запуск управляющего узла. Соединение устанавливается с помощью получения `ROS_MASTER_URI` по `tcp/ip` протоколу. Связь между управляющим узлом и узлами агентов реализована с помощью сервиса «`manager_service`», который осуществляет классическую для ROS реализацию соединения. Каждый из агентов следует протоколу запросов, которые отличаются уникальными ключами. В список ключей входит запрос для получения информации об участии агента в акте планирования «`requirements_name`», запрос для получения информации об имени главного агента «`requirements_major`», запрос ожидания вырожденного сценария подплана «`requirements_sub`», запрос предоставления подплана по вырожденному сценарию «`requirements_solved`», запрос целевого решения «`requirements_wait_sol`» и запрос предоставления целевого решения «`requirements_solution`». Главный, наиболее опытный агент осуществляет синтез вырожденного подплана, вызывая библиотеки «`mapcore`» и «`mapmulti`», а также осуществляя синтез пространственных ситуаций. Далее, каждая часть вырожденного сценария передается, с помощью управляющего узла, соответствующему агенту, который публикует подплан, с помощью запроса «`requirements_solved`». После окончания сбора всех подпланов, главный агент публикует решение с помощью «`requirements_solution`», которое рассыпается всем агентам, в ответ на запрос «`requirements_wait_sol`».

4.2 Перемещение робототехнической платформы

Робототехническая платформа МП-РМ (см. рис. 19) оборудована двигателями постоянного тока, взаимодействие с которыми было установлено с помощью библиотеки GPIO. Функционал библиотеки позволяет управлять модулем драйвера L298N. Интерфейс библиотеки передает сигналы «Высокий» и «Низкий» на сервоприводы, регулировка скорости вращения моторов производится с помощью ШИМ (Широтно-импульсная модуляция) сигнала. Перемещение платформы осуществляется по принципу выбора направления вращения колесной базы, управление которой абстрагировано до функций, описанных в модуле «robot/utils.py». Также, в этом модуле присутствует функциональная адаптация взаимодействия с манипулятором МП-РМ, с помощью библиотеки Adafruit_PCA9685, которая регулирует подачу ШИМ сигнала на сервоприводы манипулятора. Набор функций «MotorAB_Brake», «MotorAB_Direction_Forward», «MotorAB_Direction_Backward», «MotorAB_Direction_Right», «MotorAB_Direction_Left» принимают в качестве аргументов скорость вращения сервоприводов, функции «MotorAB_Direction_Suck», «MotorAB_Direction_Unsuck» логическое состояние вакуумной помпы робототехнической платформы. Каждая из функций вызывается на промежуток времени, достаточный для перемещения робототехнической платформы в требуемое положение. Время перемещения робототехнической платформы на одинаковое расстояние по разным поверхностям отличается из-за различной силы трения колес о материал пола и проскальзывания колесной базы. В экспериментах было использовано время, полученное эмпирически. В процессе эксплуатации помпы робототехнической платформы была установлена недостаточность её мощности, и она была заменена на электромеханический захват. Взаимодействие робототехнической платформы и объектов среды было осуществлено с помощью реализаций функций «MotorAB_Direction_Pickup» и «MotorAB_Direction_Stack», которые последовательно вызывают изменение состояния сервоприводов до необходимого для взаимодействия. Требуемые состояния сервоприводов манипулятора были заранее заданы.



Рисунок 19. Платформа МП-РМ.

Платформа МП-РМ оборудована сервоприводами без обратной связи, также отсутствует возможность отслеживания GPS-сигнала для контроля местоположения платформы. При перемещении по различным типам поверхностей происходило накопление ошибки, которое скомпенсировано с помощью реализации ROS-узла распознавания aruco-меток (тип QR-кодов, используемый в робототехнике). Aruco-метки были закреплены на потолке полигона и реализовали значения точных координат платформы в зашифрованном виде (см. рис. 20). Для их распознавания структура МП-РМ была дополнена дополнительной платой Raspberry Pi 3 и дополнительной камерой.

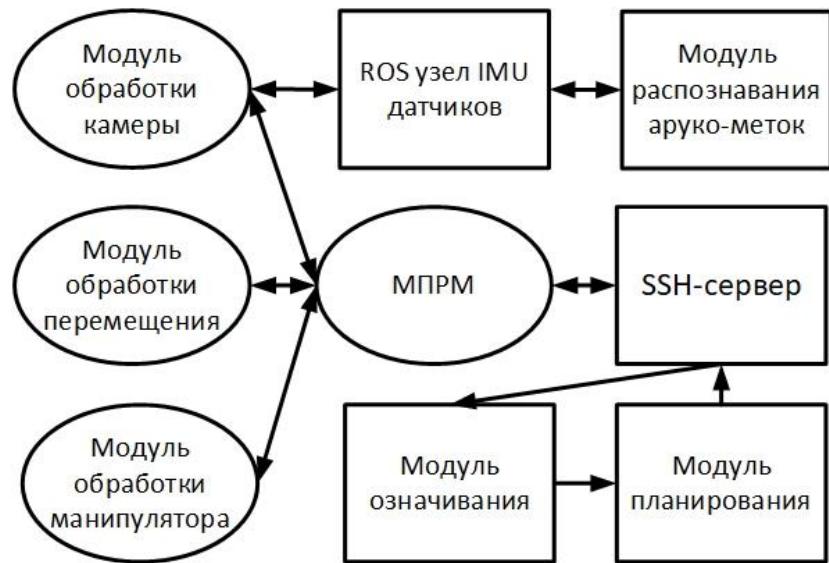


Рисунок 20. Схема робототехнической платформы.

Также были использованы функции распознавания QR-кодов с фронтальной камеры платформы, IMU датчики наклона и поворота. Для установления требуемого формата координат

платформы и их взаимосвязи с спланированными, а также для запуска функций перемещения платформы был разработан модуль SSH-сервера.

