

ОТКРЫТЫЕ СЕТИ АГЕНТОВ

В. И. ГОРОДЕЦКИЙ¹, О. В. КАРСАЕВ², В. В. САМОЙЛОВ³,
С. В. СЕРЕБРЯКОВ⁴

Санкт-Петербургский институт информатики и автоматизации РАН

СПИИРАН, 14-я линия ВО, д. 39, Санкт-Петербург

¹<gor@iias.spb.su>, ²<ok@iias.spb.su>, ³<samovl@iias.spb.su>

⁴<sergey_s@iias.spb.su>

УДК 004.89

Городецкий В. И., Карсаев О. В., Самойлов В. В., Серебряков С. В. **Открытые сети агентов** // Труды СПИИРАН. Вып. 4. — СПб.: Наука, 2007.

Аннотация. Данная работа представляет реализованную авторами P2P агентскую платформу, экземпляры которой, установленные в узлах сети поверх стандартного P2P сервиса, образуют распределенную базу знаний, предназначенную для организации семантического P2P взаимодействия агентов. Прикладные агенты, в свою очередь, устанавливаются в узлах сети поверх экземпляров агентской платформы. В основу разработки положены функциональная архитектура, разработанная рабочей группой FIPA в качестве предложения для последующей программной реализации и стандартизации. Разработанная программная реализация платформы поддерживается также механизмом парных взаимодействий агентов на основе сообщений разработанных форматов, а также парных коммуникаций узлов сети. Такой механизм парных взаимодействий агентов также разработан авторами. Роль, функции и существо процессов функционирования этой платформы поясняется на примерах двух приложений, которые сами по себе являются достаточно важными с практической точки зрения. Эти же приложения использованы для верификации основных решений, предложенных в работе. — Библ. 26 назв.

UDC 004.89

Gorodetsky V. I., Karsaev O. V., Samoilov V. V., Serebryakov S. V. **Open networks of agents** // SPIIRAS Proceedings. Issue 4. — SPb.: Nauka, 2007.

Abstract. This paper presents developed by authors P2P agent platform, a special software product. Instances of P2P agent platform are deployed in the nodes of P2P network over standard P2P service and altogether produce distribute knowledge database, intended for organizing of semantic agents' interaction with each other. Agents of applied system are deployed in the nodes of P2P network over instances of P2P agent platform. The functional architecture, intended for further implementation and standardization, developed by FIPA working group was taken as basis of development. Implemented program product supports such mechanisms as P2P agents' interacting on the basis of ACL — agent communication language, white and yellow pages. These mechanisms are also were developed by authors of this paper. The role, functions and functioning process are shown using two implemented examples, which are quite important from practical point of view. These examples were also used to test and verify main system mechanisms, proposed in this paper, in particular, registering and deregistering of agents and their services in white and yellow pages, distributed searching of agents and their services in the network. — Bibl.26 items.

1. Введение и мотивация

Создание систем распределенных вычислений на основе парных взаимодействий (P2P computing), использующих недавно разработанную сетевую архитектуру распределенных систем, в настоящий момент является объектом усиленного внимания со стороны науки и промышленных компаний. Изначально такая архитектура была предложена «для совместного использования ресурсов компьютеров (данных, хранилищ данных, процессорного времени) путем прямого обмена данными между заинтересованными компонентами без использования посредников или центрального сервера, или каких-либо еще спе-

циальных средств» [2]. В таком подходе узлы сети выполняют одинаковые роли, выступая в различных сценариях или в роли клиента, или в роли сервера [20]*. Идея распределенных P2P вычислений явилась движущей силой для возникновения новых идей, архитектур и возможностей в проектировании и программной реализации современных распределенных приложений, состоящих из автономных компонент. Одной из таких возможностей является интеграция многоагентных систем (МАС) и архитектур, использующих P2P вычисления.

Во многих случаях многоагентные системы могут рассматриваться как множество взаимодействующих автономных компонент, а потому объединение агентов МАС в сеть, узлы которой взаимодействуют в P2P стиле, может привести к архитектуре, предоставляющей принципиально новые возможности для создания *открытых* МАС, в которых множество функционирующих агентов динамично за счет того, что агенты могут покидать МАС и присоединяться к ней в любой момент времени. Следует заметить, что последнее требование характерно для многих современных, зачастую «критических» приложений. В качестве примеров таких приложений можно привести системы удаленного мониторинга и объединения информации в задачах наблюдения за большими географическими регионами для предсказания и выявления природных аномалий и катастроф (цунами, ураганов, землетрясений и т.п.) [22]. В настоящий момент быстро развивается рынок встроенных систем с мобильным доступом, построенных на основе архитектуры СОА, используемых в широком круге приложений. Например, именно на базе такой архитектуры разрабатывается концепция «Smart Home» («Умный дом»), которая предполагает создание интеллектуальной инфраструктуры для управления, в том числе удаленного (например, с помощью Интернет доступа) всеми домашними устройствами, климатом и т.п. P2P агентские системы представляются весьма перспективными для создания систем антитеррористической борьбы. То же самое можно уверенно утверждать для практически неограниченного множества военных приложений, решающих задачи оперативного мониторинга больших географических регионов. Многие задачи распределенной диагностики сложных технических объектов масштаба атомных электростанций могут эффективно решаться с помощью P2P агентских систем. Список потенциальных приложений можно продолжить.

Современные исследования показывают, что для практической реализации идеи открытой P2P МАС необходимо создание программной компоненты (*middleware*), поддерживающей *прозрачное взаимодействие агентов*. Такая компонента далее называется «*P2P агентской платформой*», которая должна состоять из множества идентичных экземпляров, установленных в узлах сети агентов, должна выполнять роль посредника между прикладными агентами открытой МАС и коммуникационной компонентой, реализующей P2P маршрутизацию сообщений. Термин «прозрачное взаимодействие агентов» означает, что агенты ничего не обязаны знать о том, каким образом реализуется адресация (маршрутизация) и выполнение семантических запросов, могут не знать о том, какой именно агент или какие именно агенты выполняют их запросы, могут мало знать или совсем ничего не знать о том, какие агенты и с какими функциональностями вообще существуют в сети и присутствуют ли они в сети реально в текущий момент времени или выключены. С другой стороны, компонента, поддерживающая P2P маршрутизацию сообщений (далее — P2P провайдер),

* Понятие (отношение) *Client-server* является атрибутом транзакций, а не компьютеров или других устройств [2].

может ничего не знать о том, какие агенты установлены в узлах сети и сколько их в каждом узле и в сети в целом, и какими функциональностями они обладают. Другими словами, «прозрачность» взаимодействия агентов прежде всего означает, что агенты не должны что-либо знать о том, как и кем предоставляется транспортный сервис, а P2P провайдер может не знать, какой именно паре агентов он предоставляет этот сервис.

В соответствии с концепцией FIPA* обмен сообщениями между агентами реализуется использованием соответствующего сервиса, называемого FIPA Message Transport Service (MTS) [8]. Для P2P сетей агентов, FIPA MTS реализуется над некоторым P2P транспортным протоколом, например, JXTA, Bluetooth OBEX или каким-либо другим протоколом, используемым P2P провайдером. По аналогии с P2P технологиями распространения контента [2] обеспечение прозрачности коммуникаций может выполняться сетью программных компонент, инсталлированных «поверх» P2P провайдера. Множество распределенных компонент такой сети можно рассматривать в качестве распределенных сервисов P2P провайдера, которые позволяют реализовывать поиск сервисов и агентов на удаленных устройствах в средах [11], не имеющих управляющего сервера. Именно это множество распределенных компонент назван ранее P2P агентской платформой (P2P АП).

Задача разработки функциональной архитектуры P2P АП в конце 2005 г. была возложена на рабочую группу FIPA, которая официально называется Nomadic Agent Working Group 6 (NA WG6) [11], которая в своем первом документе [11] отметила, что в настоящее время «в мире не существует базовой P2P агентской платформы и FIPA намерена сделать шаг в этом направлении, предоставив спецификации и ускорив разработку ее программной реализации. Задачей рабочей группы является предоставление спецификаций для P2P агентов, установленных на мобильных устройствах (*nomadic P2P агентов*) и информацию о программных разработках, способных функционировать на малогабаритных мобильных устройствах». Предварительный документ, содержащий описание первого варианта функциональной архитектуры P2P АП был выпущен в начале 2006 г. К сожалению, до сих пор в известной авторам литературе не было информации о конкретной программной реализации такой платформы, а потому описываемая далее ее реализация является пионерской.

Данная статья представляет описание разработанной авторами P2P АП, которая реализует базовые обязательные компоненты функциональной архитектуры платформы, предложенной группой NA WG 6 [10]. В текущей реализации сетевая P2P коммуникационная инфраструктура поддерживается P2P провайдером, разработанным авторами. Описываемая далее P2P АП была экспериментально проверена на ряде приложений, включающих такие системы, как: P2P агентская система обнаружения вторжений в компьютерную сеть, описанная в работе [12], интеллектуальная сенсорная P2P сеть для предсказания возникновения течения El Nino [17], описанная в [13], а также система коллективного распознавания наземных объектов, попадающих в поле зрения распределенной системы наблюдения, развернутой на множестве малогабаритных беспилотных летательных аппаратов (БЛА). Последняя система состоит из множе-

* FIPA (Foundation for Interaction of Physical Agents) — это общественная организация, созданная сообществом специалистов в области многоагентных систем для разработки научно обоснованных предложений по стандартизации языков, моделей, интерфейсов и т.д. в области многоагентных систем [26]. В настоящее время многие документы FIPA рассматриваются промышленными компаниями в качестве де-факто стандартов.

ства агентов, установленных на каждом БЛА. Каждый агент обучен обнаруживать только объекты определенного класса при его наблюдении под определенным ракурсом (спереди, сзади, слева, справа). Агенты такой открытой системы рассматриваются как провайдеры конкретных классификационных сервисов, которые могут быть доступны другим агентам системы на основе парных коммуникаций. Два последних примера описаны в последних разделах данной работы.

Статья в целом далее организована следующим образом. Во втором разделе приводится краткий обзор работ по рассматриваемой тематике. В третьем разделе дается краткое описание функциональной архитектуры P2P АП, предложенной NA WG 6 [10]. В четвертом разделе описывается разработанная авторами архитектура и программная реализация P2P АП, а также производится ее сравнение с архитектурой, предложенной группой NA WG 6. В пятом разделе представлен используемый формат описания сервисов, который совместим с форматом FIPA. В шестом разделе приводятся описания разработанных прикладных систем, реализованных программно с использованием разработанной P2P АП. Они использовались для тестирования платформы, а также для содержательного пояснения существа ее работы. Примеры включают систему коллективного распознавания наземных объектов и систему мониторинга конкретного географического региона. Заключение содержит основные результаты работы и перспективы использования разработанной P2P АП.

2. Состояние проблемы и известные результаты

Задача совместного использования идей MAC и P2P систем в последнее время становится объектом пристального внимания исследователей. В работе [14] авторы описывают MAC, решающую задачу композиции распределенных веб-сервисов. В построенной модели поиск агентов и сервисов реализуется в P2P стиле. MAC реализована в виде программных компонент, использующих для коммуникации на нижнем уровне структурированную P2P сеть *Chord*. В работе [24] авторы рассматривают агентскую систему для мобильных P2P приложений. Разработанная система, названная *FRAGme2004*, является трехуровневой. На нижнем уровне располагается провайдер P2P, на среднем уровне располагаются системные агенты, управляющие сетевыми ресурсами, а на верхнем уровне располагаются компоненты, специфичные для прикладной задачи. Агенты P2P узла сети рассматриваются как компоненты, управляющие ресурсами.

