

Открытая многоагентная среда: концепция и платформа

И.Г. Харин, А.В. Яновский

Аннотация. Рассмотрены основные проблемы организаций агентов в открытой многоагентной среде. Описаны концепция и методы построения платформы для открытой многоагентной среды, способы моделирования организаций агентов с использованием XMPP- и HTTP-протоколов. Показано, что XMPP-протокол позволяет реализовать динамическую природу этих организаций. Разработаны архитектура и основные компоненты платформы eSocialGRIDs для распределенной открытой многоагентной среды.

Ключевые слова: агент, многоагентная система, организация агентов, архитектура платформы, открытая многоагентная среда, социальные приложения, мгновенные сообщения, облако сервисов.

Введение

В настоящее время наблюдается бурное развитие персонализации и социализации сетевых ресурсов. При этом онлайн-социальная активность пользователей настолько возросла, что возникает необходимость ее обслуживания в достаточно короткий промежуток времени. В частности, растет потребность в социальных приложениях с поддержкой коммуникаций реального времени. В качестве примера можно привести системы обмена “мгновенными сообщениями” в социальных сетях Facebook, Skype или Google+, а также саму социальную сеть, полностью основанную на имитации обмена “мгновенными сообщениями” - Twitter. В этих системах естественно участие различных программных агентов, от служб перевода в Google Gtalk, до виртуальных собеседников (chatterbots) и рекламных роботов. Уже ясно, что в ближайшем будущем будут широко востребованы подобные системы с программными агентами в качестве активных участников образующихся сообществ. Для описания на языке вычислительных процессов комплекса возникающих при этом проблем наиболее перспективны подходы на основе концепции многоагентных систем (МАС) открытого типа.

В связи с вышесказанным, в последнее время особенно активно исследуются МАС смешанного состава, т.е. когда составляющими могут быть роботы, люди или команды людей. Такие системы называются виртуальными организациями [1]. К ним можно отнести так называемые GRID-системы. Более того, в последние годы в области вычислительной технологии и коммуникаций (WWW, e-commerce, мобильных приложений и т.д.) возникает новая парадигма прикладных вычислений: “вычисление как взаимодействие”. Примером является организация прикладного вычисления в виде сервиса. Т.е. вычисление организуется среди вычислительных сущностей, объединённых в организацию. Вычисление здесь становится, по сути, социальной активностью, а не единичной, что приводит к новым формам понимания, разработки и управления вычислительными системами. Технология МАС, особенно открытых МАС, имеет нужные характеристики, которые демонстрируют ее потенциал для реализации парадигмы “вычисление как взаимодействие”, как в виде формальных моделей, так и программного обеспечения.

Очевидно, что динамические организации агентов, которые самонастраиваются с целью

получения выгоды в некотором контексте, имеют большие возможности практических применений. Такие организации могут возникать в тех системах, где компоненты-агенты координируются между собой и другими компонентами (не агентами), чтобы образовать составной сервис, примером являются GRID-системы. Другими словами, социальные факторы в организациях МАС – все более и более важны, чтобы структурировать взаимодействия в динамичных открытых системах. Как отметил Л. Гассер [2], почти все предложенные агентные архитектуры имеют общую проблему: каким образом обходиться с наборами взаимодействующих агентов на высшем уровне абстракции – т.е., как обходиться с организациями, подобно тому как это делается, например, с агентами, входящими в организацию. Для решения этой проблемы в области программной инженерии необходимо модернизировать концепцию проектирования агентов и методы разработки соответствующих платформ, адаптируя определение агентов с учетом возможности динамического создания из них самих других агентов. Развитие программного обеспечения в этом направлении требует новых подходов и инструментов для описания и поддержки эволюции многоагентных инфраструктур, обеспечения их роста и изменения во время работы с учетом характеристик открытых систем.

Главная цель этой статьи — разработать концепцию и принципы функционирования универсальной открытой многоагентной среды, в рамках которой смогут существовать и взаимодействовать различные МАС, построенные на основе разных многоагентных платформ. Исходя из последовательного формального понятия об агентах и МАС, мы строим концепцию открытой и закрытой многоагентной среды (МС). Также мы рассмотрим методы моделирования организаций агентов и архитектуру платформы для открытой МС, где коммуникации программных объектов подобны общению в реальном времени между людьми с помощью XMPP-протокола. В данном подходе исследуются возможности конструирования организаций с использованием HTTP-протокола на основе одной или более организационных единиц, которые строятся с помощью соответствующих XMPP-сущностей. Рассматриваются методы регистра-

ции и защиты в таких системах. Определяется новая архитектура открытых многоагентных систем, в рамках которой для решения текущих задач объединяются агенты и сервисы, взаимодействуя в реальном времени между собой. В рамках этой архитектуры разработаны базовые компоненты платформы eSocialGRIDs (electronic social grids). Рассматривается основная функциональность платформы и методы ее использования для моделирования организаций.

1. Формальные рамки описания МАС

Очертим формальные рамки используемого подхода описания МАС и уточним понятие распределенных масштабируемых многоагентных сред. Будем называть “сущностями” на высшем уровне абстракции, подобно [3], все составные части рассматриваемых систем, т.е. всё, что можно представить фреймом в некоторой формальной модели таких систем. При этом система в данный момент времени – это один из вариантов набора сущностей из заданного универсума. Обратим внимание, что при таком подходе сущностями также можно считать и связи между частями, и акты взаимодействия.

Известно, что в любой системе можно выделить две основные части: это способные к активности сущности (объекты) [3], которые могут изменять состояние среды, и пассивные сущности (не объекты), которые не могут воздействовать на другие сущности. В частности, очевидно, агенты – активные сущности, а значит, являются объектами. Но в отличие от простых объектов, агенты являются объектами, которые имеют возможности для назначения им, или усваивания ими целей, что определяет их действия на собственном окружении в поддержку этих целей. Надо отметить, что в данной статье принято, как и в [3], что простой объект отличается от агента отсутствием целей и функциональности для выстраивания своей активности при достижении этих целей. К тому же объект не может воспринимать среду, т.е. его взаимодействие с нею не обусловлено внутренней целью или мотивацией.

