

# Повсеместные вычисления на основе открытой многоагентной среды: проект eSocialGRIDs

И. Г. Харин

*Институт Сцинтилляционных Материалов НАН Украины, 60 проспект Ленина, Харьков 61001, Украина*

igor@isc.kharkov.com

**Аннотация.** *Рассмотрена проблема создания глобальной программной инфраструктуры для поддержки распределенных приложений в рамках парадигмы повсеместных вычислений. Последовательно описана концепция и методы построения такой инфраструктуры на основе платформы для открытой многоагентной среды. Особенностью предлагаемого решения является ориентация на технологию построения многоагентных систем, основывающуюся на контекстных организациях агентов. С учетом динамического характера инфраструктуры для повсеместных вычислений рассмотрен комплекс основных проблем организаций агентов в открытой многоагентной среде (отношения и коммуникацию между агентами, структуру, задачи и поведение многоагентных систем). Показано, что протокол XMPP позволяет естественным образом обеспечить динамическую природу организаций агентов и динамический характер повсеместных коммуникаций в распределенной открытой вычислительной среде. Как результат этой работы разработаны архитектура и основные компоненты eSocialGRIDs платформы для поддержки повсеместных вычислений в социальной среде, которая является объединением различных сообществ людей и программных агентов.*

## Ключевые слова

Повсеместные вычисления, коммуникации, агент, многоагентная система, организация агентов, архитектура платформы, открытая многоагентная среда, социальные сети, мгновенные сообщения.

## 1 Введение

Современные вычислительные системы достигли уже достаточно высокого уровня и, в основном, успешно справляются с решением достаточно сложных задач управления и получения информации, однако их очевидный недостаток состоит в том, что они "живут" и используются независимо друг от друга, в результате многие функции многократно дублируются и вычислительные ресурсы используются неэффективно. Уже сейчас в своей повседневной жизни человек использует все больше и больше компьютеров.

Эти компьютеры достаточно разнообразны, они обладают различными ресурсами памяти и процессоров, работают в различных операционных средах, используют различные физические каналы связи, соединяются с различными источниками данных по различным протоколам, предоставляют пользователю различные сервисы.

Но в большей или меньшей степени они работают изолированно, например, компьютеры, входящие в GRID-систему, работают согласованно в GRID-облаке, но могут использоваться только определенным, принятым только для GRID образом и только пользователями, вошедшими в соответствующую GRID-систему, сервисы предоставляемые GRID технологией не могут взаимодействовать с сервисами предоставляемыми другими системами.

Пользователь может работать с такими системами только поочередно, что его не устраивает. Потребность интеграции разнообразных компьютерных, информационных и коммуникационных ресурсов в единую среду в настоящее время является одним из главных факторов, определяющих современные тенденции развития в области компьютерных технологий.

Например, в современной медицине существуют десятки компьютерных и информационных систем, решающих задачи диагностики, систем контроля и обработки данных, и т.д. Врачу доступны мобильные

сервисы, например, доступ в сервисную службу, когда ему необходимо контролировать обработку данных, поставить диагноз, проконтролировать прием препаратов и т.д.

Врач чувствовал бы себя гораздо комфортнее, если бы все эти системы были интегрированы в единую среду. Тогда при возникновении проблемы с пациентом автоматически передавалась бы соответствующая информация в мобильное устройство врача, и предлагался сценарий решения.

После принятия предварительных решений врачом (возможно вместе с коллегами) с помощью соответствующих сервисов, он мог бы вместе с коллегами пообщаться с пациентом в реальном времени и предписать ему ряд действий для лечения, возможно при помощи находящихся по близости врачей. Для описания на языке вычислительных процессов комплекса возникающих проблем наиболее перспективны подходы, основывающиеся на концепции многоагентных систем (МАС) открытого типа.

В связи с вышесказанным последнее время особенно активно исследуются МАС смешанного состава, т.е. когда составляющими могут быть роботы, люди или команды людей. Такие системы, как хорошо известно, называются виртуальными организациями [1]. Сюда, например, можно отнести так называемые Grid-системы. Кроме того, последние годы в области вычислительной технологии и коммуникаций (WWW, e-commerce, мобильных приложений и т.д.) возникает технологически новая парадигма прикладных вычислений: "вычисление как взаимодействие".

Один из наиболее известных примеров использования такой точки зрения на вычисления - это 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, проектирования, разработки и управления вычислительными системами. Соответственно, технология МАС, особенно открытых МАС, имеет нужные характеристики и потенциал для реализации парадигмы "вычисление как взаимодействие", как в виде формальных моделей, так и программного обеспечения.

