МНОГОАГЕНТНЫЕ СИСТЕМЫ И СРЕДСТВА ИХ РАЗРАБОТКИ

КАРСАЕВ О.В., КОНЮШИЙ В.Г.

УДК 681.3

Карсаев О.В., Конюший В.Г. **Многоагентные системы и средства их разработки**

Аннотация. В статье рассматриваются вопросы практического использования многоагентных технологий. Приводится описание специфических свойств многоагентных систем, определяющих области их практического применения, а также – примеры успешного использования многоагентного подхода для решения нескольких важных практических задач. Основное внимание в статье уделяется проблеме взаимодействия агентов с внешней средой. В связи с этим приводится описание примеров внешней среды нескольких многоагентных систем. Статья также содержит описание среды MASDK, поддерживающей полный жизненный цикл разработки прикладных многоагентных систем. С учетом направленности статьи основное внимание при описании среды также уделяется описанию предлагаемых в ней решений, используемых для реализации механизмов взаимодействия агентов с внешней средой.

Ключевые слова: многоагентные системы, технология разработки, внешняя среда агентов.

Karsaev O.V., Konyushiy V.G. **Multiagent Systems and its Development Tools**

Abstract. The paper presents practical usage items of multiagent technology. It includes description of agent-based system specific features determining an area of their practical application and successful examples of agent-based approach usage for decision of some important practical tasks. The main attention of the paper is paid to the problem of interaction of agents with environment. In this relation description of some multiagent systems' environments is given. As well the paper includes description of MASDK tool supporting completed life cycle of multiagent systems' development. Taking into account the focus of the paper the main attention of the tool description is paid to presentation of decisions that are used for realization of interaction agents with environment.

Keywords: multiagent systems, development technology, agents' environment.

1. Введение. Тенденции развития технологий разработки программного обеспечения, с одной стороны, определяются потребностями индустриального сообщества и возрастающими возможностями вычислительных средств. В частности, интенсивное развитие Интернет и Web-технологий постепенно приводит к качественному пересмотру концепции современного бизнеса, в котором основной тенденцией становится повышение роли виртуальных предприятий (организаций), перенос бизнеса в Интернет, интенсивное использование информационно-телекоммуникационной среды в качестве инфраструктуры бизнеса.

С другой стороны, качественно новые практические возможности открываются в связи с развитием новых направлений и использованием новых технологий и методологий разработки прикладных про-

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

граммных систем. Одним из таких направлений является разработка прикладных программ в виде многоагентных систем (МАС). МАС позволяют существенно расширять возможности и области применения программных систем на практике. Эти возможности, прежде всего, определяются тем, что *организация и свойства* МАС, в отличие от прочих подходов к разработке программного обеспечения, позволяют воспроизводить *организацию и свойства* реально существующих систем более естественным образом.

Активное использование многоагентной технологии на практике в настоящее время сдерживается рядом причин, основными среди которых являются следующие. Первая причина состоит в отсутствии одной (единой) достаточно зрелой и общепринятой методологии проектирования и разработки многоагентных систем. Среди развивающихся в настоящее время одной из наиболее зрелых методологий (на основании индекса цитируемости и по мнению авторов данной статьи) является методология Gaia [8]. Вторая причина отчасти связанна с первой и состоит в том, что к настоящему времени пока нет достаточно зрелого и широко используемого CASE-средства для автоматизации проектирования и разработки многоагентных систем. В связи с этим следует отметить, что в различных сравнительных обзорах многоагентных технологий одним из основных недостатков методологии Gaia указывается именно отсутствие CASE-средств, поддерживающих ее использование для разработки многоагентных систем на индустриальном уровне. Это обстоятельство, в частности, было для авторов данной статьи определенным мотивом разработки подобного средства. В течении двух последних лет прототип такой среды (MASDK 4.0) был разработан в СПИИРАН при поддержке Министерства науки и образования в рамках Федеральной целевой программы "Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2012 годы". Описание среды и ее программная реализация доступны на Web-сайте [2]. Описание среды как средства поддержки методологии Gaia опубликовано в работе [6].

МАС, как правило, предполагают наличие внешней (окружающей) среды агентов. Проектирование и разработка компонент внешней среды и реализация взаимодействия агентов с этими компонентами составляет неотъемлемую, а в ряде случаев весьма значительную часть разработки законченных прикладных систем. Несмотря на это, вопросы взаимодействия агентов с внешней средой пока еще исследованы в недостаточной мере, и это составляет третью по важности причину,

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

сдерживающую индустриализацию многоагентных технологий. Примером одного из немногих законченных решений является решение, разработанное компанией Magenta [7]. Однако, это решение в своей основе базируется на клиент-серверной архитектуре, что также является определенным ограничением в плане реализации возможностей МАС.

Целью настоящей статьи является общее описание жизненного цикла разработки прикладных систем с помощью среды MASDK с основным акцентом на то, какие решения в ней предлагаются для решения задачи взаимодействия агентов с внешней средой. Статья организована следующим образом. В разделе 2 приводится описание основных свойств МАС, характеризующих их возможности и преимущества и определяющих области их применения. Конкретные примеры использования многоагентной технологии на практике приведены в разделе 3. В разделе 4 приводится описание примеров моделей внешней среды для двух прикладных МАС. В разделах 5 и 6 описываются основные инвариантные (многократно используемые) решения, реализованные в среде MASDK в виде готовых компонент. При этом в разделе 6 описываются предлагаемые решения, которые непосредственным образом используются в разрабатываемых приложениях для реализации взаимодействия агентов с внешней средой. Раздел 7 содержит общее описание жизненного цикла разработки приложений с помощью среды MASDK.