В работе [23] авторы описывают агентскую P2P систему, предназначенную для поиска информации, в которой агенты взаимодействуют в P2P стиле. В работе [4] агенты MAC организованы в гибридную P2P сеть. Агенты сети в любой момент могут покинуть сеть и войти в нее, а узлы P2P сети представлены мобильными устройствами, на которых располагаются агенты. В ряде работ, например в [25, 26, 1], авторы проектируют сами узлы P2P сети в виде программных агентов, которые интегрируют в себе аспекты прикладных задач.

Общим свойством рассмотренных работ является то, что авторы не разделяют сетевой уровень P2P инфраструктуры и уровень прикладных агентов, интегрируя агентскую и P2P составляющие в единое целое, тем самым создавая приложения, которые могут использоваться, главным образом, в рамках конкретной прикладной задачи, а их адаптация к другим приложениям потребует значительных усилий.

Идеи использования желтых страниц, как это предлагается в документах FIPA, реализованы в работе [18]. Авторы этой работы рассматривают множество взаимодействующих друг с другом многоагентных систем, причем каждая из них располагается на своем узле сети и использует сервис своего P2P провайдера. В предложенной системе отсутствует промежуточный слой, называемый агентской платформой. Соответствующие сервисы (поиск агентов и сервисов) выполняются специальными агентами (Facilitator). В работе [3] в качестве коммуникационной P2P среды используется JXTA инфраструктура. Компонента, реализующая некоторый сервис, который публикуется посредством JXTA, называется в работе агентом. Таким образом, в работе рассматривается MAC распределенных сервисов, доступных посредством инфраструктуры JXTA.

В работе [21] авторы делают шаг вперед, интегрируя MAC технологии в сетевую инфраструктуру P2P. Коммуникация между агентами происходит через специального агента — «агента Пира». Этот агент использует инфраструктуру JXTA для коммуникации с агентами, расположенными на разных пирах. В терминах ФИПА этот агент реализует то, что должна реализовывать компонента MTS (message transport system — система отправки сообщений) агентской платформы. Происходит это следующим образом: «агент Пира» смотрит, кому адресовано сообщение. Если сообщение адресовано агенту, находящемуся на локальном пире, то он просто передает ему это сообщение. Если же агент-получатель зарегистрирован на удаленном пире, то используя инфраструктуру JXTA, отправляет сообщение «агенту Пира» удаленного узла. Тот, получив сообщение, смотрит, кому оно адресовано, и передает сообщение соответствующему агенту.

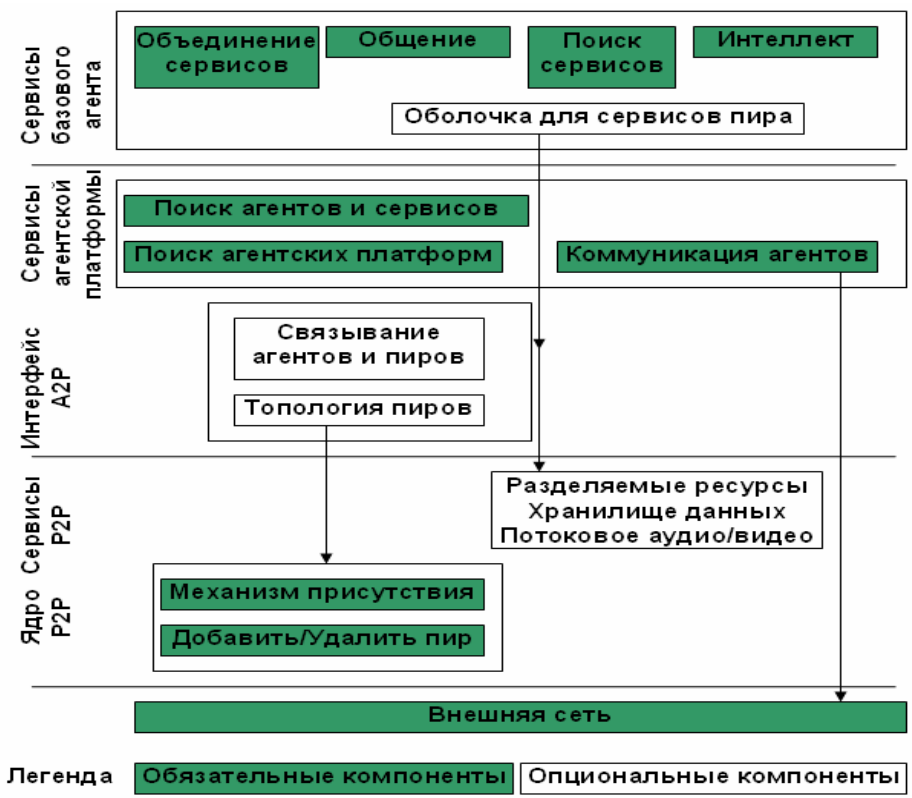
Таким образом, несмотря на то что P2P MAC в настоящее время являются объектами интенсивных исследований, к настоящему времени предложены, главным образом, решения для конкретных, частных задач. Подход, предложенный в [21], является наиболее интересным и значительным шагом в направлении создания стандартных решений, которые могли бы быть использованы уже многократно при создании различных приложений агентских P2P систем.

Аналогичная идея, реализованная на более абстрактном, более общем уровне, является предметом исследований и разработок в данной работе. Ее целью является полное разделение уровня прикладных агентов и уровня P2P коммуникационного сервиса с помощью некоторого промежуточного программного обеспечения, которое может рассматриваться как программная реализация функциональной архитектуры, предложенной для этих целей FIPA NA WG 6 [10]. По имеющейся у авторов информации она является первым решением в области P2P агентских технологий, которое независимо от конкретного приложения. Соответствующее программное обеспечение реализовано как распределенная сеть экземпляров агентских платформ, установленных поверх сети P2P провайдеров. Экземпляр агентской P2P платформы, в рамках построенной реализации, рассматривается просто как потребитель коммуникационного P2P сервиса. С другой стороны, сеть экземпляров агентских платформ по существу реализует идею распределенной базы знаний о возможностях (сервисах) различных агентов P2P сети, которая и обеспечивает прозрачность коммуникаций агентов, основанную на «семантической» маршрутизации. Естественно, что соответствующий экземпляр агентской платформы должен быть зарегистрирован в качестве клиента P2P провайдера. Это позволяет агентам, установленным поверх конкретного экземпляра агентской платформы, взаимодей-

ствовать с другими агентами, расположенными на других экземплярах платформы, используя P2P инфраструктуру сети. Разработанная P2P АП поддерживает поиск сервисов и агентов, в то время как в таком сценарии прикладные агенты ответственны только за реализацию поведения, специфичного для прикладной задачи. Заметим, что множество прикладных агентов может рассматриваться как агентская сеть, инсталлированная поверх агентской платформы. В последующих разделах данной работы очерченные выше особенности предложенного решения рассматриваются более подробно.

3. Функциональная архитектура P2P агентской платформы для мобильных устройств, предложенная FIPA NA WG

Функциональная архитектура P2P АП, которая рассчитана на ее использование в мобильных устройствах [10], представлена на рис. 1. Как уже отмечалось, эта архитектура предложена рабочей группой FIPA NA WG 6 [11] в каче-



• Рис. 1. Функциональная архитектура, предложенная Nomadic Agents Working

стве исходного варианта. Эта архитектура построена по трехуровневой схеме. Нижний уровень архитектуры реализует функции P2P провайдера, т.е. является серверной частью программы, которая предоставляет своим клиентам услуги (сервис) по отправке и приему сообщений по протоколам, поддерживающим P2P маршрутизацию. Далее для краткости эту часть рассматриваемой функциональной архитектуры будем называть *пиром*. Множество пиров, установленных в различных узлах сети, например TCP/IP сети, которая реализует транспортный уровень, можно рассматривать как виртуальную («оверлейную») сеть, развернутую поверх «обычной» (например, поверх TCP/IP) сети.

Программные компоненты пира, в свою очередь, образуют двухуровневую подсистему. На верхнем уровне пира находится компонента, которая является его серверной частью. На нижнем уровне пира находится компонента, которая на рис. 1 названа *P2P ядром*.

Ядро пира включает в себя два программных механизма. Первый из них должен сигнализировать о том, является ли пир в текущий момент времени узлом P2P сети, т.е. «присутствует» ли данный пир в P2P сети или он отключен от нее («Механизм присутствия»). Второй механизм предназначен для управления списком контактов, т.е. списком «соседних» пиров («Добавить/Удалить пир»). Этот механизм позволяет добавлять в список соседей или удалять из него те или иные пиры. Следует обратить внимание на то, что эта компонента только реализует этот механизм, однако запускаться он может различным образом. В абстрактной архитектуре, рассматриваемой в данном разделе, вопрос о том, какая компонента архитектуры принимает решение об изменении списка соседей, не решается. Предполагается, что этот вопрос может решаться различным образом в зависимости от «политики управления», принятой в каждой

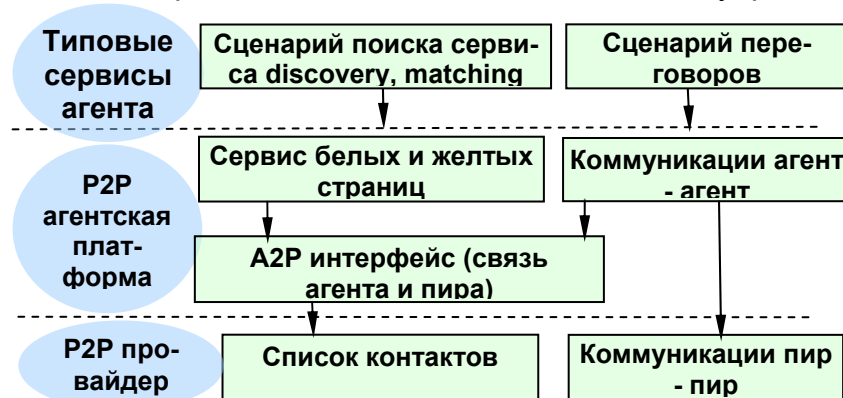


Рис. 2. Функциональная архитектура P2P агентской платформы и P2P провайдера.

конкретной реализации данной абстрактной архитектуры. Кроме P2P сервисов пир может обладать и другими сервисами.

Под сервисами пира понимаются некоторые его опциональные возможности, которыми он может обладать, например, пир может хранить

файлы, устанавливая прямые каналы для потокового аудио/видео и т.п. Заметим, что пир как сервер может предоставлять сервисы не только агентам, работающим в P2P сети, но и другим программам, являющимся его клиентами.

Средний уровень рассматриваемой функциональной архитектуры образует собственно *P2P агентская платформа*. Компоненты этой платформы разделяются на два подуровня. Первый из них образуют сервисы агентской платформы, например такие, как поиск сервисов и агентов в P2P сети, поиск других экземпляров агентских платформ и реализация коммуникаций между агентами. Заметим, что язык коммуникаций определяется документом FIPA ACL [5]. Как уже отмечалось, механизм, реализующий коммуникации между агентами, в документах FIPA называется *Message Transport Service (MTS)*. В соответствии с документом [10], компоненты, реализующие поиск сервисов и агентов, включают в себя системы управления агентами (*Agent Management System, AMS*) и справочники сервисов (*Directory Facilitator, DF*). Компонента *AMS* ответственна за ведение и реализацию *сервиса белых страниц*, которые содержат список агентов, зарегистрированных на платформе. Она также ответственна за поддержание жизненного цикла агентов. Компонента *DF* предоставляет агентам *сервис желтых страниц*. Множество желтых страниц, расположенных на разных экземплярах агентской платформы, реализуют сервис *распределенных желтых страниц*. По аналогии с оверлейной P2P сетью, множество экземпляра-

ров агентской платформы образуют оверлейную сеть, развернутую поверх P2P сети.