Динамические организации агентов, которые самонастраиваются с целью получения выгоды в данном контексте, становятся все более важными. 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 Gasser [2], почти все предложенные агентные архитектуры имеют общую проблему: каким образом обходится с наборами взаимодействующих агентов на высшем уровне абстракции – т.е., как обходится с организациями, так же как, с агентами.

Для этого в области программной инженерии необходимо решить несколько трудных проблем, связанных с развитием концепции агентов и разработки соответствующих платформ, адаптируя рекурсивное определение агентов с учетом возможности динамического создания агентов (организаций агентов) самими агентами.

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 сред.

Главная идея этой статьи обосновать концепцию и принципы функционирования универсальной открытой многоагентной среды, в рамках которой смогут существовать и взаимодействовать различные МАС построенные на основе разных многоагентных платформ. На основе 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 с использованием HTTP-протокола организаций на основе одной или более организационных единиц, которые в свою очередь строятся с помощью соответствующих XMPP- сущностей. Рассматриваются методы регистрации 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 и агентом-заказчиком. В рамках этой архитектуры разработаны базовые компоненты платформы eSocialGRIDs (electronic social grids). Рассматривается основная функциональность платформы и методы ее использования для моделирования организаций.

## 2 Повсеместные вычисления

Повсеместные вычисления — это парадигма организации вычислений, которая предполагает повсеместное использование компьютеров в форме, незаметной для пользователя [3]. В рамках многоагентного подхода такое поведение вычислительной системы естественным образом можно реализовать через контекстно организуемого агента, с которым пользователь взаимодействует во время обращения к устройствам, обеспечивающим интерфейс с вычислительной системой.

Повсеместные вычисления предполагают дальнейшее развитие персонализации в использовании вычислительных систем. При развитии персонализации агентные технологии будут все более востребованы массовым потребителем: для сбора информации, распространения своей информации, контроля, анализа “на лету”, различного вывода информации и т.д.

Для того, чтобы поддерживать функционирование множества гетерогенных устройств (компьютеров, карманных компьютеров, сотовых телефонов, встроенных систем и др.) и установленных на них программ, работающих под управлением различных операционных систем необходима система коммуникаций, основанная на едином протоколе передачи данных.

Таким образом, создание инфраструктуры повсеместных вычислений подразумевает и среду повсеместных коммуникаций. В этой среде могут совместно использоваться различные коммуникационные протоколы (Ethernet, WiFi, Bluetooth, и др.), по которым связываются различные устройства сети, например, настольные и карманные компьютеры, сотовые телефоны, устройства бытовой техники, сенсоры и т.п. Эта среда должна организоваться таким образом, что взаимодействующие сущности (объекты) “не замечали” этих особенностей.

Кроме того объекты для своевременной реакции на события должны иметь возможность коммуникаций в реальном времени. При этом коммуникации реального времени нуждаются в достаточной гибкости настроек под конкретный объект и персонализированной настройке для людей. Это предполагает также, что коммуникации должны быть привязаны к контексту (иначе не понятно как их инициировать и как управлять).

Таким контекстом является, например, статья со ссылкой на многопользовательский чат (конференцию) для обсуждения статьи. Пользователю необходима поддержка экспертов, агентов или других пользователей в рамках соответствующего сообщества, которое сформировано по общности рода деятельности его членов. Таким образом, нужна глобальная всепроникающая среда “мгновенных” сообщений для эффективного применения многоагентных систем (да и вообще их функционирования) для массового потребителя.

В рамках нашего подхода для формирования среды повсеместных коммуникаций предлагается снабжать соответствующие объекты функциями приема и отправки сообщений на основе XMPP-протокола. При отсутствии возможности снабдить такими функциями конкретный объект, предлагается подключать к нему промежуточные компоненты для конвертации сообщений в формате для данного объекта в XMPP-формат и взаимодействия с соответствующими сервисами, образующими ядро коммуникационной среды. XMPP – это основанный на XML, открытый, свободный для использования протокол для мгновенного обмена сообщениями и информацией о присутствии в режиме, близком к режиму реального времени.

Изначально спроектированный, легко расширяемый, протокол, помимо передачи текстовых сообщений, поддерживает передачу голоса, видео и файлов по сети. С точки зрения клиентской части, выбор XMPP протокола особенно важен в случае мобильных устройств. XMPP позволяет держать TCP соединение открытым и получать сообщения асинхронно. При этом клиент не расходует сотовый трафик на ожидание сообщений в отличие от, например, HTTP протокола, а обмен сообщениями гораздо удобнее и более гибок в плане организации конференций, безопаснее и дешевле SMS.