2. Характерные свойства МАС. Основополагающей специфической особенностью многоагентных технологий является переход от пассивных сущностей, описываемых в виде классов объектов в объектно-ориентированном подходе, к активным сущностям, которые описываются в прикладных системах в виде агентов. Агенты прикладных систем в соответствии с их предназначением могут представлять самые разнообразные сущности предметных областей. Например, такими сущностями могут быть *люди, организации, поставщики, заказчики, транспортные средства, станки, проекты, заказы, продукты и т.п.* Под активностью сущностей подразумевается способность агентов воспроизводить в программных системах поведение этих сущностей. При этом под поведением материальных и виртуальных объектов подразумевается бизнес-логика, которая определяется целью их существования или использования. Например, целью использования грузового транспортного средства может являться обеспечение рентабельности его использования. В таком случае поведение агента транспорт-

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

ного средства может подразумевать поиск, анализ, выбор и заключение выгодных контрактов по выполнению грузоперевозок.

Указанная особенность многоагентных систем неразрывно связана с воспроизведением следующих основных свойств, присущих широкому классу реально существующих систем.

Автономность. Поведение агентов прикладных систем определяется их ролью в системе. При этом агенты могут иметь свои индивидуальные цели, отличные от целей системы в целом. В соответствии с этим они способны самостоятельно планировать собственное поведение для выполнения роли и/или для достижения своих целей, осуществляя самоконтроль над своими действиями и внутренним состоянием.

Децентрализация. Агенты прикладных систем, подобно тому, как это происходит в реальных системах, в рамках автономного поведения наделяются полномочиями принимать решения, а не только действовать по чьим-то командам. В соответствии с этим агенты могут проактивно (по собственной инициативе) принимать решения на выполнение тех или иных действий в зависимости от текущей ситуации.

Взаимодействия. Автономность и децентрализация не исключает, а напротив, предполагает активные и интенсивные взаимодействия между антами. Поведение агентов предполагает обмен информацией, координацию действий, согласование решений, и т.п. Например, агенты транспортных средств могут согласовывать между собой, какой заказ будет выполняться каким грузовиком. Способность взаимодействия агентов имеет принципиальное значение, поскольку именно от него в большинстве случаев зависит возможность обеспечения и качество реализации всех других свойств. Следует заметить, что эффективность взаимодействия элементов реальных систем (например, подразделений производственных предприятий) самым непосредственным образом предопределяет эффективность реализации бизнес-процессов и функционирования систем в целом. Принципиальное значение этого свойства нашло свое отражение в том, что современные тенденции в разработке программных систем, на которых, в том числе, основаны и многоагентные системы, называют «вычислениями, основанными на взаимодействиях».

Открытость. Множество сущностей (объектов, субъектов), формирующих реальные системы, в большинстве случаев может меняться во времени. Следствием этого является необходимость постоянного поиска новых (в частности, производственных) связей. Например, для производственных предприятий характерен постоянный поиск новых поставщиков и потребителей. При этом априори неизвест-

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

но, с кем из них в будущем будут заключаться контракты. По аналогии с этим состав агентов прикладных систем может постоянно обновляться. Когда в системе появляются новые агенты, они способны «знакомится», т.е. устанавливать связи с другими агентами, которые могут предоставлять им необходимые сервисы, или наоборот, потреблять предоставляемый ими сервис.

Распределенность. Использование многоагентных технологий является естественным подходом для разработки приложений в виде распределенных систем. В данном случае подразумеваются как физически распределенные системы, например, относящиеся к классам P2P или B2B, так и распределенные вычисления. Распределение агентов (автономных вычислительных процессов) на разных вычислительных устройствах/процессорах является естественным подходом к реализации распределенных вычислений, что может существенно повышать производительность прикладных систем. Тем не менее, это вовсе не умаляет преимуществ использования многоагентного подхода для решения прикладных задач, когда агенты функционируют на одном вычислительном устройстве в псевдопараллельном режиме.