Второй уровень агентской платформы образуется компонентой, которая в архитектуре FIPA NA WG 6 называется *A2P интерфейсом (Agent-to-Peer interface)*. Ее можно рассматривать как некоторое промежуточное программное обеспечение, реализующее взаимодействие агентов пира с пиром, т.е. как интерфейс, связывающий агентов и пир, на котором агенты установлены.

Верхний уровень функциональной архитектуры, представленной на рис. 1, содержит описание стандартных сервисов агентов, созданных на основании соответствующих спецификаций FIPA.

4. Реализация P2P платформы агента

4.1. Функциональная архитектура

Архитектура разработанной реализации P2P АП представлена на рис. 2. Она в основных чертах следует архитектуре, предложенной FIPA NA WG, в частности, она реализует все ее обязательные компоненты.

Программные компоненты платформы также построены по трехуровневой схеме (рис. 2). *На нижнем ее уровне* находится пир, т.е. узел P2P сети, который предоставляет свои сервисы потребителям. В текущей реализации пир предоставляет своим клиентам P2P каналы связи с другими пирами*, а также реализует механизм управления списком своих контактов. Напомним, что список контактов содержит имена соседних пиров и их адреса, т.е. тот список узлов P2P сети, с которыми данный пир имеет прямые коммуникационные каналы.

Средний уровень реализует P2P АП, которая предоставляет сервисы белых и желтых страниц агентам, зарегистрированным на платформе. В текущей реализации агентская платформа поддерживает поиск агентов в P2P сети и сервисов. Поиск реализуется как распределенный поиск, который использует протокол, известный под названием *gossiping* [15], поддерживающий P2P маршрутизацию сообщений.

На верхнем уровне располагаются агенты прикладной системы, зарегистрированные на P2P АП.

Точно так же, как P2P сеть образует «оверлейную сеть», установленную поверх TCP/IP сети, множество экземпляров P2P агентской платформы образует «оверлейную сеть» установленную поверх P2P сети (см. рис. 3). В этой сети множество экземпляров P2P АП играет роль распределенной базы знаний. Каждый экземпляр платформы содержит знания об агентах, которые на ней установлены, о сервисах, предоставляемых этими агентами. В частности, желтые страницы содержат информацию о сервисах агентов экземпляра платформы, представленную в стандартной форме, в терминах общей онтологии сети агентов, и об именах агентов, которые соответствующие сервисы предоставляют. Экземпляр агентской платформы может также содержать знания о некоторых других экземплярах платформы. Белые страницы содержат информацию об адресах агентов, способных предоставлять сервисы, перечисленные в желтых страницах. Аналогично, множество агентов, установленных на экземплярах

* На нижнем уровне можно использовать существующие реализации P2P провайдера, например JXTA или WiFi OВEX. В описываемой реализации используется P2P реализация, разработанная С. В. Серебряковым, одним из авторов данной работы.

агентской платформы, образуют «оверлейную сеть» третьего уровня, которая установлена поверх сети экземпляров агентской платформы. Трехслойная архитектура открытой сети агентов демонстрируется на рис. 3.

Важно, что в начальном состоянии желтые и белые страницы могут быть вообще пустыми, а их заполнение может происходить уже в процессе функционирования сети, когда агенты объявляют о своих сервисах, запрашивают тот или иной сервис или получают ответы на свои запросы о сервисах. (Форматы соответствующих сообщений будут рассмотрены далее). Таким образом, содержание желтых и белых страниц может постоянно обновляться в процессе работы сети. При этом агентская платформа может быть дополнительно снабжена компонентой, которая отслеживает потребности установленных на ней агентов, накапливает соответствующую информацию и, привлекая алгоритмы обучения, оптимизирует содержание желтых и белых страниц экземпляра платформы в соответствии с предметно-ориентированными показателями качества работы системы. Параметрами платформы, определяющими конфигурацию связей между агентами сети в целом, могут быть объем желтых и белых страниц, порядок хранения записей в ней, предпочтения по выбору сходных сервисов, которые используются агентами, установленными на платформе, и другие. По сути эта компонента может использоваться для динамического формирования профилей установленных на ней агентов и для адаптации атри-

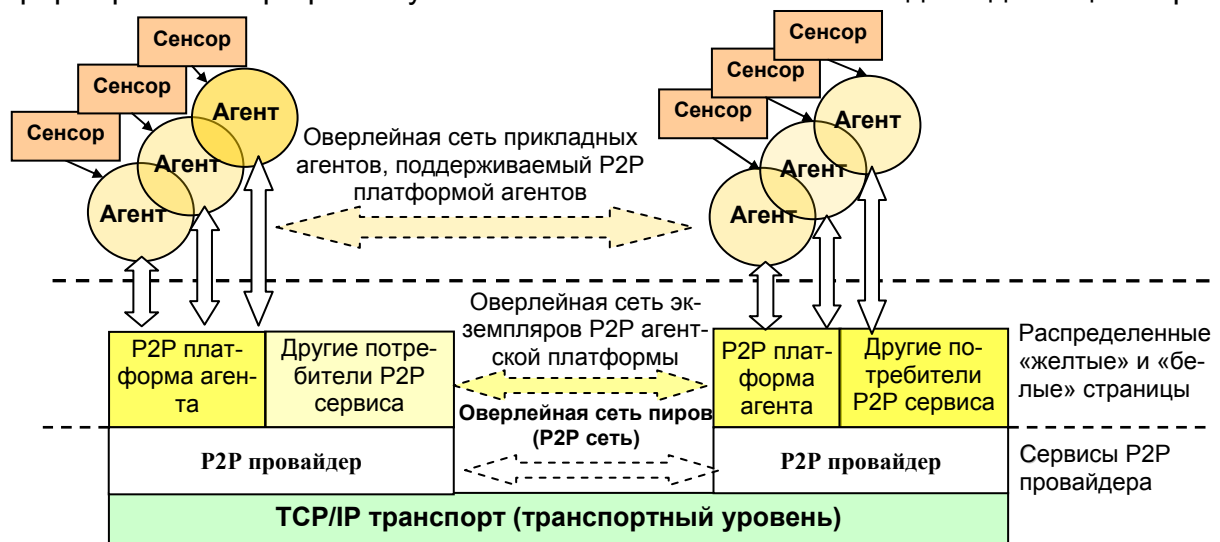


Рис. 3. Оверлейные сети, построенные над TCP/IP, обеспечивающие «прозрачность» взаимодействия прикладных агентов.

бутов самой платформы с целью оптимизации ее функционирования в интересах собственных агентов и, возможно, в интересах сети в целом. Возможность динамического формирования содержания белых и желтых страниц становится особенно важным свойством в случае *открытой сети агентов*, в которой агенты и узлы сети могут появляться в ней и уходить из нее в любой момент времени.

4.2. Взаимодействие агентской платформы с P2P провайдером

Общая схема взаимодействия компонент разработанной функциональной архитектуры представлена на рис. 4. С точки зрения пира экземпляр агентской платформы — это один из потребителей предоставляемого им P2P сервиса.

Если некоторое приложение желает получить доступ к P2P сети, то оно должно получить доступ к пиру через специальный программный механизм, после чего зарегистрироваться на нем. При регистрации на пире любой потребитель указывает свой уникальный идентификатор и тип. Идентификатор однозначно определяет потребителя в сети, а тип указывает на некоторое функциональное описание этого потребителя. Например, одним из типов потребителей P2P сервиса является «*agent-platform*» (агентская платформа), другим типом может быть «*file-storage*» (сервис хранения файлов) и т.п.

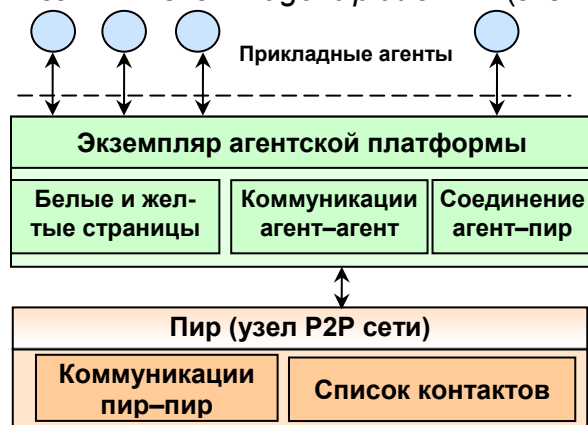


Рис. 4. Взаимодействие между агентами, агентской платформой и пиром на локальном хосте.

Как видно из описания, потребитель сервисов и P2P провайдер «слабо» связаны друг с другом. Это означает, что потребитель в принципе может функционировать и без использования P2P сервиса, например, в случае если все агенты прикладной системы располагаются на одном экземпляре агентской платформы.

Рассмотрим взаимодействие потребителя с пиром более подробно (рис. 5). Потребитель взаимодействует с пиром не напрямую, а через специальную компоненту *фабрику пиров* (*peer factory*). Данная компонента по запросу потребителя предоставляет последнему запрашиваемый пир (фактически, передается указатель на интерфейс пира). Генератор пиров по запросу потребителя может создавать пиры двух типов: *пир уровня приложения* (*in-proc-peer*) и *пир уровня системы* (*out-of-proc-peer*). Пир уровня приложения создается фабрикой пиров и находится в адресном пространстве процесса — потребителя сервиса. Преимуществом данного подхода является быстрое действие. Его недостатком является то, что такой пир не является «разделяемым», т.е. только потребитель, создавший данный пир, может зарегистрироваться на нем. Второй тип пира, пир уровня системы, создается системой через механизм *СОМ*. Преимуществом данного подхода является то, что такой пир может использоваться и другими приложениями системы. Однако его недостатком является проигрыш в быстродействии, т.к. дополнительно используется механизм межпроцессного взаимодействия между потребителем и пиром. Отметим, что этот механизм скрыт от потребителя, и реализуется фабрикой пиров.

4.3. Взаимодействие агентов с агентской платформой

Связь между агентами и агентской платформой, в отличие от связи агентской платформы с пиром, является сильной. Это означает, что агенты не могут функционировать вне агентской платформы. Платформа полностью управляет жизненным циклом агента, т.е. она создает его, инициализирует, устанавливает

на платформе, деинсталлирует и удаляет. Уровни взаимодействия агент/агентская платформа/пир представлены на рис. 4.

При создании агента экземпляром платформы происходит обмен интерфейсами между платформой и агентом. Через интерфейс платформы агент получает доступ к сервисам платформы, а через интерфейс агента платформа может, например, уведомлять агента о новых сообщениях. Агентам доступны такие сервисы агентской платформы, как отправка сообщений другим агентам, поиск конкретных сервисов и агентов, регистрация сервисов на желтых страницах, регистрация самого агента на белых страницах и т.д.

В разработанной платформе желтые и белые страницы являются обязательными компонентами каждого экземпляра агентской платформы. Они реализованы как агенты. Эти агенты могут образовывать коалиции с агентами таких же типов, расположенными на других экземплярах платформы, которые в свою очередь расположены на пирах — соседях исходного пира. Такие коалиции позволяют прикладным агентам использовать алгоритмы поиска сервисов и агентов, разработанные протоколы маршрутизации сообщений в P2P сетях, например *flooding* или *gossiping* [15].