Многоагентная технология сегодня предлагает наиболее перспективные модели и технологические средства для реализации крупномасштабных приложений, состоящих из распределенных автономных объектов, которые взаимодействуют для решения собственных и общих задач приложения.

Для многоагентных систем разработана стандартная абстрактная архитектура [4], которая поддерживает семантический поиск сервисов, а также агентов, которые владеют нужным сервисом, что очень важно в открытых сетях.

В данной работе предлагается концепция использования MAS, интегрированная с концепцией архитектуры для открытой MS, ориентированной на поддержку организаций агентов, что открывает новые возможности для реализации систем, работающих в среде повсеместных вычислений. Но для практической реализации такого подхода необходимо создание стандартной программной инфраструктуры. Последующий материал данной работы посвящен описанию формальной модели и разработанной на ее основе платформы для MS, а также описанию возможностей поддержки организаций агентов.

## 3 Формальные рамки описания MAS

Очертим формальные рамки используемого подхода описания MAS и уточним понятие распределенных масштабируемых многоагентных сред. Будем называть “сущностями” на высшем уровне абстракции, подобно [5], все составные части, рассматриваемых систем, т.е. всё, что можно представить фреймом в некоторой

формальной модели таких систем. Система в данный момент времени – это один из вариантов набора сущностей из заданного универсума. Обратим внимание, что связи между частями и акты взаимодействия тоже можно представлять сущностями при таком подходе.

Хорошо известно, что в любой системе можно выделить две основные части: это способные к активности сущности (объекты) [5], которые способны 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 есть объекты, 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 [5], что простой объект отличается от агента отсутствием целей и сопутствующей функциональности для выстраивания своей активности при достижении этих целей. К тому же объект не может воспринимать среду, т.е. он взаимодействует с ней, но эта способность не обусловлена внутренней целью или мотивацией.

Итак, в этом рассмотрении (как и во многих других [1,5-8]) мы считаем, агентом сущность, которая интерпретирует состояние среды и действует на основании этой интерпретации сообразно определенному представлению о системе, частью которого он является, сообразуясь со своими целями.

При несколько более слабых условиях, чем в [5] 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).

Пункт 2 из перечисленных, можно обозначить словом “видеть агента”, т.е. один агент видит другого, как нечто обладающее целью, а не как простую сущность или объект.

С одной стороны при определенных условиях МАС может рассматриваться, как единая сущность – агент. С другой стороны, формально любую связанную совокупность агентов, можно рассматривать, как сущность с некоторым интегральным представлением о мире и обобщенной целью, производными от представлений и целей отдельных агентов соответственно.

В связи с этим, есть смысл 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 [6], 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 [6] структурном подходе из [5]:

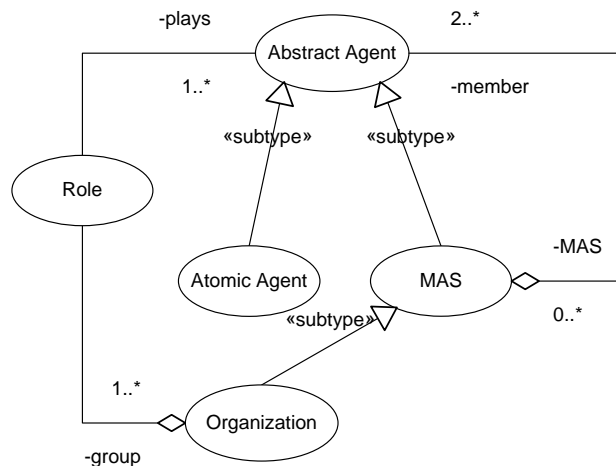
**Определение 3.1.** АА – это атомарная сущность или система сущностей, которая рассматривается как целое (в смысле представимости некоторым фреймом) в виде объекта, воспринимающего свое окружение. В зависимости от восприятия этот объект интерпретирует ситуацию и выполняет действия. Эти действия АА зависят 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 3.2.** МАС - это как минимум совокупность двух или более АА, которые взаимодействуют, чтобы решить проблемы и достичь целей выше индивидуальных способностей и знаний каждого АА.

Эти определения связаны между собой, выражая понятия АА и МАС друг через друга, и в основе имеют понятие атомарного агента, которое совпадает с традиционными определениями агента [1, 5-8]. Ниже для краткости мы будем иногда использовать термин агент в смысле упомянутого выше понятия АА. 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.

Далее, чтобы избежать путаницы в определениях и утверждениях, будем всегда явно использовать термин АА в соответствующих случаях. Понятие абстрактный агент находится на более высоком уровне абстракции, чем атомарный агент или МАС. В тоже время термин АА существует только на стадии формального рассмотрения, при моделировании реальными сущностями, он должен быть заменен, либо атомарным агентом, либо МАС.



