

Интеллектуальная система планирования поведения коалиции робототехнических агентов с STRL архитектурой*

Г. А. Киселёв

Институт проблем искусственного интеллекта ФИЦ ИУ РАН, Москва, Россия

Аннотация. Работа посвящена вопросам программной реализации STRL архитектуры когнитивного агента для группы робототехнических платформ. Рассмотрена задача синтеза коалиционного и индивидуального пространственных планов поведения агентов. Представлены результаты по адаптации методов теории знаковой картины мира при построении иерархических систем управления на базе мобильной робототехнической платформы. Проведен ряд экспериментов по построению совместных коалиционных планов, включающих действия по перемещению в пространстве и по манипуляции объектами.

Ключевые слова: когнитивные агенты, иерархического планирование, геометрическое планирование, псевдофизические логики, знаковый подход, коалиции агентов, робототехника.

DOI 10.14357/20718632200203

Введение

Когнитивные архитектуры призваны повысить степень автономности робототехнических платформ и позволить им выполнять комплексные задачи без участия человека. Функционал архитектуры должен обеспечивать сложное поведение агента, реализовывать внутригрупповую коммуникацию и оптимизацию деятельности коллектива агентов. Интеллектуальный подход к построению архитектур такого рода является актуальной задачей и требует решения комплекса междисциплинарных проблем. Обычно, структура когнитивных архитектур представлена тремя уровнями контроля поведения – стратегическим, тактическим и реактивным [1]. Реализация уровней осуществлена в одной из наиболее перспективных на данный

момент архитектур - STRL [2], на основе которой были продемонстрированы особенности взаимодействия методов интеллектуального управления, планирования траекторий и планирования поведения. Однако в предыдущих работах, посвященных архитектуре, мало внимания уделялось построению реальных интеллектуальных программных систем, а также особенностям реализации архитектуры на колесных роботах. Целью данной работы является разработка и описание программной реализации архитектуры STRL на робототехнической платформе МП-РМ. Для достижения этой цели были реализованы функции планирования поведения когнитивного агента с архитектурой STRL и представлены методы решения ряда технических проблем, возникающих при работе с датчиками и актуаторами реального робота.

* Работа выполнена при частичной финансовой поддержке РФФИ, проекты №17-29-07051 и №18-29-22047.

Каждый из уровней архитектуры STRL отвечает за выполнение соответствующих функций:

- Стратегический – хранение, приобретение и использование знаний об окружающей среде, целеполагание, планирование и обучение. Этот уровень отвечает за связь между агентами и координацию действий в группе. Особенностью уровня является наличие знакового [4] представления окружающей среды, позволяющего реализовать такие когнитивные функции как целеполагание и распределение ролей в коллективе.
- Тактический – картирование, локализация и планирование траектории перемещений.
- Реактивный – выработка управляющего сигнала для двигателей агента, расчет геометрических ограничений на перемещение.

В рамках адаптации для мобильной робототехнической платформы, функционал уровней архитектуры был разделен на модули, представленные на Рис. 1. Модули сенсоров и взаимодействия с окружающей средой были вынесены в отдельные разделы, так как при адаптации архитектуры к МП-РМ был выявлен ряд проблем при использовании стандартных реализаций и проведена работа по их устранению. Основными проблемами являются неточность перемещений и локализации робота, ошибки распознавания окружающих объектов, неуспешные взаимодействия с ними. Навигация робототехнических платформ требует точной обработки и предсказания ошибок при перемещении и локализации робота. У большинства когнитивных архитектур, реали-

зация робототехнических экспериментов заканчивается неудачно из-за несвоевременного выявления и обработки ошибок такого рода, накопление которых может привести к поломке или столкновению платформы с различного рода препятствиями. Решением проблем локализации служит постоянная оценка собственного местоположения роботом, исходя из данных от различных датчиков (лидары, кинекты, GPS-приемники, камеры, УЗ датчики и т.д.) и перепланирование в случае сильного отклонения от заданного курса.

План поведения синтезируется планировщиком, который основан на принципах распространения активности в знаковой картине мира. Планировщик позволяет сокращать перебор при поиске применимых в настоящий момент действий благодаря активации только тех действий, которые непосредственно связаны с объектами текущей ситуации. Процесс планирования поведения и способность внутреннего психологически-правдоподобного отображения окружающей среды позволяет описывать робота в качестве когнитивного агента. Агент способен решать задачи не только классического вида, но и использовать иерархические методы описания подпланов, сохранять и переиспользовать опытные знания для активации сложных действий (описывающих прецеденты деятельности), а также оперировать пространственными данными, сохраняя планы высокого уровня детализации в качестве действий более низкого уровня (подробнее в разделе 2). Различные способы формулирования заданий планировщику

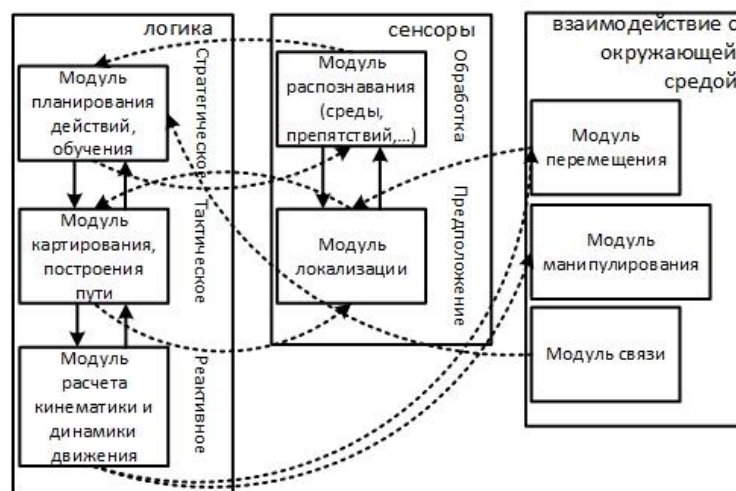


Рис. 1. Схема робототехнической платформы

призваны упростить коммуникацию агентов, а также организацию коллективной деятельности. В качестве иллюстрации способности описываемой архитектуры синтезировать план поведения рассмотрен ряд экспериментальных задач взаимодействия с объектами окружающей среды, в рамках которых агент планирует как самостоятельно, так и в коалиции с другими агентами. Конечный план действий формируется на основе протокола аукциона планов, в рамках которого агенты голосуют за наиболее приемлемый для них по ряду признаков план. Рассмотрен пример синтеза плана гетерогенной коалицией агентов, в которой ограничения на деятельность агентов присутствует в классическом предикативном описании задачи, а пространственные характеристики на геометрическом уровне описания.

Далее статья организована следующим образом. В разделе 1 приводится краткий обзор когнитивных архитектур и их применение в робототехнике, приведены основные способы планирования поведения роботов и разобран их функционал. В разделе 2 описан алгоритм синтеза иерархического плана поведения агента, разобран классический и иерархический способ генерации действий, а также приведен пример формирования и сохранения прецедентного действия для его переиспользования в последующих задачах. Также, в разделе приведено описание пространственного псевдофизического представления окружающей среды агентом и описана реализация псевдофизической логики в рамках алгоритма планирования поведения. Разобран способ коммуникации агентов с пространственным представлением окружающей среды. В разделе 3 представлена реализация схемы STRL-архитектуры для робототехнической платформы МП-РМ и детально описана программная реализация. В разделе 4 даны результаты и описание модельных экспериментов, а также их сравнение с аналогичными решениями планировщика действий.