4.3 Вывод главы 4

В процессе адаптации алгоритма планирования поведения для робототехнической платформы была разработана реализация архитектуры STRL, позволяющая взаимодействовать с объектами окружающей среды. Был выявлен ряд недостатков имеющегося оборудования и произведена деятельность по их программному и инженерному компенсированию. Успешно проведен ряд экспериментов, подтверждающих возможность следования плану поведения агента со ЗКМ для реализации задач перемещения объектов в условиях реальной среды. Была установлена проблема формирования последовательности атомарных поддействий для точной реализации абстрактных действий алгоритма планирования поведения. В качестве примера решения проблемы была предложена реализация алгоритма обучения с подкреплением и проведены эксперименты в синтетических средах.

Заключение

В работе были получены следующие основные результаты.

1. Разработан иерархический алгоритм планирования поведения агента со знаковой картиной мира. Результаты проведенных с алгоритмом экспериментов подтвердили преимущества семиотического представления знаний агента, которые заключаются в возможности ограничивания выбора доступных агенту действий условиями задачи, одновременно дополняя возможности агента знаниями о предыдущих итерациях планирования. Разработанный алгоритм реализует механизм синтеза как классического плана действий, так и уточнение иерархически заданных задач.

2. Разработан алгоритм планирования поведения когнитивного агента, учитывающий пространственную составляющую знаний. Описан биологически и психологически правдоподобный способ представления пространственных характеристик двумерной среды. Способ представления знаний адаптирован к ограничениям знакового подхода, что позволило синтезировать детальные и одновременно понятные человеку планы действий, за время, сравнимое с временем, которое требуется общизвестным планировщикам для синтеза абстрактного плана поведения без детального описания под действий. Исследован процесс формирования фокуса внимания агента, знаковое представление фокуса внимания формируется биологически правдоподобным способом и отражает современные исследования этого вопроса.

3. Исследованы протоколы коммуникаций агентов в многоагентных системах и разработан психологически правдоподобный многоагентный алгоритм планирования поведения. Описана структура представления личности агента в собственной картине мира, знаковое представление которой получило отражение в знаке «Я», а представление остальных агентов коалиции в знаке «Они». Знаковая абстракция описания коалиции агентов предоставила возможность выбора агентов для координации при составлении плана поведения и обеспечила создание современного механизма неразглашения личных знаний агента в процессе коммуникации и динамическое распределение ролей в процессе деятельности. Создание пространственного многоагентного механизма планирования основано на психологическом процессе осознавания личности через оценивание отношений присутствия себя, отдельно от остальной группы агентов и рефлексивного оценивания возможностей других агентов.

4. Алгоритм планирования поведения адаптирован для робототехнической платформы и проведены робототехнические эксперименты. Создание пространственного алгоритма планирования, основанного на логике снижения вычислительных затрат агентов на синтез плана поведения позволило дополнить многомодульную робототехническую архитектуру

STRL модулем планирования поведения. Разделение архитектуры на ROS-пакеты и разработка соответствующих модулей распознавания и управления двигателями привела к реализации многоагентного поведения в условиях робототехнического полигона. Были разработаны соответствующие механизмы управления как перемещением агентов, так и коммуникациями, а также взаимодействием с объектами окружающей среды. Адаптация алгоритмов обучения с подкреплением к деятельности агентов позволила реализовать атомарное представление под действий и фиксацию промежуточных результатов робототехнической платформы. Были успешно проведены робототехнические эксперименты, модельно подтвердившие основные тезисы, описывающие преимущества рассматриваемого подхода.

Список литературы

1. Киселёв, Г. А., & Панов, А. И. (2018). Знаковый подход к задаче распределения ролей в коалиции когнитивных агентов. Труды СПИИРАН, 2(57), 161-187.
<https://doi.org/10.15622/sp.57.7> (Sign-based Approach to the Task of Role Distribution in the Coalition of Cognitive Agent. In; SPIIRAS Proceedings pp. 161-187)
2. Киселев, Г. А. (2020). Интеллектуальная система планирования поведения коалиции робототехнических агентов с STRL архитектурой. Информационные Технологии и Вычислительные Системы, 21–37. <https://doi.org/10.14357/20718632200203>
3. Kiselev G.A., Panov A.I. (2017) Synthesis of the Behavior Plan for Group of Robots with Sign Based World Model. In: Ronzhin A., Rigoll G., Meshcheryakov R. (eds) Interactive Collaborative Robotics. ICR 2017. Lecture Notes in Computer Science, vol 10459. Springer, Cham. https://doi.org/10.1007/978-3-319-66471-2_10
4. Autyugulov, A., Kiselev, G., & Panov, A. I. (2018). Task and Spatial Planning by the Cognitive Agent with Human-like Knowledge Representation. Interactive Collaborative Robotics, (16). https://doi.org/10.1007/978-3-319-99582-3_1
5. Kiselev G., Panov A. (2019) Hierarchical Psychologically Inspired Planning for Human-Robot Interaction Tasks. In: Ronzhin A., Rigoll G., Meshcheryakov R. (eds) Interactive Collaborative Robotics. ICR 2019. Lecture Notes in Computer Science, vol 11659. Springer, Cham. https://doi.org/10.1007/978-3-030-26118-4_15
6. Kiselev G., Panov A. (2020) Psychologically inspired model for constructing a hierarchy of cognitive agent actions (In Press). In: Ronzhin A., Rigoll G., Meshcheryakov R. (eds) Interactive Collaborative Robotics. ICR 2020. Lecture Notes in Computer Science, (In Press).
7. Киселев, Г. А., & Панов, А. И. (2016). STRIPS постановка задачи планирования поведения в знаковой картине мира. Информатика, Управление и Системный Анализ: Труды IV Всероссийской Научной Конференции Молодых Учёных с Международным Участием. Тверской государственный технический университет.
8. Киселёв Г.А., Панов А.И. Планирование действий коалицией агентов: коммуникационный аспект// Четвертый Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ-2017, 5-6 октября 2017 г., г. Казань, Республика Татарстан, Россия): тр. семинара. /

под ред. Е.А. Магида, В.Е. Павловского, К.С. Яковлева – Казань: Центр инновационных технологий, 2017. – 240 с. С.204-215.