**Рис.1.** Абстрактный Агент. На данном уровне абстракции, показанные на этой диаграмме, сущности соответствуют фреймам, которые обозначены овалами

Определение 1.2 расширяет традиционное представление о MAS, учитывая, что MAS может состоять из меньших MAS. Это позволяет описывать системы, в которых строительными блоками являются взаимодействующие MAS, сотрудничающие для достижения глобальных целей (здесь глобальная цель – это цель всей системы как целого). Таким образом, рекурсивная структура AA может быть представлена в виде UML-диаграммы (см. рис.1). Агентная организация является частным случаем MAS и понимается, как социальная сущность, составленная из определённого числа AA, которые обслуживают несколько отдельных задач. Агентная организация структурирована, следуя определенной топологии и коммуникативным отношениям для того, чтобы достигать главных целей организации [9]. Организации агентов предполагают существование глобальных целей вне любых индивидуальных AA, и они существуют независимо от отдельных AA.

Во многих работах среду включают [1] по определению в состав MAS, в других не включают [5]. Из вышеприведенного определения можно сделать вывод, что это вопрос удобства, зависящего от рассматриваемой предметной области, так как искусственную систему, которая содержит набор связанных AA, иногда удобно в целом саму рассматривать как MAS. Заметим, что рекурсивному определению AA это никак не противоречит, что говорит об успешном разрешении в рамках подобного подхода некоторой неоднозначности понятий MAS и MC. В то же время, если, например, MAS – 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 MAS. И вообще часто удобно рассматривать MAS и конкретную среду отдельно. Соответственно, при таком подходе, получается, что конкретная MAS привязана скорее к типу среды, а не к конкретному экземпляру среды. Поэтому когда мы будем анализировать поведение данного AA в среде, предполагая, что он должен учитывать цели других AA, мы будем называть такую среду MC. При этом в каждом конкретном случае, анализируя систему, мы будем явно определять те сущности, которые составляют определенную MAS, а остальное будем называть MC. Учитывая вышесказанное, можем дать следующее определение.

**Определение 3.3. Многоагентная среда (MC)** – это такая среда, где созданы условия для жизнедеятельности, по крайней мере, двух AA, и поведение хотя бы одного AA при достижении им своих целей зависит от целей другого.

Существует множество практических реализаций вычислительных MC. Одна из перспективных платформ для MC – eSocialGRIDs, описываемая в данной работе. К ней наиболее близка по принципам построения одна из известных агентных платформ SPADE [10]. eSocialGRIDs и SPADE, как и большинство агентных систем, используют спецификации FIPA<sup>1</sup>.

## 4 Открытые MAS и MC

Интуитивно понятно, что MAS, как и MC могут быть открытыми и закрытыми. При моделировании проблемных сред всегда стоит вопрос, какую архитектуру, открытую или закрытую выбрать. Замкнутую

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

систему обычно легче построить, поддерживать, однозначно интерпретировать результаты и использовать. С другой стороны открытая архитектура более гибкая 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 и архитектур. Типичным представителем такой открытой вычислительной системы является Интернет.

В подходе, представленном в данной статье, как было упомянуто выше, архитектура всей моделируемой (или проектируемой системы) складывается из архитектур МАС и МС. Уточним вначале понятие открытости и закрытости для МАС.

**Определение 4.1. 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.

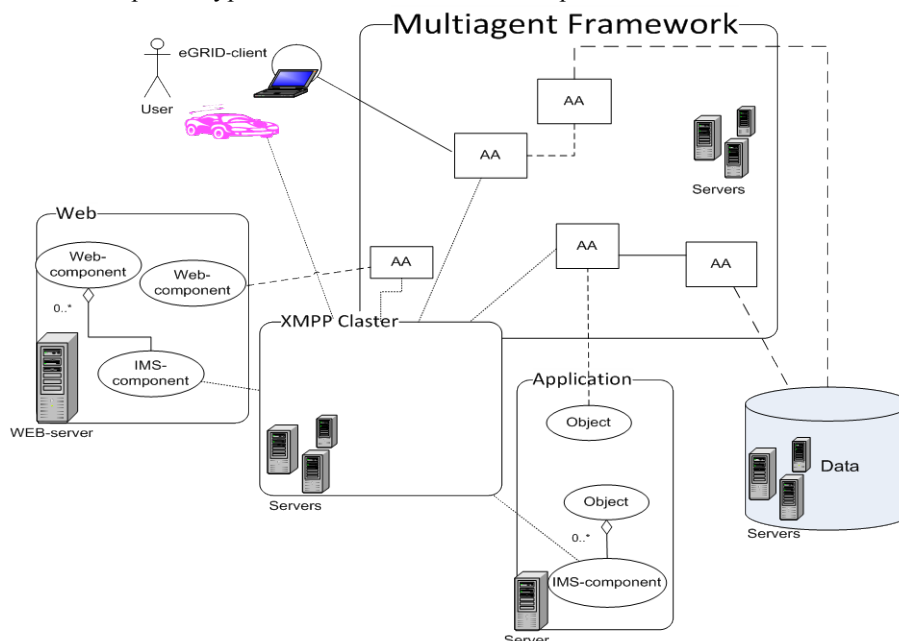
Открытые МАС предназначены, прежде всего, для существования в динамически изменяющейся структуре окружения, где новые компоненты могут быть интегрированы в систему, или могут быть удалены из нее, а также условия функционирования компонент могут непредсказуемо измениться. Интуитивно понятно, что функционировать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