1. Когнитивные архитектуры в управлении роботом

Помимо рассматриваемой реализации архитектуры STRL, процессы планирования и при-

ятия решений, целеполагания и обучения частично реализованы и описаны в когнитивных архитектурах HiPOP [5], SOAR [6, 7], BDI [8], LIDA [9] и др. В архитектуре HiPOP роботы выполняют коалиционные действия по решению когнитивных задач, в число которых входит поиск и сопровождение цели, объезд препятствий и др. Основным планировщиком архитектуры является иерархическая версия алгоритма POP [10]. Также используется простая временная сеть STN [11], которая позволяет представить общий план действий в виде последовательности временных интервалов, ограничивающих перемещения робота. В статье [5] рассматривается случай коалиционного слежения за целью и приведены эксперименты для 12 роботов в симуляции и 8 роботов в реальной среде. Также, как и в данной статье, авторы [5] столкнулись с рядом проблем при реализации перемещений роботов в неидеальных условиях реальной среды, среди которых были выделены: ветер, вызывающий отклонение роботов, ограниченный срок работы аккумуляторов и проблемы со связью. Время действия роботов в реальной среде превысило спланированное время на 73 процента, а целевое состояние достигалось не каждый раз. Настоящая реализация архитектуры STRL позволяет осуществить не только перемещение роботов, но и манипулирование объектами окружающей среды, а также использовать информацию о предыдущих итерациях планирования, сохраненных в опыте агента. Способность оперировать пространственной псевдофизической логикой предоставляет возможность алгоритму планирования описываемой архитектуры, в отличие от POP алгоритма, составлять иерархические и понятные для человека планы поведения роботизированного агента. Использование опыта, в качестве основополагающего элемента строения архитектуры представлено в архитектуре SOAR.

В SOAR цикл планирования осуществляется с помощью правил оценивания и выбора промежуточных состояний (основываясь на опыте), а также поиска применимых в этих состояниях операторов. Правила оперируют 2 типами знаний – декларативными (об окружающей среде) и процедурными (кодирующими обработку декларативных знаний). Архитектура SOAR в

базовой реализации не учитывает фокус внимания [12] агента, но в статье [7] описана реализация архитектуры на робототехнической платформе, в рамках которой агент формирует внутреннее представление окружающей среды. Внутреннее представление формируется посредством целенаправленного рендеринга окружающих объектов, а процесс планирования осуществляется для ограниченного набора данных об окружающей среде. Также, архитектура SOAR может быть использована в качестве совещательного органа для построения пути роботизированного агента [6], реализуя возможность выбора наиболее подходящего маршрута, основываясь на знаниях конфигурации робота. В отличие от STRL, в SOAR отсутствует возможность оперирования геометрическим представлением об объектах среды и физических ограничениях робота в процессе синтеза плана действий. Этот факт существенно ограничивает архитектуру и требует построения множества планов действий, выбор среди которых может быть осуществлен, основываясь на прецедентных знаниях агента и данных с других уровней восприятия среды. В многоагентную составляющую архитектуры SOAR входит рефлексивное представление других агентов, их целей, предпочтений и способов коммуникаций. Подобная структура также присутствует в роботизированных архитектурах, основанных на BDI.

Архитектуры, основанные на BDI способе представления знаний, достигают целей (желаний) применяя правила (убеждения) к воспринимаемым знаниям об окружающей среде. Роботизированные решения, использующие BDI, также, как и реализации STRL архитектуры, основывают деятельность на многоагентном подходе к планированию и учитывают прецеденты взаимодействий агентов в процессе синтеза плана поведения. В BDI, как и в стратегическом уровне STRL, присутствует иерархичность представления знаний, что позволяет выстраивать планы различных уровней абстракции. В статье [8] приводится описание реализации робототехнической архитектуры SAMAL, основанной на BDI способе представления знаний агентами. Авторы рассмотрели ролевое взаимодействие 1 робота Pioneer, 4 AmigoBots и ActivityBots. Целью деятельности

являлось картирование местности с помощью сонаров маломощных роботов и камеры робота Pioneer. Все когнитивные вычисления роя производились централизованно на Pioneer, а деятельность остальных агентов была направлена на реализацию реактивного поведения. Авторы столкнулись с проблемами зашумленности датчиков, неточности перемещения и локализации роботов, маломощность большинства агентов роя не позволяла им реализовать своевременное распознавание объектов среды. Также, как и в [5], в [8] не реализована деятельность агентов по взаимодействию с объектами окружающей среды, что снизило сложность задачи. Основным отличием архитектур, основанных на BDI от STRL, LIDA и других архитектур, основополагающей функцией когнитивного цикла которых является функция планирования поведения, является отсутствие реализаций механизмов уменьшения ветвления при выборе действий в процессе синтеза плана (моделирование внимания, использование множества эвристических правил и т.д.).

Системы с архитектурой LIDA имеют 3 фазы когнитивного цикла взаимодействия с окружающей средой - фаза понимания (распознавание объектов и локализация), фаза внимания (перемещение объектов в фокус внимания) и фаза действий и обучения. В фазе понимания присутствует сенсорный шум и неопределенность, работа над которыми была произведена в [9]. В статье описан байесовский подход к прогнозированию местоположения робота Atlas, относительно его предыдущего местоположения и объектов окружающей среды. Авторы не описывают полноценный способ взаимодействия архитектуры и робота, но предлагают байесовскую непараметрическую модель кластеризации для структурирования представлений объектов в рамках предметной метрики для формирования представления окружающей среды роботом. Из-за требований архитектуры к когнитивному синтезу действий агента, основной проблемой LIDA является низкая скорость реакции на изменения внешней среды и требовательность к ресурсам. В отличие от STRL в архитектуре отсутствует возможность частично сворачивать модули работы с окружающими объектами до бессознательных про-

Табл. 1. Сравнение когнитивных архитектур

	STRL	HIPOP	SOAR	CAMAL	LIDA
Планирование коллективного поведения групп агентов	Да	Да	Да	Да	Нет
Персонафицированность знаний	Да	Нет	Да	Да	Да
Моделирование внимания	Да	Нет	Нет	Нет	Да
Использование и сохранение опыта	Да	Да	Да	Да	Да
Представление пространственных знаний агента	Да	Нет	Да	Да	Да
Использование автоматических процедур	Да	Да	Нет	Да	Нет
Разработчик	ФИЦ ИУ РАН, РФ	Университет Тулузы, Франция	Университет Корнеги Меллон, США	Университет Халла, Англия	Университет Мемфиса, США

цедур, которые могут быть выполнены мало-мощными роботами, а также не описан способ коммуникации агентов. Сравнительная характеристика описанных выше архитектур представлена в Табл. 1.

При планировании перемещений и действий роботизированного агента в помещениях с большим количеством объектов требуется соответствующая адаптация алгоритмов. Примерами таких алгоритмов являются планировщики [12-15]. Эксперименты в симуляционной и реальной средах выявили проблему недостаточной скорости построения плана действий и был разработан прецедентный (основанный на опыте) [16-19] подход. Прецеденты деятельности в реальной среде не всегда могут быть полностью адаптированы к настоящей ситуации, в связи с чем, в статье [12] было выдвинуто предложение выявлять сценарии деятельности. Сценарии выявляются с помощью представления экземпляров решения задачи планирования в виде векторов оценок при расчете плана с заданными ограничениями. С помощью оценивания применимости планов выявлялись наиболее значимые ограничения, которые ускоряли синтез нового плана, частично уточняя действия перед планированием (в процессе планирования заполнялись остальные детали). Прецедентный способ планирования поведения является распространенным методом ускорения синтеза плана и модуль синтеза плана поведения описываемой в данной работе архитектуры также использует накопленные знания для ус-

корения поиска решения задачи. Алгоритм планирования архитектуры STRL может использовать не только полностью совпадающий с настоящей задачей опыт действия, но и прецеденты, описывающие частичное решение поставленной задачи.

2. Синтез иерархического плана поведения STRL-агента

Стратегическая составляющая архитектуры STRL основана на семиотическом представлении знаний агента [20, 21], базовым элементом которой является знак [22-24]. Использование знаковой парадигмы в качестве представления знаний об элементах окружающей среды было выбрано исходя из исследований нейрофизиологов и психологов [25, 26] относительно представления знаний человеком. Знаковая парадигма архитектуры учитывает различные типы информации об одном и том же объекте и использует их для перехода от классического для искусственного интеллекта символического способа представления знаний к гибриднему, который отвечает принципам робототехники настоящего времени.

Знак описывается кортежем, в который включаются четыре основные компоненты $s = \langle n, p, t, a \rangle$, где n – компонента имени, с помощью которого задается знак, p – компонента образа, t – компонента значения и a – компонента смысла. Каждая из компонент знака отвечает за свой вид представления информации об

описываемой сущности. Компонента p является описанием характерных признаков сущности (при геометрической постановке задачи – это описание координат участка местности, либо последовательность поддействий абстрактного действия (в том числе в классическом случае)). Компонента m – отвечает за доступные обобщенные сценарии использования сущности коллективом агентов, а компонента a – определяет роль представленной знаком сущности в действии планирующего субъекта. Смыслы знака синтезируются в процессе деятельности агента с описываемой сущностью и являются актуализацией в рамках настоящей задачи значений этого знака. Знаком может быть представлен как статический объект, так и действие.

