

Karsaev Oleg Vladislavovich

Institution of the Russian Academy of Science St. Petersburg Institute for Informatics and Automation RAS.

E-mail: ok@iias.spb.su.

39, 14 line, Saint Petersburg, 199178, Russia.

Phone: +78123233570.

Head of Laboratory; Cand. of Eng. Sc.

Usupov Rafael Midhatovich

E-mail: ok@iias.spb.su

Corresponding Member of RAS, Director of SPIIRAS.

Skobelev Petr Olegovich

Institute of Control of Complex systems of Russian Academy of Science.

E-mail: petr.skobelev@gmail.com.

61, Sadovaya Street, Samara, 443020, Russia.

Phone: +78463323927.

Professor of Academy of Volga State University of Telecommunication and Informatics, Senior Scientists of Intelligent Systems Laboratory.

УДК 681.3

В.И. Городецкий, С.В. Серебряков, Д.В. Троцкий

**СРЕДСТВА СПЕЦИФИКАЦИИ И ИНСТРУМЕНТАЛЬНОЙ ПОДДЕРЖКИ
КОМАНДНОГО ПОВЕДЕНИЯ АВТОНОМНЫХ АГЕНТОВ***

Приводится краткий анализ состояния исследований в области теории, моделей, программных инфраструктур и языков, предназначенных для проектирования и программной реализации командного поведения автономных агентов, и предлагается высокоуровневый язык спецификации командной работы агентов и кратко описываются предметно-независимые программные инструментальные средства, поддерживающие функционирование автономных команд агентов. Предложенный язык и разработанная инфраструктура демонстрируются на примере программного прототипа командной работы автономных подводных роботов, выполняющих миссию по патрулированию трехмерного подводного пространства морского порта с целью обеспечения его безопасности.

Автономные агенты; командная работа; сценарные знания; целенаправленное планирование; автономные подводные роботы.

V.I. Gorodetsky, S.V. Serebryakov, D.V. Trotsky

**KNOWLEDGE-BASED SPECIFICATION LANGUAGE AND REUSABLE
SOFTWARE SUPPORTING AUTONOMOUS AGENTS' TEAMWORK**

The paper analyses the existing theories, models, software infrastructures and teamwork-oriented programming languages in order to match their expressive capabilities to the advanced frameworks and infrastructures and proposes high-level knowledge-based teamwork specification language and reusable domain-independent software components intended to support the development and implementation of autonomous agent teamwork systems. The developed language and reusable software are validated through the software prototyping the teamwork of autonomous underwater robots patrolling the 3D underwater space of a sea port.

Autonomous agents; teamwork; scenario knowledge; goal-oriented planning; autonomous underwater robots.

* Работа поддерживается грантом РФФИ № 10-07-00026а.

1. Введение. В основе понятия автономности агента лежит его способность самостоятельно анализировать ситуацию и на этой основе выбирать и реализовывать стратегию поведения, которая приводит к наилучшему выполнению миссии, возложенной на него. В зависимости от типа приложения понятие миссии агента может варьироваться. Наиболее часто под миссией автономного агента понимается достижение одной или нескольких целей, которые могут отождествляться с достижением одного или нескольких целевых состояний. Такое понимание миссии агента используется в данной работе. Подчеркнем, что свойство автономности агента означает, что извне ему может только ставиться задача. Выбор стратегии ее решения относится к области компетенции самого агента. Аналогично понимается миссия коллектива автономных агентов с тем отличием, что в этом случае она подразумевает достижение одной или многих целей (целевых состояний) их совместными скоординированными усилиями.

В общем случае существуют различные ситуации, когда агенты должны координировать свое поведение. Обычно коллективное поведение агентов определяется дополнительными соглашениями, которые принято называть *взаимными обязательствами (commitments)*, и именно они определяют тип объединения агентов в группу, среди которых принято различать альянсы, коалиции и команды. Наиболее сильными являются взаимные обязательства *команды агентов*. В этом случае агенты решают общую задачу, а команда действует как один агент. При этом ни у одного агента нет собственных *глобальных* целей, отличных от целей команды, а динамически формируемые локальные цели агентов должны быть направлены на достижение общих целей команды.

Данная работа посвящена моделям *командной работы автономных агентов* при выполнении некоторой миссии без внешнего вмешательства. Команда агентов должна самостоятельно решать такие задачи, как построение плана выполнения миссии путем декомпозиции глобальной цели на частные подцели и распределения частных задач по их достижению между агентами команды. Она должна выполнять мониторинг процесса работы и отслеживать зависимости между задачами, решаемыми членами команды, обеспечивать их синхронизацию, своевременно предсказывать и разрешать конфликты, нарушения ограничений и обрабатывать нештатные ситуации. Все эти задачи должны решаться в реальном времени и в распределенном варианте, а информация, которой обмениваются агенты команды, должна постоянно поддерживать *самоорганизующийся* стиль исполнения командной работы [21].

Можно привести много практически важных задач, которые в настоящее время могут быть эффективно решены только с использованием модели командной работы автономных физических и программных агентов. Примерами являются командная работа автономных роботов, охраняющих подводное пространство порта, роботов военного назначения, решающих совместно различные задачи на театре военных действий, команд беспилотных летательных аппаратов наблюдения, решающих различные задачи противоборства в киберпространстве и другие.

Сложности практической реализации идей командной работы автономных агентов, например, в приложениях упомянутого выше типа обусловлены многими факторами, в частности, такими, как:

- ◆ непредсказуемая динамика внешней среды, сознательное противодействие соперничающей стороны (если она имеется);
- ◆ ограниченность знаний каждого агента о внешней среде и о других агентах команды, что влечет необходимость распределенной оценки текущей ситуации;

- ◆ многообразии вариантов сценария коллективного поведения и распределения ролей между агентами при выполнении миссии, зависимость возможных сценариев от предыстории работы, от наличия ресурсов и специализации агентов, что обуславливает необходимость перераспределения ролей в реальном времени;
- ◆ динамика сетевой организации команды агентов, когда коммуникационные средства имеют ограниченную дистанцию достижимости, что требует наличия специальной программной инфраструктуры для поддержки взаимодействия агентов и специальных коммуникационных протоколов.

Проблемы командного поведения автономных агентов и, в частности, командного поведения агентов, управляющих автономными роботами, находятся в центре внимания исследователей и разработчиков в области искусственного интеллекта и робототехники, начиная с конца 1980-х гг., когда была опубликована фундаментальная работа [4]. Эта работа и по настоящее время во многом определяет направления исследований в данной области.

Центральными задачами в области командной работы являются задачи динамического планирования и перепланирования многошагового процесса выполнения миссии команды, динамического распределения ролей и подзадач между ее членами в контексте текущей ситуации, а также задачи координации поведения агентов команды. Эти задачи чрезвычайно сложны, и, несмотря на большие усилия многих исследователей, нуждаются в дальнейших разработках.

Анализ известных результатов в рассматриваемой области показывает, что, несмотря на хорошо разработанную теорию командной работы, ее практическая реализация встречается с большими трудностями. Они обусловлены, наряду с иными причинами, слабостью выразительных средств языков формального описания планов совместной работы агентов команды в динамической среде. Построенный план всегда требуется пересматривать в динамике его исполнения с учетом текущего контекста, состояния внешней среды, агентов команды и действий оппонировающей команды при ее наличии.