**Определение 4.2. Открытая многоагентная среда** – это пространство, 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. В противоположность **закрытой МС**, которая характеризуется неизменными условиями функционирования объектов и фиксированными типами сущностей.

Таким образом, в нашем подходе полная закрытость всей системы возникает, когда закрыты все МАС в закрытой МС, в противоположность полностью открытой системе, складывающейся соответственно из открытых систем. Между ними существуют промежуточные варианты открытости, определяющиеся комбинациями открытых и закрытых многоагентных систем с открытой, или закрытой МС.

## 5 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 АА и реализующую эту архитектуру платформу eSocialGRIDs. Общий вид архитектуры eSocialGRIDs показан на рис.2.



**Рис.2.** Архитектура eSocialGRIDs. Пунктирные, штриховые и сплошные линии обозначают коммуникационные, слабые (не постоянные) и сильные связи объектов соответственно.

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 агентами 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 агентами разрешено 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, тем более для выхода за границы облака. Соответственно, сама социальная коммуникация основана на этих двух протоколах. В такой архитектуре любой объект, умеющий посылать или принимать сообщения, может получить свой уникальный идентификатор Jabber ID (JID) посредством AA. AA может, конечно, зарегистрироваться сам.

Основой поддержки виртуальных организаций в этой платформе являются контактные списки, разбитые на группы (группы общих контактов и группы управляемые самим агентом), а также многопользовательские чаты (конференции). В предлагаемой архитектуре AA любого типа, может получить JID, который в данном подходе играет роль идентификатора агента (в общепринятом смысле) в системе для установления соединения с платформой.

Изначально IMS (Instant Messages Service) предназначался для поддержки коммуникаций между людьми. Очевидно, что использование IM (Instant Messages) - технологии для обеспечения коммуникации агентов - это естественный шаг в развитии платформ реализующих MAS. Можно сказать, что поведение AA с JID подобно поведению пользователя, который посылает и принимает сообщения, использует ресурсы и сервисы платформы. Как показано в работе [11], агенты могут обмениваться информацией в соответствии с FIPA. Для этого достаточно встраивать исходное сообщение агента в XMPP-сообщение.

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

## 6 XMPP-сущности для поддержки организаций

Главная цель описываемой в этой работе концепции состоит в обеспечении пользователей эффективной моделью, средствами построения и поддержки организаций агентов. В нашем рассмотрении мы используем довольно практичную классификацию по типам организационных структур Л. Гассера [12]. Согласно этому подходу имеется 4 типа организаций агентов: *централизованные и рыночно-подобные организации, сообщества с правилами поведения, плюралистические сообщества.*

Не будем приводить подробное описание перечисленных типов, это можно посмотреть в [12]. Отметим лишь, что в данной статье мы придерживаемся организационной мета-модели из [9], основанной (как и данное рассмотрение) на понятии AA, в рамках которой воспроизводятся все перечисленные выше типы организаций. В таком подходе используется концепция организационной единицы (OE), которая описывает возможные группы (структурные единицы) членов организации. Эти единицы имеют специфическую внутреннюю структуру; они определяют несколько ролей, определяющих функциональность (предлагаемые и необходимые сервисы) и цели, представляющих ожидания организации для каждой роли; они включают ресурсы и приложения, которые могут быть доступны определенным членам организации; они определяют некоторые нормы контроля поведения членов.