3. Примеры использования МАС на практике. Известно достаточно много примеров практических задач, с решением которых существующие классические подходы справиться не в состоянии. В частности, здесь достаточно вспомнить практические проблемы, которые сводятся к решению задачи оптимизации. К настоящему времени разработано достаточно большое количество различных методов («спуска с горы», «имитации отжига», генетических и др.) для решения задач этого класса. Тем не менее, возможность использования этих методов на практике имеет существенные ограничения, которые в каждом отдельном случае определяются соотношением размерности поискового пространства, ограничениями времени на поиск решения задачи, и производительностью вычислительного устройства. Однако, на практике реальные проблемы все-таки «как-то» решаются людьми, а МАС, как это уже упоминалось выше, в основе своей «пытаются» повторить свойства и организацию реальных систем. Поэтому, их использование в таких случаях в перспективе может обеспечить достижение качественно нового уровня решений. В качестве иллюстрации этого утверждения далее в этом пункте рассматриваются примеры использования многоагентного подхода для решения двух разных практических задач: организации воздушного движения (ОВД) в воздушном пространстве аэропорта (ВПА) и управления заказами такси.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Основными требованиями ОВД являются обеспечение выполнения расписания прибытия и убытия воздушных судов (ВС) и соблюдение норм безопасности между ВС во время прохождения ими ВПА. В настоящее время ОВД в зоне аэропорта осуществляется диспетчерской службой. ВПА разбито на несколько зон, в каждой из которых за ОВД отвечает диспетчер зоны. Диспетчер зоны в режиме реального времени получает текущую информацию о параметрах движения ВС в своей зоне и координирует дальнейшее развитие текущей ситуации, выдавая экипажам ВС соответствующие указания о необходимых изменениях параметров движения. По сути, поведение диспетчера сводится к решению задачи оптимизации - он оценивает множество возможных вариантов продолжения движения каждым ВС и выбирает наилучшие на его взгляд варианты. Решение этой задачи в настоящее время выполняется без какой-либо компьютерной поддержки, и качество ее решения полностью определяется субъективными факторами - уровнем подготовки диспетчера, его состоянием, степенью напряженности текущей ситуации и т.п. Возможным следствием этого может являться нерациональное использование воздушного пространства, влекущее задержку или опоздание рейсов. В связи с этим задача поиска новых принципов ОВД является весьма актуальной. В том числе это предполагает разработку и использование соответствующего программного обеспечения. Многоагентный подход в данном случае является одним из наиболее перспективных решений. В частности, пример такого решения, реализующий концепцию «свободных полетов» (free flight), был разработан авторами настоящей статьи [1]. В разработанной системе экипажу каждого ВС, находящегося в ВПА, назначается агент-ассистент. Эти агенты формируют возможные планы движения свих ВС и, на основании P2P взаимодействия между собой, принимают скоординированные решения. Предложенный подход дает сразу несколько качественно новых преимуществ, которые в целом позволяют достигнуть более рационального использования ВПА и существенно снизить нагрузку диспетчеров. Такие выводы были сделаны квалифицированными экспертами данной предметной области, и подтверждены результатами экспериментальных исследований.

МАС планирования и оптимизации использования ресурсов такси была разработана компанией Magenta по заказу компании Аддисон Ли (Лондон) [5]. Задача характеризуется следующими количественными параметрами. В день поступает более 13 тысяч заказов, таксопарк насчитывает около 2000 машин, оснащенных средствами GPS-навигации, прием заказов выполняет 130 диспетчеров. В разработан-

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

ной системе для каждого заказа и каждой машины назначается свой агент, и планирование выполнения заказов выполняется в процессе их взаимодействия. Среди основных функциональных возможностей разработанной системы в первую очередь следует отметить возможность перепланирования исполнителей заказов. Целесообразность решения этой задачи определяется тем, что после события получения заказа и до момента принятия решения по его выполнению (назначения машины) имеется интервал времени. В течение этого интервала времени поступление других заказов может существенно менять текущую ситуацию и влечь за собой изменение начального решения.

До внедрения этой системы все управление заказами выполнялось диспетчерами компании. Внедрение многоагентной системы позволило получить следующие организационные и экономические эффекты. 98,5% заказов теперь планируются автоматически, без участия диспетчеров; число заказов, выполненных не вовремя, сократилось в 3,5 раза; на 22,5% уменьшился холостой пробег такси; каждое такси теперь выполняет по 2 дополнительных поездки в неделю при тех же затратах времени и горючего. В целом это обеспечило получение дополнительной прибыли компании около 5 миллионов долларов в год.

Описание других успешных примеров использования многоагентного подхода для решения практически важных задач можно найти на сайте Agentlink – европейского проекта по координации исследований в области МАС [3].

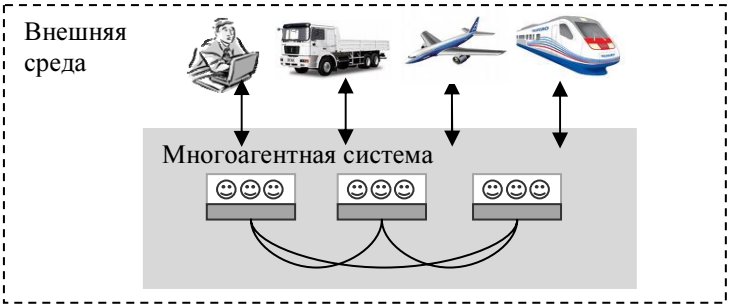

Рис.1. Подсистемы прикладных многоагентных систем

4. Внешняя среда МАС. Прикладные МАС, как правило, состоят из двух подсистем (Рис. 1). Первой и основной подсистемой является собственно сама многоагентная подсистема, которая реализует основную содержательную функциональность приложения. При этом, агенты являются представителями сущностей предметного мира, описание

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

которых также является неотъемлемой частью прикладной системы. Описание сущностей относится ко второй подсистеме, в целом описывающей внешнюю (окружающую) среду агентов. Эта подсистема состоит из компонент, выполняющих по отношению к агентам функции «передачи им данных из внешней среды» и «исполнение действий агентов во внешней среде». Этими компонентами могут быть приложения пользователя или различного рода устройства (сенсоры, датчики, контроллеры и т.п.). Далее в этом пункте приведены примеры двух подсистем разного типа, описывающих внешнюю среду двух разных МАС.