Итак, мы считаем агентом (как и в [1, 3-6]) сущность, которая интерпретирует состояние среды и действует на основании этой интерпре-

тации сообразно определённым представлениям о системе, частью которого он является, сообразуясь со своими целями.

При несколько более слабых условиях, чем в [3] здесь принято, что МАС - любая система, которая содержит:

1) два или более агента;

2) по крайней мере, одно соотношение между двумя агентами, где действия одного агента зависят от цели другого (согласовываются, или конкурируют, или как-то комбинируются).

С одной стороны, при определенных условиях МАС может рассматриваться, как единая сущность – агент. С другой стороны, любую связанную совокупность агентов можно рассматривать формально, как сущность с некоторым интегральным представлением о мире и обобщенной целью, производными, соответственно, от представлений и целей отдельных агентов. В связи с этим целесообразно обобщить определение агента, чтобы учесть следующие варианты:

- агент как атомарный агент – это атомарная или составная сущность, у которой каждая отдельная компонента не является сама по себе агентом;

- агент как МАС – это система сущностей, содержащая в своем составе больше одного агента.

Данное обобщение и было сделано в [4], посредством введения рекурсивного понятия об абстрактном агенте (АА) и соответствующего определения для МАС. Приводим формально более корректный вариант рекурсивного определения АА и МАС, основывающийся на альтернативном по отношению к [4] структурном подходе из [3]:

Определение 1.1. АА – это атомарная сущность или система сущностей, которая рассматривается как целое (в смысле представимости некоторым фреймом) в виде объекта, воспринимающего свое окружение. В зависимости от восприятия этот объект интерпретирует ситуацию и выполняет действия. Эти действия АА зависят от его целей и меняют систему, частью которой он является. АА может быть атомарным агентом (атомарная или составная сущность, у которой каждая отдельная компонента не является сама по себе агентом) или МАС.

Определение 1.2. МАС — это совокупность двух или более АА, из которых как минимум два взаимодействуют, чтобы решить проблемы и достичь целей свыше индивидуальных способностей и знаний каждого АА.

Эти определения связаны между собой, выражают понятия АА и МАС друг через друга, и в основе имеют понятие атомарного агента, которое совпадает с традиционными определениями агента [1, 3-6]. Ниже для удобства мы будем иногда использовать термин “агент”, имея в виду определенный АА, или некоторую характеристику АА-а, или сущность, связанную с АА, например, программный агент, реестр агентов и т.д. Это будет означать, что имеется в виду либо МАС, либо атомарный агент, что удобно при анализе в пренебрежении структурой агента или при начальном моделировании. Заметим, что многие известные общие утверждения, куда входит термин “агент”, верны именно при употреблении этого термина в смысле АА. Таким образом, в подобных случаях, используя понятие АА, нам не придется дважды записывать схемы и утверждения: один раз для атомарных агентов, другой раз для МАС.

Понятие АА находится на более высоком уровне абстракции, чем атомарный агент или МАС. В то же время термин АА существует только на стадии формального рассмотрения. При моделировании реальными сущностями он должен быть заменен либо атомарным агентом, либо МАС.

Определение 1.2 расширяет традиционное представление о МАС, учитывая, что МАС может состоять из “меньших” МАС. Это позволяет описывать системы, в которых строительными блоками являются взаимодействующие МАС, сотрудничающие для достижения глобальных целей (здесь глобальная цель – это цель всей системы как целого). Таким образом, рекурсивная структура АА может быть представлена в виде UML-диаграммы на Рис.1. Здесь организация агентов является частным случаем МАС и понимается, как социальная сущность, составленная из определённого числа АА, которые обслуживают несколько отдельных задач. Агентная организация структурирована, следуя определенной топологии и коммуникативным отношениям для того, чтобы

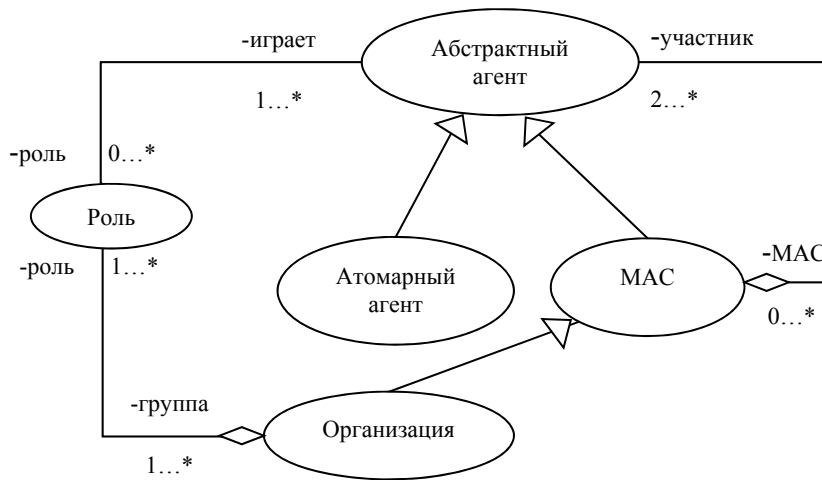


Рис. 1. Абстрактный агент

На данном уровне абстракции сущности соответствуют фреймам, которые обозначены овалами

достигать главных целей организации [7]. Организации агентов предполагают существование глобальных целей вне любых индивидуальных АА, входящих в организацию, и они существуют независимо от отдельных АА.

В одних работах среду включают по определению в состав МАС [1], в других не включают [3]. Из вышеприведенного определения можно сделать вывод, что это вопрос удобства, зависящего от предметной области, так как искусственную систему, которая содержит набор связанных АА, иногда удобно в целом саму рассматривать как МАС. Заметим, что рекурсивному определению АА это никак не противоречит, что говорит об успешном разрешении в рамках подобного подхода некоторой неоднозначности понятий МАС и МС. В то же время, если, например, МАС – это система M , представляющая из себя космический корабль, типа робот-носитель, и роботов-сборщиков данных, предназначенных для исследования планетарной среды (атмосферы и поверхности планеты), то весьма неудобно рассматривать объединение M и такой среды в качестве МАС. Вообще говоря, часто удобно рассматривать МАС и конкретную среду отдельно. При таком подходе следует, что конкретная МАС привязана скорей к типу среды, а не к конкретному экземпляру среды. Поэтому, когда мы будем анализировать поведение данного АА в среде, предполагая, что он должен учитывать цели других АА, мы будем называть такую среду МС. При этом в каждом

конкретном случае, анализируя систему, мы будем явно определять те сущности, которые составляют определенную МАС, а остальное будем называть МС. Учитывая вышесказанное, можем дать следующее определение.