Компоненты знака представлены специальной структурой – каузальной матрицей [27] - структурированным набором ссылок на другие знаки и элементарные признаки, которые связаны со структурой описания объекта. Такой тип матриц позволяет связывать компоненты знаков и формировать семантические сети, отношения на каждой из которых различны (например, на сети m – объект-роль, на сети p – часть-целое, а на сети a – коалиция – участник коалиции). Каузальные матрицы имеют разделенную на две части структуру, левая часть отвечает за условия активации описываемой сущности, а правая за эффекты. В простейшем случае, каузальную матрицу можно описать с помощью кортежа $z = \langle e_1, e_2, \dots, e_t \rangle$

длины t событий e_i . Каждое из событий на сети образов описывает распознанный элемент в момент времени t_n , на сетях значений и личностных смыслов – ссылку на другой знак, признак или функцию агента. Описанная структура матрицы способствует разделению матриц на два типа: на объектные и процедурные. Процедурные матрицы описывают действия и процессы, и включают не пустое множество событий эффектов. Компоненты знака, как правило, описываются трехмерными матрицами, каждый слой которых является уникальным прецедентом распознавания объекта, либо описанием процедуры.

2.1. Формирование плана действий роботизированного агента

На стратегическом уровне архитектуры STRL присутствует рекурсивный цикл планирования, отвечающий за синтез состояний и действий агента в рамках процесса достижения поставленных целей. Агент рефлексивно представляет свои возможности в контексте психологически правдоподобной структуры – знака «Я», а возможности других агентов в виде знака «Они» (Рис. 2). Семантические составляющие знака «Я» способствуют представлению самосознания агента и описывают его функциональные возможности (на сети смыслов), ролевое, либо видовое представление агента в коллективе (на сети значений) и сенсорные качества при их наличии (на сети образов). Знак

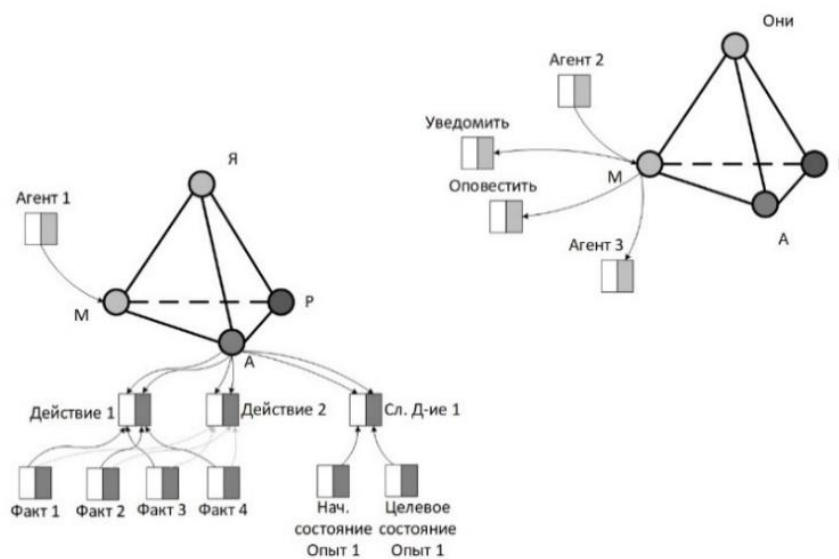


Рис. 2. Знаки «Я» и «Они»

«Они» представляет психологически правдоподобное рефлексивное представление агентов коалиции и способствует установлению протокола коммуникаций между ними, посредством активации знаков «Уведомить» и «Оповестить».

Реализация когнитивной функции планирования поведения в архитектуре STRL направлена на выявление последовательности абстрактных действий, основываясь на знаниях об ограничениях агентов. Далее план уточняется пространственными составляющими и воспроизводится на робототехнических платформах.

Синтез плана состоит из процедур пополнения картины мира данными о задаче, совершенения коммуникационных актов с другими агентами при их наличии и непосредственно рекурсивной процедуры поиска доступных планов [28, 29] (Рис. 3). Процедура поиска может быть вызвана как для синтеза плана в рамках классической нотации, так и для поиска поддействий при уточнении иерархической задачи. Вначале проверяется условие закливания алгоритма, которое может возникнуть при планировании от начальной ситуации к конечной. Далее задаётся глубина распространения активности по семиотической сети знаков, которая может варьироваться в рамках различных задач и вычисляется в цикле рассуждений. После этого генерируются матрицы возможных действий, множество которых может быть ограничено заранее (такая ситуация возникает

при уточнении иерархической задачи, при необходимости активации заранее заданного целевого состояния агента). Следующим шагом является отбор действий, применимых в настоящей ситуации. Далее следует итерация по всем отображенным действиям, которая начинается с попытки найти поддействия (актуально для сложных действий). Предпоследней процедурой алгоритма является синтез знака новой ситуации, который добавляется в план. Завершается итерация проверкой активации целевой ситуации, если она не активна, то процедура поиска плана рекурсивно вызывается ещё раз.

После окончания процесса генерации доступных планов агент производит отбор наиболее приемлемого для него плана по критериям минимизации длины конечного плана, минимизации задействованных в плане агентов (актуально для многоагентного случая, также рассматривается эвристика максимизации длины цепочки последовательных действий выполняемых каждым из агентов, которая требуется для снижения нагрузки в рамках использования протокола коммуникации агентов). После этого осуществляется процесс активации алгоритма сохранения опытных знаний. Алгоритм начинается с итерации по всем знакам, в рамках которой отбираются знаки не входящих в план ситуаций и удаляются их смыслы и образы. Далее знаки ситуаций, включенные в тупиковые ветви планов, изымаются из множества сохра-

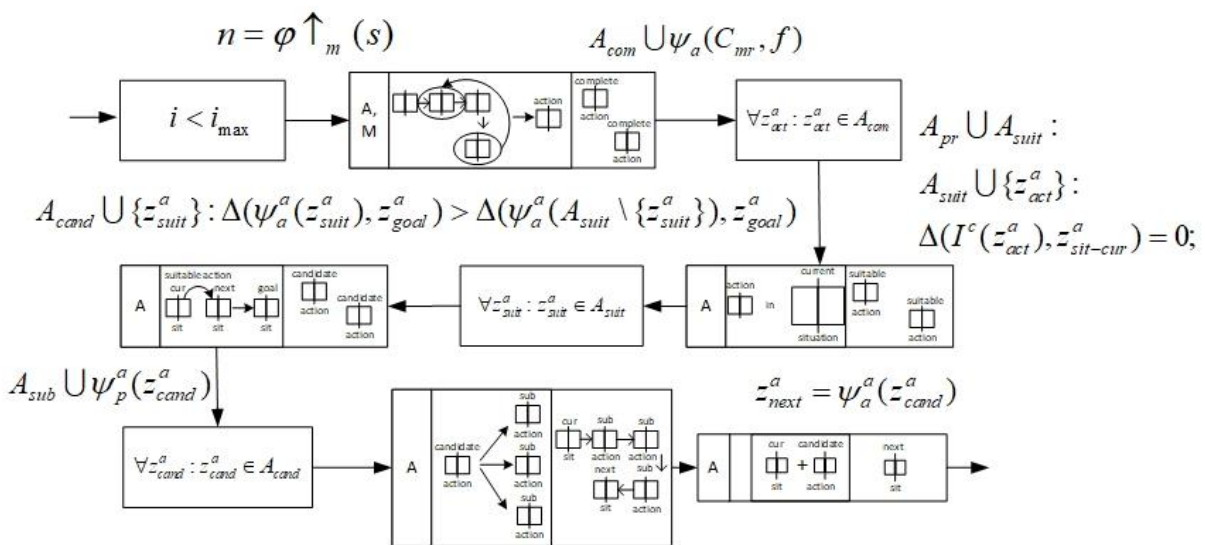


Рис. 3. Синтез плана действий агента с архитектурой STRL

няемых знаков. Отбираются знаки действий, у которых удаляются те матрицы смыслов, которые не входят в выбранный план. Также удаляются все смыслы и образы объектов, которые не входят в действия и ситуации плана. Процедура оканчивается созданием знака сложного действия, для которого на сети смыслов создаётся процедурная матрица, в множество условий которой входит ссылка на знак начальной ситуации плана, в множество эффектов – ссылка на знак целевой ситуации плана. На сети образов описывается список поддействий плана.