В заключение данного подраздела еще раз отметим суть интеграции P2P сети и сети экземпляров агентской платформы. Данную интеграцию можно рассматривать с двух сторон. С одной стороны, агентская платформа использует P2P сеть для коммуникаций с различными экземплярами агентских платформ без использования сервера маршрутизации, а через экземпляры агентских платформ с установленными на них агентами, расположенными, возможно, на различных устройствах. Таким образом, каждый P2P провайдер предоставляет стандартный коммуникационный сервис, связывающий различные экземпляры агентских платформ в одну распределенную агентскую платформу. С другой стороны, интеграцию P2P сети и сети экземпляров агентской платформы можно рассматривать как внутреннее развитие возможностей многоагентных систем. Как уже было отмечено, агенты белых и желтых страниц путем формирования коалиций фактически могут создавать соответствующие открытые P2P сети агентов в сети экземпляров агентской платформы, которые делают возможным поиск агентов и сервисов. Очевидно, что такая интеграция вносит принципиально новые возможности для построения больших масштабируемых многоагентных систем.

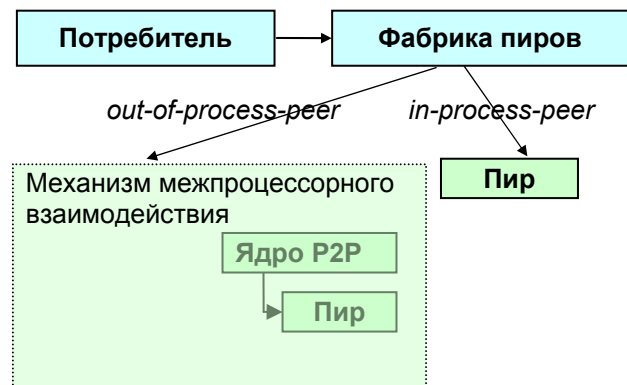


Рис. 5. Функциональная архитектура пира.

5. Сервисы, поиск сервисов и форматы сообщений

Любую функциональность, которую агент может предоставить в распоряжение других агентов, можно рассматривать как сервис, предоставляемый данным агентом другим агентам MAC. Разработанный формат описания сервисов базируется на документе FIPA [7].

5.1. Формат описание сервисов

Любой сервис, в соответствии с принятой FIPA моделью, характеризуется именем, типом и свойствами. Имя сервиса является уникальным именем в агентской среде и служит однозначным его идентификатором. Типом сервиса является некое концептуальное описание сервиса, говорящее о его предназначении. Множеству «похожих» сервисов обычно приписывается одинаковый тип. Свойства сервиса описывают его на более детальном уровне, различая однотипные сервисы, которые представляются в виде множества экземпляров понятий из онтологии агента. Например, если в онтологии агента определены некоторые понятия *Notion1*, *Notion2* и *Notion3*, то в псевдокоде описание сервиса будет иметь следующий вид:

```
service-description //описание сервиса
service-name //имя сервиса
service-type //тип сервиса
properties //свойства сервиса
  notion1_a //экземпляр понятия Notion1
  notion1_b //экземпляр понятия Notion1
  notion2_a //экземпляр понятия Notion2
  notion3_a //экземпляр понятия Notion3
  notion3_b //экземпляр понятия Notion3
```

В разработанной версии формального описания сервисов понятия, которые могут быть использованы для его описания, должны быть простыми, т.е. не наследоваться ни от каких-либо других понятий. Аналогично, атрибутами понятий должны быть простые типы — *integer*, *real*, *string*, *boolean*, но не другие типы. В соответствии с принятыми ограничениями, некоторое понятие в псевдокоде может быть описано, например, следующим образом:

```
Notion //имя понятия
  Attribute //атрибут
    Name //имя атрибута
    Type //тип атрибута
    Value //значение атрибута
  Attribute //атрибут
    Name //имя атрибута
    Type //тип атрибута
    Value //значение атрибута
```

...

Как видно из приведенного примера, каждое понятие характеризуется некоторым множеством атрибутов. Атрибуты в свою очередь характеризуются именем атрибута, его типом и конкретным значением.

В существующей версии описания сервисов передаются прикладными агентами экземпляры Р2Р агентской платформы в виде *xml* документа.

```
<service-description>
  <service-name>
    Classification_Target_1
  </service-name>
  <service-type>Classification</service-type>
  <properties>
    <notion>
      <name>Cost</name>
      <attribute>
        <name>Cost</name>
        <type>real</type>
        <value>12.232</value>
      </attribute>
    </notion>
    <notion>
      <name>ClassificationTarget</name>
      <attribute>
        <name>ClassificationTarget</name>
        <type>string</type>
        <value>Object_1</value>
      </attribute>
    </notion>
    <notion>
      <name>Experience</name>
      <attribute>
        <name>Experience</name>
        <type>real</type>
        <value>3.34</value>
      </attribute>
      <attribute>
        <name>Assistances</name>
        <type>integer</type>
        <value>2</value>
      </attribute>
      <attribute>
        <name>AvAssistExp</name>
        <type>real</type>
        <value>2.32</value>
      </attribute>
    </notion>
  </properties>
</service-description>
```

Рис. 6. Пример описания сервиса, формируемого агентом желтых страниц.

Рассмотрим пример. Пусть агент предоставляет сервис типа **Classification** (классификация) с именем **Classification_Object_1**. Пусть в онтологии агента определены следующие понятия, которыми может быть охарактеризован данный сервис:

– понятие **Cost** — стоимость сервиса, которое содержит единственный атрибут **Cost** — стоимость выполнения данного сервиса:

Notion Cost

Attribute Cost: cost (real);

– понятие **ClassificationTarget** (объект классификации), которое содержит единственный атрибут **ClassificationTarget** — объект классификации:

Notion ClassificationTarget

Attribute ClassificationTarget: target (string)

– понятие **Skill** — уровень качества рассматриваемого сервиса, которое содержит такие атрибуты, как **Experience** (опыт), **Assistances** (количество ассистентов) и **AvAssistExp** (усредненный опыт ассистентов):

Notion Skill

Attribute Experience : experience (real)

Attribute Assistances : count (integer)

Attribute AvAssistExp : av_exp (real)

Тогда описание сервиса, которое должен подготовить агент прикладной системы для агента желтых страниц, будет иметь вид, представленный на рис. 6.

Дополнительные ограничения на формат описания сервиса следующие.

- 1) Идентификаторы имени и типа сервиса, а также имена понятий и атрибутов не должны содержать пробелов.
- 2) Возможными значениями для типов атрибутов являются «**real**», «**integer**», «**boolean**» и «**string**»; любое имя (сервиса, понятия, атрибута, типа сервиса) не может совпадать с зарезервированным ключевым словом «**AND**».
- 3) Понятие обязано содержать хотя бы один атрибут. Поля «**service-name**» и «**service-type**» узла «**service-description**» обязательны,
- 4) Поля «**name**» узлов, «**notion**» и «**attribute**» обязательны.
- 5) Поля «**type**» и «**value**» узла «**attribute**» обязательны.

Табл. 1 приводит соответствие типов атрибутов и их возможных значений.

Таблица 1

Соответствие типов и возможных значений атрибутов в описаниях сервисов

Тип атрибута	Возможные значения
boolean	true, True, TRUE, false, False, FALSE
integer	Произвольное число типа int (integer)
real	Произвольное число типа double (real)
string	Любое строковое значение, которое может быть сохранено в типе языка C++ std::wstring

5.2. Формат поискового запроса на сервис

Любой агент прикладной системы может вызывать существующие методы платформы для поиска необходимых ему сервисов. Поиск сервиса может происходить по имени, типу или некоторому поисковому запросу. Очевидно, что в случае поиска по поисковому запросу ограничение по имени не должно задаваться. В этом случае следует использовать соответствующий метод платформы, а также можно использовать поле типа.

Поисковый запрос представляет собой строку определенного формата, которая задает ограничения по атрибутам понятий, описывающих искомый сервис. Ограничения задаются в виде равенств/неравенств. Возможно использование следующих операторов:

'<', '<=', '>', '>=', '==', '!=',

семантика которых традиционна.

Доступ к атрибутам понятий осуществляется с помощью специальной конструкции `NotionName.AttributeName`.

Доступ к полю типа сервиса и имени сервиса осуществляется при помощи зарезервированных ключевых слов `service-type` и `service-name`.

Ограничения задаются в одинарных кавычках ". Различные ограничения разделяются при помощи ключевого слова **AND**, перед которым и после которого должны быть пробелы. Ниже дается пример поискового запроса для вышеописанного примера:

```
ServiceType == 'Classification' AND Cost.Cost <= '5' AND
ClassificationTarget.ClassificationTarget == 'Object_1' AND Experience.Experience > '3'
AND Experience.Assistances >= '1' AND Experience.AvAssistExp > '0.6'
```

5.3. Форматы сообщений

При поиске сервиса формируется сообщение со следующим содержанием*:

```
<service-search>
  <search-query>query</search-query>
</service-search>
```

Пример сообщения, имеющего целью поиск сервиса, дается ниже:

```
<service-search>
  <search-query>ServiceType == 'Classification' AND Cost.Cost <= 15 AND
  ClassificationTarget.ClassificationTarget == 'Object_1'
</search-query>
</service-search>
```

Сообщение, которое ожидает агент в ответ на запрос сервиса, может иметь следующий формат:

```
<service-search>
  <search-reply>
    <agent-description>
      <agent-name>AgentName</agent-name>
      <agent-state>AgentState</agent-state>
```

* В терминологии многоагентных систем содержание сообщения принято именовать «контентом» (content)

```

</locator>
  <transport-type>Type</ transport-type >
  <transport-specific-address>Address</ transport-specific-address>
  < transport-specific-property>Property</ transport-specific-property>
</locator>
<agent-services>
  <service-description>
    <service-name>...</service-name>
    <service-type>...</service-type>
    <properties>
      ...//notions ...
    </properties>
  </service-description>
</agent-services>
</agent-description>
<agent-description> ...
</agent-description>
</search-reply>
</service-search>.

```

При поиске агентов формируются сообщения следующего типа:

```
<agent-search><name>AgentName</name></agent-search>
```

Ниже приводится пример конкретного сообщения:

```

<agent-search>
  <name>UIAgent</name>
  <name>ClassifierAgent</name>
  <name>ManagerAssistant</name>
</agent-search>

```

В ответ на такое сообщение агент может получить сообщение следующего содержания:

```

<agent-search>
  <search-reply>
    <agent-description>
      <agent-name>AgentName</agent-name>
      <agent-state>AgentState</agent-state>
      <locator>
        <transport-type>Type</ transport-type >
        <transport-specific-address>Address</ transport-specific-address>
        <transport-specific-property>Property</ transport-specific-property>
      </locator>
    </agent-description>
  </search-reply >
</agent-search>.

```

6. Примеры приложений

6.1. Распределенное обнаружение наземных объектов

В данном примере* рассматривается задача коллективного распознавания наземных объектов, в которой отдельные решатели взаимодействуют в P2P стиле. Предполагается, что система наблюдения наземной обстановки являет-

* Исходные данные, использованные в данном примере, сформированы искусственно на основании серий инфракрасных изображений реальных наземных сцен.

ся распределенной. Она состоит из множества сенсоров — инфракрасных видеокамер, установленных на беспилотных летательных аппаратах. На борту каждого летательного аппарата имеется множество агентов, каждый из которых обучен распознаванию объектов одного конкретного типа при его наблюдении под определенным ракурсом, а именно, или спереди, или сзади, или справа, или слева.