Данная работа преследует три основные цели. Первая из них – это краткий критический обзор существующих теорий командной работы, разработанных моделей и предметно-независимых инструментальных средств их поддержки, а также языков спецификации конкретных приложений. Вторая цель – дать краткие сведения о разработанном высокоуровневом языке спецификации командной работы автономных агентов, основанном на знаниях, и о механизме целенаправленного планирования на основе простого механизма синтаксического вывода. Третья цель – описать разработанные предметно-независимые программные средства поддержки командной работы автономных агентов. Полученные в работе результаты демонстрируются на примере работы команды автономных агентов, управляющих поведением подводных роботов, решающих задачу патрулирования трехмерного подводного пространства морского порта.

Дальнейшие четыре раздела работы (разделы 2–5) посвящены последовательному рассмотрению перечисленных выше проблем. В заключении отмечаются новые результаты, а также планируемые исследования.

2. Анализ состояния исследований в области автономной командной работы. Основная особенность командной работы автономных агентов состоит в том, что они должны функционировать как один агент в ситуации, когда каждый агент имеет ограниченную информацию о собственной команде, о сопернике и о внешней среде и реализует *собственные намерения* с помощью *индивидуальных действий*, выполняемых параллельно или последовательно с действиями других агентов. Существующие теории и модели командной работы направлены на решение возникающих при этом проблем. Рассмотрим их кратко.

2.1. Теории командной работы. К настоящему времени существует несколько теорий командной работы автономных агентов. Одна из них – это "Теория общих намерений", которая предложена в фундаментальной работе [4]. Она формулирует базовые понятия и общие рамки, определяющие командное поведение, характер взаимодействия и информационного обмена членов команды. Вторым подходом – это "Теория общих планов", предложенная в работах [9], [10]. Некоторые базовые положения, связанные с понятиями общих намерений, обязательств и соглашений, были сформулированы в работах [11] и [12].

Теория общих намерений [4] оперирует двумя типами понятий. К первому типу относятся так называемые *ментальные* понятия, в число которых входят *события, убеждения, цели и взаимные убеждения агентов*. Понятия второго типа используются для описания *поведения команды агентов*. К ним относятся *командная (общая) цель, индивидуальные обязательства и соглашения агента, индивидуальные намерения, а также общие (коллективные) обязательства и намерения команды агентов*.

Понятие общей цели является одним из ключевых понятий теории. Эта цель формально описывается с помощью пропозиционального утверждения p , и при этом цель считается достигнутой, если это утверждение принимает значение *истина*. В некоторых ситуациях цель может оказаться недостижимой. Например, если команда подводных роботов имеет целью идентифицировать неизвестный подвижный объект, то цель эта может быть недостижимой, если скорость обнаруженного объекта больше, чем максимальная скорость подводных роботов. Цель может также стать нерелевантной, если некоторое утверждение q перестает быть истинным. Например, если подводные роботы имеют целью идентифицировать неизвестный объект в их зоне патрулирования, но в процессе преследования объект вышел из зоны патрулирования, то цель командной работы подводных роботов становится нерелевантной ситуации.

Индивидуальное обязательство – это та цель, достижение которой агент взял на себя по поручению команды. Соглашениями (*conventions*) называются условия, при истинности которых агент может отказаться от своих индивидуальных обязательств. Индивидуальное намерение агента – это действие (или их последовательность), которое он *намерен* выполнить в определенных условиях в соответствии с обязательствами.

Общие обязательства команды определяются так, чтобы они описывали действия группы агентов, как одного агента. Для их описания простого объединения индивидуальных обязательств агентов недостаточно, так как агенты могут расходиться в своих убеждениях относительно истинности p и q . И тогда некоторые агенты будут продолжать выполнение своих обязательств, когда цель, например, уже достигнута, и когда другие агенты команды уже отказались от выполнения своих обязательств. Общие обязательства команды – это объединение индивидуальных обязательств, которое поддерживается осведомленностью всех агентов относительно статуса цели.

Теория общих намерений формулирует три базовых утверждения, определяющих принципы, которым должны следовать агенты команды:

- ◆ **Утверждение 1.** Агент придерживается командной цели и принятых соглашений до тех пор, пока не станет истинным хотя бы одно из следующих условий: (а) он убежден, что цель достигнута (утверждение p истинно); (б) цель является недостижимой (пропозиция q истинна) или (в) цель стала нерелевантной.
- ◆ **Утверждение 2.** Агент придерживается своей текущей цели, если выполнено хотя бы одно из следующих утверждений: (а) он убежден, что цель

еще не достигнута (p остается истинным); (б) он убежден, что цель достигнута, но это факт еще неизвестен некоторым членам команды и потому его текущей подцелью становится довести до них эту информацию.

- ◆ *Утверждение 3.* Команда агентов придерживается общей цели и стремится достичь ее, если выполнены все следующие условия: (а) все члены команды убеждены, что цель не достигнута; (б) все члены команды знают, что все они стремятся достичь этой цели и (в) все члены команды придерживаются этой общей цели до тех пор, пока хотя бы один агент придерживается своих соглашений в соответствии с утверждением 1.

Любая постановка задачи командной работы должна принимать во внимание динамику плана, поскольку планы всегда развиваются во времени и обычно в пространстве. Агенты начинают с частичного плана и расширяют его вплоть до получения полного плана. Поскольку убеждения агента могут быть ложными, а мир может изменяться как при планировании, так и при исполнении действий в соответствии с частичным планом, то в некоторых условиях может потребоваться пересмотр плана. По этим же причинам описание компонент модели командной работы могут быть неполными. Например, агент может иметь только частичный план действий; может быть не назначен агент для некоторого действия, а потому может отсутствовать индивидуальный план для него и т.д. Неполной может быть информация, полученная из внешней среды и других агентов. Эти аспекты в теории общих намерений не отражены. Они более глубоко представлены в "Теории общих планов".

Теория общих планов ("Shared Plans Theory", [10]) строится иначе, чем теория общих намерений. Она не использует такие понятия, как *общее намерение* и *общая цель*. Ее базовыми понятиями являются *групповой план* и *индивидуальные ментальные понятия агентов*, а согласование индивидуальных действий и планов групп агентов выполняется другими средствами. Эта теория в большей мере ориентирована на построение структур конкретных планов, чем на формирование команды и дисциплины взаимодействия.

Под *групповым планом* в этой теории понимается план совместного выполнения некоторого множества действий *группой* агентов. Частными случаями группы являются команда агентов в целом, а также отдельный агент. Групповой план содержит в себе *условия согласования* действий отдельных агентов и групп агентов. Эти условия представляются понятиями *действие* и *предписание*. Под *действием* авторы понимают абстрактную, достаточно сложную сущность, которая описывается такими свойствами, как *тип* и *время* действия, *группа агентов* или *агент*, исполняющий действие, *другие объекты*, вовлеченные в выполнение действия и т.д. Функции *предписание* ставится в соответствие каждому действию. Она задает множество инструкций, определяющих, каким именно образом действие должно выполняться. В общем случае *предписание* представляется иерархической структурой, узлами которой являются *предписания* более низкого уровня. Предполагается, что агент имеет библиотеку предписаний для действий различных типов, эта библиотека со временем может обновляться, а различные агенты могут иметь различные библиотеки.

Теория общих планов обладает рядом недостатков, которые приводят к трудностям ее использования на практике. В ней не конкретизируется способ построения конкретного общего плана. Далее, некоторый агент по какой-то причине может оказаться неспособным выполнить назначенное ему действие (вышел из строя, исчерпал ресурс и т.п.). Состояние внешней среды может измениться так, что действие не может быть выполнено. Эти ситуации в теории общих планов не рассматриваются.