2.2. Пространственно-геометрическая интерпретация

Для пополнения картины мира агента знаниями об окружающей среде было использовано описанное в [29-30] знаковое представление псевдофизической логики Д.А. Пospelova [31], которое способствовало формализации знаний агента об объектах среды и их местоположении, описанном с помощью пространственных примитивов. Алгоритм использует два уровня представления среды – локальный, являющийся фокусом внимания агента (ту ситуацию, в рамках которой синтезируются непосредственные действия) и глобальный (текущее нечеткое представление агента о состоянии объектов и субъектов на карте). Фокус внимания агента состоит из 9 клеток, в центральной находится агент. Местоположению каждой из клеток соответствует направление (слева, справа, сверху,

слева-сверху...), которое позволяет описать положение объектов и пространство для перемещения внутри фокуса внимания. Размер фокуса внимания динамичен и рассчитывается исходя из рассуждений агента относительно загруженности окружающего пространства. При прохождении сильно загруженных мест, фокус внимания может уменьшиться, позволяя агенту перемещаться, аккуратно объезжая предметы. Глобальная карта местности состоит из 9 областей, размер которых статичен. Местоположение областей относительно агента рассчитывается динамически в процессе перемещений по карте и требуется для формализации описания направления к удаленным от агента объектам и субъектам (Рис. 4).

Пространственные отношения позволяют агенту динамически отображать расстояния и соответствуют психологически подобным рассуждениям человека относительно местоположения окружающих объектов. Для изменения размера фокуса внимания агента используются две процедуры – уточнение и абстрагирование. При уточнении, центральная клетка агента разделяется на 9 сегментов (если не достигнут минимальный размер клетки – диаметра агента) и увеличиваются пространственные отношения. В качестве примера можно рассмотреть отношения «близко-далеко» при деятельности агента в условиях абстрактного полигона в абстрактном городе. При описании местоположения агента, выезд из полигона будет иметь от-

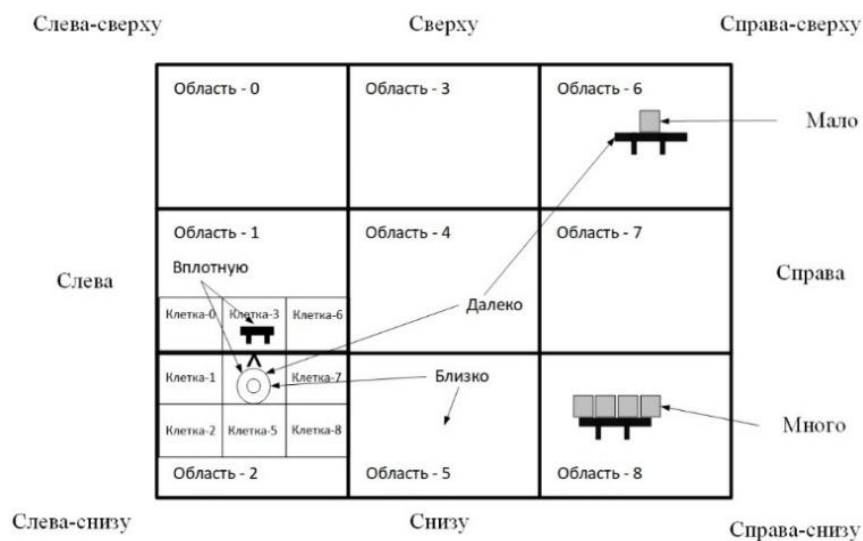


Рис. 4. Описание пространственных знаний агентом

ношение «близко», а перемещение агента к границе города (при условии расположения полигона в центре города) будет иметь отношение «далеко». Но при увеличении фокуса внимания до размеров страны (если она состоит больше, чем из 1 города) перемещение агента внутри города будет описываться отношением «близко», а перемещение между городами будет представлено отношением «далеко». Процедура абстрагирования описывает увеличение размера фокуса внимания агента и реализуется с помощью объединения клеток фокуса внимания агента в центральную клетку (если клетки пустые, так как центральная клетка должна быть заполнена только агентом и, если размер фокуса не достиг размера карты) и способствует уменьшению пространственных отношений.

Фокус внимания агента представлен знаком конкретизированной ситуации и представлен на сетях смыслов и образов описанием местоположения объектов относительно агента, также на сети образов фокус внимания имеет дополнительную каузальную матрицу позволяющую учитывать точные координаты клеток (Рис. 5). Подобным является описание знакового представление карты, состоящее из ссылок на знаки областей и их местоположение относительно агента.

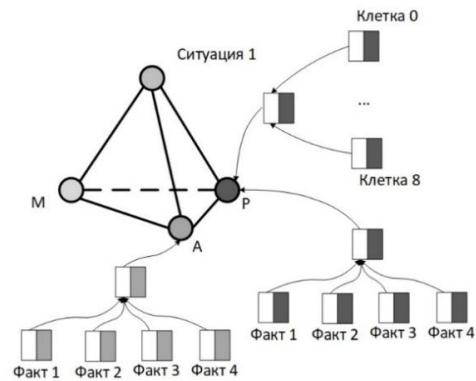


Рис. 5. Описание пространственной ситуации

Связь точных координат объектов со знаковым представлением осуществляется посредством процедуры означивания (по англ. Grounding Рис. 6), в рамках которой происходит считывание карты местности, разбиение карты на области и формирование первоначального фокуса внимания агента. Эта процедура работает и в обратную сторону – по окончании синтеза оптимального плана присутствует возможность осуществить проецирование знакового представления окружающей среды на точные координаты, по которым возможно осуществлять перемещение роботизированной платформы в условиях реальной среды.

```

1.  $plan, signs := GROUND(map, struct, plan)$  Function  $GROUND(map, struct, plan)$ :
2.  $signs = \{ \}$  # процедура возвращает текущее представление плана агентом и знаковое представление среды
3. For  $el$  in  $struct$ : # для каждого элемента описания задачи
4.  $signs \cup s = actsign(el)$  # алгоритм активирует знаковое представление
5.  $s_{focus}, s_{map} = markup(map, signs)$  # Генерируется знаковое представление фокуса внимания агента и знаковое представление карты.
6. If not  $plan$ :
7. Return  $\_, signs$ 
8. Else:
9. For  $s_{action}, s_{state}$  in  $plan$ : # если план уже синтезирован, то
10.  $coords = \varphi_p \downarrow (s_{state})$  # спускаемся вниз по сети образов и получаем координаты точные клеток
11.  $motor = \varphi_p \downarrow (s_{action})$  # спускаемся вниз по сети образов и получаем процедуры выполнения действий агента
12.  $plan\_review \cup \{coords, motor\}$  # обновляем представление плана
13. Return  $plan\_review, signs$ 

```

Рис. 6. Описание процедуры означивания

Первым этапом работы алгоритма пространственного планирования является расчёт абстрактной последовательности действий в рамках классического представления без учёта перемещений агента (поднять такой-то объект, сделать с ним что-то, поднять другой...), после этого агент пополняет свою картину мира пространственными знаниями о решаемой задаче и производит синтез более детального, пространственного плана. Процедура поиска планов дополнилась описанием начальной и конечной карты, активация которых позволяет отличить одинаковые фокусы внимания (пустые в простейшем случае). После выбора действий, которые наиболее быстро приведут к целевой ситуации в пространственном алгоритме происходит проверка условий необходимости уточнения настоящей ситуации (настоящая версия алгоритма настроена так, чтобы менять уровень абстракции фокуса внимания только в крайнем случае, так как это ресурсоемкая процедура). Основными условиями уточнения являются: отсутствие возможных действий (либо их нулевая эвристическая оценка, которая свидетельствует о том, что ни одно из синтезированных действий не способно приблизить целевую ситуацию), уровень абстракции целевой ситуации существенно ниже уровня абстракции настоящей ситуации. Далее следует проверка условий выполнимости процедуры абстрагирования, которая необходима для ускорения перемещения агента в слабозагруженной местности. Основными критериями для неудовлетворения условий применения процедуры являются присутствие объектов в фокусе внимания агента, присутствие объектов непосредственно по направлению перемещения агента при абстрагированном фокусе внимания и уточнение фокуса внимания на предыдущем шаге (Рис. 7).