Определение 1.3. Многоагентная среда (МС) – это такая среда, где созданы условия для жизнедеятельности, по крайней мере, двух АА, и поведение хотя бы одного АА при достижении им своих целей зависит от целей другого.

Существует множество практических реализаций вычислительных МС. Одна из перспективных платформ для МС – eSocialGRIDs, описываемая в данной работе. К ней наиболее близка по принципам построения известная агентная платформа SPADE [8]. eSocialGRIDs и SPADE, как и большинство агентных систем, используют спецификации FIPA¹.

2. Открытые МАС и МС

МАС, как и МС могут быть открытыми и закрытыми. При моделировании проблемных сред всегда стоит вопрос, какую архитектуру, открытую или закрытую, выбрать. Закрытую систему обычно легче построить, поддерживать, однозначно интерпретировать результаты и использовать. Преимущество открытой архи-

¹ FIPA (Foundation for Intelligent Physical Agents) – это объединение специалистов в области МАС, с 2005 года функционирует как один из комитетов IEEE по стандартизации (<http://www.fipa.org>).

тектуры заключается в большей гибкости для применения с точки зрения дальнейшего развития. Открытые системы могут легко развиваться без существенных переделок при необходимости учета новых факторов. Также открытая архитектура позволяет решать широкий спектр проблем, объединяя в единую систему разрозненные МАС и проблемные среды. Так, в открытых системах могут сосуществовать неоднородные АА, разработанные различными коллективами с использованием различных языков программирования и архитектур.

В описанном выше подходе архитектура всей моделируемой (или проектируемой системы) складывается из архитектур МАС и МС. Уточним вначале понятие открытости и закрытости для МАС.

Определение 2.1. Открытые МАС – это такие МАС, которые позволяют различным типам АА подсоединяться к системе и выходить из нее. В открытой МАС поведение и взаимодействие отдельных АА преимущественно не предсказуемо, в противоположность **закрытым МАС**, где множество типов АА определено до создания этой системы, а взаимодействие и поведение ее компонент предопределено.

Открытые МАС предназначены, прежде всего, для существования в динамически изменяющейся структуре окружения, где новые компоненты могут быть интегрированы в систему или удалены из нее, а условия функционирования компонент могут непредсказуемо изменяться. Очевидно, что функционировать в таких условиях закрытой системе гораздо трудней, чем открытой. Уточним теперь понятие открытой МС.

Определение 2.2. Открытая МС – это МС, в котором есть основные условия для жизнедеятельности различных типов АА: система, организующая жизненный цикл АА, каналы передачи данных, реестр агентов, средства интерфейса и пр. В эту среду внедряются и удаляются новые АА, в том числе взаимодействующие друг с другом, интегрируются и удаляются новые компоненты, непредсказуемо меняются условия функционирования объектов. Наоборот, **закрытая МС** характеризуется неизменными условиями функционирования объектов и фиксированными типами входящих в ее состав сущностей.

Из указанных выше определений кроме открытой и закрытой МАС следует возможность промежуточных вариантов (комбинаций открытых и закрытых МАС с открытой или закрытой МС).

3. МАС на основе организаций

Практический смысл описываемой в этой работе концепции состоит в обеспечении пользователей эффективной моделью, средствами построения и поддержки организаций агентов. Будем использовать классификацию по типам организационных структур Л. Гассера [9]. Согласно этому подходу организации в МАС делятся на 4 группы. Не будем приводить подробное описание перечисленных групп, это можно посмотреть в [9], остановимся лишь кратко на самых общих их чертах, необходимых для дальнейшего рассмотрения. Наиболее однозначно определена и наименее динамична структура у *централизованной организации*, фактически представляющей собой иерархию с жесткими вертикальными связями. Более динамична структура у *рыночно-подобных* организаций.

Поскольку в IT-индустрии исторически и на данный момент все еще наиболее распространено предоставление услуг по контракту, то рассмотрим подробнее организации *рыночного типа*. Они основаны на контрактном принципе: агент-менеджер рассылает запрос (предложение) группе агентов. Некоторые АА предлагают свои услуги. Тогда агент-менеджер выбирает одного (или нескольких) из них и заключает контракт. Организации такого типа обычно образуются из независимых АА-ов, которые решают свои собственные задачи.

Сообщество с правилами поведения — это коллектив агентов, которые взаимодействуют и группируются согласно определенным правилам, как например, пользователи в социальной сети. Независимые АА могут образовывать *плюралистическое сообщество* для совместного решения задач, после чего результаты будут доступны другим АА. Те в свою очередь проверяют, отвергают или используют эти результаты для своих задач. Здесь типичным примером служит деятельность экспертных групп и сообществ, обеспечивающих потребности пользователей, обра-

тившихся к сервисам, предоставляемым данной вычислительной системой.

Из вышеизложенного ясно, что в данной статье, как и в [4] в основе понятия организации агентов используется понятие об абстрактном агенте. Концепция организационной единицы описывает, как формируются организации, из определенных групп АА-ов. Эти группы - Организационные Единицы (ОЕ)[7] имеют специфическую внутреннюю структуру; определяют несколько ролей, задающих функциональность (предлагаемые и необходимые сервисы) и цели, представляющие ожидания организации для каждой роли; включают ресурсы и приложения, которые могут быть доступны определенным членам организации; ОЕ определяют некоторые нормы контроля поведения их участников. В данной статье мы основываемся на организационной мета-модели из [7], в рамках которой ОЕ рассматривается с четырех точек зрения: структурной, функциональной, социальной и динамической.

Структурное представление определяет, что есть статические компоненты организации, т.е. все те элементы, которые не зависят от остальных, задействованных при выполнении вычислений. Таким образом, система строится из ОЕ, которые иногда могут рассматриваться как глобальные сущности (действующие в качестве АА-ов), или как групповые сущности или части организации. Однако надо не забывать, что в рекурсивном аспекте ОЕ может включать другие ОЕ, а значит, она должна иметь в своей основе некоторые атомарные или базисные организационные структуры. Мы рассматриваем только три базисные структуры, внутри которых отношения членов организации подразумевают иерархичную, командную и плоскую структуру. С помощью рекурсивной композиции, начиная с этих ОЕ, можно строить более сложные и законченные структуры, такие как матрица, федерация, коалиции или сообщества [10]. Таким образом, в рамках концепции ОЕ воспроизводятся все перечисленные выше типы организаций.