Таким образом, система строится из OE, которые иногда могут рассматриваться как глобальные сущности (действующие в качестве абстрактных агентов), или, как групповые сущности или части организации. Однако надо не забывать, что в рекурсивном аспекте OE может включать другие OE, а значит должны иметь в своей основе некоторые атомарные или базисные организационные структуры. В [9] OE рассматриваются только три базисные структуры, названные "**simple hierarchy**", "**team**" и "**flat unit**", внутри которых отношения членов подразумевают иерархическую, командную и плоскую структуру соответственно. С помощью рекурсивной композиции, начиная с этих единиц, можно строить более сложные и законченные структуры, такие как матрица, федерация, коалиции или сообщества [13].

В XMPP есть два наиболее полезных механизма для поддержки организаций агентов, которые предоставляет IMS, это список контактов (СК, по-английски roster) и многопользовательский чат (по-английски MUC - multi users chat). На основе СК реализуется важная особенность IMS – 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.

Два или более объекта могут установить контакт друг с другом, после определенного процесса переговоров, который может быть автоматизирован. 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 группы 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, добавляет туда контакты и управляет 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” (см. 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, пользователи 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 JID может присутствовать в нескольких таких группах.

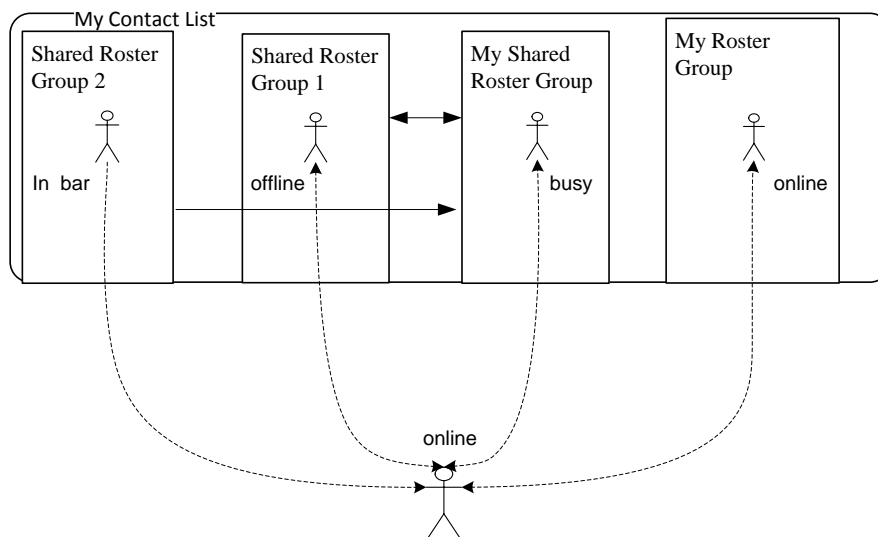


Рис.3. Нотификация присутствия показана направленными к адресатам стрелками: штрихованными для одного пользователя и сплошными для групп общих контактов.

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 агентах 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» (САРТСНА). 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. Например, 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 зарегистрированных в системе агентов) 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 AA хочет получать уведомления от определенного сервиса, благодаря MUC механизму каждый агент, 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.

## 7 Организационная модель в eSocialGRIDs

Универсальный способ в eSocialGRIDs моделирования организационных единиц – это использование контекстных MUC, что означает наличие для конкретной конференции семейства URI<sup>2</sup>, 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 того что в [12] существенными деталями. Эта конфигурация определяется кортежем  $C = \langle N, M, A, V, I, S, T \rangle$ , где  $N \in \text{String}$  – уникальное название данной организации,  $M \in \{\text{moderated}, \text{unmoderated}\}$  определяет, используется ли функционал модерации в данной организации, предназначенный для контроля коммуникации объектов,  $A \in \{\text{open}, \text{member-only}\}$  определяет варианты доступности организации для объектов,  $V \in \{\text{public}, \text{hidden}\}$  определяет видимость организации для объектов на общих основаниях,  $I \in \{\text{anonymous}, \text{semi-anonymous}, \text{non-anonymous}\}$  доступность реальных регистрационных данных участников организации,  $S \in \{\text{password\_protected}, \text{captcha\_protected}, \text{unsecured}\}$  определяет уровень защиты,  $T \in \{\text{temporary}, \text{persistent}\}$  означает, что для поддержки организации используется временный или постоянный MUC.

Значения параметра  $T$  выбираются в зависимости от интенсивности функционирования организации. Так, если интенсивность общения в организации изначально низка настолько, что в MUC часто продолжительное время 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 организации, 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 организации с помощью постоянного MUC в принципе особый случай, необходимый, когда время, затраченное на создание конференции, может сказаться на качестве функционирования организации, или хранение всех параметров конфигурации MUC затруднено.