9. Киселев, Г. А., & Панов, А. И. (2018). Семиотическое представление пространственных отношений для задачи интеллектуального перемещения. IV Всероссийская научно-техническая конференция «Интеллектуальные системы, управление и мехатроника». Материалы Всероссийской научн.-техн. конф., Севастополь 29-31 мая 2018 г., 91–97. Retrieved from <https://elibrary.ru/item.asp?id=35263035>
10. Kiselev G., Kovalev A., Panov A.I. (2018) Spatial Reasoning and Planning in Sign-Based World Model. In: Kuznetsov S., Osipov G., Stefanuk V. (eds) Artificial Intelligence. RCAI 2018. Communications in Computer and Information Science, vol 934. Springer, Cham. <https://doi.org/10.1007/978-3-030-00617-4>
11. Андрейчук А.А., Киселев Г.А., Яковлев К.С. Интеграция методов планирования поведения и планирования траектории // Семнадцатая Национальная конференция по искусственному интеллекту с международным участием. КИИ-2019 (21–25 октября 2019 г., г. Ульяновск, Россия). Сборник научных трудов. В 2 т. – Ульяновск: УлГТУ, 2019. – Т.1. – 2019. – 258 с. С. 66-74.
12. Киселев Г.А., Андрейчук А.А., Панов А.И., Яковлев К.С. Взаимодействие методов планирования в знаковой картине мира и планирования пути // XII мультиконференция по проблемам управления (МКПУ-2019) : Материалы мультиконференции (Дивноморское, Геленджик, 23-28 сентября 2019 г.) в 4 т. Т.1. – Ростов-на-Дону; Таганрог: Издательство Южного федерального университета, 2019. – 216 с. С. 80-82. <https://elibrary.ru/item.asp?id=41445882>
13. Осипов, Г. С., Панов, А. И., & Чудова, Н. В. (2014). Управление поведением как функция сознания. I. Картина мира и целеполагание. Известия Российской Академии Наук. Теория и Системы Управления.
14. Осипов, Г. С., Панов, А. И., & Чудова, Н. В. (2015). Управление поведением как функция сознания. II. Синтез плана поведения. Известия Российской Академии Наук. Теория и Системы Управления.
15. Леонтьев А.Н. Деятельность. Сознание. Личность. М.: Политиздат, 1975.
16. Fodor, J. A. (1975). The language of thought.
17. Узнадзе, Д. Н. Психология установки. М.: Питер, 2001.
18. Асмолов, А. Г. Деятельность и установка. М.: Издательство Московского университета, 1979.

19. Вачнадзе Э.А. О некоторых особенностях художества душевнобольных. Тбилиси, 1979, 32с.
20. Поспелов Д.А. Ситуационное управление. М.: Теория и практика.: Наука, 1986.
21. Coull J.T., Frackwitiik R.S.J., Frith C.D. Monitoring for target objects; activation of right frontal and parietal cortices with increasing time on task // *Neurophysiol.* 1998. V. 36. N 12. p.p. 1325-1334.
22. Goldberg M.E., Colby C.L. Oculomotor control and spatial processing // *Current opinion in neurobiology.* 1992.N.2. P. 198.
23. Подвигин Н.Ф., Макаров Ф.Н., Телепин Ю.Е. Элементы структурно-функциональной организации зрительно-глазодвигательной системы. JL: Наука, 1986.251с.
24. Daniel, K., Nash, A., Koenig, S., & Felner, A. (2010). Theta*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research*, 39, 533-579.
25. Harabor, D. D., & Grastien, A. (2011). Online graph pruning for pathfinding on grid maps. In *Proceedings of The 25th AAAI Conference on Artificial Intelligence*, pp.1114-1119.
26. Яковлев, К. С. (2020). Aa-Sipp: Алгоритм Планирования В Среде С Динамическими Препятствиями. Искусственный Интеллект И Принятие Решений, 49–59. <https://doi.org/10.14357/20718594200105>
27. Erdem, U. M., & Hasselmo, M. E. (2014). A biologically inspired hierarchical goal directed navigation model. *Journal of Physiology* Paris, 108(1), 28–37. <https://doi.org/10.1016/j.jphysparis.2013.07.002>
28. <https://www.humanbrainproject.eu/en/> (Переход по ссылке осуществлён 21.05.2020).
29. Morris, R.G.M., Garrud, P., Rawlins, J.N.P., O'Keefe, J., 1982. Place navigation impaired in rats with hippocampal lesions. *Nature* 297 (5868), 681–683, <http://dx.doi.org/10.1038/297681a0>
30. Steele, R.J., Morris, R.G.M., 1999. Delay-dependent impairment of a matching-to- place task with chronic and intrahippocampal infusion of the NMDA-antagonist D-AP5. *Hippocampus* 9 (2), 118–136
31. Steffenach, H.-A., Witter, M., Moser, M.-B., Moser, E.I., 2005. Spatial memory in the rat requires the dorsolateral band of the entorhinal cortex. *Neuron* 45 (2), 301– 313
32. Milford, M., & Wyeth, G. (2010). Persistent navigation and mapping using a biologically inspired slam system. *International Journal of Robotics Research*, 29(9), 1131–1153. <https://doi.org/10.1177/0278364909340592>
33. Milford, M., & Schulz, R. (2014). Principles of goal-directed spatial robot navigation in biomimetic models. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1655), 20130484–20130484. <https://doi.org/10.1098/rstb.2013.0484>