Функциональное представление описывает предназначение и поведение каждой ОЕ. Предназначение определяет: глобальные цели, кто есть заинтересованные стороны, которые взаимодей-

ствуют с организацией, что является результатами, поставляемыми ОЕ, как они употребляются клиентами и, что организации необходимо от вспомогательных сервисов и ресурсов. Кроме того, любая организация агентов имеет при своем создании тот или иной глобальный информационный контекст — заданную извне предметную задачу, которую необходимо решить, или цель, которую надо достичь. В некоторых случаях контекст определяет причину, по которой конкретную организацию или ОЕ вообще надо создавать. Информационный контекст уточняет задачи данной ОЕ, дополняя ее конкретными данными.

На оставшихся двух из вышеупомянутых четырех точек зрения остановимся более кратко, так как для дальнейшего рассмотрения достаточно самого общего их определения (подробнее можно посмотреть в [7]). **Социальное представление** описывает роли и социальные отношения между АА, разбитые на три типа: управление, мониторинг и порядок информационного обмена. **Динамическое представление** определяет шаблоны для построения сервисов ОЕ, которые способны управлять всеми своими структурными и динамическими компонентами.

4. Масштабируемая архитектура для открытых МС

В соответствии с приведенным выше анализом мы предлагаем к рассмотрению новую архитектуру для открытых МС с набором базовых моделей организаций агентов и реализующую эту архитектуру платформу eSocialGRIDs. Общий вид архитектуры eSocialGRIDs показан на Рис.2.

В основе архитектуры eSocialGRIDs лежат кластеры Jabber-серверов (ejabberd), использующие XMPP протокол, которые могут быть в свою очередь организованы в “облако”. В этих кластерах коммуникация между АА и другими объектами может, кроме XMPP-сообщений, производиться также посредством распределенной базы данных, так, что все объекты имеют возможность доступа к общим актуальным данным. Отметим, что это даже еще более сильное взаимодействие, чем XMPP протокол, но такое взаимодействие между АА разрешено

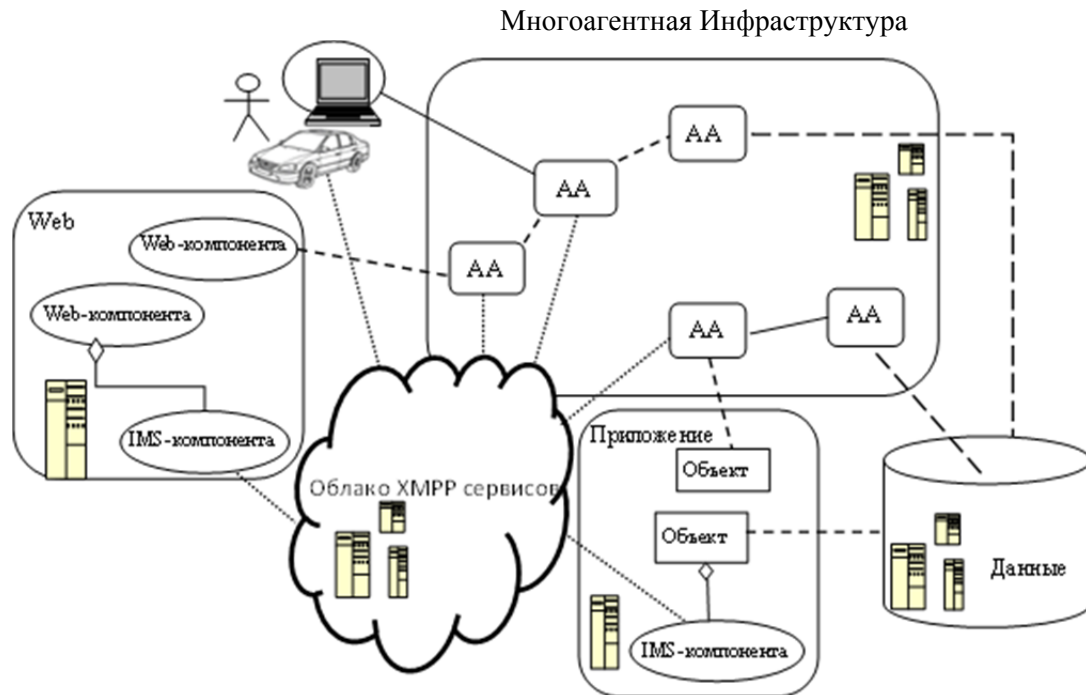


Рис.2. Архитектура eSocialGRIDs

Пунктирные, штриховые и сплошные линии обозначают коммуникационные, слабые (не постоянные) и сильные связи объектов соответственно.

лишь в пределах кластера. Каждый кластер может строиться независимо и соединяться с необходимой степенью открытости с главной Jabber-сетью. Для выхода за границы кластера предпочтительно использовать XMPP и HTTP протоколы, и особенно для выхода за границы облака. Соответственно, коммуникация объектов основана, прежде всего, на этих двух протоколах. В такой архитектуре любой объект, умеющий посылать или принимать сообщения, может получить свой уникальный идентификатор Jabber ID (JID) посредством АА. АА может, конечно, зарегистрироваться сам. Основой поддержки виртуальных организаций в этой платформе являются контактные списки, разбитые на группы (группы общих контактов и группы, управляемые самим АА), а также многопользовательские чаты (конференции). В предлагаемой архитектуре АА любого типа может получить JID, который в данном подходе играет роль идентификатора агента (в общепринятом смысле) в системе для установления соединения с платформой.

Изначально IM (Instant Messaging) предназначался для поддержки коммуникаций между

людьми. Очевидно, что использование IM технологии для обеспечения коммуникации агентов - это естественный шаг в развитии платформ, реализующих MAC. Можно сказать, что поведение АА с JID подобно поведению пользователя, который посылает и принимает сообщения, использует ресурсы и сервисы платформы. Как показано в работе [11], АА могут обмениваться информацией в соответствии с FIPA. Для этого достаточно встраивать исходное сообщение АА в XMPP-сообщение.