В поле зрения видеокамеры каждого летательного аппарата попадает определенная область земной поверхности, причем эти области для разных аппаратов могут быть пересекающимися. Таким образом, различные видеокамеры могут получать информацию об одних и тех же объектах, которые, как правило, в таких случаях будут наблюдаться ими под различными углами. Следовательно, одни и те же наземные объекты могут обнаруживаться различными агентами, которые могут быть расположены как на борту одного и того же летательного аппарата, так и на различных аппаратах.

В рассматриваемой модели каждый отдельный летательный аппарат интерпретируется как пир, который может связываться с соседними пирами (летательные аппараты) на основе беспроводной связи и обмениваться с ними

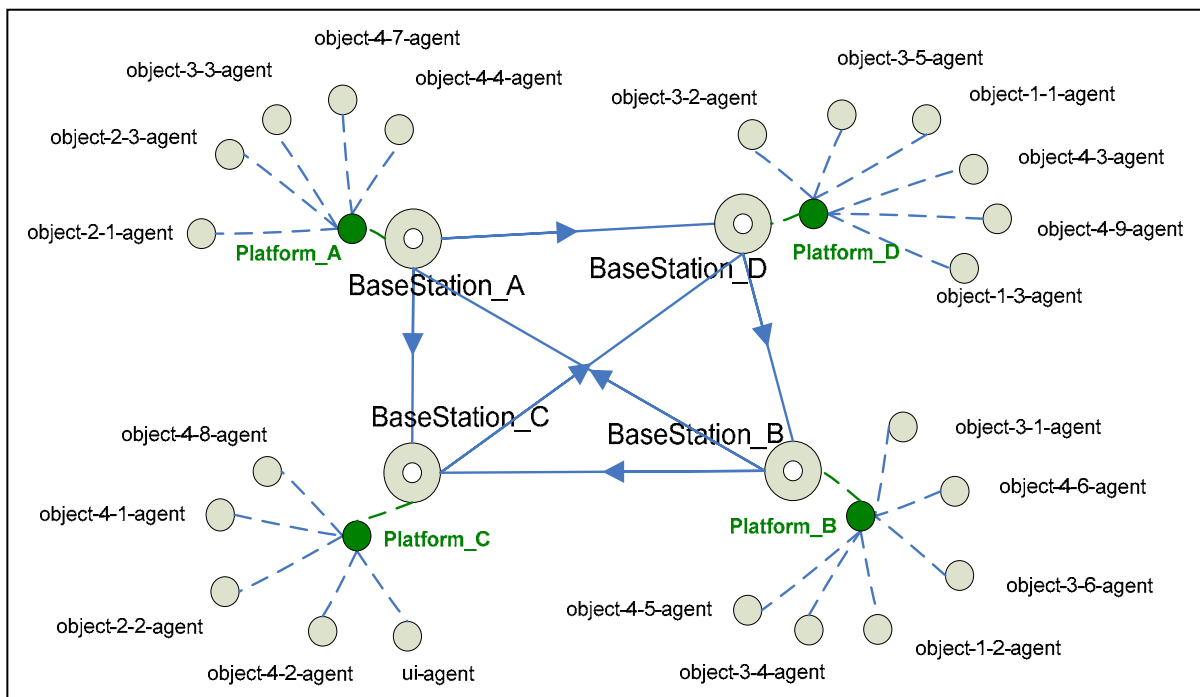


Рис. 7. Открытая P2P сеть агентов, ориентированная на сервис, предназначенная для распределенного распознавания наземных объектов.

информацией, а множество всех летательных аппаратов образуют P2P сеть.

Предполагается, что на каждом летательном аппарате установлен экземпляр агентской платформы. Все экземпляры агентской платформы (они имеют сервисы белых и желтых страниц) вместе с сервисами пиров, обеспечивающими P2P коммуникации в сети летательных аппаратов, реализуют «прозрачное» взаимодействие агентов различных летательных аппаратов в соответствии со схемой, представленной ранее на рис. 3.

Функциональность каждого агента по распознаванию наземных объектов определенного типа, если они наблюдаются под заданным ракурсом, рассматривается как сервис, который агент может использовать сам и предоставлять его по запросу любого другого агента P2P сети.

В примере, реализованном программно, P2P сеть состоит из четырех пиров (летательные аппараты), которые названы 'BaseStation' A, B, C и D соответственно. На рис. 7 представлена топология сети, которая отражает отношение «соседства» пиров (летательные аппараты) в некоторый момент времени, а также указывает состав агентов, установленных на экземпляре агентской платформы каждого пира. Всего в примере распределенной системы обработки данных распределенной системы наблюдения используются 22 агента*.

Сценарий взаимодействия агентов в процессе обработки информации следующий (предполагается, что желтые и белые страницы всех экземпляров P2P агентской платформы сети содержат необходимую информацию). Агент, когда он обнаруживает объект «своего» типа (т.е. объект, на обнаружение которого обучен агента классификатор), запрашивает информацию от других агентов, которые обучены распознаванию объектов такого же типа, обнаружили ли они в заданной области «свои» объекты, наблюдая тот же район, возможно, под другим ракурсом. После получения ответов агент объединяет полученные в ответ решения, используя метод голосования на основе простого большинства, и в случае положительного решения отправляет информацию агенту, реали-

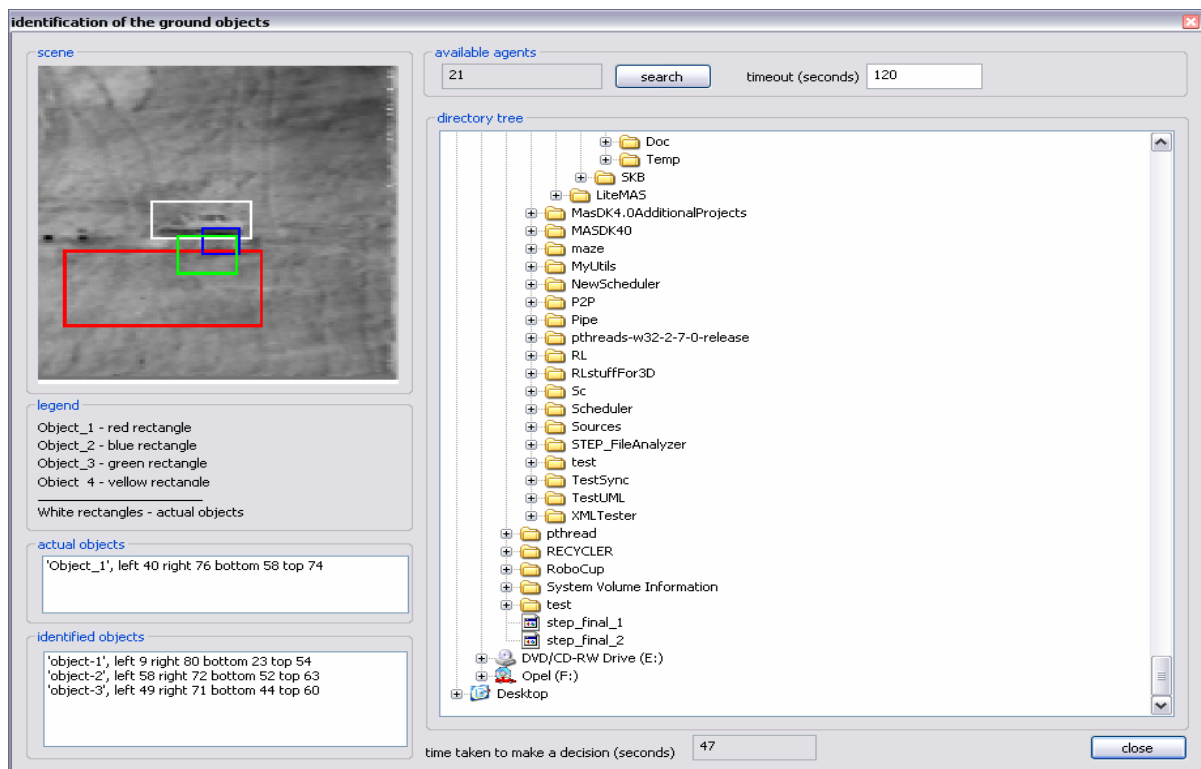


Рис. 8. Интерфейс пользователя, который отображает наблюдаемую область земной поверхности и объекты, обнаруженные различными компонентами системы наблюдения.

зующему интерфейс пользователя, который в свою очередь отображает полученные результаты на экране монитора. На рис. 8 нанесены четыре прямоугольника, внутри которых агенты предполагают наличие объекта.

Результаты численного моделирования процессов распределенного обнаружения наземных объектов описанной выше распределенной системы подтверждают, что даже при столь простом алгоритме объединения решений, как метод голосования на основе простого большинства, последний дает удовле-

* Среди них 21 агент предназначен для обнаружения наземных объектов. Это агенты-классификаторы, а ещё один агент реализует интерфейс пользователя (рис. 8).

творительные результаты по точности работы системы распределенной классификации в целом. Сами результаты здесь не приводятся и не анализируются в связи с тем, что этот аспект работы системы выпадает из основного содержания данной работы.

Отметим, что программный код, задающий структуру связей в P2P сети, распределение экземпляров агентских платформ и агентов по платформам был сгенерирован автоматически с помощью специального инструментального средства, которое было разработано авторами [8]. Краткие сведения об этом инструментарии приводятся в Приложении.

6.2. Предсказание появления течения El Nino

Течение El Nino периодически возникает в экваториальной зоне Тихого океана, и его появление обычно вызывает негативные природные последствия в экваториальной зоне американского континента. Так, в 1981–1982 гг. это течение создало серьезные проблемы по всему миру, при этом в некоторых странах (Перу, США) оно привело к разрушительным наводнениям. Для предсказания появления этого течения в середине 1980-х годов в экваториальной зоне Тихого океана от берегов Австралии и до берегов Америки была развернута глобальная сенсорная сеть, состоящая из 71 сенсора, установленного на буях. Каждый сенсор периодически измеряет несколько локальных параметров океана и атмосферы и передает эти данные в центр обработки, где выполняется централизованная обработка, имеющая целью обнаружить появление течения El Nino.

Целью представляемой далее программной разработки является демон-

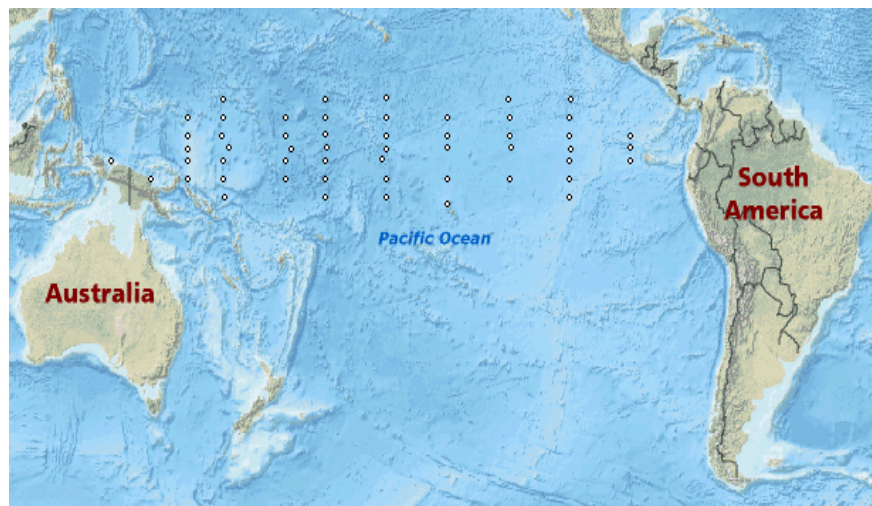


Рис. 9. Расположение буев в экваториальной зоне Тихого океана.