Оценивая обе теории, можно утверждать, что они определяют только общие, но очень важные рамки командной работы, например, базовые понятия и их взаи-

мосвязь, протоколы взаимодействия, необходимый информационный обмен и т.п., и рассматривают командную работу с различных точек зрения. Дальнейшие исследования и разработки, основанные на их практическом использовании, показали, что эти теории хорошо дополняют друг друга. В частности, разработанные позднее конкретные модели командной работы всегда в той или иной форме комбинируют результаты обеих теорий. Это, в частности, имеет место в широко известной модели STEM [22] и разработанной на ее основе архитектуры и инструментальной поддержки Teamcore ([23], [16]), а также в модели RETSINA ([20], [21]), которые кратко описываются далее.

2.2. Модели командной работы и инструментальные средства для их поддержки. Разработки в области моделей командной работы были начаты в середине 1990-х гг. Наиболее известными среди них являются модели, описанные в работах [17] (модель COLLAGEN), [13] (модель ADEPT), [3] (модель COOL), [22], [23], [16] (модель STEM и инструментальная среда Teamcore), а также в [20], [21] (модель RETSINA). Две последние модели получили наиболее глубокое развитие, а потому и наибольшее признание. Рассмотрим их.

Модель STEM [22] построена на основе комбинирования идей теории общих намерений и теории общих планов. На основе теории общих намерений в ней формируются базовые блоки модели командной работы. Модель общих намерений дает возможность явно представить план совместной работы агентов команды в виде отношения на множестве действий агентов и их групп, а также индивидуальный и групповой вклад агентов в его выполнение. Этот план используется как руководство для рассуждений о координации и коммуникациях, а также позволяет на основе мониторинга командной работы обнаруживать и своевременно предотвращать ошибки исполнения плана.

В модели STEM согласованная работа агентов команды при выполнении плана и его мониторинге обеспечивается решением четырех задач:

1. Декомпозиции общих намерений команды на общие намерения подгрупп агентов и индивидуальные намерения агентов. Эта декомпозиция выполняется без указания конкретных исполнителей действий, реализующих намерения. Общий план в дальнейшем доводится до агентов, что формирует у них единые взаимные убеждения о плане выполнения командной работы.
2. Информационное обеспечение агентов команды в целях мониторинга командной работы. В соответствии с теорией общих планов каждый агент должен знать, какое именно намерение сейчас реализуется тем или иным агентом или группой агентов, но не как именно это намерение реализуется.
3. Работа команды в случае, когда общее намерение оказывается недостижимым, когда формируется общее намерение перепланировать работу.
4. Информационный обмен при управлении последовательностью исполнения действий плана. Если некоторое намерение достигнуто, то агент, который имеет информацию об этом, всегда знает также и следующий оператор, но команда не может приступать к его выполнению, пока все ее члены, ответственные за его выполнение, не будут осведомлены о выполнении предшествующего оператора.

Позднее модель STEM была использована в архитектуре и инструментарии для быстрого объединения гетерогенных агентов в команду, названного TEAMCORE ([23], [16]). Основная особенность этого инструментария состоит в том, что в нем агент состоит из двух частей. Одна из них является предметно-зависимой, а другая – предметно-независимой. Предметно-независимая часть агента называется *teamcore-агент*. Она играет роль "обертки" для предметно-зависимой части агента, наделяя агента в целом способностью к командной рабо-

те. Модель TEAMCORE наделена способностью к переговорам и динамическому распределению ролей в команде. Детальное описание ее архитектуры и программного инструментария с примерами приложений можно найти в [16].

Модель RETSINA и соответствующая ей архитектура ([20], [21]) построена, главным образом, на основе теории общих намерений, которая реализуется в форме кооперативного интерфейсного агента, инкапсулирующего способность рассуждать о намерениях. Эти рассуждения используются для того, чтобы (а) в нужные моменты времени определять релевантного получателя критической информации и передавать ему эту информацию; (б) отслеживать зависимости между задачами, исполняемыми членами команды; (в) своевременно распознавать конфликты и нарушения ограничений, (г) предлагать решения для разрешения конфликтов; (д) выполнять мониторинг работы членов команды.

В модели RETSINA любой агент, кроме решения запланированных задач, должен быть в состоянии получать дополнительные задачи и подцели от других агентов, передавать другим агентам информацию о ходе выполнения своих задач, выполнять мониторинг действий команды в целом и уметь делегировать свои или чужие задачи другим членам команды. Эти задачи решаются интерфейсным агентом, реализующим эти функции с помощью коммуникаций. Как и в Teamcore, в модели RETSINA все члены команды имеют возможность выполнять мониторинг командной деятельности для синхронизации работы и порядка выполнения подзадач, соблюдать выполнение своих обязательств и выполнять работы по достижению командной цели.

Обе описанные архитектуры, Teamcore и RETSINA, использовались для ряда практических разработок, главным образом, в военных интересах.

2.3. Языки спецификации поведения автономных агентов, участвующих в командной работе. Модели и инструментальные средства, описанные в предыдущем подразделе, ориентированы на поддержку процессов программирования компонент повторного использования, которые ответственны, главным образом, за организацию командной работы при планировании и распределении ролей на множестве агентов, за процессы исполнения плана и мониторинг работы членов команды и т.п. Такие инструментарии, например, TEAMCORE, RETSINA и др. оставляют значительный объем работ для ручного программирования всего того, что относится к конкретному приложению, в частности, к программированию агентов, управляющих поведением роботов. Далее дается краткий обзор языков, которые разрабатывались для целей программирования поведения агентов, управляющих коллективом роботов.

Применительно к отдельным автономным роботам первые разработки подобных языков были начаты в 1990-х гг., и вначале они не ориентировались на модели коллективного поведения. Примером может служить язык Colbert [14]. В последующем расширения таких языков были положены в основу языков спецификации агентов команды за счет введения механизмов координации действий, а также использования коммуникационных протоколов. Рассмотрим два наиболее известные языка такого назначения.

Язык XABSL (Extensible Agent Behavior Specification Language), предложенный в [15] и развитый в [18], разработан для описания коллективного поведения роботов в условиях, когда внешняя среда частично наблюдаема и обладает непредсказуемой динамикой. Его формальную основу составляет модель иерархии машин состояний. Он включает в себя следующие компоненты [18]:

- ♦ модулярную архитектуру, представленную множеством параллельных машин состояний, и язык спецификации для ее описания;

- ◆ компилятор, генерирующий документацию и промежуточный код, который затем запускается на исполнение;
- ◆ C++ библиотеку, используемую для исполнения программы.

Машина состояний описывает некоторую простую последовательность действий, а каждое действие авторами называется *опцией*. Опция исполняется тогда, когда состояние соответствующей машины состояний активно. Некоторые состояния могут отвечать вложенным опциям, т.е. машине состояний, которая запускается при активации этих состояний. Таким образом, машины состояний в языке XABSL могут быть организованы иерархически. Опции могут иметь атрибуты, значения которых задаются вызывающей опцией. Заметим, что одновременно могут вызываться несколько опций, что реализует идею параллельного исполнения действий агентами. При запуске иерархически организованных опций они структурируются в виде дерева, которое называется деревом активации опций, в корне которого находится *корневая опция*. Все опции описываются ациклическим направленным графом (рекурсивные операции в этом языке запрещены), который называется *графом опций*. Заметим также, что иерархическая организация машин состояний позволяет использовать одну и ту же опцию в различных контекстах, путем ее вызова из разных опций.