Таким образом, eSocialGRIDs может служить хорошей практической основой глобальной инфраструктуры для интеграции разрозненных многоагентных сред в единую MC. В этой MC АА общаются между собой через основную коммуникативную компоненту платформы eSocialGRIDs, основанную на XMPP-протоколе. Это позволяет в рамках единой системы значительно расширить возможности использования MAC различного рода. Инфраструктура такой MC может строиться путем присоединения "облака" eSocialGRIDs-сервисов к уже существующей системе ресурсов.

5. XMPP-сущности для поддержки организаций

В XMPP есть два наиболее полезных механизма для поддержки организаций агентов — это список контактов (СК, по-английски roster) и многопользовательский чат (по-английски MUC - multi users chat). На основе СК реализуется важная особенность XMPP – возможность нотификации присутствия (НП) любого пользователя из списка. Механизм НП позволяет пользователю менять свой статус в сети (т.е. “Доступен”, “Занят”, “Отключен” и т.д.), чтобы уведомлять свои контакты о своей готовности. Вообще говоря, пользователем может быть соответствующий объект, необязательно АА или человек.

После определенного процесса переговоров, который может быть автоматизирован, два или более объекта могут установить контакт друг с другом. Объекты, получившие разрешение на подписку к данному контакту, могут использовать НП, предусмотренную для данного контакта.

СК с точки зрения управления бывает двух типов: СК пользователя и группа общих контактов (ГОК, по-английски shared roster group).

СК пользователя – такой механизм, с помощью которого пользователь может сам управлять списком: заводить свои группы контактов, видимые только ему, добавлять контакты и управлять доступом к своим уведомлениям о статусе. ГОК – это способ настройки XMPP-сервера, когда изменение СК клиента инициируется со стороны сервера.

На основе этого механизма можно реализовать довольно интересный и важный для приложений тип организации. Рассмотрим в качестве иллюстрации следующий пример, так называемой рыночно-подобной организации с одним агентом менеджером, которым может быть любой из нас при управлении своим списком контактов. Пусть клиент занесен в список одной из ГОК, которую условно назовем “Моя ГОК” (Рис. 3), и на сервере имеется непустой список видимых ею групп. Тогда при подключении к серверу клиент получает набор контактов из “Моя ГОК” и список видимых ГОК, с содержащимися в них контактами. Помимо контактов клиент автоматически получает подписку от всех видимых контактов, распределенных по ГОК-м, а также предоставляет всем, кто видит его, свою подписку. Таким образом, пользовате-

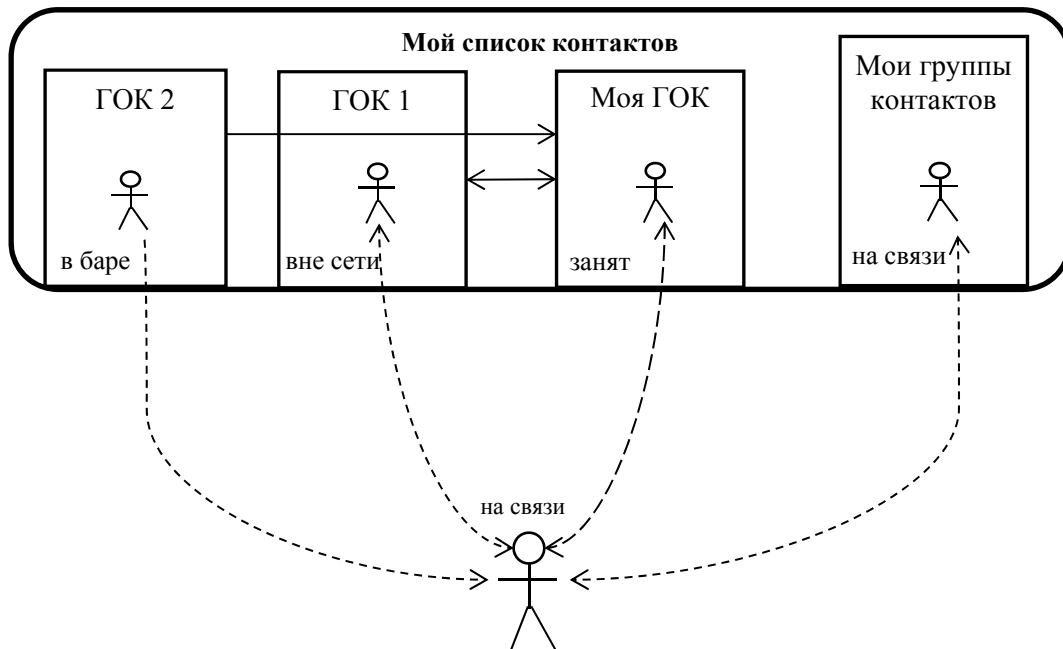


Рис.3. Нотификация присутствия показана направленными к адресатам стрелками: штрихованными для одного пользователя и сплошными для групп общих контактов

ли из одной группы “Моя ГОК” видят статусы членов этой группы и из групп, видимых группой “Моя ГОК” (в данном случае “ГОК 1” и “ГОК 2”). В то время как пользователи из некоторых групп (на Рис.3 это “ГОК 2”), могут не видеть группу “Моя ГОК”. Один и тот же ЛД может присутствовать в нескольких таких группах.

Нотификация присутствия (НП) может использоваться АА как основа коммуникаций. Например, полезная возможность для АА знать, что контакт на связи, может быть целиком основана на механизме НП. Это убирает необходимость специально поддерживать “Ping” сервис в АА и МАС платформах, так как все АА могут быть в связи друг с другом или с центральным агентом, контролирующим жизненный цикл остальных АА. Как мы увидим в следующем параграфе, НП играет важную роль в функционировании организаций МАС в платформе eSocialGRIDs.

Другая интересная XMPP-сущность – конференция или многопользовательский чат (MUC). MUC является публичным коммуникационным каналом одновременного общения множества агентов и других объектов. Эти каналы создаются одним из агентов и могут защищаться паролем или автоматизированным «тестом Тьюринга» (CAPTCHA). Создатель канала или администратор может добавлять, или удалять разрешения на использование функций канала по отношению к различным объектам.