После синтеза всех доступных пространственных планов агент запускает процедуру выбора наиболее приемлемого плана, идентичную процедуре, рассмотренной в разделе 3.1. Конечный пространственный план является цепочкой ситуаций и действий различного уровня абстракции. Действия по абстрагированию и уточнению ситуации также добавляются в план и имеют структуру схожую со структурой сложного действия – их каузальная матрица на сети смыслов

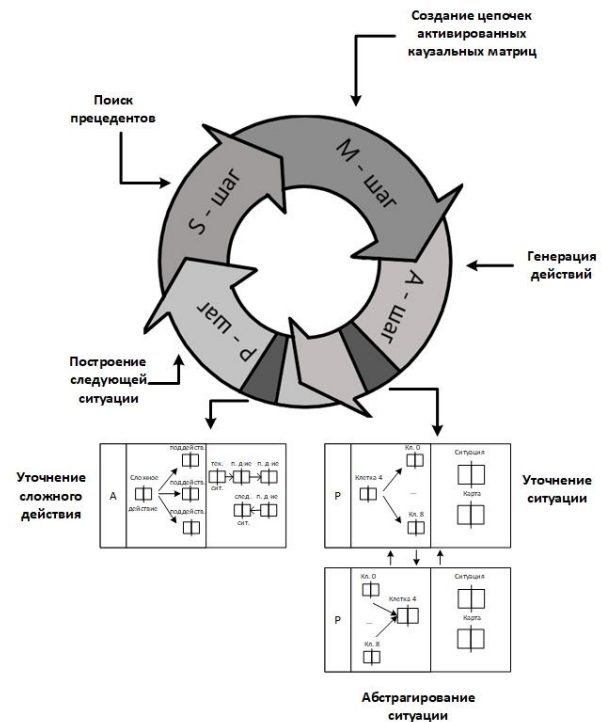


Рис. 7. Процесс синтеза пространственного плана

связывает ситуации различного уровня абстракции и позволяет заново выстроить логику действий агента. Алгоритм сохранения опыта агента дополняется процедурами, позволяющими находить и сохранять подпланы различного уровня абстракции. Итерация происходит по каждому из поддействий абстрактного плана. Если в подплане, который уточняет абстрактное действие существует как действия по уточнению ситуации, так и действия по абстрагированию, то сохраняются дополнительные подпланы между ними. Накопление опыта позволяет агенту переиспользовать сложные последовательности поддействий, например, при многократном прохождении загруженного объектами участка местности (проезд через узкий коридор, объезд препятствия).

2.3. Синтез многоагентного плана поведения

Процесс синтеза многоагентного плана начинается с выбора агента координатора, которым становится наиболее опытный агент. Опытность агента оценивается посредством минимизации новых знаний, которыми агент должен пополнить свою картину мира для решения поставленной задачи. После этого происходит процесс синтеза плана в рамках клас-

сической, не пространственной постановки. Синтез плана поведения осуществляется посредством генерации всех доступных планов агентами, с использованием их представлений об ограничении деятельности других агентов. Ограничения деятельности описаны согласно правилам описания поля :constraints языка разметки MLPDDL 3. В качестве примера ограничений можно рассмотреть 2 распространенных домена планирования – «мир блоков» и «логистика». В рамках первого домена, агенты будут иметь ограничения на подъем блоков определенного вида (некоторые агенты могут взаимодействовать сразу с несколькими видами блоков) и целью задачи является оптимальное выстраивание башни из блоков. Для второго домена ограничения задаются на перемещения агентов (например, самолет летает между городами, но не может переместиться за пределы аэропорта внутри города, а грузовики перемещаются только внутри городов). Целью задачи является перемещение грузов из начальных позиций в целевые, учитывая различные возможности средств перемещения по поднятию грузов. Каждый из агентов выбирает наиболее приемлемый по ряду вышеописанных критериев план и отправляет агенту координатору для аукциона планов. План, за который проголосовало наибольшее количество агентов дополняется агентом-координатором пространственными примитивами. Дополнение происходит с помощью создания пространственных начальных и конечных ситуаций для каждого из действий и отсылки кортежа из ситуаций и действия агенту, ответственному за выполнение действия. Агент синтезирует пространственный подплан из действий, ограниченных перемещениями, поворотами и необходимым присланным действием и отправляет синтезированный подплан обратно агенту координатору. Итогом цикла планирования является список подпланов, объединение которых описывает пространственный многоагентный план действий.

3. Адаптация STRL архитектуры к МП-РМ

Приведенный выше алгоритм планирования применен в решении робототехнических задач

по перемещению объектов на физическом полигоне. В качестве платформы для реализации робототехнического агента были выбраны платформы МП-РМ, имеющие следующие характеристики:

1. одноплатный микрокомпьютер – Raspberry Pi 3B;
2. количество ведущих двигателей: 2 или 4 в зависимости от модификации платформы;
3. тип ведущего двигателя: ДПТ с редуктором, 12в, 6000 об/мин;
4. понижающий коэффициент редуктора: 1:56;
5. драйвер управления двигателями: двухканальный на базе L298N;
6. аккумулятор: Литий-полимерный 3-х секционный 11,1В 2200 мА-ч;
7. манипулятор с пятью сервоприводами RDS3135;
8. актуатор с вакуумной помпой;
9. RPLidar A2M4.

Выбор робототехнической платформы обусловлен наличием платформы МП-РМ, в ФИЦ "Информатика и управление", которая обладает необходимым набором функций для реализации предложенной архитектуры, алгоритмов и методов.

Взаимодействие с колесной базой происходит с помощью языка Python и библиотеки GPIO, функционал которой позволяет управлять модулем драйвера L298N для двигателей постоянного тока. Унифицированность интерфейса взаимодействия с драйвером позволяет передавать сигналы по типу «Высокий» и «Низкий» на каждый привод. Регулировка скорости вращения моторов производится посредством подачи на выходы ШИМ сигнала. Поворот платформы осуществляется за счет выбора направления вращения колесной базы. Основной проблемой разработки сценария перемещений была неоднозначность выбора IN для соединения с микроконтроллером.

Управление предустановленным манипулятором платформы осуществляется библиотекой Adafruit_PCA9685, контролирующей подачу ШИМ сигнала на сервоприводы манипулятора, включение и выключение помпы происходит за счет GPIO. В процессе экспериментов, вакуумная помпа показала низкую эффективность при взаимодействии с пористыми объектами (бума-

га, дерево) и на некоторых платформах была заменена на актуатор-зажим. Также была выявлена проблема отсутствия обратной связи сервоприводов RDS3135. При отсутствии обратной связи от сервопривода невозможно осуществить обучение манипулятора динамическому поведению для взаимодействия с различными типами объектов без дополнительного оборудования. Проблема может быть частично решена наличием двух камер, фиксирующих положение актуатора – камеры на самом манипуляторе и камеры на корпусе, направленной на объект. Камера в задней части робота (входящей в зону работы манипулятора) отсутствовала, а получение координат с камеры в передней части робота и поворот на месте давали ошибочную картину местоположения объекта. Также была выявлена проблема работы с объектами, находящимися на возвышенности – столе, полке или же верхними блоками в башне из блоков. Камера корпуса робота не охватывала их местоположение, а поиск объектов с помощью камеры манипулятора осуществить не удалось, по причине малого рабочего времени сервоприводов. Решением этих проблем является коалиционное взаимодействие роботов, в рамках которого описание местоположения объектов осуществляется посредством робота с более высоким расположением камеры, либо при использовании БПЛА.