34. Steckel, J., & Peremans, H. (2013). BatSLAM: Simultaneous Localization and Mapping Using Biomimetic Sonar. *PLoS ONE*, 8(1). <https://doi.org/10.1371/journal.pone.0054076>
35. Maddern, W., Milford, M., & Wyeth, G. (2012). CAT-SLAM: probabilistic localisation and mapping using a continuous appearance-based trajectory. *The International Journal of Robotics Research*, 31(4), 429–451. <https://doi.org/10.1177/0278364912438273>
36. Epstein, S. L., Aroor, A., Sklar, E. I., & Parsons, S. (2013). Navigation with Learned Spatial Affordances, 1–6. Retrieved from <http://www.compsci.hunter.cuny.edu/~epstein/papers/CogSciFinal.Epstein.pdf>
37. Epstein, S. L., Aroor, A., Evanusa, M., Sklar, E. I., & Parsons, S. (2015). Spatial abstraction for autonomous robot navigation. *Cognitive Processing*, 16, 215–219. <https://doi.org/10.1007/s10339-015-0713-x>
38. Madl, T., Franklin, S., Chen, K., & Trapp, R. (2018). A computational cognitive framework of spatial memory in brains and robots. *Cognitive Systems Research*, 47, 147–172. <https://doi.org/10.1016/j.cogsys.2017.08.002>
39. Kaelbling, L. P., & Lozano-Pérez, T. (2013). Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, 32(9–10), 1194–1227. <https://doi.org/10.1177/0278364913484072>
40. Konidaris, G., Kaelbling, L. P., & Lozano-Perez, T. (2018). From skills to symbols: Learning symbolic representations for abstract high-level planning. *Journal of Artificial Intelligence Research*, 61, 215–289. <https://doi.org/10.1613/jair.5575>
41. Макаров Д. А., Панов А. И., Яковлев К. С. STRL: многоуровневая система управления интеллектуальными агентами // В кн.: Пятнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2016 (3-7 октября 2016г., г.Смоленск, Россия): Труды конференции Т. 1. Смоленск : Универсум, 2016. С. 179-188.
42. Franklin, Stan et al. “The Lida architecture: adding new modes of learning to an intelligent, autonomous, software agent.” (2006).
43. Laird, J. E., Lebiere, C., & Rosenbloom, P. S. (2017). A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. *AI Magazine*. <https://doi.org/10.1609/aimag.v38i4.2744>
44. Sun, R. 2016. Anatomy of the Mind: Exploring Psychological Mechanisms and Processes with the Clarion Cognitive Architecture. New York, NY: Oxford University Press
45. Ritter, FE, Tehranchi, F, Oury, JD. ACT-R: A cognitive architecture for modeling cognition. *WIREs Cogn Sci*. 2019; 10:e1488. <https://doi.org/10.1002/wcs.1488>
46. Choi, D., & Langley, P. (2018). Evolution of the ICARUS Cognitive Architecture. *Cognitive Systems Research*. <https://doi.org/10.1016/j.cogsys.2017.05.005>

47. Epstein, S. L., Aroor, A., Sklar, E. I., & Parsons, S. (2013). Navigation with Learned Spatial Affordances. 1–6. Retrieved from <http://www.compsci.hunter.cuny.edu/~epstein/papers/CogSciFinal.Epstein.pdf>
48. Davis, D. N., & Ramulu, S. K. (2017). Reasoning with BDI robots: From simulation to physical environment - Implementations and limitations. *Paladyn*. <https://doi.org/10.1515/pjbr-2017-0003>
49. Bechon, P., Lesire, C., & Barbier, M. (2020). Hybrid planning and distributed iterative repair for multi-robot missions with communication losses. *Autonomous Robots*. <https://doi.org/10.1007/s10514-019-09869-w>
50. Caro, M. F., Josvula, D. P., Gomez, A. A., & Kennedy, C. M. (2018). Introduction to the CARINA Metacognitive Architecture. *Proceedings of 2018 IEEE 17th International Conference on Cognitive Informatics and Cognitive Computing, ICCI*CC 2018*, (October), 530–540. <https://doi.org/10.1109/ICCI-CC.2018.8482051>
51. Хорошевский В.Ф, Гаврилова Т.А., Базы знаний интеллектуальных систем. М: Питер, 2000.
52. V.Stefanuk, A.Zhzhikashvili Principles of Rapid Prototyping for Expert Problems. Proceedings of the Workshop "International Technology Transfer in Russian Federation: Challenges and Opportunities." European Network of Excellence on Advanced Methodologies and Tools for Manufacturing Systems (AMETMAS-NoE). Editors P.P.Groumpos, J.V.Poduraev, V.L.Stefanuk, Moscow State University of Technology "STANKIN", 1998, pp.77-81.
53. Roy, D. (2005). Semiotic schemas: A framework for grounding language in action and perception. *Artificial Intelligence*, 167(1–2), 170–205. <https://doi.org/10.1016/j.artint.2005.04.007>
54. Yang, F., Lyu, D., Liu, B., & Gustafson, S. (2018). Peorl: Integrating symbolic planning and hierarchical reinforcement learning for robust decision-making. *IJCAI International Joint Conference on Artificial Intelligence, 2018-July*, 4860–4866.
55. M. Leonetti, L. Iocchi, and P. Stone. A synthesis of automated planning and reinforcement learning for efficient, robust decision-making. *Artificial Intelligence*, 241:103–130, 2016.
56. A. Bokovoy, K. Muravyev and K. Yakovlev, "Real-time Vision-based Depth Reconstruction with NVidia Jetson," *2019 European Conference on Mobile Robots (ECMR)*, Prague, Czech Republic, 2019, pp. 1-6, doi: 10.1109/ECMR.2019.8870936.
57. Wijmans, E., Kadian, A., Morcos, A., Lee, S., Essa, I., Parikh, D., ... Batra, D. (2019). DD-PPO: Learning Near-Perfect PointGoal Navigators from 2.5 Billion Frames. 1–21. Retrieved from <http://arxiv.org/abs/1911.00357>