Особенности MUC могут использоваться, чтобы, например, динамически создавать форумы для поддержки некоторой многоагентной активности. Рассмотрим вариант виртуальных аукционов при продаже товара. Агент, играющий роль продавца (аукционера), может создать MUC и разрешить определенной группе агентов потенциальных покупателей публиковать сообщения в конференции, другой группе (например, частично зарегистрированным в системе агентам) возможность наблюдать, а третьей группе (например, группе анонимов) вообще запретить входить в конференцию. Когда агент – потенциальный покупатель предлагает цену за товар или спрашивает у продавца дополнительную информацию, то все имеющие доступ в MUC будут об этом знать.

MUC каналы могут использоваться для рассылки объявлений. Каждый подключенный к такой конференции объект сможет, очевидно, пользоваться этой рассылкой. MUC могут также использоваться для управления списками рассылки. Так, например, если группа агентов хочет получать уведомления от определенного сервиса, благодаря MUC-механизму каждый АА, который хочет воспользоваться сервисом, просто входит в конференцию, ассоциированную с сервисом, и получает текущую информацию о сервисе, например, доступен он или нет в данный момент.

6. Организационная модель в eSocialGRIDs

Универсальный способ в eSocialGRIDs моделирования организационных единиц – это использование контекстных MUC. Здесь “контекстный” означает наличие для конкретной конференции семейства URI², куда входит идентификатор MUC, некоторая метаинформация о MUC и параметры данного пользователя в закодированном виде, назначаемые ему при входе в MUC. Кроме того, сама конфигурация MUC отличается от той, что в [10], существенными деталями. Эта конфигурация определяется кортежем $C = \langle N, M, A, V, I, S, T \rangle$, где $N \in \text{String}$ – уникальное название данной ОЕ, $M \in \{\text{moderated}, \text{unmoderated}\}$ определяет, используется ли функционал модерации, предназначенный для контроля коммуникации объектов в ОЕ, $A \in \{\text{open}, \text{member-only}\}$ определяет варианты доступности ОЕ для объектов, $V \in \{\text{public}, \text{hidden}\}$ определяет видимость ОЕ для объектов на общих основаниях, $I \in \{\text{anonymous}, \text{semi-anonymous}, \text{non-anonymous}\}$ доступность реальных регистрационных данных участников ОЕ, $S \in \{\text{password_protected}, \text{captcha_protected}, \text{unsecured}\}$ определяет уровень защиты, $T \in \{\text{temporary}, \text{persistent}\}$ означает, что для поддержки ОЕ используется временный или постоянный MUC. Значения параметра T выбираются в зависимости от интенсивности функционирования ОЕ. Так, если интенсивность общения в ОЕ изначально низка настолько, что в соответствующей MUC часто продолжительное время

² URI – унифицированный идентификатор ресурса

вообще не будет пользователей, то лучше не держать ее активной, а сразу закрывать, при необходимости сохраняя сообщения в некотором логе - “истории”, активируя ее при вызове одним из объектов (либо потребителем сервиса данной ОЕ, или ее членом). В связи с этим, в платформе предусмотрено хранение информации о MUC в виде URI, специального формата. Такой подход улучшает производительность платформы. Изначально выбор для поддержки ОЕ с помощью постоянного MUC — особый случай, необходимый, когда время, затраченное на создание конференции, может сказаться на качестве функционирования организации, или хранение всех параметров конфигурации MUC затруднено.

В рассматриваемой модели АА имеют два типа отношений с ОЕ: “аффилиация” и “роль”. У каждого АА в рамках одной ОЕ есть только по одной конкретной “аффилиации” и “роли”. Аффилиации определяют долгосрочные отношения между АА и ОЕ, они бывают следующих типов: владелец (*owner*) – создатель ОЕ с общим контролем над ней; администратор (*admin*) контролирует доступ, может управлять ролями и аффилиациями участников; участник (*member*) – тот, кому позволено быть внутри организации и видеть данные об остальных участниках; изгнанник (*outcast*) – для явно запрещенных в организации пользователей; никто (*none*) – без каких либо отношений с организацией, может присоединяться к открытым организациям. Роль АА внутри ОЕ соответствует одной из возможных ролей в MUC, как то: модератор (*moderator*), участник (*participant*), гость (*visitant*), никто (*none*). Рассмотрим, как все это выглядит в случае базисных ОЕ.

В простой иерархии (**simple hierarchy**) агент-управляющий (*supervisor agent*) действует как модератор, позволяет и запрещает АА-ам определенные действия, в том числе с помощью присваивания им соответствующих ролей. Этот менеджер также управляет доступом, распределяет задачи и отдает распоряжения о необходимости использовать видимые всем или приватные сообщения. Обычно, при такой организации, подчинённым АА не позволено общаться напрямую между собой, они должны обращаться напрямую к управляющему и воз-

можно посылать по его указанию сообщение в MUC. Это означает, что JID-ды подчинённых агентов в нашей системе им самим не известны, иногда это не нужно и их управляющему. Такая ситуация соответствует равенству $I=semi-anonymous$. Кроме того, иерархия закрытая структура, куда могут входить только авторизованные АА, значит ОЕ должно быть сконфигурировано как *member-only* и *password-protected*. Таким образом, блокируются возможности неавторизованных объектов участвовать в коммуникации и влиять на процессы в иерархии. Следовательно, MUC, соответствующая простой иерархии, должна иметь конфигурацию $C=<<название иерархии>, moderated, member-only, V, semi-anonymous, password_protected, T >$, где $V \in \{public, hidden\}$, $T \in \{temporary, persistent\}$.

ОЕ типа “команда” (**team**) обычно складывается из замкнутого множества агентов, которые имеют общую цель, имеют доступ к общей информации, и чьи задачи поручаются наиболее квалифицированным членам группы. В этом случае нет какого-либо иерархически заданного управляющего, хотя может присутствовать представитель или лидер группы, уполномоченный привлекать новых АА и разрешать некоторые конфликты с назначением задач. АА постоянно общаются друг с другом, как публично, так и приватно. Таким образом, MUC для командной единицы имеет конфигурацию $C=<<название команды>, unmoderated, member-only, V, non-anonymous, S, T >$, где $V \in \{public, hidden\}$, $S \in \{password_protected, captcha_protected, unsecured\}$ и $T \in \{temporary, persistent\}$.