Описанная выше схема робота позволила создать набор скриптов (Рис. 8), отвечающих за реализацию действий по перемещению и взаимодействию с окружающими объектами. Перемещение робота осуществляется подачей ШИМ сигнала заранее заданной частоты, расстояние контролируется временем подачи сигнала. Отсутствие датчиков контроллеров колесной базы повлияло на точность перемещения робота, ошибка была наибольшей при повороте на месте. Было установлено, что ошибка вызвана разницей трения колес о материалы пола (где-то резина, где-то дерево), корректировка выполнена за счет взаимодействия с aruko-метками. Распознавание aruko-меток происходит с помощью ROS-пакета `clever` и библиотеки `roslibpy`, позволяющей считывать телеметрию динамически посредством обращения к соответствующему ROS-сервису. Робототехническая платформа выполняет действие по

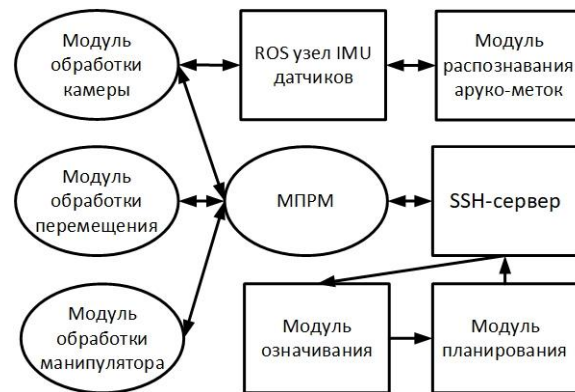


Рис. 8. Схема реализации STRL-архитектуры на МП-PM

перемещению и повороту, далее сверяет координаты с телеметрией и выполняет дополнительные перемещения пока положение платформы не станет соответствовать спланированному. Итерационный процесс вычисления направления робота может быть заменен на динамический, в рамках которого платформа будет автоматически изменять угловую скорость в процессе выполнения действия, а не после его окончания. Динамический процесс возможно осуществить при замене бортового одноплатного компьютера Raspberry более мощными аналогами (например, Nvidia Jetson). Был проведен эксперимент по определению направления робота с помощью магнитометра, а также камеры, распознающей метки направлений. Данные магнитометра были противоречивыми и зависели от местоположения платформы, а распознавание меток, находящихся в отдалении по горизонтали от робота, не было точным и происходило с опозданием, в связи с невысокой вычислительной мощностью микроконтроллера.

В рамках реализации экспериментов с робототехнической платформой был создан протокол коммуникаций, основанный на SSH-сервере, запуск которого осуществим на любой из выбранных платформ (например, на той, которая была выбрана в качестве агента-координатора). При исполнении протокола агентом-координатором производится деятельность по интерпретации синтезированных действий планировщика в команды роботам, деятельность по обработке ошибок выполнения плана и по запуску перепланирования в случае возникновения необрабатываемого исключения

при исполнении действий. Доступ к каждому из управляющих контроллеров роботов производится по SSH каналу с помощью библиотеки paramiko. При замене управляющих контроллеров на более мощные может быть настроен доступ в рамках ROS модели коммуникаций, wifi-ssh соединение может быть заменено на более устойчивое к помехам радио-сообщение, при сохранении логической составляющей модуля. Все эксперименты были проведены из расчёта реализации поведения агента, синтезированного только планировщиком поведения STRL-архитектуры, не были учтены временные ограничения действий (выполнение действий не распараллелено), взаимодействие с объектами осуществлялось по заранее заданным сценариям манипулятора (поиск оптимального сценария взаимодействия происходил вне экспериментов).

С использованием роботизированной платформы МП-РМ были осуществлены запуски приведенных в разделе 4 пространственных экспериментов. Исполнение синтезированного плана позволило достичь целевого состояния в 80 процентах случаев, в оставшихся 20 процентах исполнению помешали сбои в работе SSH соединений роботов и отсутствие точных данных о местоположении робота.

3. Экспериментальное исследование

В рамках демонстрации возможностей по синтезу плана алгоритмом планирования STRL-

архитектуры был проведен ряд экспериментов с задачами в иерархической постановке, с многоагентным способом планирования и с планированием с псевдофизической пространственной логикой. Работоспособность алгоритма была оценена по времени выполнения задач и по затраченной памяти. Эксперименты проводились на ноутбуке ASUS ZenBook с характеристиками: Intel® Core™ i7-8565U CPU @ 1.80GHz 1.99GHz 16Gb ОЗУ 1Tb SSD.

На Рис. 9 представлены результаты работы алгоритма планирования архитектуры STRL при решении задач в контексте иерархического способа задания цели и приведено его сравнение с популярным планировщиком PANDA (слева) и результаты синтеза многоагентного плана с ограничениями на деятельность каждого из агентов в контексте MAPDDL 3 справа. Планировщик PANDA был выбран для сравнения, так как является одним из немногих и самым широкоизвестным планировщиком иерархических задач планирования, который использует описание задачи и домена планирования на языке PDDL 3 и свободно распространяет дистрибутив планировщика. Дистрибутив планировщика PANDA был получен с сайта разработчиков. Для описания примера деятельности многоагентного планировщика также приведены результаты работы планировщика как с использованием опыта, так и без него. Количество экспериментов обусловлено демонстрацией возможностей алгоритма планирования поведения когнитивной архитектуры STRL находить решение не только когнитивно-

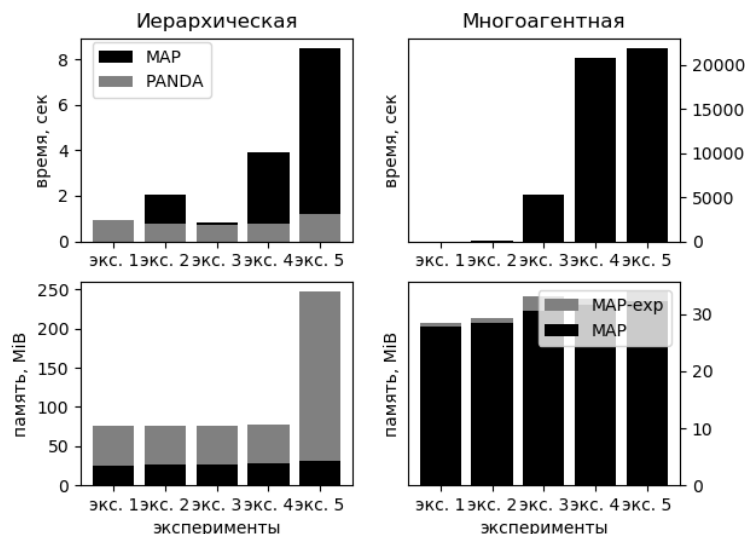


Рис. 9. Синтез плана иерархических и многоагентных задач планирования

сложных пространственных задач, но и задач классического планирования действий, затрачивая сравнимое количество ресурсов с известным планировщиком.

Исходя из гистограммы следует, что алгоритм планирования поведения STRL затрачивает больше времени на синтез плана, но расходует меньше памяти, чем планировщик PANDA, который используется в случаях реального производства и имеет заранее скомпилированный код. Также, описываемый алгоритм поддерживает многоагентную версию как иерархического, так и классического планирования, оценка результатов которых была отражена на рисунке справа. В рамках рассматриваемых примеров для многоагентного случая были использованы задачи из домена «Мир блоков» и задачи перемещения груза внутри многокомнатного помещения для тестирования синтеза плана при иерархической постановке. Эти домены были выбраны исходя из близости рассматриваемых задач к реальным робототехническим задачам. Сложность задач заключалась в увеличении количества блоков, грузов и комнат. Более детальное описание присутствует по ссылкам [32-34] в разделе «benchmarks».