страция иного, более перспективного и эффективного подхода к решению данной задачи, в котором используется описанная выше парадигма открытых сетей агентов, взаимодействующих «через» разработанную распределенную P2P агентскую платформу и осуществляющих распределенную

обработку данных и принятие решений с использованием архитектуры, ориентированной на сервис. При этом предполагается, что в «центр» такая система периодически передает только результаты принятия решений.

Для эмуляции процессов функционирования этой системы используются данные массива ТАО [17], собранные с помощью экваториальной системы буев в период времени с 1980 по 1998 год.

Географическое распределение буев по поверхности Тихого океана представлено на рис. 9. Как уже указывалось, на каждом бую размещены сенсоры,

измеряющие положение буя, скорость и направление ветра, относительную влажность, а также температуры воздуха и поверхности океана.

В разработанной модели открытой сети агентов для распределенной обработки данных множество буев рассматривается как множество узлов (пиров) P2P сети, взаимодействующих друг с другом по радиоканалам. Каждый сенсор рассматривается как источник «сырых» данных для агента, расположенного на этом бую. Как и в предыдущем примере, множество пиров с установленными на них экземплярами P2P агентской платформы предоставляют агентам возможность «прозрачного» взаимодействия с агентами, расположенными на «сосед-

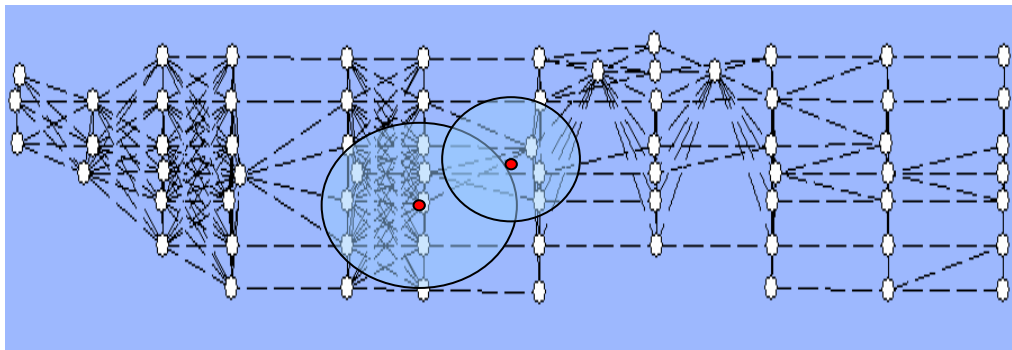


Рис. 10. Топология связей P2P сети, связывающей отдельные буи.

них» буях, т.е. на буях, находящихся в зоне достижимости радиосвязи. В такой системе каждый агент может обрабатывать не только данные, полученные со своих сенсоров, но также и данные, полученные сенсорами соседних буев и обработанных установленными на нем агентами. В разработанной модели предполагается, что мощность радиопередатчиков такова, что один буй (пир) может взаимодействовать с другими буями, удаленными на расстояние не более чем заданное значение, т.е. находящимися внутри круга некоторого фиксированного радиуса. Дополнительные ограничения по расстоянию могут быть наложены, например, исходя из экономии энергии батарей, питающих сенсоры и систему обработки.

На рис. 10 представлена структура P2P сети, которая использовалась в качестве модели, положенной в основу разработанной программной системы. На рисунке белыми кружками обозначены буи (пиры), а черными пунктирными линиями указаны существующие коммуникационные каналы между буями. Предполагалось, что максимальное расстояние, на котором способны связываться буи, равняется 1675 км. При таком предположении у каждого буя имеется от 5 до 16 соседей при общем количестве буев равном 71.

На данные, собранные с одного конкретного буя, могут влиять различные локальные возмущения, такие как неисправность сенсора, погодные аномалии; некоторые данные могут быть пропущены. Чтобы обеспечить устойчивость системы обработки данных, получаемых сенсорами буев, к таким возмущениям, в процессе своего функционирования каждый буй усредняет данные своих и соседних сенсоров. Это реализуется агентами в P2P стиле с использованием сервисов, предлагаемых агентами системы.

Система функционирует следующим образом. Один из агентов системы, играющий роль интерфейса пользователя и симулятора глобального (единого для сети сенсоров) времени, рассылает через свой P2P сервис всем агентам значение времени (дату) в текущий такт. Агенты на эту дату вычисляют среднее значение измеряемых параметров, взаимодействуя со своими соседями. После вычисления усредненных значений полученные данные отправляются агенту

визуализатору для отображения. Интерфейс пользователя представлен на рис. 11.

В заключение данного подраздела отметим, что в описанной программной системе пока не решалась собственно задача обнаружения течения El Niño, поскольку это требует более глубокого изучения имеющихся данных, построения алгоритмов обучения распознаванию ситуации, которая характеризуется пространственными и временными атрибутами, разработки самого метода обучения распознаванию таких ситуаций на основе парных коммуникаций агентов.

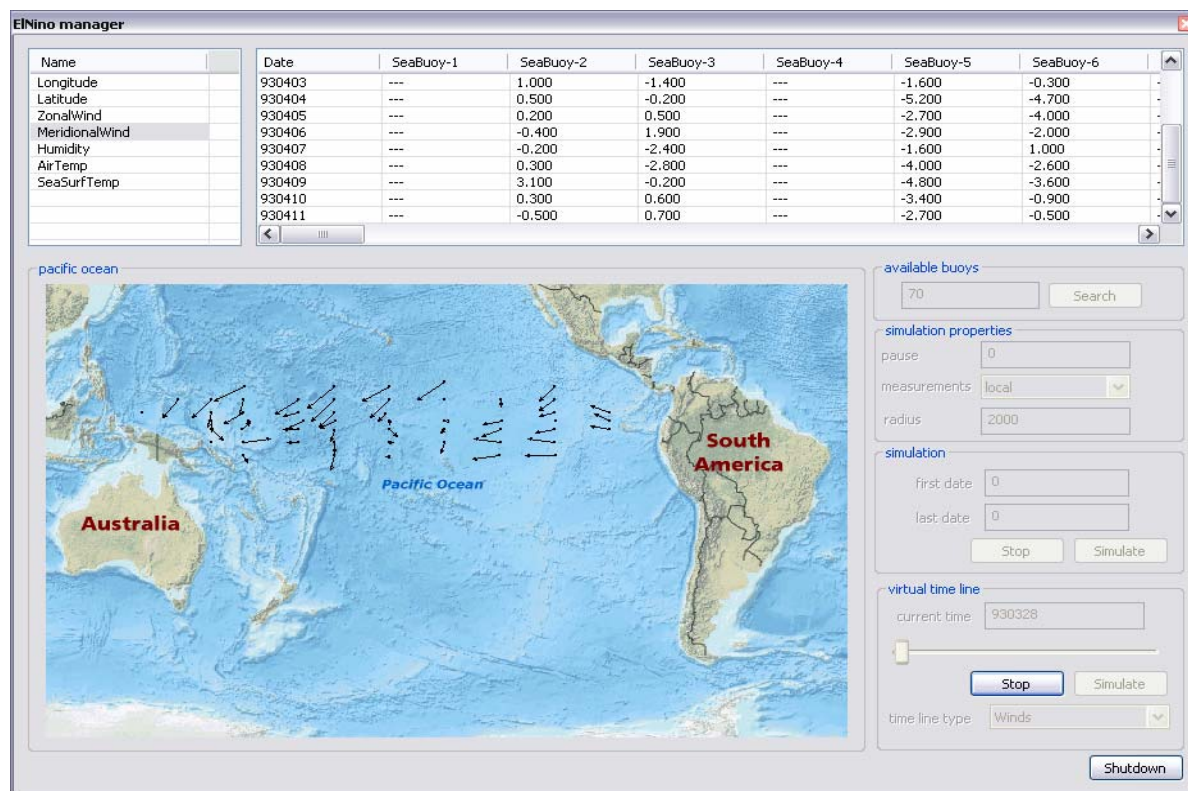


Рис. 11. Пользовательский интерфейс.

Готовых решений для такой задачи пока не существует, а потому эта проблема требует дополнительных исследований. Однако последняя задача и по своей сути выпадает из основного содержания данной работы.

7. Заключение

Основная проблема, решаемая в данной работе, — это архитектура и программная реализация открытых сетей агентов, взаимодействующих на основе парных коммуникаций (P2P коммуникации). В работе показано, что обеспечение открытости таких сетей может быть достигнуто за счет промежуточного программного обеспечения, которое называется агентской P2P платформой. Функциональная архитектура такой платформы, а также архитектура компонент, обеспечивающих ее взаимодействие с уровнем P2P провайдера и с уровнем прикладных агентов, была в общих чертах предложена рабочей группой FIPA NA WG 6 в начале 2005 года, однако ее детальная разработка и программная реализация впервые предложена в данной работе.

Агентская P2P платформа обеспечивает интеграцию агентских технологий, технологий, ориентированных на сервис, и P2P технологий. С одной сторо-

ны, в таких интегрированных системах приложения могут пользоваться функциональностью и возможностями P2P систем, такими как коммуникационные каналы между узлами сети, распределенный поиск узлов и их сервисов и т.п., через стандартные программные интерфейсы. С другой стороны, агентские системы, рассматриваемые как некоторое множество распределенных компонент, взаимодействующих друг с другом, оказываются способными к асинхронным вычислениям и функционированию в кооперативной и (или) конкурентной среде, могут эффективнее достигать желаемого результата. Стандартизированная P2P инфраструктура, используемая как коммуникационная среда для экземпляров агентских платформ, позволяет создавать распределенные приложения, расположенные на различных гетерогенных устройствах, таких, как компьютеры, сотовые телефоны, персональные компьютеры, карманные компьютеры и т.п. Она оказывается способной использовать для связи между узлами различные коммуникационные технологии, такие, как *TCP/IP*, *SOAP*, *SMTP*, *WIFI*, *BLUETOOTH*, одновременно поддерживая динамическую популяцию агентов, которые в любой момент времени могут покидать сеть и присоединяться к ней. Наконец, интеграция P2P принципов и агентской платформы позволяет создавать масштабируемые MAC, в которых число агентов может исчисляться тысячами и миллионами.

Разработанный программный прототип состоит из:

- 1) P2P провайдера, который реализован в виде узла, способного к взаимодействию с другими узлами;
- 2) агентской P2P платформы, которая реализована как приложение в среде Windows XP, способное управлять жизненным циклом агента и поддерживать сервисы белых и желтых страниц);
- 3) базового агента (generic agent), реализующего низкоуровневое взаимодействие с агентской P2P платформой.

Основные выводы работы подтверждены экспериментальной разработкой прикладных открытых сетей агентов, в частности интеллектуальной сенсорной сети агентов, решающей задачу обнаружения океанского течения El Nino, и распределенной системы наблюдения за земной поверхностью для кооперативного обнаружения объектов определенных классов. Разработанная платформа использовалась также и в ряде других приложений, которые не упомянуты в статье. К ним относится система обнаружения вторжений в компьютерную сеть, а также система управления воздушным движением в районе аэропорта в нестандартных и опасных ситуациях, в частности, когда в районе аэропорта появляется самолет, захваченный террористами.