58. Mishkin, D., Dosovitskiy, A., & Koltun, V. (2019). Benchmarking Classic and Learned Navigation in Complex 3D Environments. Retrieved from <http://arxiv.org/abs/1901.10915>
59. A. Clegg et al. AI Habitat. <https://aihabitat.org/> (Переход по ссылке осуществлён 21.05.2020).
60. Aaron Chong et al. Robot Operation System <https://www.ros.org/> (Переход по ссылке осуществлён 21.05.2020).
61. Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*. <https://doi.org/10.1109/TRO.2015.2463671>
62. M. Labb  and F. Michaud, "RTAB-Map as an Open-Source Lidar and Visual SLAM Library for Large-Scale and Long-Term Online Operation," in *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019. (Wiley)
63. К.Г. Юнг Структура психики и архетипы. М.: Академический проект, 2015.
64. Чуганская А. А., & Чудова, В. Н. (2019). Психологические исследования сенсорной уверенности и социальных аттитюдов как активаторов и механизмов затухания в знаковой картине мир. *Вестник экспериментального образования*, с 27–35.
65. Erickson, Z., Clever, H. M., Turk, G., Liu, C. K., & Kemp, C. C. (2018). Deep Haptic Model Predictive Control for Robot-Assisted Dressing. *Proceedings - IEEE International Conference on Robotics and Automation*, 4437–4444. <https://doi.org/10.1109/ICRA.2018.8460656>
66.  zkil, A. G., Fan, Z., Dwids, S., Aana s, H., Kristensen, J. K., & Christensen, K. H. (2009). Service robots for hospitals: A case study of transportation tasks in a hospital. *Proceedings of the 2009 IEEE International Conference on Automation and Logistics, ICAL 2009*, (September), 289–294. <https://doi.org/10.1109/ICAL.2009.5262912>
67. Madl, T., Franklin, S., Chen, K., Trappl, R., & Montaldi, D. (2016). Exploring the structure of spatial representations. *PLoS ONE*. <https://doi.org/10.1371/journal.pone.0157343>
68. Hanford, S. D., & Long, L. N. (2014). Development of a mobile robot system based on the soar cognitive architecture. *Journal of Aerospace Information Systems*, 11(10), 714–725. <https://doi.org/10.2514/1.I010191>
69. Puigbo, J. Y., Pumarola, A., & Tellez, R. (2013). Controlling a general purpose service robot by means of a cognitive architecture. *CEUR Workshop Proceedings*, 1100, 45–55.
70. Kirk, J. R., & Laird, J. E. (2019). Learning hierarchical symbolic representations to support interactive task learning and knowledge transfer. *IJCAI International Joint Conference on Artificial Intelligence*, 2019-August, 6095–6102. <https://doi.org/10.24963/ijcai.2019/844>
71. Echeverria, G., Lemaignan, S., Degroote, A., Lacroix, S., Karg, M., Koch, P., Lesire, C., & Stinckwich, S. (2012). Simulating complex robotic scenarios with MORSE. In *International*

Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR), Tsukuba, Japan.

72. Wilm Decr' et al. Orocoss. <https://www.orocos.org/> (Переход по ссылке осуществлён 21.05.2020).
73. Rao, A., & Georgeff, M. (1995). BDI Agents: From Theory to Practice. Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95).
74. Кулинич А.А. Модель командного поведения агентов в качественной семиотической среде. Часть 1. Качественная среда функционирования. Основные определения и постановка задачи (2017). Искусственный Интеллект И Принятие Решений.
75. Кулинич А.А. Модель командного поведения агентов в качественной семиотической среде. Часть 2. Модели и алгоритмы формирования и функционирования команд агентов. (2018). Искусственный Интеллект и Принятие Решений.
<https://doi.org/10.14357/20718594180103>
76. В. И. Варшавский, М. В. Мелешина, М. Л. Щетлин, “Организация дисциплины ожидания в системах массового обслуживания с использованием модели коллективного поведения автоматов”, Пробл. передачи информ., 4:1 (1968), 73–76 mathnet; V. I. Varshavskii, M. V. Meleshina, M. L. Tsetlin, “Priority Organization in Queueing Systems Using a Model of the Collective Behavior of Automata”, Problems Inform. Transmission, 4:1 (1968), 58–60
77. Kaliaev I.A., Gaiduk A.R., Kapustyan S.G. И . А . Каляев , А . Р . Гайдук , С . Г . Капустян Самоорганизация в мультиагентных системах С. 14–20.
78. Jennings N.R. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions // Artificial Intelligence. 1995. № 2 (75). С. 195–240.
79. Milind Tambe Towards Flexible Teamwork // Journal of Artificial Intelligence Research. 1997
80. Barbara J. Grosza,*, Sarit Kraus Of F., Agents C. Artificial Intelligence 2010. № 95 (86). С. 19–20.
81. Бурдун И.Е., Бубин А.Р. Исследования и разработки в области мобильной робототехники стационарного применения (краткий технический обзор зарубежных публикаций) // Морские информационно-управляющие системы. 2012. № 1. С. 46–56.
82. В. О. Корепанов, Д. А. Новиков Метод рефлексивных разбиений в моделях группового поведения и управления 2011. № 1. С. 21–32.
83. Dmitry A. Novikov, G. Chkhartishvili C. Graph of a Reflexive Game and Bélasses-léttres p. 16–28.
84. Тарасов В.Б. Искусственная жизнь и нечеткие эволюционные многоагентные системы - основные теоретические подходы к построению интеллектуальных организаций// Известия РАН: Теория и системы управления. – 1998. – №5. – С.12-23.

85. Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – М.: Эдиториал УРСС, 2002. – 352 с
86. Stefanuk V.L. Behavior of Multi-Agent Systems: the Coordination Paradigm Artificial Intelligence News, 4, 1997, pp. 92-104. [82]
87. Stefanuk V.L. Local Organization of Intellectual Systems. Models and Applications. Moscow: Fizmatlit, 2004, 349 P. [83]
88. Городецкий В.И. и др. Прикладные многоагентные системы группового управления // Искусственный интеллект и принятие решений. 2009. № 2. С. 3–24.
89. Городецкий В.И., Каравес О.В. Самоорганизация группового поведения кластера малых спутников распределенной системы наблюдения // Известия ЮФУ. Технические науки. 2017. Т. 187. № 2. С. 234–247.
90. Granatyr J. et al. Trust and Reputation Models for Multiagent Systems // ACM Comput. Surv. 2015. vol. 48. no. 2. pp. 1–42.
91. Huynh T.D., Jennings N.R., Shadbolt N.R. An integrated trust and reputation model for open multi-agent systems // Autonomous Agents and Multiagent Systems. 2006. vol. 13. no. 2. pp. 119–154.
92. Sabater J., Sierra C. Review on computational trust and reputation models // Artificial Intelligence Review. 2005. № 1 (24). С. 33–60.
93. Serrano E., Rovatsos M., Botia J. A qualitative reputation system for multiagent systems with protocol-based communication // Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems. 2012. vol. 1. pp. 307–314
94. Brafman, R. I. (2015). A Privacy Preserving Algorithm for Multi-Agent Planning and Search. Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015), 1530–1536.
95. Jakubův J., Tožička J., Komenda A. Multiagent Planning by Plan Set Intersection and Plan Verification // Proceedings of the International Conference on Agents and Artificial Intelligence. SCITEPRESS - Science and Technology Publications. 2015. pp. 173–182.
96. Primeau N. et al. Improving task allocation in risk-aware robotic sensor networks via auction protocol selection // 2016 IEEE 20th Jubilee International Conference on Intelligent Engineering Systems (INES). 2016. pp. 21–26.
97. Холодкова А.В. Применение агентов в модели договорных сетей // Системы обработки информации. 2012. № 4(102). С. 142-145.
98. Lorini E., Verdicchio M. Towards a Logical Model of Social Agreement for Agent Societies // Coordination, Organizations, Institutions and Norms in Agent Systems V. 2010. pp. 147–162.

99. The Foundation for Intelligent Physical Agents – FIPA. URL: <http://www.fipa.org> (дата обращения: 19.05.2020).
100. Осипов Г.С., Панов А. И. Синтез рационального поведения когнитивным семиотическим агентом., 2020 (В печати).
101. Gerevini, A., & Long, D. (2006). Preferences and Soft Constraints in {PDDL3}. Proceedings of the ICAPS-2006 Workshop on Preferences and Soft Constraints in Planning.
102. <https://github.com/aibasel/pyperplan/> (ссылка проверена 16.06.2020)
103. Drew McDermott, et.al. (1998). The First International Planning Competition. <http://icaps-conference.org/index.php/Main/Competitions>
104. Cox, M. T., Alavi, Z., Dannenhauer, D., Eyorokon, V., Munoz-Avila, H., & Perlis, D. (2013). MIDCA: A Metacognitive, Integrated Dual-Cycle Architecture for Self-Regulated Autonomy. Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16).
105. Prof. Dr. Susanne Biundo-Stephan et al. PANDA <https://www.uni-ulm.de/en/in/ki/research/software/panda/panda-planning-system/>
106. Grea, A., Matignon, L., Aknine, S. HEART. In: International Conference on Automated Planning and Scheduling, pp. 17–25. (2018).
107. Francesco Rovida et al. eBT&SkiROS <https://github.com/frovida/ebt>
108. Sutton, S. S., Barto A.G.: Reinforcement Learning: An Introduction (1998)
109. Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. Machine Learning. <https://doi.org/10.1007/bf00992698>
110. Mnih, V., Kavukcuoglu, K., Silver, D. et al. Human-level control through deep reinforcement learning // Nature 518, 2015
111. Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D. de Las, ... Riedmiller, M. (2018). **美国** Google Deepmind 2018 DeepMind Control Suite.

Приложение А. Когнитивные архитектуры

Структура	STRIL	LIDA	SOAR	CLARION	ACT-R
3 уровня стратегический, тактический реактивный.	3 фазы - 1. понимания и 2. внимания и 3. действий обучения.	3 фазы - 1. понимания и 2. внимания и 3. действий обучения.	Набор связанных вычислительных модулей. Акцент на функциональност и эффективности.	4 подсистемы ориентированные на - 1. ACS действия 2. NACS память 3. MS мотивацию 4. MCS внимание и осознавание.	Модульность. Каждый модуль имеет буфер, через который запрашивает информацию. В отличии от SOAR описывается более подробный когнитивный цикл.
Способ представления	(с) - знаковая модель (т) - метрические топологические графы (р) - нелинейные регуляторы.	Зависит от кодлеты + сети схем, векторы многомерного модульного составного представления и др.	Декларативный - чанки (куски, вектора) и процедурный (продукции). Знания краткосрочные и долгосрочные.	Знания в каждой из подсистем представлены в явном и неявном виде. Явный - системы правил, не явный - нейросети.	Декларативный - чанки (куски, вектора) и процедурный (продукции). Знания краткосрочные и долгосрочные.
Процесс планирования поведения и принятия решений	основан на принципах распространения активности.	Иерархический, представлен быстро повторяющимися когнитивными циклами.	Иерархия процессов обработки подцелей. Процессы распараллелены.	Иерархичность заключается разделении когнитивных действий по каждой и подсистеме.	Иерархичность процесса заключается в комплексности работы модулей.
Обучение	Опыт планирования сохраняется и переиспользуется. Абстрактные действия планировщика уточняются с помощью обучения без учителя.	Опыт сохраняется и переиспользуется. Образуются новые когнитивные связи.	Обучение на опыте достижения подцели. Декларативная и процедурная память пополняема.	Индуктивное и дедуктивное.	И декларативная, и процедурная память пополняемы.