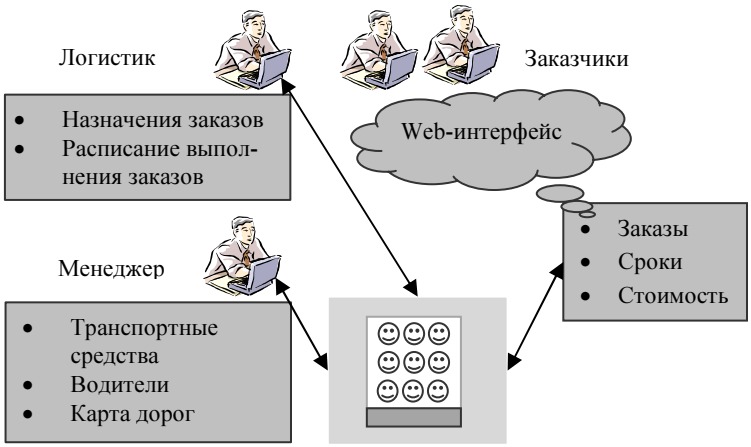

Рис.2. Пример внешней среды для МАС

Первым примером является окружающая среда МАС, предназначенной для решения задач транспортной логистики. Прикладная система предназначена для планирования использования транспортных средств компании для выполнения поступающих заказов на перевозку различного рода грузов. В данном случае окружающая среда МАС, например, может состоять из следующих трех пользовательских приложений (Рис.2). Первое приложение является рабочим местом «Менеджера» компании, и предназначается для описания транспортных средств компании, водителей и дополнительной информации, например, – карты дорог региона, в котором выполняются заказы на перевозки грузов. Второе приложение является рабочим местом «Логистика» компании и предназначается для отображения результатов работы МАС, а именно – для отображения формируемого плана выполнения

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

заказов. Эти два приложения по сути своей являются «десктопными» и размещаются в рамках локальной сети офиса компании. Третье приложение предназначается для ввода и контроля выполнения заказов «Заказчиками». Эти возможности предоставляются заказчикам посредством web-доступа к прикладной системе компании.

В данном примере агенты МАС могут представлять следующие сущности предметного мира: транспортные средства, водители и заказы. По мере описания в подсистеме внешней среды новых экземпляров сущностей в подсистеме МАС происходит добавление новых агентов соответствующих типов, и начинается функционирование этих агентов. Например, при добавлении новых заказов создаются агенты заказов. Возникновение этих событий инициирует поиск решений – поиск свободных ресурсов (транспортных средств и водителей) и определение времени выполнения этих заказов. Решение этих задач инициируют агенты в соответствии с описанным в МАС поведением. Это могут быть как агенты заказов, так и агенты транспортных средств или водителей. Поиск решения предполагает взаимодействие агентов. Например, агент заказа посылает запрос агентам транспортных средств, получает от них предложения о выполнении заказа, выбирает из них наилучшее предложение, и посылает агенту выбранного транспортного средства уведомление о том, что он согласен на его предложение. Это решение, если это предусмотрено правилами функционирования прикладной системы, может быть подвергнуто ревизии со стороны «Логистика» и/или «Заказчика», и далее передано на исполнение «Менеджеру» компании.

Вторым примером является модель окружающей среды МАС, предназначенной для управления работой автоматизированной сборочной линии, а именно – сварки кабин грузовых автомобилей (Рис.3). Этот пример взят из работы [4], которая посвящена детальному описанию специализированной методологии проектирования таких систем в виде МАС, и возможностей их практического применения. В данном примере сборочная линия функционирует следующим образом. На линии используется несколько «автоматически управляемых тележек». По территории сборочной линии тележки могут передвигаться между отдельными станциями, расположенными в разных зонах, в которых выполняются те или иные операции, по определенным маршрутам и в определенном направлении. В зонах загрузочной станции (зоны 0101 .. 0110) выполняется «ручная» погрузка комплектующих элементов одной кабины на одну тележку. На станции контроля (зона 0201) выполняется проверка наличия всех комплектующих, необходимых для

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

сборки кабины. Сборка кабин выполняется на роботизированных сварочных станциях (зоны 0301..0305). Собранные кабины снимаются с тележек на разгрузочной станции (зона 0401). Также в определенные моменты времени тележки могут находиться в зонах ожидания (зоны 0070..0077). Во всех зонах сборочной линии расположены контроллеры. В настоящее время управление сборочными линиями выполняется программами, которые исполняются контроллерами. Под управлением в данном случае понимается составление расписания выполнения операций на станциях линии с учетом их занятости, и обеспечения бесконфликтного передвижения «тележек» между этими станциями. Эф-

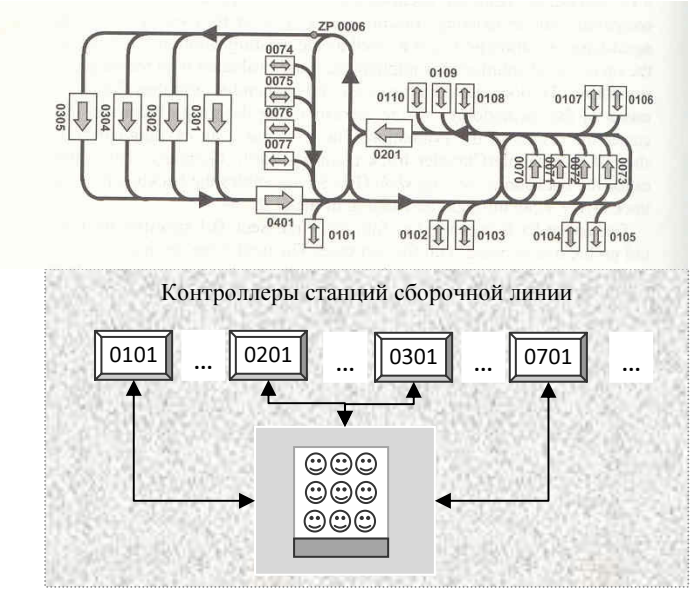

Рис.3. Модель окружающей среды для MAC

фективность реализованной логики управления в конечном итоге определяет производительность сборочной линии, т.е. сколько кабин может быть собрано за определенный интервал времени.