Приложение. Генератор топологии сети

Генератор топологии сети представляет собой специальное программное средство (программный инструментарий), при помощи которого можно автоматически генерировать конфигурационные файлы для экземпляров агентских платформ и пиров с целью создания P2P сетей агентов требуемой топологии.

Работа пользователя при описании конфигурационных файлов включает в себя несколько шагов. На первом шаге специфицируется идентификатор будущей P2P сети агентов. Этот идентификатор фактически указывает имя каталога, где будут располагаться необходимые файлы (библиотеки агентов, возможно, конфигурационные файлы для пира, каталоги, где будут записываться логи, фиксирующие значения заданного множества атрибутов, используемые далее

для улучшения характеристик работы пира и самоорганизации сети в целом путем использования алгоритмов машинного обучения и т.п.). Имя этого каталога потребуется также при реализации различных задач поиска. На рис. 12 изображен соответствующий диалог.

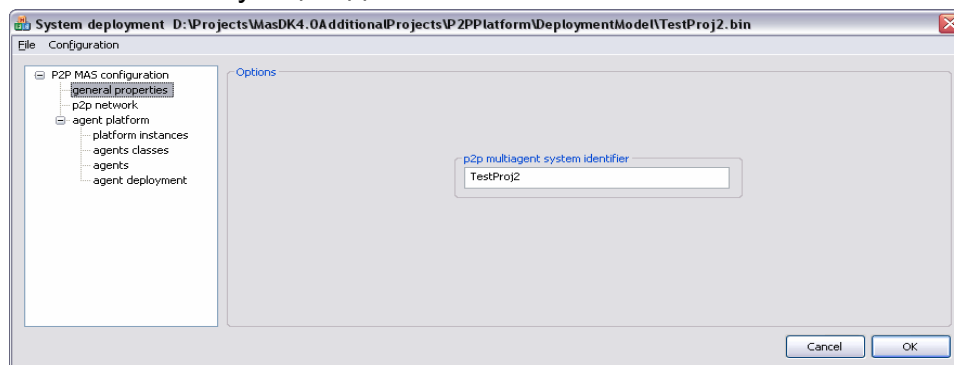


Рис. 12. Спецификация идентификатора P2P сети агентов.

На следующем шаге специфицируется топология P2P сети. При этом определяются пиры (узлы P2P сети), связи между узлами, а также локаторы каждого пира. Под локатором пира понимается адрес, т.е. спецификация протокола, по которому данный пир может быть доступен. Примерами адресов могут быть протоколы TCP/IP, UDP, SMTP и т.п. В текущей версии, имеется поддержка только TCP/IP протокола. Для спецификации адреса TCP/IP необходимо ввести имя хоста, имя порта и идентификатор пира. Рассматриваемое приложение может автоматически генерировать пиры на основании шаблона имени, а также автоматически генерировать топологию сети.

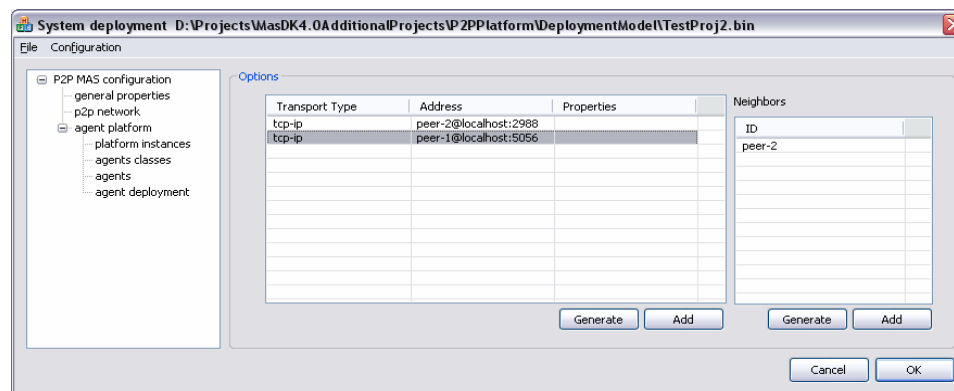


Рис. 13. Спецификация топологии P2P сети.

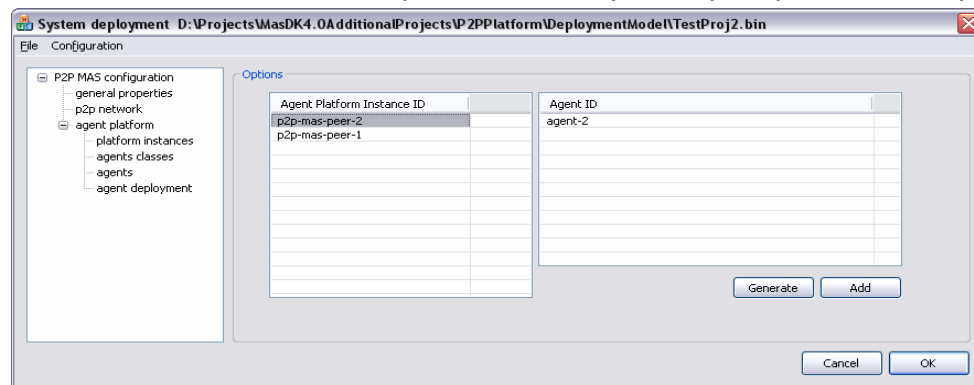


Рис. 14. Спецификация экземпляров агентских платформ.

При этом гарантируется, что сгенерированная сеть всегда будет сильно связанной, т.е. в ней всегда будет существовать путь от любого случайно выбранного пира к любому другому пиру. Соответствующий диалог показан на рис. 13.

* P2P сеть можно представить в виде орграфа (граф, в котором ориентированы все ребра). Пусть $G(V,E)$ — орграф, V_1 и V_2 — его вершины. Говорят, что две вершины V_1 и V_2 сильно связаны в орграфе G , если существует путь (ориентированная цепь) из V_1 в V_2 и из V_2 в V_1 . Если все вершины орграфа сильно связаны, то такой орграф называется сильно связанным.

На следующем шаге необходимо специфицировать топологию сети агентов. Под сетью агентов понимается множество экземпляров агентских платформ, расположенных на пирах, и прикладных агентов, установленных поверх экземпляров агентских платформ. Соответственно задачей этого шага является спецификация множества экземпляров агентских платформ и их распределение по пирам. В текущей версии генератора поддерживается исключительно автоматическая генерация. Это означает, что для каждого пира будет генерироваться ровно один экземпляр агентской платформы.

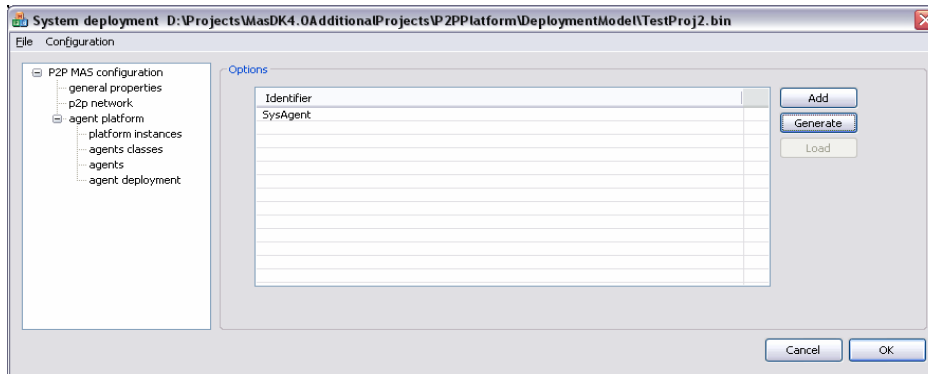


Рис. 15. Спецификация класса агента.

При этом идентификатор каждого экземпляра будет состоять из последовательности 'p2p-mas-', к которой должен добавляться идентификатор пира, например *p2p-mas-peer1*. Соответствующий диалог представлен на рис. 14.

На следующих шагах выполняется спецификация агентов и указывается их распределение по экземплярам агентских платформ. Каждый агент специфицируется указанием его класса и уникального идентификатора. Класс агента, или его тип, это понятие, которое широко используется в технологии MAS. Класс агента определяется множеством ролей, которые он может выполнять в различных протоколах работы много-агентной системы.

Идентификатор агента одно-значно задает агента в сети агентов. В

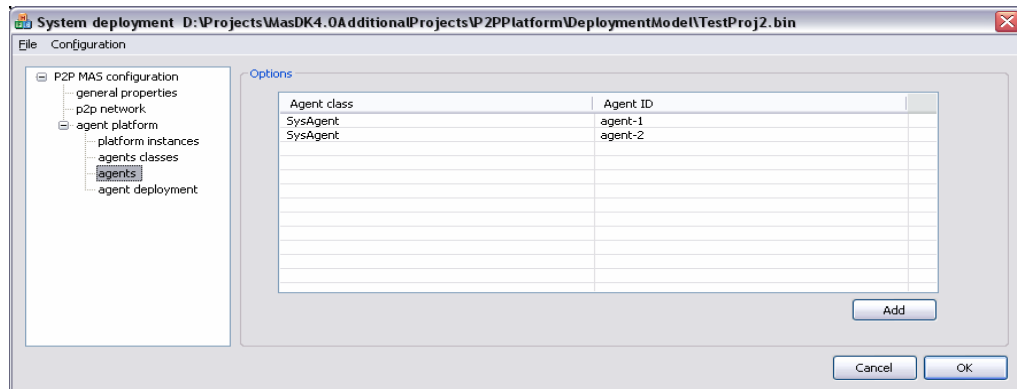


Рис. 16. Спецификация агентов MAS.

дальнейшем имя класса агента используется для создания его экземпляров в сети, т.е. для создания конкретных программных агентов.

Правило задания идентификатора агента простое: библиотека соответствующего агента должна иметь имя 'agent-class.dll' и соответственно находиться в определенном каталоге, а именно в одном из подкаталогов каталога с идентификатором проекта сети агентов. Соответствующие диалоги для спецификации класса агента и агентов показаны на рис. 15 и 16. Для добавления нового экземпляра агента необходимо указать его класс из введенного ранее списка и указать идентификатор (имя) агента.

На последнем этапе определяется распределение агентов по экземплярам агентских платформ. Это распределение может выполняться как в ручном, так и в автоматическом режиме. В ручном режиме для каждого экземпляра агентской платформы необходимо указать множество агентов, которые будут располагаться на ней. В автоматическом режиме агенты равномерно распре-

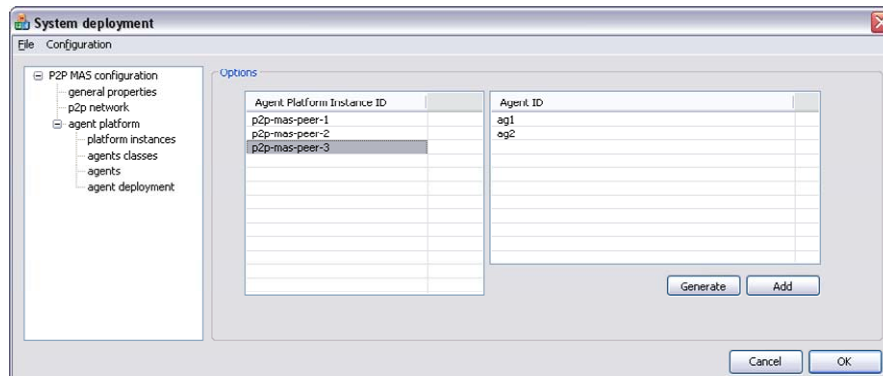


Рис. 17. Спецификация распределения агентов MAS.