	STR	LIDA	SOAR	CLARION	ACT-R
Полнение знаний на основе данных сенсоров	Производится на тактическом и реактивном уровнях. Используются SLAM по карте, распознавание объектов для манипулирования.	Зависит на и от реализации фазы понимания. Расширяется с помощью кодлов.	Представлена от модулем обработки визуальных данных	описание отсутствует.	Реализации модуля распознавания и визуализации.
Неопределенность данных	Обусловлено представлением о возможностях манипулирования с объектами в процессе обучения.	Существуют примеры байесовской интерпретации фазы понимания.	Правила выбору операторов в процессе планирования генерируются с помощью RL.	Обусловлено неявным представлением данных. Используются алгоритмы RL/нейросети.	Используются допущения к представлению задачи.
Персонифицированность знаний	Представлено на стратегическом уровне знаками "Я" и "Они". Используется рефлексия (авторефлексия) и 2 рода.	Представлено в перцептивной памяти рефлексией 1 и 2 рода.	Описана в субъективность и лекларативных процедурных знаний агента.	Достигается посредством метакогнитивного уровня осознания.	Отсутствует в классической реализации
Моделирование внимания агента	На стратегическом уровне представлена иерархическим фокусом внимания агента.	Представлена соответствующей фазой когнитивного цикла.	Представлена робототехническими дополнениями к внутреннему представлению окружающей среды.	достигается деятельностью подсистемы MS и MCS.	Представлена сегментацией событий для робототехнических реализаций.

	STRIL	LIDA	SOAR	CLARION	ACT-R
Использование автоматических процедур	Деятельность тактического и реактивного уровня.	Используются кодлыты. Зависит от реализации.	Явное представление отсутствует. Зависит от реализации.	Представлена неявным представлением	Различается зависимости от реализации архитектуры
Способ коммуникации между агентами	Протокол коммуникации, аукцион планов.	Явно зависит от реализации.	Используются модули коммуникации, основанные на рефлексивном восприятии других агентов.	Часть системы ACS. Явно не описан.	Представлен модулем коммуникации.
Применение	Робототехника, интеллектуальные ассистенты.	Робототехника, симуляции когнитивных агентов, агентов, медицинская диагностика.	Робототехника, моделирование человеческого поведения для тестирования	Симуляции когнитивных агентов	Моделирование человеческого поведения для тестирования
Разработчик	ФИЦ ИУ РАН, Россия	Университет Мемфиса, США	Университет Корнеги Мелон, Питтсбург, США	Ренсселлерский политехнический институт, США	Университет Корнеги Мелон, Питтсбург, США
Виды экспериментов	Роботы turtlebot-2 (симуляция) - поднятие и перемещение объектов.	MPI-PM, Atlas в симуляции - по перемещению и роботов	Эксперименты PR2, Fetch и др - поднятие объектов, совещательная деятельность при перемещении автотранспорта в симуляции.	Описание экспериментов отсутствует	Взаимодействие с Unity - генерация логики агентов в аркадных играх. Предсказание скорости распознавания дорожных знаков водителем в симуляции.

	STRL	LIDA	SOAR	CLARION	ACT-R
Ссылка на описание	<p>Макаров Д. А., Панов А. И., Яковлев К. С. STRL: многоуровневая система управления интеллектуальными агентами // Пятнадцатая национальная конференция по искусственному интеллекту и интеллектуальному международным Участием КИИ-2016 Труды конференции Г. 1. Смоленск : Универсум, С. 179-188.</p>	<p>Franklin, Stan et al. “THE LIDA ARCHITECTURE: ADDING NEW MODES OF LEARNING TO AN INTELLIGENT, AUTONOMOUS, SOFTWARE AGENT.” (2006).</p> <p>intelligence, cognitive science, neuroscience, and robotics. AI Magazine. https://doi.org/10.1609/aimag.v38i4.2744</p>	<p>Laird, J. E., Lebriere, C., & Rosenbloom, P. S. (2017). A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. AI Magazine. https://doi.org/10.1609/aimag.v38i4.2744</p>	<p>Sun, R. 2016. Anatomy of the Mind: Exploring Psychological Mechanisms and Processes with the Clarion Cognitive Architecture. New York, NY: Oxford University Press</p>	<p>Ritter, FE, Tehranchi, F, Oury, JD. ACT-R: A cognitive architecture for modeling cognition. WIREs Cogn Sci. 2019; 10:e1488. ps://doi.org/10.1002/wcs.1488</p>
Ссылка на код и ссылка на архитектуру	<p>http://strl-robotics.ru/cog-is.edu/tutorial/-isa/map-core</p>	<p>http://ccrg.cs.memphis.edu/articles/downloads/soar-suite/227-soar-suite-9-6-0</p>		<p>http://www.clarioncognitionarchitecture.com/</p>	<p>http://act-r.psych.cmu.edu/</p>