Операционная семантика языка XABSL задается дискретным циклом исполнения. В течение каждого цикла обновляются состояния всех опций дерева опций, включая корневую опцию. Дискретный цикл исполнения может быть либо постоянным, либо может управляться событиями. Опция, запустившая некоторую опцию, ответственна за контроль окончания ее работы и вызов следующей опции. Важно отметить, что машины состояний XABSL, в отличие от языка Colbert, ответственны только за выбор действия агента, но модели самих действий реализуются библиотекой.

Язык XABSL позволяет описывать поведение отдельного агента, управляющего роботом, а также имеет некоторые средства для описания коллективного поведения агентов [18], к которым относится, например, специальный атрибут машины состояний, указывающий, какое количество агентов могут одновременно исполнять соответствующую опцию. Например, если речь идет о проходе через дверь, то такой атрибут указывает максимальное число роботов, которые могут одновременно выполнять это действие, чтобы не вызвать пробку. Еще одно средство предназначено для указания на то, что действия некоторого множества агентов должны быть синхронизированы, т.е. они в результате действия должны прийти в некоторое состояние опции одновременно. Более сложные акты координации в этом языке не могут быть специфицированы машинами состояний напрямую. Они могут быть реализованы через обмен сообщениями. Однако эти возможности достаточно ограничены.

Следует заметить, что язык XABSL разрабатывался для описания командной работы с хорошо определенными правилами типа командной работы роботов-футболистов [19], и это определяет область его использования.

Язык PNP (Petri Net Plan) был предложен в работе [25] и далее развит в [5] и [26]. В его основе лежит широко известная модель сети Петри. Известно, что эта модель обладает достаточно выразительной графической формой представления сложных процессов. Для нее разработаны средства формальной верификации различных свойств, что привлекает к ней (не всегда заслуженно) внимание разработчиков моделей поведения, в особенности, в задачах управления потоками взаимодействующих процессов (работ). Например, существует несколько расширений языка сетей Петри для спецификации бизнес-процессов [24]. Примерами являются язык WN (Workflow Net) [2], и язык YAWL (Yet Another Workflow Language) [1].

Авторы PNP позиционируют его как достаточно выразительное средство для спецификации планов поведения агентов команды в частично наблюдаемой динамической среде. Он рассматривается ими как графический язык для описания динамических систем, способный выразить ряд специфических особенностей коллективного поведения. К ним относятся такие, например, как учет продолжительности обычных действий (в сетях Петри это невозможно) и зависимых действий. PNP имеет возможности по представлению циклов, параллельности, прерываний, возможности описания реакции системы на ошибки. Язык PNP имеет также специальные средства координации работы агентов, к которым относятся операторы жесткой (*hard-synchronization*) и мягкой (*soft-synchronization*) синхронизации, которые позволяют средствами языка представить некоторые аспекты командного плана. Они описаны в [26]. Модульность языка обеспечивается использованием понятия субплана, который в тексте ссылается вместо действия на некоторую сеть Петри. Понятие субплана реализует то же самое, что и иерархия машин состояний в языке XABSL.

Операционная семантика PNP базируется на операционной семантике сетей Петри, но дополнительно использует два понятия для перехода, а именно, понятие *возможного перехода* и *выполнимого перехода*. Возможный переход имеет тот же смысл, что и возможный переход в сети Петри. Выполнимый переход – это возможный переход, для которого выполнено событийное условие φ , которым этот переход помечен. Таким образом, операционная семантика языка PNP задается тремя компонентами: (1) структурой экземпляра сети Петри; (2) начальной ее маркировкой и (3) потоком управления, который определяет выполнение событийных условий φ на переходах. Очевидно, что операционная семантика языка PNP аналогична той, что используется в хорошо известных языках WN и YAWL описания потоков работ [24].

В последней версии PNP (2008) предполагается, что план разрабатывается заранее, и каждому его действию заранее назначается агент, который будет его исполнять, т.е. каждому действию плана жестко ставится в соответствие идентификатор ответственного агента. Затем весь план сообщается всем агентам, так что каждый из них может автоматически определить, какие действия и в каком контексте он должен исполнять. В результате координация агентов сводится к синхронизации действий агентов в соответствии с общим планом. Синхронизация описывается оператором жесткой синхронизации *hardsync*. Он используется тогда, когда пара роботов, управляемых агентами, должна начинать и оканчивать некоторое действие строго в одно и то же время. Например, если два робота должны приблизиться к некоторому объекту, а затем вместе перенести его в некоторую точку, то их действия синхронизируются агентами с помощью оператора *hardsync*. С помощью оператора мягкой синхронизации *softardsync* описывается частичный порядок на множестве действий. Оба оператора могут реализовываться с помощью обмена сообщениями, если они имеют общую коммуникационную платформу. Детали их использования описаны в [26].

Анализ языка PNP показывает, что он обладает практически теми же возможностями, что и язык XABSL, а использование формальной модели сети Петри не дает ему каких-либо преимуществ. Например, операционная семантика в PNP реализуется сложнее, чем это свойственно модели машины перехода состояний. Почти все возможности, требуемые для описания индивидуальных действий роботов (иерархическое описание плана действий, синхронизация, прерывание, параллельное исполнение действий, представление зависимых действий и др.) имеются в обоих языках. Оба языка имеют потенциальные возможности по использованию протокола общих намерений для координации поведения в стиле командной работы, хотя эта возможность не заложена в средства этих языков. Использование же

стандартных средств коммуникации на базе языка ACL и распределенной программной инфраструктуры типа JADE для поддержки взаимодействия агентов потребует дополнительных усилий и ручного программирования интерфейсов. Оба языка имеют, хотя и различные, но достаточно ограниченные возможности по спецификации командной работы *автономных* агентов и не в состоянии описать командную работу с той степенью общности, которая предусмотрена в моделях TEAMCORE и RETSINA.

Но самый главный недостаток обоих языков состоит в том, что в них детальный план поведения агентов разрабатывается заранее. Оба языка не имеют средств распределения и перераспределения функций между агентами в динамике исполнения командной работы

3. СКВ: Язык спецификации командной работы автономных агентов, основанный на знаниях.

3.1. Мотивация. Анализ современного состояния исследований и последних публикаций в области теории, моделей, программных инструментариев и языков спецификации приложений применительно к командной работе автономных агентов показывает, что две проблемы являются предметом основных исследований, а именно:

- 1) планирование целенаправленной командной работы автономных агентов и ее динамическое перепланирование, необходимость в котором вызывается динамичностью внешней среды и другими причинами;
- 2) динамическое распределение и перераспределение компонент плана, обусловленное теми же факторами, а также возможностью потери агентом некоторых способностей, например, из-за выхода из строя управляемых объектов (роботов), досрочного израсходования ресурса, и др.

Обе задачи тесно взаимосвязаны и обычно решаются совместно. В инструментариях Teamcore и RETSINA предполагается, что при планировании и перепланировании иерархическая структура общих намерений строится на основе экспертных знаний, затем узлам этой структуры ставятся в соответствие конкретные действия, которые затем распределяются для исполнения на множестве агентов в соответствии с их способностями. При этом разработка структуры общих намерений и построение плана их достижения не поддерживается инструментарием и выполняется вручную, хотя процесс последующего программирования плана поддерживается им. В разд. 2 было отмечено, что существующие языки спецификации командной работы XABSL и PNP не имеют средств реализации процесса перепланирования в реальном времени.