Далее (Рис. 10) представлены примеры решаемых агентом задач в случае геометрического (пространственного) планирования. В левой части (1) отображена одноагентная задача, в рамках которой агенту требовалось доехать до блока «б» и переместить его к блокам «а» и «в», а целью являлось составление башни «а-б-в». В случае отсутствия опыта агент затратил 243,3 секунды на синтез плана и 25,6 MiB памяти, в случае с уже имеющимся опытом, агент потратил 50,5 секунды для синтеза плана. В правой части (2) отображена задача, в которой взаимодействовали 2 агента: первый расчистил путь второму, переместив мешающий проезду блок большого размера в отведенное ему место, а агент с более точным манипулятором доставил в ранее отгороженную область требуемый предмет. На синтез многоагентного плана было затрачено 3191,4 секунды и 110,8 MiB памяти без опыта и 48,2 секунды и 150 MiB памяти в случае имеющегося опыта. Уточненные действия по перемещению и взаимодействию были в картине мира соответствующих агентов, каждый агент имел только

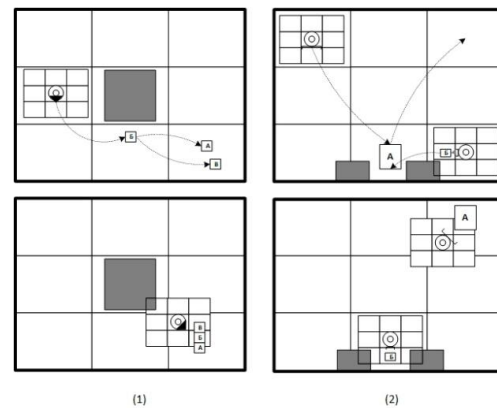


Рис. 10. Пример одноагентного и многоагентного решения геометрической задачи планирования

абстрактное представление о возможностях кооператора. Описываемые эксперименты были выбраны в качестве демонстрации работы всех имеющихся модулей архитектуры STRL, которые позволили синтезировать планы поведения, выбрать наиболее приемлемый план в многоагентном случае, оценить собственные местоположения агентов, объезжать препятствия, придерживаться заранее спланированному курсу, взаимодействовать с блоками и сохранять успешность синтезированной и реализованную последовательность действий в качестве опыта планирования.

Исходя из вышеизложенных примеров следует, что алгоритм планирования STRL архитектуры может быть использован на робототехнических платформах малой мощности, не требовательных к временным затратам (Рис. 11). Использование накопленного опыта предоставляет возможность существенно снизить время на синтез плана поведения агентов и осуществить оптимальный выбор стратегии взаимодействия между агентами.

Заключение

В статье рассмотрены основные особенности программной реализации систем управления когнитивным агентом с STRL архитектурой, которая представлена в виде интеллектуальной системы планирования поведения коалиции робототехнических агентов. Рассмотрен иерархический метод синтеза плана и его дополнение пространственными примитивами. Результаты реализации интеллектуальной системы на базе робототех-



Рис. 11. МП-РМ на робототехническом полигоне

нических платформ МП-РМ Зарница показали эффективность предложенных решений по сравнению с планировщиком PANDA. В дальнейших исследованиях агент с архитектурой STRL будет дополнен уточнением действий до минимального уровня абстракции за счет синтеза поддействий алгоритмами обучения с подкреплением. Будут рассмотрены способы динамической генерации эвристики планирования и эвристики выбора коалиции в многоагентном случае. Предполагается, что применение сценариев поведения агента увеличит скорость синтеза плана за счет уменьшения требований к точности описания текущей ситуации.

Литература

- Emelyanov S. and etc. (2015) Multilayer cognitive architecture for UAV control. *Cognitive System Research*, 34.
- 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.
- <https://zarnitza.ru/catalog/mekhatronika-i-robototekhnika/robototekhnika/sredne-spetsialnye-uchebnye-zavedeniya/ustanovka-po-izucheniyu-mobilnyx-robotizirovannyx-sistem-na-baze-mobilnoj-platformy-i-robot-manipulyatora-mp-rm-104/>
- Osipov, G.S., Panov, A.I., Chudova, N. V.: Behavior control as a function of consciousness. *I. World model and goal setting. J. Comput. Syst. Sci. Int.* 53, 517529 (2014).
- Bechon, P., Barbier, M., Grand, C., Lacroix, S., Lesire, C., & Pralet, C. (2018). Integrating planning and execution for a team of heterogeneous robots with time and communication constraints. 1091–1097.
- Benjamin, D. P., Li, T., Shen, P., Yue, H., Zhao, Z., & Lyons, D. (2018). Spatial understanding as a common basis for human-robot collaboration. *Advances in Intelligent Systems and Computing.* https://doi.org/10.1007/978-3-319-60384-1_3J.
- R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, Y. Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060, 2004.
- Thome, R., Fedor, C., Sustersic, J., & Ukah, U. (2015). Non-linear Decision Making for Robust Navigation in Role Based Autonomy.
- Davis, D. N., & Ramulu, S. K. (2017). Reasoning with BDI robots: From simulation to physical environment - Implementations and limitations. *Paladyn*, 8(1), 39–57. <https://doi.org/10.1515/pjbr-2017-0003>
- Madl, T., Franklin, S., Chen, K., & Trappl, 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>
- H. Younes and R. Simmons, “VHPOP: Versatile heuristic partial order planner,” *JAIR*, vol. 20, pp. 405–430, 2003.
- R. Dechter, I. Meiri, and J. Pearl, “Temporal Constraint Networks,” *Artificial Intelligence*, vol. 49, no. 1-3, pp. 61–95, 1991.
- Kim, B., Wang, Z., Kaelbling, L. P., & Lozano-Pérez, T. (2019). Learning to guide task and motion planning using score-space representation. *International Journal of Robotics Research*, 38(7), 793–812. <https://doi.org/10.1177/0278364919848837>
- Berenson D, Abbeel P and Goldberg K (2012) A robot path planning framework that learns from experience. In: *IEEE Conference on Robotics and Automation*.
- Dantam NT, Kingston Z, Chaudhuri S and Kavraki L (2017) Incremental task and motion planning: A constraint-based approach. In: *Robotics: Science and Systems*.
- Dey D, Liu TY, Sofman B and Bagnell JA (2012 a) Efficient optimization of control libraries. In: *AAAI Conference on Artificial Intelligence*.
- Hoda’l J and Dvor’ak J (2008) Using case-based reasoning for mobile robot path planning. *Journal of Engineering Mechanics* 15(3): 181–191.
- Jetchev N and Toussaint M (2013) Fast motion planning from experience: Trajectory prediction for speeding up movement generation. *Autonomous Robots* 34(1–2): 111–127.
- Kaelbling LP and Lozano-Pérez T (2013) Integrated task and motion planning in belief space. *The International Journal of Robotics Research* 32(9–10): 1194–1227.

19. Kim B, Kaelbling LP and Lozano-Pe' rez T (2017) Learning to guide task and motion planning using score-space representation. In: IEEE Conference on Robotics and Automation.
20. Осипов Г.С., Панов А.И. Отношения и операции в знаковой картине мира субъекта поведения // Искусственный интеллект и принятие решений. 2017. № 4.
21. Panov A.I. Behavior Planning of Intelligent Agent with Sign World Model // Biol. Inspired Cogn. Archit. 2017. Vol. 19. P. 21–31.
22. Osipov, G.S., Panov, A.I., Chudova, N. V.: Behavior control as a function of consciousness. I. World model and goal setting. J. Comput. Syst. Sci. Int. 53, 517529 (2014).
23. Osipov, G.S., Panov, A.I., Chudova, N. V.: Behavior Control as a Function of Consciousness. II. Synthesis of a Behavior Plan. J. Comput. Syst. Sci. Int. 54, 882896 (2015).
24. Osipov, G.S.: Sign-based representation and word model of actor. In: Yager, R., Sgurev, V., Hadjiski, M., and Jotsov, V. (eds.) 2016 IEEE 8th International Conference on Intelligent Systems (IS). pp. 2226. IEEE (2016).
25. Леонтьев А.Н. Деятельность. Сознание. Личность. М.: Политиздат, 1975.
26. Брунер Дж. Психология познания. За пределами непосредственной информации. М.: Прогресс, 1977. 413 с.
27. Панов, А. И. (2018). Формирование образной компоненты знаний когнитивного агента со знаковой картиной мира. Информационные технологии и вычислительные системы, (4).
28. Киселёв Г. А., & Панов, А. И. (2017). Знаковый подход к задаче распределения ролей в коалиции когнитивных агентов. Труды СПИИРАН. <https://doi.org/DOI/10.15622/sp.57.7>
29. Андрейчук А. А., Киселев Г. А., & Яковлев, К. С. (2019). Интеграция Методов Планирования Поведения и Планирования Траектории. Труды 17 Национальной Конференции с Международным Участием КИИ-2019. Том 1, стр 66-75.
30. Kiselev, G., & Panov, A. (2019). Hierarchical psychologically inspired planning for human-robot interaction tasks. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). https://doi.org/10.1007/978-3-030-26118-4_15.
31. Поспелов Д.А. Ситуационное управление: теория и практика. М.: Наука, 1986. 288 с.
32. <https://github.com/cog-isa/map-core.git>
33. <https://github.com/cog-isa/map-multi.git>
34. <https://github.com/cog-isa/map-spatial.git>

Киселёв Глеб Андреевич. Институт Проблем Искусственного Интеллекта Федерального государственного учреждения "Федеральный исследовательский центр "Информатика и управление" Российской академии наук" г. Москва, Россия. Младший научный сотрудник. Количество печатных работ: 10. Область научных интересов: когнитивные архитектуры, планирование поведения, многоагентные коммуникации. E-mail: kiselev@isa.ru

Intelligent Behavior Planning System for a Coalition of Robotic Agents with STRL Architecture

G. A. Kiselev

The Institute of Problems of Artificial Intelligence of the Federal State Institution "Federal Research Center "Computer Science and Control "of the Russian Academy of Sciences" Moscow, Russia.