В работе [4] предлагается разрабатывать системы управления сборочных линий в виде MAC. Основным объектом данной работы является детальное описание разработанной методологии DACS (Designing agent-based control systems) проектирования подобных систем. В соответствии с этой методологией в данном случае предлагается использо-

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

вание агентов, соответствующих тележкам и автоматизированным сварочным станциям. Эти агенты взаимодействуют между собой по предопределенным протоколам и формируют планы действий соответствующих им сущностей. Контроллеры сборочной линии в данном случае являются компонентами внешней среды агентов. Отличительной особенностью данного примера от предыдущего является то, что компонентами окружающей среды агентов являются физические устройства – программируемые контроллеры, которые являются для агентов одновременно и сенсорами (передают агентам данные о состоянии станций) и эффекторами (исполняют команды агентов).

Следует отметить, что разработанная методология DACS была апробирована инженерами, которые разрабатывали реально функционирующую систему управления для данной линии, т.е. составляли программы для контроллеров. По оценкам этих инженеров такой подход имеет ряд преимуществ по сравнению с традиционным подходом. Во-первых, использование многоагентного подхода позволяет существенно упрощать и ускорять решение задачи разработки системы управления. Во-вторых, использование этого подхода может обеспечивать достижение более высокой производительности таких линий. В качестве недостатка было отмечено отсутствие в настоящее время CASE-средств разработки, поддерживающих разработанную методологию.

5. Абстрактная архитектура агентов. В этом и последующих разделах рассматриваются вопросы разработки прикладных МАС с помощью инструментальной среды MASDK. Архитектура агентов (Рис.4), разрабатываемых с помощью этой среды, состоит из следующих четырех компонент.

Поведение. Поведение агентов описывается с помощью одного или нескольких сценариев поведения. Каждый сценарий поведения выполняется в отдельном потоке управления. Это означает, что агент может одновременно выполнять несколько сценариев поведения в псевдопараллельном режиме.

Сценарии поведения агентов могут инициироваться тремя различными способами:

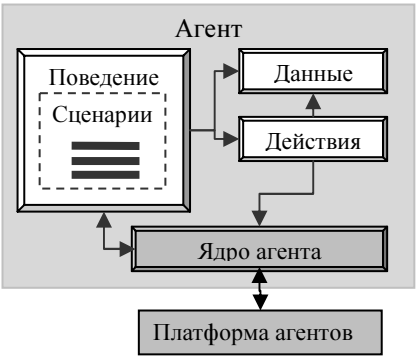

Рис.4. Абстрактная архитектура агента

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

- Триггером проактивного поведения,
- Сообщением от другого агента,
- Сообщением от компоненты окружающей среды.

Сценарии поведения состоят из множества узлов, в которых предусматривается выполнение тех или иных действий.

Действия. Эта компонента содержит библиотеку (или библиотеки) процедур, реализующих действия агентов, соответствующих узлам сценариев поведения.

Данные. В этой компоненте хранятся «персональные» данные агента, модифицируемые им в процессе функционирования.

Ядро агента. Инвариантная (по отношению к предметной функциональности агента) компонента, реализующая базовый (предметно-независимый) уровень поведения агента. Эта компонента, в частности, инициирует выполнение сценариев поведения, обрабатывает прерывания их исполнения, а также обеспечивает взаимодействия агента с платформой агентов (для посылки / получения сообщений, для публикации / поиска сервисов и т.п.).

6. Среда функционирования агентов. Специфика агентов, как автономных программных сущностей, состоит в том, что они могут функционировать только в определенной «функциональной среде», которая по отношению к агентам должна обеспечивать выполнение следующих функций и сервисов. Во-первых, среда должна управлять жизненным циклом агентов, а именно – управлять созданием и запуском, остановом и удалением агентов. Во-вторых, среда должна предоставлять агентам сервисы, обеспечивающие их взаимодействие между собой и взаимодействие с компонентами внешней среды.

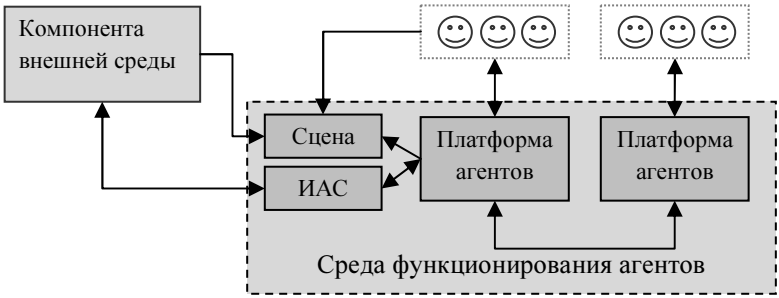

Рис.5. Среда функционирования агентов

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

В предлагаемой технологии основной компонентой, формирующей такую среду, является «платформа агентов». Для выполнения ряда функций и сервисов по управлению жизненным циклом агентов, а также для обеспечения взаимодействия агентов с внешней средой платформа использует две вспомогательных компоненты (Рис.5) – «сцену» и «интерфейсы активных сущностей» (ИАС). Необходимость использования вспомогательных компонент определяется в каждом конкретном случае в процессе проектирования прикладной системы.