Цель данной работы состоит в том, чтобы частично заполнить указанный пробел. В ней предлагается модель и инструментальная программная поддержка для автоматического решения задач планирования и перепланирования в реальном времени. Основная идея модели состоит в использовании экспертных знаний о способностях агентов по решению тех или иных задач, а также о множестве возможных сценариев достижения той или иной цели в зависимости от начального состояния команды агентов. Разработанный инструментарий включает в себя следующие компоненты:

- 1) формальный язык и механизм синтаксического вывода для описания и вывода знаний о множестве возможных сценариев достижения командой агентов тех или иных целей в зависимости от ее начального состояния;
- 2) исполнительный механизм (он называется ядром инструментария), который реализует операционную семантику процесса планирования и процесса исполнения плана; он также реализует механизм вывода, решающий задачу построения множества потенциальных планов поведения команды, ведущих к достижению заданного множества целей;

- 3) FIPA-совместимую распределенную агентскую платформу [7], реализующую механизм парных (P2P) взаимодействий; она поддерживает сервисы белых и желтых страниц по распределенному поиску агентов и сервисов, а также коммуникационный сервис P2P. При необходимости агентская платформа может быть расширена сервисом доступа к веб-сервисам;
- 4) сервер внешней среды, поставляющий агентам информации о ее состоянии, о состояниях членов команды, состоянии исполнения плана, а также о состоянии и действиях команды противника, если она имеется; в качестве прототипа при создании данного сервера выступал хорошо известный Robocup 2D-сервер [19], который используется в международных соревнованиях по футболу роботов;
- 5) эмулятор распределенной сети компьютеров, на которых работают агенты, управляющие поведением команды роботов; эта программная компонента используется в том случае, когда командная работа множества автономных агентов симулируется на одном компьютере.

Разработанная модель и поддерживающий ее программный инструментарий называются SKBF (*Scenario Knowledge-Based Framework*), а формальный язык, лежащий в его основе, назван языком сценарных знаний (*SKBL-Scenario Knowledge-Based Language*). Далее дается краткое их описание.

3.2. Язык сценарных знаний. Он включает в себя четыре компоненты:

- ◆ *синтаксическую компоненту*, которая описывает иерархически структурированное множество имен (идентификаторов) частично упорядоченных состояний и действий (подсценариев, если действия не являются примитивными);
- ◆ *механизм синтаксического вывода*, который позволяет решать задачи планирования и перепланирования в реальном времени;
- ◆ *семантическую компоненту*, описывающую (1) множество предусловий и пост-условий, поставленных в соответствие узлам синтаксической компоненты, (2) семантику действий, соответствующих узлам синтаксической компоненты и (3) структуры данных, описывающих состояния;
- ◆ *операционную семантику* компонент языка, реализуемую ядром инструментария; она управляет порядком исполнения действий плана на основе потока событий, генерируемых агентами и внешней средой.

Синтаксическая компонента задается двудольным ациклическим направленным графом, содержащим узлы двух типов:

- ◆ *узлы*, представляющие *действия*, примитивные или сложные (они называются подсценариями);
- ◆ *узлы*, представляющие *состояния*, понимаемые также как *цели* или подцели.

Как и в сетях Петри, в синтаксической компоненте каждый узел может иметь направленные дуги, соединяющие его только с узлами другого типа. Входные дуги действий отвечают тем состояниям, которые должны быть достигнуты для того, чтобы действие могло быть потенциально выполненным. Узел-действие может иметь несколько входных дуг. Выходная дуга узла-действия указывает на состояние, которое достигается при успешном его выполнении. Узел-действие может иметь только одну выходную дугу.

В отличие от этого, узел-состояние может иметь несколько входных дуг и несколько выходных. При этом входные дуги узла-состояния исходят из тех узлов-действий, результатом успешного выполнения которых может быть достижение данного состояния. Что касается выходных дуг, то узел-состояние может также иметь их несколько, так как достижение одного и того же состояния может быть необходимым условием для выполнения разных действий.

В синтаксической компоненте языка имеется еще один тип узлов, отвечающих понятию *активной сущности*, под которой понимается, например, человек, который может посылать команде автономных агентов задания на выполнение той ли иной миссии, или быть потребителем информации о состоянии исполнения миссии и/или о результате. Подчеркнем, что наличие в модели активной сущности не означает наличие внешнего вмешательства в процесс исполнения автономной миссии (операции) командой агентов. Активная сущность может только указать, какую именно миссию (возможно, с атрибутами) должна выполнить команда агентов, но не как именно ее следует выполнять. Построение плана выполнения миссии, мониторинг исполнения, а также перепланирование и распределение задач плана между агентами должно оставаться внутренней задачей команды.

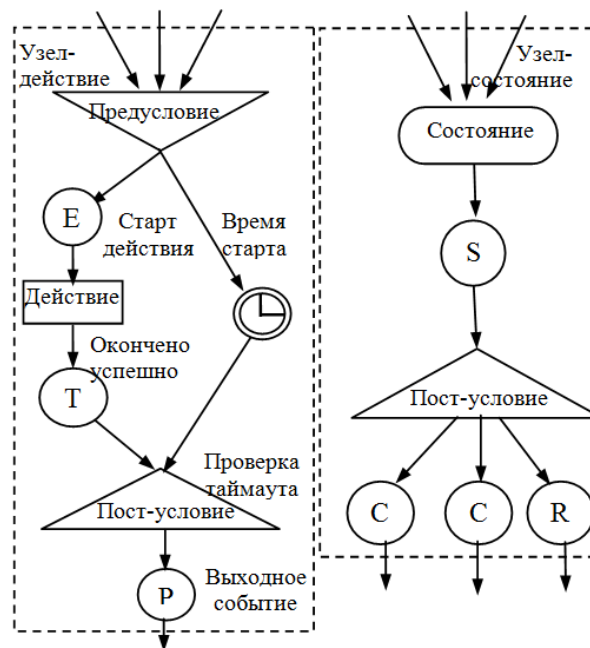


Рис. 1. Семантика узлов разных типов базы сценарных знаний, представленная в принятой стандартной графической нотации OMG

Для конкретного приложения и конкретной команды агентов сценарная база знаний должна описываться с такой детальностью, в которой примитивными действиями могут являться способности агентов. Например, для команды подводных роботов, выполняющих миссию по охране подводного пространства морского порта, под способностями роботов, которыми управляют агенты, могут пониматься, например, "Переход между заданными точками трехмерного подводного пространства" с заданным ограничением на время перехода. Другими примерами могут быть "Идентификация подводного объекта", "Избежание столкновений с подводными или надводными объектами", "P2P-коммуникация с другими агентами или объектами внешней среды", "Запрос сервиса" и т.д. Заметим, что способность агентов команды к взаимодействию и коммуникациям в P2P-стиле рассматривается в языке как одна из обычных способностей агентов. То же самое касается протоколов, которые используются в процессе переговоров для оптимизации распределения задач между агентами, например, протокола контрактных сетей или протоколов того или иного варианта аукциона.

Механизм синтаксического вывода использует свойства частичного порядка на множестве состояний и действий. При планировании он используется в двух вариантах, а именно, в варианте “сверху вниз” (*top down*) и в варианте “снизу вверх” (*bottom up*).

Таким образом, синтаксическая компонента языка SKBL описывает высокоуровневые знания о допустимых вариантах планов достижения тех или иных целей в терминах способностей автономных агентов и объектов, которыми они управляют. Синтаксическая компонента знаний¹ должна быть общим знанием всех агентов команды, и именно на ее основании агенты могут решать задачи планирования, командного мониторинга исполнения общего плана и координации своих действий в соответствии с принятым распределением задач. Отметим, что синтаксическая модель и механизм вывода описываются в языке формально. Здесь они не приводятся из-за ограничения объема работы.