С одной стороны AA имеют два типа отношений с организациями: “аффилиация” и “роль” в данном MUC. В других MUC эти AA могут иметь другие аффилиации и роли. Аффилиация определяет долгосрочные отношения между AA и организацией, они бывают следующих типов: владелец (*owner*) – создатель организации 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 AA внутри организации определяет внутри MUC его текущее отношение с конкретной OE, соответствующее одной из возможных ролей в MUC, как то: модератор (*moderator*), участник (*participant*), гость (*visitant*), никто (*none*). Рассмотрим, как все это выглядит в случае базисных OE.

В простой иерархии (**simple hierarchy**) агент-управляющий (*supervisor agent*) действует как модератор, позволяет и запрещает агентам 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 сообщения или приватные сообщения. Обычно, при такой организации, подчинённым AA не позволено общаться напрямую между собой, они должны обращаться напрямую к управляющему и возможно 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  $I = \text{semi-anonymous}$ . Кроме того, иерархия закрытая структура, куда могут входить только авторизованные AA, значит OE должно быть сконфигурировано как *member-only* и *password-protected*. Таким образом, блокируются возможности неавторизованных объектов участвовать в

<sup>2</sup> URI – унифицированный идентификатор ресурса

коммуникации и влиять на процессы в иерархии. Следовательно, MUC, соответствующая простой иерархии, должна иметь конфигурацию  $C = \langle \langle \text{название иерархии} \rangle, \text{moderated}, \text{member-only}, V, \text{semi-anonymous}, \text{password\_protected}, T \rangle$ , где  $V \in \{public, hidden\}$ ,  $T \in \{temporary, persistent\}$ .

Командный вариант OE (**team**) обычно складывается из замкнутого множества AA, которые имеют общую цель, имеют доступ к общей информации, и чьи задачи поручаются наиболее квалифицированным членам группы. Там нет какого-либо иерархически заданного управляющего, хотя там может быть представитель или лидер группы, уполномоченный привлекать новых AA и разрешать некоторые конфликты с назначением задач. AA постоянно общаются друг с другом, как публично, так и приватно. Таким образом, MUC для командной единицы имеет конфигурацию  $C = \langle \langle \text{название команды} \rangle, \text{unmoderated}, \text{member-only}, V, \text{non-anonymous}, S, T \rangle$ , где  $V \in \{public, hidden\}$ ,  $S \in \{\text{password\_protected}, \text{captcha\_protected}, \text{unsecured}\}$  и  $T \in \{temporary, persistent\}$ .

В плоско организованной OE (**flat unit**) среди AA присутствует полная анархия, каждый AA никого не контролирует. Следовательно, любой может войти в OE и общаться со всеми другими пользователями. В этом случае конфигурация MUC  $C = \langle \langle \text{название организации} \rangle, \text{unmoderated}, \text{open}, \text{public}, \text{non-anonymous}, \text{unsecured}, T \rangle$ , где  $T \in \{temporary, persistent\}$ .

Рыночно подобные организации могут моделироваться как с помощью MUC, так и с помощью СК. СК – это механизм, который обычно используется с целью пометки некоторых взаимоотношений, определяющих правила обмена сообщениями между участниками соответствующей группы в списке контактов. Использование этого механизма позволяет, например, фиксировать некоторые отношения, описывающиеся в протоколе XFN<sup>3</sup>, отношении объектов в организациях на основе контракта и т.д.

Группы общих контактов в основном используются в eSocialGRIDs, как форма устойчивых социальных групп, или “рынков” услуг (в дальнейшем будем называть такие организации просто 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 агенты или сервисы. Такие организации моделируются командным вариантом OE (**team**) и конфигурируются на сервере.

Независимые агенты могут присоединиться к соответствующей группе по запросу. Агент, с функциональностью управляющего, получив задачу, заключает контракты на выполнение задач с агентами из видимых групп. Каждая группа по смыслу соответствует некоторому типу “рынка” услуг, и соответствующий AA может там заказать себе решение задачи, также как и ему могут предложить решить задачу в рамках того круга, куда он входит.

Автономный агент может использовать также свой СК пользователя, разбивая его на группы, чтобы сформировать свои круги (круги здесь соответствуют группам в СК), куда входят объекты, выбранные данным агентом по определенным критериям. Члены этого списка 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 представленный данным автономным агентом человек следит за членами этой организации - различными устройствами в своем доме. Этот случай относится к простой иерархической OE (**simple hierarchy**), формируемой и управляемой исключительно самим AA, создателем этой организации.

Итак, как мы видели из вышеизложенного, СК удобны для моделирования устойчивых организаций MAC 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 членов, возникают и меняются отношения между объектами и т.д., по сравнению с иерархическим и рыночным типами организаций.