6.1. Компонента «сцена». Компонента «сцена», в процессе функционирования прикладной системы, используется для управления конфигурацией агентов. Она является пассивной компонентой, содержащей данные, необходимые для реализации функциональности платформы агентов. «Сцена» является предметно-ориентированной компонентой, ее модельное описание разрабатывается в процессе проектирования прикладной системы. Это описание содержит:

- множество понятий онтологии, экземпляры которых могут быть представлены на сцене,
- сопоставление понятий онтологии классам агентов, и
- (при необходимости) множество различного рода отношений, которые устанавливаются между агентами и, соответственно, между сущностями предметного мира.

В процессе функционирования МАС компонента «сцена» используется компонентой внешней среды для описания экземпляров понятий онтологии (сущностей предметного мира), что составляет исходные данные для функционирования платформы агентов. Платформа агентов при запуске обращается к этой компоненте и на основании данных «сцены» создает соответствующих агентов.

Добавление новых сущностей, а также удаление или изменение описания ранее созданных сущностей может происходить после запуска платформы. Поэтому платформа обращается к компоненте «сцена» не только при запуске, но и в случае возникновения перечисленных событий. Это означает, что платформа может обеспечивать динамическое изменение конфигурации агентов в процессе функционирования.

В ряде случаев конфигурация агентов прикладных систем может включать список агентов, временем функционирования которых является все время функционирования прикладной системы. Это подмножество агентов может быть либо определено проектировщиком в процессе описания модели «сцены», либо описано пользователем при-

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

кладной системы при первой сессии редактирования «сцены» с помощью соответствующей компоненты внешней среды.

Данные, описываемые на «сцене» с помощью компоненты внешней среды, являются исходными данными для создания агентов. В процессе функционирования агенты могут использовать и изменять эти данные, либо непосредственно на «сцене», либо в своей рабочей памяти с последующей модификацией данных на «сцене». Таким образом, в результате функционирования агентов в компоненте «сцена» формируется результат решения задачи, который далее по назначению используется соответствующими компонентами внешней среды.

6.2. Компонента «Интерфейс активной сущности». В предыдущем пункте описан механизм взаимодействия агентов с внешней средой на основе формирования и обновления данных в компоненте «сцена». Наряду с этим также предлагается использование еще одного механизма, предполагающего взаимодействие агентов с компонентами внешней среды в соответствии с описанием в среде MASDK 1) протокола взаимодействия и 2) интерфейса активной сущности (ИАС) - интерфейса компоненты внешней среды. На основании этого описания генерируется программный код интерфейса компоненты, который используется в качестве готового решения при разработке компоненты. Если компонента внешней среды уже является готовым (функционирующим) модулем (устройством, программой, и т.п.), в этом случае рассматривается задача встраивания разрабатываемого интерфейса активной сущности в программную реализацию компоненты.

Этот механизм взаимодействия, в частности, предназначается для использования в тех случаях, когда компоненте внешней среды сопоставляется одна сущность предметного мира, которая может вести «активный диалог с агентом» в соответствии со сценарием описанного протокола. При этом полагается, что такой диалог может инициироваться как сущностью (компонентой внешней среды), так и агентом.

Выбор механизма для обеспечения взаимодействия агентов с внешней средой – использование компоненты «сцены» или компоненты «интерфейса активной сущности», осуществляется в процессе проектирования в соответствии со спецификой постановки задачи и с учетом предпочтения разработчика. В ряде случаев возможно использование как одного, так и другого механизма. Например, в прототипе МАС, предназначенном для управления воздушным движением в зоне аэропорта, задачей компоненты внешней среды является отображение текущей ситуации (Рис.6). При этом она может взаимодействовать с

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

агентами экипажей ВС как посредством использования компоненты ИАС, так и посредством компоненты «сцена».

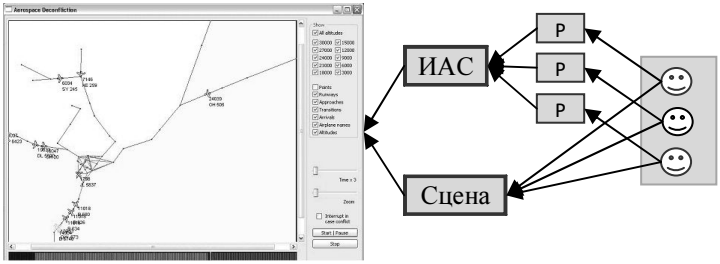

Рис.6. Пример взаимодействия агентов и компонент окружающей

6.3. Функционирование агентов. В двух предыдущих пунктах были описаны функции, которые обеспечиваются платформой агентов с использованием компонент «сцены» и «интерфейса активной сущности». Остальные функции и сервисы (знакомство агентов, взаимодействия между собой и останов агентов) обеспечиваются платформой агентов без использования вспомогательных компонент.

После запуска агента его поведением управляет компонента «ядро агента» (Рис.4). Эта компонента при возникновении соответствующих событий (срабатывание правил проактивного поведения, поступление сообщений от других агентов или от компонент окружающей среды) инициирует выполнение соответствующих сценариев поведения. Она также выполняет обработку прерываний выполнения сценариев, которые, в частности, могут быть связаны с ожиданием ответных сообщений при взаимодействиях агентов с другими агентами или компонентами внешней среды. В процессе функционирования агента выше перечисленные функции выполняются платформой агентов следующим образом.

Знакомство агентов. После создания и запуска агент еще «ни с кем не знаком». Он не знает про других агентов, и другие агенты не знают про него. Знакомство происходит уже в процессе функционирования с помощью сервиса «желтых страниц», предоставляемых экземплярами платформы агентов. С помощью этого сервиса агенты публикуют, и, при необходимости, обновляют описание предоставляемых ими сервисов, а также выполняют поиск требуемых им сервисов, предоставляемых другими агентами.