Семантическая компонента языка SKBL описывает семантику действий, соответствующих узлам плана, структуры данных, представляющих состояния, предусловия и пост-условия, поставленные в соответствие узлам базы сценарных знаний. На рис. 1 показана внутренняя структура узлов, соответствующих действиям и состояниям. Элементами семантики являются также потоки событий. Некоторые из событий показаны на рис. 1.

Операционная семантика реализуется, как обычно, с помощью потока событий, инициируемого агентами, а также поступающего из внешнего мира. Они несут информацию о процессе исполнения плана и событиях, происходящих во внешнем мире. Каждое событие имеет тип, имя, время инициации и, возможно, другие атрибуты. Заметим, что здесь идет речь об операционной семантике процесса управления командной работой. Что касается операционной семантики исполнительных механизмов агентов (роботов), то здесь этот уровень не рассматривается. Заметим, что она может быть реализована так, как это предусмотрено, например, в языке XABSL, описанном в разд. 2.

Язык SKBL в настоящее время, в основном, реализован. Основные усилия (кроме традиционного тестирования) в настоящее время направлены на реализацию его графической поддержки, которая позволит реализовать визуальный стиль программирования. Только после этого можно будет дать ему итоговую оценку более определенно.

4. Командная работа подводных роботов: Содержание демонстрационного примера. Прежде чем описывать другие компоненты разработанной инфраструктуры для поддержки технологии, опишем содержательный пример, который использовался для тестирования и отладки. Он демонстрирует, как практически используется разработанный язык и другие компоненты инструментальной инфраструктуры. В нем рассматривается поведение команды автономных подводных роботов (АПР), выполняющих миссию патрулирования трехмерного подводного пространства в акватории морского порта.

Предполагается, что АПР² движутся вдоль заданных траекторий, сканируя подводное пространство в передней зоне с помощью сонаров с заданной диаграммой направленности, дистанцией обнаружения объектов, а также дистанцией, позволяющей проводить детальное обследование объектов с соблюдением собственной безопасности. Предполагается также, что АПР воспринимает гидроакустический сигнал с круговой диаграммой направленности, а также имеет гидроакустический канал связи ограниченной дальности. Заметим, что ввиду динамического

¹ Естественно, что она должна быть представлена в терминах понятий общей онтологии, однако этот аспект в работе не рассматривается из-за ограничений объема работы.

² В реализованном примере рассматривается группировка из шести АПР.

изменения конфигурации сети АПР расстояние между ними постоянно изменяется, а потому конфигурация сети связи между АПР также постоянно изменяется. Иначе говоря, каждый АПР в зависимости от времени может связываться с различными множествами АПР, а периодически вообще может ее не иметь. Для того, чтобы в любой момент времени каждый АПР мог иметь связь со своей группировкой, а также с пунктом управления, патрулируемое пространство снабжается несколькими придонными гидроакустическими связными буями (ПГАС), которые могут играть роль посредников при коммуникации АПР-группировки между собой, а также с пунктом управления. Полагается, что связь АПР с пунктом управления возможна только через ПГАС.

Связь компонент группировки АПР и пункта управления в условиях динамики конфигурации сети связи организуется по протоколам парных взаимодействий (P2P). То же самое касается и протоколов взаимодействия АПР и ПГАС при решении содержательных задач. Эти возможности в разработанной инфраструктуре практически реализованы с помощью распределенной агентской платформы P2P, которая разработана авторами данной работы в 2007 г. [7] в полном соответствии с абстрактной архитектурой FIPA [6]. Эта платформа устанавливается как оверлейная сеть поверх TCP/IP коммуникационной платформы и обеспечивает взаимодействие автономных агентов, управляющих командным поведением АПР в сервис-ориентированном стиле. Базовыми сервисами платформы являются сервисы белых и желтых страниц, а также коммуникационный сервис P2P, поддерживающий парные коммуникации агентов.

Для команды агентов АПР разработана база сценарных знаний, формализующая экспертные знания о сценариях командной работы автономных агентов в нескольких миссиях. Дадим краткое описание одного из этих сценариев, а именно, сценария командной работы при формировании команды для идентификации неизвестного объекта, обнаруженного некоторым АПР.

Когда в процессе патрулирования некоторый АПР обнаруживает неизвестный объект, его агент принимает на себя роль *агента-инициатора* и инициирует сценарий командной работы по детальному обследованию обнаруженного объекта в целях его идентификации. При этом он сначала вычисляет координаты объекта, рассчитывает свою траекторию приближения и координаты ее конечной точки с учетом условий собственной безопасности. Затем, используя свой экземпляр агентской платформы P2P, агент посылает в широкоэвещательном режиме (в режиме бродкастинга) запрос на поиск партнера по обследованию объекта. В этом запросе он сообщает также координаты объекта, координаты точки, в которой он сам будет проводить обследование, перечень необходимых способностей партнера, а также желаемый интервал времени, когда партнер по команде должен будет решать задачу идентификации объекта. Предполагается, что зону наблюдения партнер должен выбрать сам таким образом, чтобы совместно с АПР-инициатором получить трехмерную картину объекта.

После анализа предложений агент-инициатор выбирает партнера, сообщает ему об этом и инициирует протокол общих намерений для формирования команды, принятия индивидуальных обязательств и соглашений. После этого каждый агент команды работает по выполнению своих обязательств. Агент-инициатор начинает движение по вычисленной траектории. После достижения ее конечной точки он ориентируется на обнаруженный объект и использует свою аппаратуру наблюдения для сбора детальной информации о нем и ждет информации от агента-партнера. *Агент-партнер*, по окончании протокола общих намерений, также вычисляет свою траекторию, перемещается в выбранную точку подводного пространства, ориентируется нужным образом на объект и решает задачу о типе объекта и передает его вместе с собранной информацией агенту-инициатору. Когда

инициатор получает эту информацию, он производит ее объединение с собственной информацией и принимает итоговое решение.

Если решение принято, то в зависимости от полученного результата агент-инициатор принимает решение об окончании обследования объекта и командной работы ввиду достижения цели. В этом случае агент-партнер получает информацию о том, что он свободен от обязательств и соглашений и может переходить к патрулированию. Сам агент-инициатор при этом переходит к выполнению того или иного сценария в зависимости от полученного результата.

Если агент-инициатор не может принять надежного решения о типе обнаруженного объекта, то он сообщает об этом агенту-партнеру, и они продолжают командную работу по обследованию объекта.

Напомним, что оба агента, инициатор и партнер, имеют общую базу сценарных знаний, и их общение сводится к распределению ролей и к работе в соответствии со сценарием, координируемой с помощью обмена сообщениями. Напомним также, что посылка сообщения того или иного содержания, а также взаимодействие агентов по тому или иному протоколу рассматривается в языке SKBL как обычное действие, описанное в сценарной базе знаний.

Отметим еще один важный аспект сценарной базы знаний. Она описывается в терминах онтологии, которая известна всем членам команды агентов. Эта онтология имеет две части. Одна из них представляет предметно-независимые понятия, которые описывают такие понятия, как *сценарий*, *действие*, *состояние*, *событие* и тому подобные, а также предметно-независимые отношения между ними, которые используются исполнительной компонентой инструментария, называемой ядром системы. Вторая часть онтологии содержит понятия и отношения приложения, в нашем случае такой предметной областью является командная работа АПР. Обе названные компоненты онтологии были тщательно разработаны и реализованы как надстройка над сценарной базой знаний.