Как было упомянуто выше, более сложные организации можно представить комбинациями простых организационных единиц и соответственно моделировать структурами из СК и MUC.

## 8 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 в некой системе является то, что обычно называют “регистрацией”, в частности, на Интернет-ресурсах таких, как социальные сети. С регистрацией естественно связаны аутентификация и авторизация агентов в многоагентной среде. Аутентификация, как проверка подлинности AA, односторонняя или взаимная, позволяет AA за тем идентифицироваться в системе, получить определенный уровень доверия и последующего контроля со стороны

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

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 принято 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 2.0<sup>4</sup>. Эта возможность позволяет агентам 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 авторизированных, таким образом, АА.

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 агентами внешних сетей, т.е. АА-ов, 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 использования 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.

## 9 Заключение

Исходя из двух формальных моделей [5,6] 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 мультиагентной среды. В работе уточняются формальные понятия открытых и закрытых МАС, мультиагентных сред, 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 базовых 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 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 [14], которая также основывается на понятии АА. eSocialGRIDs может быть сконфигурирована для работы с агентными платформами, поддерживающими XMPP и HTTP коммуникацию.

Разработаны методы ленивой регистрации АА и других объектов в среде повсеместных вычислений, в том числе через Facebook, Google и т.д. рассмотрены методы интеграции с крупными социальными сетями, через АА.

<sup>4</sup>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>).

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 специализированных сетей и приложений, снабжая их структурами реального времени. Кроме того представленная платформа может служить мостом для широкого использования МАС в крупномасштабных приложениях, включая их в инфраструктуру повсеместных вычислений.

## Литература

- [1] V. B. Tarasov: *Ot mnogoagentnyh sistem k intellektual'nym organizacijam: filosofija, psihologija, informatika. M.: Jeditorial URSS: 348, 2002.*
- [2] L. Gasser: *Boundaries, Identity and Aggregation: Plurality Issues in MultiAgent Systems. Decentralized Artificial Intelligence: 199-212, 1992.*
- [3] M. Weiser: *The Computer for the Twenty-First Century. Scientific American: 94-102, September 1991.*
- [4] FIPA abstract architecture. [http://www.fipa.org/specs/fipa00001/XC00001J.html#\\_Toc8186403](http://www.fipa.org/specs/fipa00001/XC00001J.html#_Toc8186403).
- [5] M. d'Inverno and M. Luck: *Understanding Agent Systems, 2nd ed. Berlin: Springer: 240, 2004.*
- [6] V. Botti and A. Giret: *ANEMONA: A multi-agent methodology for Holonic Manufacturing. London: Springer: 214, 2008.*
- [7] M. Wooldridge: *An Introduction to MultiAgent Systems, 2nd ed. Glasgow: John Wiley & Sons: 484, 2009.*
- [8] S. Rassel, P. Norvig: *Iskusstvennyj intellekt, 2-e red. M.: Izdatel'skij dom "Vil'jams": 1408, 2006.*
- [9] E. Argente, V. Julian and V. Botti: *MAS Modeling based on Organization. In AOSE 2008, Heidelberg: 2009.*
- [10] SPADE (Smart Python multi-Agent Development), <http://en.wikipedia.org/wiki/SPADE>.
- [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: 1-11, 2006.*
- [12] L. Gasser: *An Overview of DAI. Distributed Artificial Intelligence: Theory and Praxis: 16-30, 1992.*
- [13] E. Argente, J. Palanca, G. Aranda, V. Julian and V. Botti: *Supporting Agent Organizations. In CEEMAS 2007, LNCS (LNAI), vol. 4696:236-245, 2007.*
- [14] 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. In AOSE 2009, Lecture Notes in Computer Science, vol. 6038:32-47, 2011.*

### Ubiquitous computing based on open multi-agent systems: the eSocialGRIDs project

**Abstract.** *The problem of global software infrastructure for support of the distributed applications within a paradigm of ubiquitous computing has been considered. The concept and the design methods for such infrastructure on the basis of a platform for the open multi-agent environment are described consistently. Feature of the proposed solution is orientation to technology of the multi-agent systems, based on the context-sensitive organizations of agents. Taking into account dynamic character of the infrastructure for ubiquitous computing the complex of the main problems of the agent organizations in an open multi-agent environment (the relations and communication between agents, structure, tasks and behavior of multi-agent systems) is considered. It is shown that the XMPP protocol allows supporting naturally the dynamic nature of the agent organizations and dynamic nature of ubiquitous communications in the distributed open computing environment. As result of this work the architecture and basic components of the eSocialGRIDs platform for the support of ubiquitous computing within the social environment which is a union of the different communities of people and software agents are developed.*