деляются по платформам. Соответствующий диалог показан на рис. 17.

В результате генерации для каждого экземпляра платформы будут созданы два конфигурационных файла, определяющих то-

пологию сети агентов и ее компоненты. Один из конфигурационных файлов задает параметры самой платформы, а другой задает параметры пира, к которому привязан соответствующий экземпляр агентской платформы.

Заметим, что оба приложения, описанные выше в данной работе, а также ряд других приложений, построенных как открытые сети P2P агентов, разрабатывались с использованием генератора, описанного в данном приложении.

Список литературы

1. Abu-Draz S., Shakshuki E. Peer to Peer Multi Agent System for Online Trading // LNCS. 2003. No. 2671. С. 611–613.
2. Androutsellis-Theotokis S., Spinellis D. A Survey of Peer-to-Peer Content Distribution Technologies // ACM Computing Surveys. 2004. Vol. 36, no. 4. P. 335–371.
3. Bertolini D., Busetta P., Nori M., Perini A. Peer-to-peer Multi Agent Systems Technology for Knowledge Management Applications. An Agent-Oriented Analysis // WOA. 2002. P. 1–6.
4. Carrillo-Ramos A., Gensel J., Villanova-Oliver M., Martin H. A Peer Ubiquitous Multi Agent Framework for Providing Nomadic Users with Adapted Information // LNCS. 2006. No. 4118. P. 159–172.
5. FIPA Agent Communication Language Specification [Электронный ресурс] // <<http://www.fipa.org/specs/fipa00061/>> (по состоянию на 15.05.2007).
6. FIPA Agent Discovery Service Specification [Электронный ресурс] // <<http://www.fipa.org/specs/fipa00095/>> (по состоянию на 15.05.2007).
7. FIPA Agent Management Specification [Электронный ресурс] // <<http://www.fipa.org/specs/fipa00023/>> (по состоянию на 15.05.2007).
8. FIPA Agent Message Transport Service Specification [Электронный ресурс] // <<http://www.fipa.org/specs/fipa00067/>> (по состоянию на 15.05.2007).
9. FIPA web site [Электронный ресурс] // <<http://www.fipa.org/>> (по состоянию на 15.05.2007).
10. FIPA P2P NA WG6. Functional Architecture Specification Draft 0.12 [Электронный ресурс] // <<http://www.fipa.org/subgroups/P2PNA-WG-docs/P2PNA-Spec-Draft0.12.doc>> (по состоянию на 15.05.2007).
11. FIPA P2P Nomadic Agents Working Group (P2PNA WG6) [Электронный ресурс] // <<http://www.fipa.org/subgroups/P2PNA-WG.html>> (по состоянию на 15.05.2007).
12. Gorodetsky V., Karsaev O., Samoilov V., Serebryakov S. Agent-based Service-Oriented Intelligent P2P Networks for Distributed Classification // International Conference "Hybrid Information Technologies" (ICHIT-2006). 2006. P. 224–233.

13. *Gorodetsky V., Karsaev O., Samoilov V., Serebryakov S.* Agent-based Service-Oriented P2P Information Fusion // Submitted to the International Conference Fusion 2007.
14. *Kungas P., Matskin M.* Semantic Web Service Composition through a P2P-based Multi Agent Environment // LNCS. 2006. No. 4118. P. 106–119.
15. *Lin N., Marzullo K., Masini S.* Gossip versus Deterministic Flooding: Low Message Overhead and High Reliability for Broadcasting on Small Networks. Technical report CS1999-0637 [Электронный ресурс] // <<http://citeseer.ist.psu.edu/563854.html>> (по состоянию на 15.05.2007).
16. *Mine T., Kogo A., Amamiya M.* ACP2P: Agent Community Based Peer-to-Peer Information Retrieval — an Evaluation // LNCS. 2006. No. 4118. P. 145–158.
17. Pacific Marine Environmental Laboratory [Электронный ресурс] // <<http://www.pmel.noaa.gov/tao/>> (по состоянию на 15.05.2007).
18. *Panti M., Penserini L., Spalazzi L., Tacconi S.* A Multi Agent System Based on the P2P Model to Information Integration. Proposal to Agentcities Task Force [Электронный ресурс] // <http://sra.itc.it/people/penserini/documents/LNCS_proposal_38.pdf> (по состоянию на 15.05.2007).
19. *Patkos T., Plexousakis D.* A Semantic Marketplace of Negotiating Agents // LNCS. 2006. No. 4118. P. 94–105.
20. Peer-to-Peer Research Group [Электронный ресурс] // <<http://www.irtf.org/charter?gtype=rg&group=p2prg>> (по состоянию на 15.05.2007).
21. *Purvis M., Nowostawski M., Cranefield S., Oliveira M.* Multi Agent Interaction Technology for Peer-to-Peer Computing Environment // LNCS. 2004. No. 2872. P. 150–161.
22. *Tate A.* The "Helpful Environment": Geographically Dispersed Intelligent Agents That Collaborate // IEEE Intelligent Systems. 2006. P. 57–61.
23. UPNP Forum [Электронный ресурс] // <www.upnp.org> (по состоянию на 15.05.2007).
24. *Wang M., Wolf H., Purvis M., Purvis M.* An Agent-Based Collaborative Framework for Mobile Peer-to-Peer Applications // LNCS. 2006. No. 4118. P. 132–144.
25. *Yu B., Singh M.* Incentive Mechanism for Agent Based Peer-to-Peer Systems // LNCS. 2004. No. 2872. P. 77–88.
26. *Zhang H., Lesser V.* Multi Agent Based Peer-to-Peer Information Retrieval Systems with Concurrent Search Sessions // ACM. 2006. P. 305–312.

ПОДХОД К ПОСТРОЕНИЮ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ ИНТЕЛЛЕКТУАЛЬНОЙ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В ОТКРЫТОЙ ИНФОРМАЦИОННОЙ СРЕДЕ*

А. В. СМЕРНОВ¹, М. П. ПАШКИН, Н. Г. ШИЛОВ, Т. В. ЛЕВАШОВА

Санкт-Петербургский институт информатики и автоматизации РАН

СПИИРАН, 14-я линия ВО, д. 39, Санкт-Петербург, 199178

¹<smir@iias.spb.su>

УДК 658.519.011.56

Смирнов А. В., Пашкин М. П., Шилов Н. Г., Левашова Т. В. Подход к построению распределенной системы интеллектуальной поддержки принятия решений в открытой информационной среде // Труды СПИИРАН. Вып. 4, — СПб.: Наука, 2007.

Аннотация. Описывается подход к интеллектуальной поддержке принятия решений, основанный на интеграции информации и знаний в контекст, описывающий задачу пользователя или требуемую ситуацию и учитывающий динамику окружающей среды. Используются контексты двух типов: абстрактный и оперативный. Задача пользователя моделируется объектно-ориентированной сетью ограничений и решается как динамическая задача удовлетворения ограничений. Совместимость с онтологической моделью представления знаний обеспечивается применением формализма объектно-ориентированных сетей ограничений для описания знаний. В основе подхода лежит двухуровневая схема поддержки принятия решений, состоящая из подготовительного и основного уровней. Для реализации подхода предполагается использовать технологию Web-сервисов. — Библ. 24 назв.

UDC 658.519.011.56

Smirnov A. V., Pashkin M. P., Shilov N. G., Levashova T. V. An approach to building distributed intelligent support system on open information environment // SPIIRAS Proceedings. Issue 4, — SPb.: Nauka, 2007.

Abstract. The paper describes an approach to intelligent decision support. The approach is based on integration of information and knowledge into context that models a problem to be solved or a situation to be presented to a decision maker. The context takes into account dynamic of the environment. Two types of context are used: abstract and operational. The problem to be solved is modelled by a means of object-oriented constraint networks and interpreted as a temporal constraint satisfaction problem. Compatibility of the problem model with ontology model is ensured through application of formalism of the object-oriented constraint networks for knowledge representation. The approach relies upon a two-level framework of decision support. The framework has two distinct main phases: the preparatory phase and the decision making phase. The approach is offered to be implemented using the technology of Web-Services. — Bibl. 24 items.

1. Введение

Целью систем поддержки принятия решений (СППР) является обеспечение пользователя данными, знаниями и моделями для анализа и решения текущих проблем [2, 3]. В современных условиях СППР сталкивается с необходимостью управления большими объемами знаний, обработки информации и знаний, предоставляемых динамически изменяющимися источниками, а также с необходимостью персонифицированной поддержки пользователей.

Процесс принятия решений охватывает три уровня решений: стратегический, тактический и оперативный. Решения стратегического уровня направлены

* Работа выполнена при финансовой поддержке РФФИ (проект № 05-01-00151), Президиума РАН (проект № 2.35) и ОИТВС РАН (проект № О-1.9)

на разработку общих направлений и определение долгосрочных целей и задач. Данные решения основываются на длительных отношениях между объектами, вовлекаемыми в процесс принятия решений. Решения тактического уровня формируются в рамках стратегических решений. В первую очередь данные решения касаются проблемы наиболее эффективного использования имеющихся ресурсов. Оперативные решения затрагивают действия, выполняемые в данный момент. Задачи, ресурсы и цели таких действий устанавливаются на стратегическом и тактическом уровнях. При генерации и принятии оперативных решений используются повторяющиеся, хорошо специфицированные процедуры [14] и решаются конкретные задачи [5].

СППР сильно связаны с окружающей средой. Они предполагают использование своевременной и точной информации (данных). Используемые данные предоставляются расположенными в окружающей среде источниками информации. В связи с этим в подходе используется модель, учитывающая отношения между СППР и окружающей средой. Окружающая среда в данной модели представляется набором данных, которые обрабатываются как информация.

Для организации интеллектуальной поддержки принятия решения недостаточно простого владения информацией. Необходимо, чтобы эта информация была интерпретируемой и могла быть включена в условие задачи или сценарий. Интерпретируемость информации обеспечивается прописыванием ее семантики. Это привело к тому, что современные СППР, предполагающие интеллектуальную поддержку, строятся как системы, основанные на знаниях [1].

2. Интеллектуальная поддержка принятия решений: контекстно-ориентированный подход

Специфика интеллектуальной поддержки принятия решений по отношению к информации и знаниям, используемым при генерации решения, заключается в том, что они должны быть релевантными текущей задаче. По отношению к пользователю (лицо, принимающее решения) интеллектуальная поддержка принятия решений должна предоставлять пользователю возможность определять, на основании чего было сгенерировано то или иное решение, и возможность оценивать альтернативные решения.

Для того чтобы учесть динамику источников информации рассматриваемый подход является контекстно-ориентированным подходом к формированию модели задачи. Для процессов, поддерживаемых информационными технологиями, контекст определяется как информация, которая может быть использована для описания ситуации, в которой находится в данный момент некоторый объект [13]. В общем случае контекст используется для определения, какая информация является релевантной в той или иной ситуации (для той или иной задачи). Учитывая направленность на интеллектуальную поддержку в рамках описываемого подхода контекст включает в себя не только информацию, но и знания, являющиеся релевантными задаче.

Предлагается контекст задачи описывать контекстами двух типов — абстрактным и оперативным. *Абстрактный контекст* представляет собой модель задачи, построенную на основании интеграции знаний проблемной области, релевантных для данной задачи. *Оперативный контекст* является экземпляром абстрактного контекста с приписанными текущими значениями переменных, входящих в модель задачи.