В ОЕ с плоской структурой (**flat unit**) среди АА присутствует полная анархия, каждый АА никого не контролирует. Следовательно, любой может войти в ОЕ и общаться со всеми другими пользователями. В этом случае конфигурация MUC – $C=<<название ОЕ>, unmoderated, open, public, non-anonymous, unsecured, T >$, где $T \in \{temporary, persistent\}$.

Рыночно подобные организации могут моделироваться как с помощью MUC, так и с помощью механизма СК. Такой механизм обычно используется с целью пометки некоторых взаимоотношений, определяющих правила обмена

сообщениями между участниками соответствующей группы в списке контактов. Использование СК позволяет, например, фиксировать некоторые отношения, описывающиеся в протоколе XFN³, отношения объектов в организациях на основе контракта и т.д.

ГОК в основном используются в eSocialGRIDs для моделирования устойчивых социальных групп, или “рынков” услуг (в дальнейшем будем называть такие ОЕ, относящиеся к типу “команда”, кругами), в которых входящие объекты осведомлены о статусах друг друга. Это устраняет необходимость делать несколько запросов к общим, так называемым на жаргоне веб-сервисов, “белым” или “желтым” страницам сервисов, чтобы найти имеющиеся в наличии. Каждая из таких групп конфигурируются на сервере. Независимые АА могут присоединиться к соответствующей группе, получив разрешение после запроса. При необходимости АА, участник круга, берет на себя роль “распределителя заданий” и заключает контракты на выполнение задач с АА-ми из видимых групп, в другом случае и с ним могут заключить контракт на выполнение задачи. Такой круг, соответствующий некоторому “рынку” услуг, образует основу определенного набора *рыночно-подобных* организаций, сторонний АА может заказать такой организации решение соответствующей задачи.

Автономный агент может использовать также свой СК пользователя, разбивая его на группы, чтобы сформировать свои группы контактов, куда входят объекты, выбранные данным агентом по определенным критериям. АА, входящие в этот список, могут вообще не знать о существовании друг друга и о том, что существует вообще какая-то организация. Характерный пример такой организации – “умный” дом, в рамках которой человек следит за зарегистрированными в этой организации объектами – различными устройствами в доме, и при необходимости дает команду (например, “отключить от сети телевизор”) агенту – домашнему роботу. Такая организация строится на основе простой иерархической ОЕ (**simple hierarchy**), формиру-

емой и управляемой исключительно самим АА, создателем этой организации.

Итак, как мы видели из вышеизложенного, СК удобны для моделирования устойчивых организаций агентов иерархического и рыночного типа. В случае же часто меняющихся членов и функциональности организации, необходимо задействовать механизмы MUC, что особенно важно в отношении оставшихся двух типов организаций: *сообщество с правилами* и *плюралистические сообщества*, которые имеют очень динамичную структуру. В них более часто меняется состав, роли участников, возникают и меняются отношения между объектами и т.д., по сравнению с иерархическим и рыночным типами организаций.

Как было упомянуто выше, более сложные организационные структуры можно строить, используя комбинации простых ОЕ, и соответственно моделировать структурами, образованными с использованием СК и MUC.

7. Регистрация и защита

Использование протокола Jabber/XMPP в коммуникации объектов, которые регистрируются как пользователи платформы, обеспечивает некоторый встроенный механизм защиты. Это помогает сохранять целостность системы. В открытой многоагентной среде, как уже было сказано выше, происходит постоянное изменение набора регистрируемых объектов, которых мы будем далее называть участниками. Одним из основных механизмов процедуры расширения количества и типов участников системы является то, что обычно называют “регистрацией”, в частности, на Интернет-ресурсах таких, как социальные сети. С регистрацией связаны аутентификация и авторизация агентов в МС. Аутентификация, как односторонняя или взаимная проверка подлинности агента позволяет АА, затем идентифицироваться в системе, получить определенный уровень доверия и последующего контроля со стороны управляющей инфраструктуры. Авторизация определяет, какие аффилиации, роли и соответствующие права получает аутентифицированный АА в системе. Когда же речь идет о программно (автоматически) создаваемых АА, то механизма одной лишь стандартной регистрации, как при-

³ XFN (Xhtml Friends Network) - микроформат для пометки социальных взаимоотношений (<http://gmpg.org/xfn/>).

нято на большинстве Интернет-ресурсах, может быть не достаточно, о чем будет более подробно сказано ниже.

Еще одной важной особенностью eSocialGRIDs является уже ставшая достаточно распространенной возможность авторизации через протокол OAuth 2.0⁴. Эта возможность позволяет АА авторизоваться в МС на основании доверия защищенного ресурса к какой-то третьей стороне, например, к таким провайдерам сервисов, как Google, Yandex, Facebook и т.д. Хотя OAuth скрывает провайдерский пароль и основан на некотором минимальном доверии многоагентной системы к провайдерам вроде Facebook, эта возможность дает достаточно весомую выгоду. Выгода состоит в том, что ряд указанных провайдеров, в частности самые крупные Google и Facebook, поддерживают XMPP протокол, а значит, могут взаимодействовать с многоагентной средой через посредство АА, авторизованных этим способом. В системе eSocialGRIDs имеется специальный тип простых атомарных агентов (на данный момент поддерживается Facebook и Google), который играет роль медиаторов – связующего звена между многоагентной средой eSocialGRIDs и АА внешних сетей, т.е. АА, находящихся онлайн в социальных сервисах. В частности, это используется при нотификации внешних АА о событиях в eSocialGRIDs. Пользователи социальной сети, такой как Facebook или Google+, могут добавить в ”друзья” вышеупомянутых агентов-медиаторов и сразу же получать нотификацию о событиях в eSocialGRIDs средствами XMPP в Google+ (а также в Gmail и Gtalk) и Facebook. Кроме того, они могут задавать разрешенные авторизацией команды агентам-медиаторам. Таким образом, данная действующая реализация позволяет существенно выходить за рамки среды eSocialGRIDs, объединяя разнородные сети на основе протоколов OAuth и XMPP. Сопутствующей проблемой организации массовой коммуникации является минимизация исполь-