Abstract. This work is devoted to the issues of software implementation of the STRL architecture of a cognitive agent for a group of robotic platforms. The problem of the synthesis of coalitional and individual spatial plans of agent behavior is considered. The results of the adaptation of the methods of the theory of the sign world model when constructing hierarchical control systems based on the mobile platform MP-RM Zarnitza are presented. A number of experiments were conducted to build joint coalition plans, including actions for moving in space and for manipulating objects.

Keywords: cognitive agents, hierarchical planning, geometric planning, pseudophysical logics, sign approach, coalitions of agents, robotics.

DOI 10.14357/20718632200203

References

1. Emelyanov S. and etc. (2015) Multilayer cognitive architecture for UAV control. Cognitive System Research, 34.
2. Makarov D. A, Panov A. I., Yakovlev K. S. STRL: multi-level control intelligent agents system // In the book: Fifteenth National Conference on Artificial Intelligence with international participation КИИ-2016 (October 3-7, 2016, Smolensk, Russia): Proceedings of the conference T. 1. Smolensk: Universum, 2016.S. 179-188.
3. <https://zarnitza.ru/catalog/mekhatronika-i-robototekhnika/robototekhnika/sredne-spetsialnye-uchebnye-zavedeniya/ustanovka-po-izucheniyu-mobilnyx-robotizirovannyx-sistem-na-baze-mobilnoj-platformy-i-robot-manipulyatora-mp-rm-104/>
4. Osipov, G.S., Panov, A.I., Chudova, N. V.: Behavior control as a function of consciousness. I. World model and goal setting. J. Comput. Syst. Sci. Int. 53, 517529 (2014).

5. Bechon, P., Barbier, M., Grand, C., Lacroix, S., Lesire, C., & Pralet, C. (2018). Integrating planning and execution for a team of heterogeneous robots with time and communication constraints. 1091–1097.
6. Benjamin, D. P., Li, T., Shen, P., Yue, H., Zhao, Z., & Lyons, D. (2018). Spatial understanding as a common basis for human-robot collaboration. *Advances in Intelligent Systems and Computing*. https://doi.org/10.1007/978-3-319-60384-1_3J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, Y. Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060, 2004.
7. Thome, R., Fedor, C., Sustersic, J., & Ukah, U. (2015). Non-linear Decision Making for Robust Navigation in Role Based Autonomy.
8. Davis, D. N., & Ramulu, S. K. (2017). Reasoning with BDI robots: From simulation to physical environment - Implementations and limitations. *Paladyn*, 8(1), 39–57. <https://doi.org/10.1515/pjbr-2017-0003>
9. 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>
10. H. Younes and R. Simmons, “VHPOP: Versatile heuristic partial order planner,” *JAIR*, vol. 20, pp. 405–430, 2003.
11. R. Dechter, I. Meiri, and J. Pearl, “Temporal Constraint Networks,” *Artificial Intelligence*, vol. 49, no. 1-3, pp. 61–95, 1991.
12. Kim, B., Wang, Z., Kaelbling, L. P., & Lozano-Pérez, T. (2019). Learning to guide task and motion planning using score-space representation. *International Journal of Robotics Research*, 38(7), 793–812. <https://doi.org/10.1177/0278364919848837>
13. Berenson D, Abbeel P and Goldberg K (2012) A robot path planning framework that learns from experience. In: *IEEE Conference on Robotics and Automation*.
14. Dantam NT, Kingston Z, Chaudhuri S and Kavraki L (2017) Incremental task and motion planning: A constraint-based approach. In: *Robotics: Science and Systems*.
15. Dey D, Liu TY, Sofman B and Bagnell JA (2012 a) Efficient optimization of control libraries. In: *AAAI Conference on Artificial Intelligence*.
16. Hoda'1 J and Dvor'ak J (2008) Using case-based reasoning for mobile robot path planning. *Journal of Engineering Mechanics* 15(3): 181–191.
17. Jetchev N and Toussaint M (2013) Fast motion planning from experience: Trajectory prediction for speeding up movement generation. *Autonomous Robots* 34(1–2): 111–127.
18. Kaelbling LP and Lozano-Pérez T (2013) Integrated task and motion planning in belief space. *The International Journal of Robotics Research* 32(9–10): 1194–1227.
19. Kim B, Kaelbling LP and Lozano-Pérez T (2017) Learning to guide task and motion planning using score-space representation. In: *IEEE Conference on Robotics and Automation*.
20. G.S. Osipov, A.I. Panov Relationships and Operations in a Sign-Based World Model of the Actor // *Scientific and Technical Information Processing*. 2018. No. 5.
21. Panov A.I. Behavior Planning of Intelligent Agent with Sign World Model // *Biol. Inspired Cogn. Archit.* 2017. Vol. 19. P. 21–31.
22. Osipov, G.S., Panov, A.I., Chudova, N. V.: Behavior control as a function of consciousness. I. World model and goal setting. *J. Comput. Syst. Sci. Int.* 53, 517529 (2014).
23. Osipov, G.S., Panov, A.I., Chudova, N. V.: Behavior Control as a Function of Consciousness. II. Synthesis of a Behavior Plan. *J. Comput. Syst. Sci. Int.* 54, 882896 (2015).
24. Osipov, G.S.: Sign-based representation and word model of actor. In: Yager, R., Sgurev, V., Hadjiski, M., and Jotsov, V. (eds.) 2016 IEEE 8th International Conference on Intelligent Systems (IS). pp. 2226. IEEE (2016).
25. Leontiev A.N. Activity Consciousness. Personality. M.: Politizdat, 1975.
26. Bruner J. Psychology of knowledge. Outside of direct information. M.: Progress, 1977. 413 s.
27. Panov, A.I. (2018). The image component formation of knowledge of a cognitive agent with sign world model. *Information technology and computer systems*, (4).
28. Kiselev, G.A., Panov, A.I.: Sign-based Approach to the Task of Role Distribution in the Coalition of Cognitive Agents. In: *SPIIRAS Proceedings*, pp. 161–187 (2018).
29. Andreichuk, A.A., Kiselev, G.A., & Yakovlev, K.S. (2019). Integration of Behavioral Planning Methods and Trajectory Planning. *Proceedings of the 17th National Conference with the International Participation of KII-2019. Volume 1*, pp. 66-75.
30. Kiselev, G., & Panov, A. (2019). Hierarchical psychologically inspired planning for human-robot interaction tasks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-030-26118-4_15.
31. Pospelov D.A. Situational management: theory and practice. M.: Nauka, 1986. 288 s.
32. <https://github.com/cog-isa/map-core.git>
33. <https://github.com/cog-isa/map-multi.git>
34. <https://github.com/cog-isa/map-spatial.git>

Kiselev G. A. The Institute of Problems of Artificial Intelligence of the Federal State Institution "Federal Research Center" Computer Science and Control "of the Russian Academy of Sciences" Moscow, Russia. Junior Researcher. Number of publications: 10. Field of scientific interests: cognitive architecture, behavior planning, multi-agent communications. E-mail: kiselev@isa.ru