5. Архитектура инструментария, его компоненты и программная реализация. Компоненты и набросок архитектуры программной инфраструктуры для поддержки командной работы автономных агентов представлены на рис. 2. Дадим краткое описание этих компонент и их функций.



Рис. 2. Архитектура и компоненты инфраструктуры для поддержки командной работы автономных агентов

Каждый АПР имеет агента – *АПР-агента*, который управляет его поведением в процессе командной работы. Эти агенты взаимодействуют между собой, а также с внешней средой через собственный экземпляр распределенной агентской платформы [7], которая обеспечивает его стандартными сервисами белых и желтых страниц, а также коммуникационным сервисом Р2Р, поддерживая также распределенный поиск сервисов и АПР, ими обладающих, и различные протоколы взаимодействия агентов. Примерами таких протоколов являются протокол поиска

сервисов и партнеров команды, протокол общих намерений, протокол распределения ролей между членами сформированной команды и др. Экземпляры агентской платформы P2P организованы в оверлейную сеть, установленную поверх *коммуникационной среды TCP/IP*.

Однако в задаче моделирования, когда все агенты могут быть установлены на одном компьютере, для эмуляции распределенной вычислительной среды между экземплярами агентской платформы и *TCP/IP*-средой должно присутствовать промежуточное программное обеспечение, называемое *Эмулятором распределенной вычислительной среды*. Фактически он эмулирует также распределенный характер вычислительной и коммуникационной сред. Смысл этой компоненты состоит в том, чтобы все исходящие и входящие сообщения агентских платформ к *TCP/IP*-уровню и обратно фиксировались этой средой и *фактически уходили* из компьютера и *входили* в него *извне*. Эмулятор позволяет также проводить мониторинг трафика сообщений, которыми обмениваются агенты. Эта компонента была также разработана авторами данной работы [8].

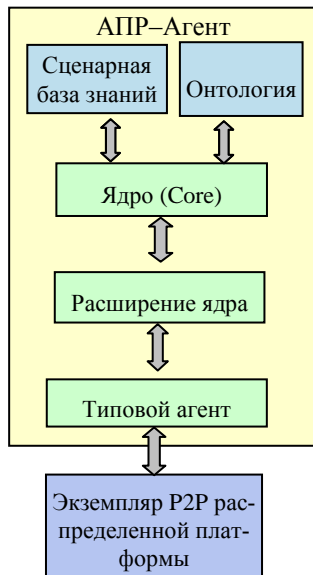


Рис. 3. Архитектура АПР-агента

Симулятор служит для моделирования внешней среды и ее взаимодействия с агентами. Он выполняет реализацию операционной семантики команд, подаваемых агентами на исполнение в АПР, и вместо физического перемещения роботов он обновляет их координаты и передает их значения соответствующим агентам. Он моделирует также появление новых объектов того или иного типа в случайно выбираемых точках подводного пространства. Эти объекты подлежат обнаружению и идентификации средствами АПР. Симулятор также моделирует работу средств обнаружения, например, область сканирования сонара, и факт обнаружения неизвестного объекта. В инфраструктуре для этих целей адаптированы идеи, реализованные в хорошо известном стандартном RoboCup 2D-сервере [19].

Визуальный интерфейс служит для отображения динамической обстановки в подводном пространстве, включая положение и движение АПР, их действий и положение неизвестных объектов. Кроме того, он обеспечивает визуальное представление состояния исполнения сценариев, подсвечивая различными цветами те элементы графического его представления,

которые имеют статус "выполнен", "выполняется", "недостигнут". Визуальный интерфейс не является обязательной компонентой инфраструктуры, он используется пользователем при отладке сценариев и других компонент инфраструктуры. Естественно, он может быть очень полезен для операторов пункта управления при мониторинге деятельности команды роботов в динамике ее исполнения.

Дадим описание архитектуры и функций одной из наиболее сложных компонент данной инфраструктуры, а именно, АПР-агента. Эта архитектура представлена на рис. 3. Ее центральной компонентой является *ядро (Core)*. *Ядро* – это предметно-независимая компонента агента. Она реализует механизмы вывода в сценарной базе знаний, представленной в терминах понятий онтологии, загружает сценарную базу знаний и онтологию в рабочую память системы. Для заданного начального состояния и цели ядро динамически строит сценарий достижения цели. Определив очередные действия сценария, ядро инициирует их исполнение, передавая задание компоненте *Расширение ядра*.

Онтология состоит из двух частей. Первая из них – это ее предметно-независимая часть. Эта часть онтологии представляет собой метазнания сценарной базы знаний. Они используются ядром системы для реализации операционной семантики, связанной с формированием текущего состояния рабочей памяти и тех действий, которые допускаются языком сценарных баз знаний для понятий, используемых механизмом вывода и управления, а также для манипуляции с понятиями *действие* и *состояние* синтаксической компоненты языка. Вторая компонента онтологии описывает понятия предметной области "*командная работа автономных подводных роботов*".

Расширение ядра – это предметно-зависимая компонента агента. Она описывает реализацию исполняемых действий в терминах предметной области, вычисляет все параметры, необходимые для выполнения действия. Действие, подготовленное к исполнению, передается компоненте "Типовой Агент".

Типовой агент описывает предметно-независимые функции управления агентом. Он моделирует возможности аппаратной части робота и отвечает за его взаимодействие с платформой, и через нее – за взаимодействие с другими агентами системы путем обмена сообщениями и передает информацию о действиях, подлежащих исполнению, "Серверу среды".

Экземпляр распределенной платформы P2P – это компонента инфраструктуры, обеспечивающая сервис желтых и белых страниц для поддержки взаимодействия агентов системы P2P, а также коммуникационного сервиса.

Заключение. Основными новыми результатами работы являются язык спецификации командной работы автономных агентов, основанный на использовании понятия сценарной базы знаний, а также программная инфраструктура, поддерживающая создание и функционирование команды автономных агентов. Предложенный язык поддерживается компилятором, а также средствами поддержки визуального программирования. Этот язык используется для описания сценарной базы знаний, содержащей множество потенциальных сценариев командной работы. Совместно с простым механизмом синтаксической база знаний используется для динамического планирования и перепланирования командной работы при заданном ее начальном состоянии и фиксированном множестве целей, которых команда должна достичь в процессе выполнения соответствующей миссии. Программная поддержка командной работы включает в себя исполнительный механизм, который называется его ядром, FIPA-совместимую агентскую платформу P2P, поддерживающую сервисы белых и желтых страниц, а также коммуникационный сервис P2P, поддерживающий обмен сообщениями между агентами. Симулятор, входящий в инфраструктуру, играет роль сервера внешней среды. В качестве прототипа симулятора использован стандартный сервер футбола роботов Robocup 2D [19].

Работоспособность полученных результатов проверена с помощью программного прототипа процесса командной работы автономных подводных роботов, патрулирующих подводное пространство морского порта.