Сервис «желтых страниц» в предлагаемой технологии имеет распределенную структуру. Это означает, что агенты публикуют свои

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

сервисы на «желтых страницах» своего экземпляра платформы агентов. Но когда агент обращается к своему экземпляру платформы с запросом на поиск нужного сервиса, экземпляр платформы производит поиск уже на всех других экземплярах платформы, «доступных» ему прямо или опосредовано (т.е., через другие экземпляры платформы).

Взаимодействия агентов между собой. Взаимодействия агентов между собой сводится к обмену сообщениями в соответствии с протоколами взаимодействия, описываемыми в моделях МАС. Отправка и прием сообщений выполняется экземплярами платформ.

При отправке сообщения агент в качестве адресата сообщения указывает своему экземпляру платформы только имя агента - получателя сообщения, который может находиться на любом экземпляре платформы.

Экземпляр платформы агентов при получении сообщения «читает» только адрес сообщения и передает его агенту – получателю.

Следует заметить, что при использовании компоненты «сцена» в определенных случаях она также может использоваться для обеспечения взаимодействия агентов в виде «доски объявлений». Однако, возможности такого способа взаимодействия агентов по сравнению с использованием протоколов взаимодействия сильно ограничены.

Останов агента. Останов агента иначе рассматривается как его переход в «спящий режим» («suspending»). Этот переход выполняется самим агентом в случае, когда нет событий, инициирующих его поведение.

Остановка экземпляра платформы выполняется только в том случае, если получено «разрешение» ото всех агентов, функционирующих на этом экземпляре платформы. Агенты выдают разрешение на остановку платформы только в том случае, если в данный момент времени они не выполняют ни одного сценария поведения, т.е. все агенты находятся в «спящем режиме».

7. Жизненный цикл разработка прикладных МАС с помощью среды MASDK. В соответствии с архитектурой прикладных МАС их проектирование и разработка в целом предполагает решение трех взаимосвязанных между собой задач:

1. проектирование и разработка подсистемы «внешней среды»,
2. проектирование и разработка собственно подсистемы МАС,
3. проектирование и разработка сценариев взаимодействия агентов МАС с внешней средой.

В соответствии с методологией Gaia [8], положенной в основу среды MASDK [6], проектирование начинается с анализа внешней сре-

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

ды и сбора требований к разрабатываемой системе (Рис.7). В частности, эта задача предполагает выявление и/или описание компонент,

Рис.7. Жизненный цикл разработки прикладной МАС

которые формируют окружающую среду агентов. Проектирование и разработка этих компонент не имеет (по крайней мере, явно выраженной) агентской специфики, поэтому для решения этих задач используются традиционные средства проектирования, например на основе языка UML. При этом, необходимость решение этих задач определяется конкретной спецификой выполняемого проекта. В ряде случаев подсистемы «внешней среды» на момент начала проекта могут быть уже предопределенны, полностью или частично. Примером этого может служить рассмотренная в пункте 4 система управления автомати-

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

зированными сборочными линиями. В этом случае множеством компонент, формирующих подсистему внешней среды, являются контроллеры таких линий. В таких случаях подсистемы «внешней среды» являются отправной точкой для разработки проектов, и в отношении их выполняется только задача анализа.

Решение второй и третьей задач выполняется с помощью средств визуального проектирования в среде MASDK. Решение этих задач в целом составляет содержание задачи «Архитектурное и детальное проектирование МАС. Генерация программного кода». Главным образом, эта задача сводится к описанию сценариев поведения класса агентов (Рис.4). Детальное описание решения этих задач приведено в «Руководстве пользователя среды MASDK» [2].

На основе модели МАС, создаваемой в среде MASDK, генерируется программный код, описывающий проблемно-ориентированное поведение классов агентов МАС. Генерируемый код интегрируется с готовой программной реализацией «абстрактной архитектуры агента» (Рис.4).

Сценарии поведения классов агентов состоят из действий, которые в модели МАС описываются на вербальном (концептуальном) уровне и специфицируются в терминах модельных переменных и параметров. В связи с этим сгенерированные библиотеки программных классов, описывающих действия классов агентов, содержат только наименования методов, соответствующих действиям, и объявления параметров и переменных. Процедуры (тела методов) разрабатываются обычным образом. Это является содержанием задачи «Развитие программного кода» (Рис.7). Такой подход, с одной стороны, позволяет гибко развивать автоматически сгенерированный программный код классов агентов за счет разработки или использования (подключения) произвольных, сколь угодно сложных процедур и/или программных компонент, реализующих действия классов агентов. С другой стороны, он позволяет сохранять полное соответствие модели МАС и программного кода классов агентов в процессе итеративной разработки прикладных систем, в течении которой как модель МАС, так и дополнительно разрабатываемый программный код могут быть подвержены многократной модификации.

Результатом разработки прикладной МАС является набор программных компонент четырех типов – компоненты внешней среды (КВС), включающие соответствующие компоненты ИАС, классы агентов, компоненты «сцен». Платформа агентов (ПА) в предлагаемой технологии поставляется в виде готовой программной компоненты.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Разворачивание (установка) разработанных и отлаженных программных компонент сводится к выполнению следующей последовательности действий.