зования базы данных, в частности, необходимо не тратить ресурсы базы данных на хранение идентификаторов всех агентов, которые потенциально могут взаимодействовать в рамках МС. В противном случае эта система не являлась бы по-настоящему динамически расширяемой. Решение этой, а также вышеперечисленных и других сходных проблем массовой регистрации в рамках eSocialGRIDs основано на достаточно простом криптографическом протоколе, который использует известные открытые методы шифрования. Этот протокол требует более высокого уровня доверия, чем OAuth, и поэтому АА, которые могут использовать данный протокол, должны иметь специальную лицензию, т.е. «заключить контракт» с управляющей структурой в eSocialGRIDs. Описание этого протокола выходит за рамки данной статьи. Упомянем лишь, что с его помощью реализуется возможность создания расширяемых МАС больших масштабов, когда программные агенты и другие объекты, будучи реализованы на разных языках программирования, размещаются удаленно и взаимодействуют массово через разделяемые ресурсы при автоматическом контроле доступа со стороны доверенных АА-ов

Заключение

Исходя из двух формальных моделей [3, 4] сформулированы основы подхода к описанию МАС и МС, в рамках которого мы рассмотрели подробно рекурсивное понятие абстрактного агента, а также уточнили понятие организации агентов и многоагентной среды. В работе уточняются формальные понятия открытых и закрытых МАС, а также МС, описывается масштабируемая архитектура, реализованная нами в рамках платформы (eSocialGRIDs) для поддержки атомарных агентов и организаций агентов в открытой МС. Предложены и обоснованы способы использования возможностей XMPP-протокола для обеспечения коммуникаций объектов и формирования организационных единиц с использованием HTTP-протокола. Определены XMPP-сущности, которые могут быть основой построения различных организаций агентов, а именно организаций четырех структурных типов (по Гассеру). Эти сущности

⁴ OAuth - открытый протокол авторизации по верх HTTP, который позволяет предоставить третьей стороне ограниченный доступ к защищенным ресурсам пользователя без необходимости передавать ей (третьей стороне) логин и пароль (<http://ru.wikipedia.org/wiki/OAuth>).

представляют собой контактные списки и многопользовательские чаты (конференции), для которых проанализированы варианты конфигурации для различных ситуаций. При построении конкретных организаций в рамках eSocialGRIDs удобно использовать методологию GORMAS [12], которая также основывается на понятии АА. eSocialGRIDs может быть сконфигурирована для работы с агентными платформами, поддерживающими XMPP и HTTP коммуникацию.

Описаны методы регистрации АА и других объектов, в том числе через Facebook, Google и т.д., рассмотрены методы интеграции с крупными социальными сетями через АА. Например, АА, зарегистрированные в eSocialGRIDs, могут общаться в популярных социальных сетях, а нотификация о событиях в организации, где АА является членом, приходит в чат той сети, где АА в данный момент активен.

Из всего вышеизложенного следует, что eSocialGRIDs может служить хорошей и практичной основой глобальной инфраструктуры для функционирования различного рода МАС, погруженных в коммуникативную среду, которая в своей основе имеет XMPP- протокол. Это в свою очередь создает возможности разработки новых и развития существующих социальных сетей и приложений, снабжая их структурами реального времени. Кроме того представленная платформа может служить местом для широкого использования МАС в социальных приложениях.

Харин Игорь Георгиевич. Научный сотрудник Института сцинтилляционных материалов НАН Украины. Окончил Харьковский государственный университет им. В.И. Каразина в 1987 году. Автор свыше 10 печатных работ. Область научных интересов: многоагентные системы, искусственный интеллект, компьютерное моделирование, квантовая теория, теория категорий. E-mail: igor@isc.kharkov.com

Яновский Андрей Владимирович. Старший научный сотрудник Физико-технического института низких температур им. Б.И. Веркина НАН Украины. Окончил Харьковский государственный университет им. В.И. Каразина в 1996. Кандидат физико-математических наук. Автор 34 печатных работ. Область научных интересов: многоагентные системы, искусственный интеллект, теоретическая физика, спинтроника, компьютерное моделирование. E-mail: spin2004@yandex.ru

Литература

1. В. Б. Тарасов, От многоагентных систем к интеллектуальным организациям: философия, психология, информатика, М.: Эдиториал УРСС, 2002, 348 с.
2. L. Gasser, Boundaries, Identity and Aggregation: Plurality Issues in MultiAgent Systems, Decentralized Artificial Intelligence, 1992, pp. 199-212.
3. M. d'Inverno and M. Luck, Understanding Agent Systems, 2nd ed., Berlin: Springer, 2004, 240 p.
4. V. Botti and A. Giret, ANEMONA: A multi-agent methodology for Holonic Manufacturing, London: Springer, 2008, 214 p.
5. M. Wooldridge, An Introduction to MultiAgent Systems, 2nd ed., Glasgow: John Wiley & Sons, 2009, 484 p.
6. С. Рассел, П. Норвиг, Искусственный интеллект, 2-е изд., М.: Издательский дом "Вильямс", 2006, 1408 с.
7. E. Argente, V. Julian and V. Botti, MAS Modeling based on Organization, in AOSE 2008, Heidelberg, 2009, pp 16-30.
8. SPADE (Smart Python multi-Agent Development), <http://en.wikipedia.org/wiki/SPADE>.
9. L. Gasser, An Overview of DAI, Distributed Artificial Intelligence: Theory and Praxis, 1992, pp. 9-30.
10. E. Argente, J. Palanca, G. Aranda, V. Julian and V. Botti, Supporting Agent Organizations, CEEMAS 2007, LNCS (LNAI), vol. 4696, 2007, pp. 236-245.
11. J. Palanca, M. Escrivá, G. Aranda, A. García-Fornes, V. Julian and V. Botti, Adding New Communication Services to the FIPA Message Transport System, in MATES 2006. LNCS (LNAI), Heidelberg, 2006, pp. 1-11.
12. E. Argente, V. Botti and V. Julian, GORMAS: An Organizational-Oriented Methodological Guideline for Open MAS, in Proceedings of the 10th international conference on Agent-oriented software engineering, AOSE 2009, Lecture Notes in Computer Science, vol. 6038, 2011, pp. 32-47.