Дальнейшие и текущие исследования и разработки имеют целью:

- 1) программную реализацию графической версии языка для спецификации командной работы автономных агентов;
- 2) разработку предметно-независимой компоненты сервера внешней среды и симулятора действий автономных агентов с целью его дальнейшего использования в других приложениях, например, при программировании спасательных, военных и гуманитарных операций, а также операций в киберпространстве;
- 3) расширение множества сценариев командной работы автономных подводных роботов. Рассматривается также ряд других задач.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Aalst, W., Hofstede, A.* YAWL: Yet Another Workflow Language // In: Information Systems. – 2005. – Vol. 30 (4). – P. 245-275
2. *Aalst, W.* The Application of Petri Nets to Workflow Management // In: Journal of Circuits, System and Computers. – 1998. – Vol. 8 (1). – P. 21-66
3. *Barbuceanu M., Fox M.* The Architecture for an Agent Building Shell. In: Wooldridge, M., Muller, J., Tambe M. (eds.) Intelligent Agents, vol. 2, Lecture Notes in Artificial Intelligence, 1037, Springer-Verlag, Heidelberg (1996).
4. *Cohen P. and Levesque H.J.* Teamwork. *Nous*, 35, 1991.
5. *Farinelli A., Iocchi, L., Nardi D., Ziparo V.A.* Assignment of Dynamically Perceived Tasks by Token Passing in Multi-Robot Systems. In: Proceedings of the IEEE, Special Issue on Multi-Robot Systems (2006).
6. FIPA Nomadic agent working Group WG6: Functional Architecture Specification Draft 012, <http://www.fipa.org/P2PNA-WG-docs/P2PNA-Spec-Draft012.doc>.
7. *Gorodetsky V., Karsaev O., Samoylov V., Serebryakov S.* P2P Agent Platform: Implementation and Testing, in Joseph, S., Despotovic, Z., Moro, G., Bergamaschi, S. (Eds.) Agent and Peer to Peer Computing, LCNS. Vol. 5319, Springer, 41–54 (2010).
8. *Gorodetsky V., Karsaev O., Samoylov V., Serebryakov S., Balandin S., Leppanen S., Turunen M.* Virtual P2P Environment for Testing and Evaluation of Mobile P2P Agents Networks. The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2008), Valencia, Spain, IEEE Computer Society, 2008. – P. 422-429.
9. *Grosz B.* Collaborating Systems. *AI Magazine*, 17(2), 1996.
10. *Grosz B. and Kraus S.* Collaborative Plans for Complex Group Actions // Artificial Intelligence. – 1996. – № 86. – P. 269-358.
11. *Jennings N.R., Varga L.Z., Aarnts R.P., Fuchs J. and Skarek P.* Transforming standalone expert systems into a community of cooperating agents // Int. Journal of Engineering Applications of Artificial Intelligence. – 1993. – № 6 (4). – P. 317-331.
12. *Jennings N.* Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems Using Joint Intentions // In: Artificial Intelligence. – 1995. – Vol. 75. – P. 195-240.
13. *Jennings N., Norman T., Faratin P.* ADEPT: An Agent-Based Approach to Business Process Management. In: ACM SIGMOD Record. – 1998. – Vol. 27(4). – P. 32-39.
14. *Konolige K.* COLBERT: A Language for Reactive Control in Saphira. In: KI '97 Proceedings of the 21st Annual German Conference on Artificial Intelligence: Advances in Artificial Intelligence, Lecture Notes in Computer Science. Springer-Verlag, London, 1997. – Vol. 1303. – P. 31-52.
15. *Lötzsch M.* XABSL - A Behavior Engineering System for Autonomous Agents. Diploma thesis. Humboldt-Universität zu Berlin (2004) Available online: <http://www.martinloetzsch.de/papers/diploma-thesis.pdf>.
16. *Pynadath Tambe M.* An automated teamwork infrastructure for heterogeneous software agents and humans. In: Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS): Special Issue on Infrastructure and Requirements for Building Research Grade Multi-Agent Systems (2002).
17. *Rich C., Sidner C.* COLLAGEN: When Agents Collaborate with People. In: Proceedings of the International Conference on Autonomous Agents (Agents'97) (1997).
18. *Risler M., von Stryk O.* Formal Behavior Specification of Multi-Robot Systems Using Hierarchical State Machines in XABSL. AAMAS08 Workshop on Formal Models and Methods for Multi-Robot Systems. Estoril (2008), see also <http://www.xabsl.de>.
19. RoboCup initiative, <http://www.robocup.org/>.
20. *Sycara K., Paolucci M., Giampapa J., van Velsen M.* The RETSINA Multiagent Infrastructure. Autonomous Agents and Multi-agent Systems, vol. 7(1) (2003).
21. *Sycara K., Sukthankar G.* Literature Review of Teamwork Models. CMU-RI-TR-06-50, November 2006, Robotics Institute Carnegie Mellon University, 2006.
22. http://www.ri.cmu.edu/pub_files/pub4/sycara_katia_2006_1/sycara_katia_2006_1.pdf.
23. *Tambe M.* Towards Flexible Teamwork // Journal of Artificial Intelligence Research. 1997. – № 7. – P. 83-124.

24. *Tambe M., Shen W.-M., Mataric M., Pynadath D. Goldberg D., Modi P.J, Qiu Z., Salemi B.* Teamwork in Cyberspace: Using TEAMCORtEo Make Agents Team-Ready / Proceedings of AAAI Spring Symposium on Agents in Cyberspace. – P. 136- 141 (1999).
25. *Weske M.* Business Process Management: Concepts, Languages, Architectures. Springer (2007).
26. *Ziparo V.A., Iocchi L.* Petri Net Plans. In: Fourth International Workshop on Modeling of Objects // Components, and Agents. – Turku, 2006. – P. 267-289.
27. *Ziparo V., Iocchi, L., Nardi, D., Palamara P. F., Costelha H.* Petri Net Plans: a Formal Model for Representation and Execution of Multi-Robot Plans. In: AAMAS, Estoril, (1). – P. 79-86. (2008).

Статью рекомендовал к опубликованию д.т.н., профессор Б.К. Гранкин.

Городецкий Владимир Иванович

Санкт-Петербургский институт информатики и автоматизации РАН.

E-mail: gor@iias.spb.su.

195349, г. Санкт-Петербург, пр. Сизова, 32, корп. 1, кв. 634.

Тел.: 88123233570; 88123011019.

Лаборатория интеллектуальных систем; главный научный сотрудник; д.т.н.; профессор.

Троцкий Денис Васильевич

E-mail: Denis.Trotsky@iias.spb.su.

Тел.: 88123233570; 88125277842.

Лаборатория интеллектуальных систем; младший научный сотрудник.

Серебряков Сергей Валерьевич

Лаборатории Hewlett-Packard в России.

E-mail: sergey.serebryakov@hp.com.

195271, г. Санкт-Петербург, пр. Кондратьевский, 79, кв. 194.

Тел.: 88123367474; 88125434231.

Научный сотрудник; к.т.н.

Gorodeckiy Vladimir Ivanovich

St. Petersburg Institute for Informatics and Automation of RAS.

E-mail: gor@iias.spb.su.

32, Sizov's Prosp., Build. 1, Ap. 634, Saint Petersburg, 195349, Russia.

Phone: 88123233570; 88123011019.

Laboratory of intellectual systems; Chief Researcher; Dr. of Eng. Sc., Professor.

Troickiy Denis Vasil'evich

E-mail: Denis.Trotsky@iias.spb.su.

Phone: 88123233570; 88125277842.

Laboratory of Intellectual Systems; Younger Scientific Employee.

Serebryakov Sergey Valer'evich

Laboratories Hewlett-Packard in Russia.

E-mail: sergey.serebryakov@hp.com.

79, Kondratmevskiy Prosp., Ap. 194, Saint Petersburg, 195271, Russia.

Phone: 88123367474; 88125434231.

Scientific Employee; Cand. of Eng. Sc.