	ICARUS	SemaFORR	CAMA	НІРОР	CARINA
Структура	Линейная структура распознавания среды и поиска навыков в долговременной и кратковременной памяти	3 уровня эвристических советчиков - 1. по ситуации (реактивные), 2. по планированию и 3. по решению разногласий между 1 и 2	Представлена соединением схемы BDI нотации, аффективных модулей и рефлексивного модуля контроля поведения.	Модули планирования (перепланирования), коммуникаций.	4 уровня: уровень 0. когнитивного агента 1. рассуждений о его деятельности 2. мета-модель рассуждений 3. мета-мета модель.цикл.
Способ представления знаний	Знания представлены в предикатной форме. Знания краткосрочные и долгосрочные.	Дескриптивы описания среды для советчиков. Также присутствует 3 вида геометрических абстракций: - трассы, области и конвейеры.	- BDI графы доверия.	В предикативной форме.	Элементами внутреннего языка M++.
Процесс планирования поведения и принятия решений	Иерархичность поиска и приобретении навыков, а также в распознавании и концептуализации ситуации.	Иерархический, основан на цикле опроса советчиков.	Стандартная иерархичность BDI-рассуждений, дополненная расширенными вероятностными оценками доверия.	Иерархическая версия алгоритма частичного планирования РОР.	Иерархичность заключается во множественном оценивании реакции окружающей среды на текущее действие
Обучение	Создание новых навыков после достижения цели без участия учителя.	Опыт планирования в эвристиках советчиков.	Агент запоминает опыт планирования и взаимодействий.	Явно не представлен	Явно не представлен. Авторы описывают обучение, как смену стратегий, основываясь на опыте.

	ICARUS	SemaFORR	CAMAL	HIPOP	CARINA
Пополнение знаний на основе сенсоров	Модуль распознавания окружения и выдвижения концепта.	Пополнение знаний с помощью HoloLens+SLAM+Е KF.	Представлено в аффективном виде.	Используется в робототехнической постановке.	Модуль восприятия датчиками и распознавания.
Неопределенность данных	Неоднозначность распознавания среды.	Обусловлено стратегиями советчиков и их представлением об выполнимости перемещения.	Обусловлена неточностью распознавания среды и вероятностной оценкой возможной взаимодействия с другим агентом.	Явно не представлена.	Явно не представлена.
Персонифицированность знаний	Описание отсутствует.	Отсутствует явное представление.	Представлена BDI графом отношений.	Отсутствует в классической реализации.	Знания персонифицированы с помощью само-модели на метауровне M1.
Моделирование внимания агента	Описание отсутствует.	Представлена деятельностью советчиков 1 и 2 уровней.	Описание отсутствует.	Представлена различными стратегиями метауровней по выявлению важных вещей при распознавании ситуаций.	

	ICARUS	SemaFORR	CAMAL	HiPOP	CARINA
Использование автоматических процедур	Для выполнения действий. Ограничены временными рамками.	Любое выполнение действий.	Обусловлено реализацией аффективного поведения. Если робот слабый используется система SCARAB.	Представлено действиями перемещений.	Присутствует в конкретных реализациях когнитивных агентов.
Способ коммуникации между агентами	Явно не описан. Зависит от робототехнической реализации.	Реактивная коммуникация. Для отслеживания деятельности группы (толпы) используются алгоритмы CUSUM-А* и Risk-A*.	С помощью протокола коммуникации и обмена данных.	Протоколы коммуникаций (обычный, восстановления плана, синхронизации...).	Явно не представлен.
Применение	Робототехника, игровая индустрия.	Робототехника, игровая индустрия.	Робототехника.	Робототехника.	Интеллектуальные ассистенты.
Разработчик	NASA, США	DARPA, США	Городской Университет Нью-Йорка, США	Университет Халла, Англия	Университет Гулзы, Франция
Виды экспериментов	Искусственные агенты в игровых средах, локализация робота Nomad 200, симуляционные эксперименты с роботом MAHRU.	Робототехнические эксперименты в симуляции.	Эксперименты на роботах AmigoBots, Pioneer Activity Bots. Картирование местности, поиск объектов.	Симуляционные и эксперименты на робототехнические роботах 3-DX, effibot перемещению местности. поиск объектов.	Система составления расписаний FunPro.
				CNRS.	LAAS-

	ICARUS	SemaFORR	CAMAL	HiPOP	CARINA	
Ссылка на описание	<p>Choi, D., & Langley, P. (2018). Evolution of the ICARUS Cognitive Architecture.</p> <p>Cognitive Systems Research.</p> <p>https://doi.org/10.1016/j.cogsys.2017.05.005</p>	<p>Epstein, S. L., Aroor, A., Sklar, E. I., & Parsons, S. (2013). Navigation with Learned Affordances.</p> <p>Retrieved from http://www.compsci.hunter.cuny.edu/~epstein/papers/CogSciFinal.Epstein.pdf</p>	<p>Epstein, S. L., Aroor, A., Sklar, E. I., & Parsons, S. (2013). Navigation with Learned Affordances.</p> <p>Retrieved from http://www.compsci.hunter.cuny.edu/~epstein/papers/CogSciFinal.Epstein.pdf</p>	<p>Davis, D. N., & Ranulu, S. K. (2017). Reasoning with BDI robots: From simulation to physical environment - implementations and limitations.</p> <p>Paladyn.</p> <p>https://doi.org/10.1015/pjbr-2017-0003</p>	<p>Bechon, P., Lesire, C., & Barbier, M. (2020).</p> <p>Hybrid planning and distributed iterative repair for multi-robot missions with communication losses.</p> <p>Autonomous Robots.</p> <p>https://doi.org/10.1007/s10514-019-09869-w</p>	<p>Caro, M. F., Josvula, D. P., Gomez, A. A., & Kennedy, C. M. (2018). Introduction to the CARINA Metacognitive Architecture.</p> <p>Proceedings of 2018 IEEE 17th International Conference on Cognitive Informatics and Computing, ICII*CC 2018, (October), 530–540.</p> <p>https://doi.org/10.1109/ICCI-CC.2018.8482051</p>
Ссылка на код и ссылка на архитектуру	<p>http://csl.stanford.edu/research/ongoing/icarus/</p>	<p>http://www.cs.hunter.cuny.edu/~epstein/</p>	<p>Описание отсутствует.</p>	<p>https://dblp.uni-trier.de/pers/b/Bechon:Patrick.html</p>	<p>https://www.researchgate.net/profile/Manuel_Pieres</p>	