1. Установка на вычислительных устройствах экземпляров платформы агентов и описание начальных связей между ними.
2. Установка программного кода классов агентов на вычислительные устройства.
3. Формирование окружающей среды. Формирование окружающей среды сводится к установке разработанных компонент окружающей среды и настройке экземпляров компонент ИАС, реализующих взаимодействия компонент окружающей среды с агентами МАС. Под настройкой понимается установление связей экземпляров компоненты ИАС с требуемыми экземплярами платформы агентов. Если в состав окружающей среды входят готовые приложения или устройства, тогда выполняется только настройка экземпляров компоненты ИАС.

Агенты МАС могут создаваться как при установке (разворачивании) прикладной системы, так и в процессе ее функционирования, на основании механизмов, описанных в подразделе 6.

9. Заключение. В данной статье рассматриваются вопросы разработки прикладных многоагентных систем с помощью инструментальной среды MASDK, развиваемой в СПИИРАН. Успешность разработки любой прикладной программой системы в значительной мере определяется качеством (функциональными возможностями, дружелюбностью, и т.п.) интерфейса, предоставляемого конечному пользователю. Многоагентные системы в этом смысле не являются исключением, но наряду с этим обладают рядом специфических особенностей. Эти особенности связаны с необходимостью взаимодействия активных сущностей двух типов: конечных потребителей системы (которыми могут быть как люди, так и различного рода устройства) и агентов. В статье рассматриваются решения, предложенные разработчиками инструментальной среды MASDK для снижения трудоемкости процесса разработки компонент взаимодействия агентов с внешней средой. Использование таких решений позволяет разработчикам сосредоточиться на описании поведения и взаимодействия элементов прикладной системы и не тратить усилия на решение сугубо технических задач.

Жизненный цикл разработки прикладных систем с помощью среды MASDK в статье приведен на достаточно общем уровне, чтобы дать минимальную необходимую информацию об использовании сре-

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

ды. Детальное описание среды и ее программная реализация доступны на Web-сайте [2].

Литература

1. *Городецкий В.И., Карсаев О.В., Конюший В.Г., Купин В.В., Самойлов В.В.* Моделирование процессов управления воздушным движением на основе многоагентных технологий // Научный вестник МГТУ ГА, серия Навигация и УВД, 2008
2. Инструментальная среда разработки многоагентных систем MASDK // http://space.iias.spb.su/products/
3. Agentlink // http://eprints.agentlink.org/view/type/project.html.
4. *Bussmann S., Jennings N., Wooldridge M.* Multiagent Systems for Manufacturing Control, Springer, 2004
5. *Claschenko A., Ivaschenko A., Rzevski G., Scobelev P.* Multi-Agent Real Time Scheduling System for Taxi Companies // Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009), Decker, Sichman, Sierra and Castelfranchi (eds.), May, 2009, Budapest, Hungary, pp. 29 – 36
6. *Gorodetsky V., Karsaev O., Samoylov V., Konushy V.* Support for Analysis, Design and Implementation Stages with MASDK // LNCS 5386, pp. 272–287, 2009
7. Magenta Technology: Software Platform V 2.1 // http://www.magenta-technology.com/docs/Magenta%20Platform%20Whitepaper_Summer07.pdf
8. *Zambonelli F., Jennings N., Wooldridge M.* Developing Multiagent systems: The GAIA methodology//ACM Transactions on Software Engineering and Methodology, 12(3), pages 417-470, 2003.

Карсаев Олег Владиславович — к.т.н., заведующий лабораторией интеллектуальных систем Учреждения Российской академии наук Санкт-Петербургский институт информатики и автоматизации РАН (СПИИРАН). Область научных интересов: многоагентные системы, инструментальные средства для создания распределенных систем, распределенные системы принятия решений и управления, машинное обучение и извлечение знаний, P2P системы, управление бизнес–процессами, планирование и управление, транспортная логистика. Число научных публикаций — более 60. ok@iias.spb.su, http://space.iias.spb.su; СПИИРАН, 14-я линия В.О., д. 39, г. Санкт-Петербург, 199178, РФ; р.т. +7(812)328-3337, факс +7(812)328-4450.

Karsaev Oleg Vladislavovich — PhD, head of Intelligent Systems Laboratory, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: multi agent systems, distributed software systems development tools, distributed decision making and control, data mining, P2P systems, business process management, planning and control, transportation logistics. The number of publications — более 60. ok@iias.spb.su, http://space.iias.spb.su; SPIIRAS, 39, 14-th Line

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

V.O., St. Petersburg, 199178, Russia; office phone +7(812)328-3337, fax +7(812)328-4450.

Конюший Виктор Григорьевич — научный сотрудник лаборатории интеллектуальных систем Учреждения Российской академии наук Санкт-Петербургский институт информатики и автоматизации РАН (СПИИРАН). Область научных интересов: многоагентные системы, инструментальные средства для создания распределенных систем, распределенные системы принятия решений и управления, планирование и управление, транспортная логистика. Число научных публикаций — 20. kvg@iias.spb.su, http://space.iias.spb.su; СПИИРАН, 14-я линия В.О., д. 39, г. Санкт-Петербург, 199178, РФ; р.т. +7(812)328-3337, факс +7(812)328-4450.

Konyushiy Victor Grigorievich — research fellow, Intelligent Systems Laboratory, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: multi agent systems, distributed software systems development tools, distributed decision making and control, planning and control, transportation logistics. The number of publications — 20. kvg@iias.spb.su, http://space.iias.spb.su; SPIIRAS, 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812)328-3337, fax +7(812)328-4450.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru