УДК 004

Г. П. Юлейси, И. И. Холод
*Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)*

# Взаимодействие в многоагентных системах интеллектуального анализа данных

*Описываются основы агентов и многоагентных систем. Обсуждается работа связи между многоагентными системами, использующими интеллектуальный анализ данных. Представлен обзор языков, которые обеспечивают связь между агентами в распределенных средах. Исследование позволяет определить оптимальное предложение, которое будет использоваться с учетом того, что потребность во взаимодействии между компонентами системы становится все более важной для совместных задач, решение которых по отдельности было бы очень дорого или даже невозможно. Подробно проиллюстрирован состав платформы для создания и взаимодействия агентов, структура канала связи и передача сообщений между агентами. Анализируются архитектуры и наиболее часто используемые языки межагентского взаимодействия. Подробно описаны аспекты модели PMML и алгоритм передачи этой модели между агентами в ACL.*

**Агент, многоагентные системы, PMML, FIPA ACL**

В настоящее время наблюдается развитие агентных систем, возникших в результате технической эволюции информационных и программно-аппаратных средств современной инфосферы. Для эффективного использования агентных приложений необходимо освоить методологию и инструменты их проектирования и эксплуатации [1].

Взаимодействие – одна из наиболее важных характеристик агентов. Агенты взаимодействуют, чтобы совместно использовать информацию, выполнять задачи, достигать общих целей. Способность общаться – свойство, которое выгодно отличает агентские от других технологий. Механизмы связи накладывают дополнительные ограничения на внутреннюю архитектуру агента, которая должна обеспечивать эффективное их использование [2].

Агентские технологии представляют новую парадигму, которая ориентирована на агенты и позволяет улучшать современные приложения. Одной из областей ее применения является интеллектуальный анализ данных (Data Mining) [3]. Это связано с тем, что три области, в которых агенты нашли широкое применение, соответствуют задачам, решаемым в системах Data Mining: интеллектуальные помощники, сборщики информации и элементы автоматического управления [4].

В системах Data Mining для обмена моделями, построенными в результате анализа данных, используется язык описания прогнозных моделей (PMML – Predictive Model Markup Language) [5]. Модели, описанные на этом языке, используются как для ассемблирования результатов анализа, так и для обмена ими в случае распределенного анализа данных.

В данной статье описывается объединение языка взаимодействия агентов (ACL – Agent Communication Language) и PMML для организации взаимодействия в многоагентных системах используемых для распределенного интеллектуального анализа данных.

**Многоагентные системы.** Многоагентные системы (МАС) включают в себя вычислительные объекты, которые реализуют заданное поведение для решения общей задачи, например анализа данных. В МАС каждый информационный объект называется агентом [6].

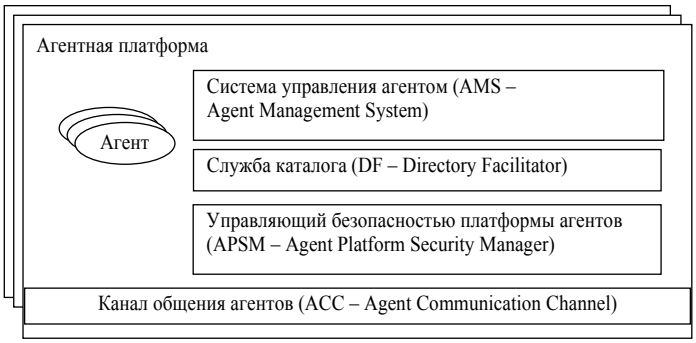


*Рис. 1*

**Агент.** Агент – это программный объект, способный воспринимать среду, в которой он функционирует, через датчики и действующий в этой среде с использованием эффекторов. Он автономен, потому что контролирует свое поведение сам и может действовать без вмешательства других агентов или людей. В настоящее время агенты имеют очень широкую область применения и существуют различные их типы (реактивный, совещательный, умный, совместный и т. д.), которые, в свою очередь, ориентированы на решение разных задач.

С целью объединения усилий и стандартизации в области разработки многоагентных систем была создана международная организация FIPA (Foundation for Intelligent Physical Agents), специализирующаяся на продвижении индустрии интеллектуальных агентов, разработки открытых спецификаций, которые поддерживают функциональную совместимость между агентами. В последние годы стандарты FIPA стали стандартами де-факто, используемыми разработчиками МАС в компьютерном сообществе и за его пределами. В 2005 г. FIPA была официально принята в качестве комитета по стандартам IEEE Computer Society. Эти стандарты влияют не только на методы взаимодействия между агентами, но и на базовую архитектуру (рис. 1), которую должна реализовывать МАС, соответствующая данному стандарту. Согласно архитектуре FIPA, средой существования агентов служат агентные платформы.

Агентная платформа – это промежуточное программное обеспечение, поддерживающее создание, интерпретацию, запуск, перемещение и уничтожение агентов. Она включает в себя следующие компоненты [7]–[9]:

– система управления агентом (AMS – Agent Management System) управляет созданием, удалением, деактивацией, возобновлением, аутентификацией и миграцией агентов, находящихся на платформе. Она обеспечивает сервис «белых страниц», который хранит текущее местонахождение МАС;

– служба каталога (DF – Directory Facilitator) реализует сервис «желтых страниц», который хранит описание агентов;

– менеджер безопасности платформы агентов (APSM – Agent Platform Security Manager) отвечает за осуществление политики безопасности на транспортном уровне и проверку выполнения операций управления агентами;

– канал связи агентов (ACC – Agent Communication Channel) использует информацию, предоставляемую системой управления агентов для маршрутизации сообщений между агентами.

Несколько агентных платформ может размещаться как на одной, так и на нескольких машинах. Все общение между агентными системами осуществляется через канал общения (рис. 1).

Для обеспечения функционирования такой системы необходима стандартизация языка взаимодействия между агентами. Для решения этой задачи в рамках FIPA был предложен язык взаимодействия агентов ACL [2].

Язык FIPA ACL определяет контент и онтологию взаимодействия, а также набор базовых концепций, используемых в сообщениях между взаимодействующими агентами. Онтология здесь выступает синонимом понятия «программный интерфейс приложения» (API – Application Programming Interface), определяя конкретный интерфейс интеллектуальных агентов.

FIPA ACL основан на теории речевых актов: сообщения предназначены для выполнения некоторых действий на основании содержимого сообщения. Он состоит из набора типов сообщений и описания их прагматики.

В отличие от таких средств, как RPC, RMI, CORBA, обеспечивающих обмен информацией между приложениями, FIPA ACL имеет более сложную семантику. В сравнении с ними FIPA ACL обладает следующими преимуществами:

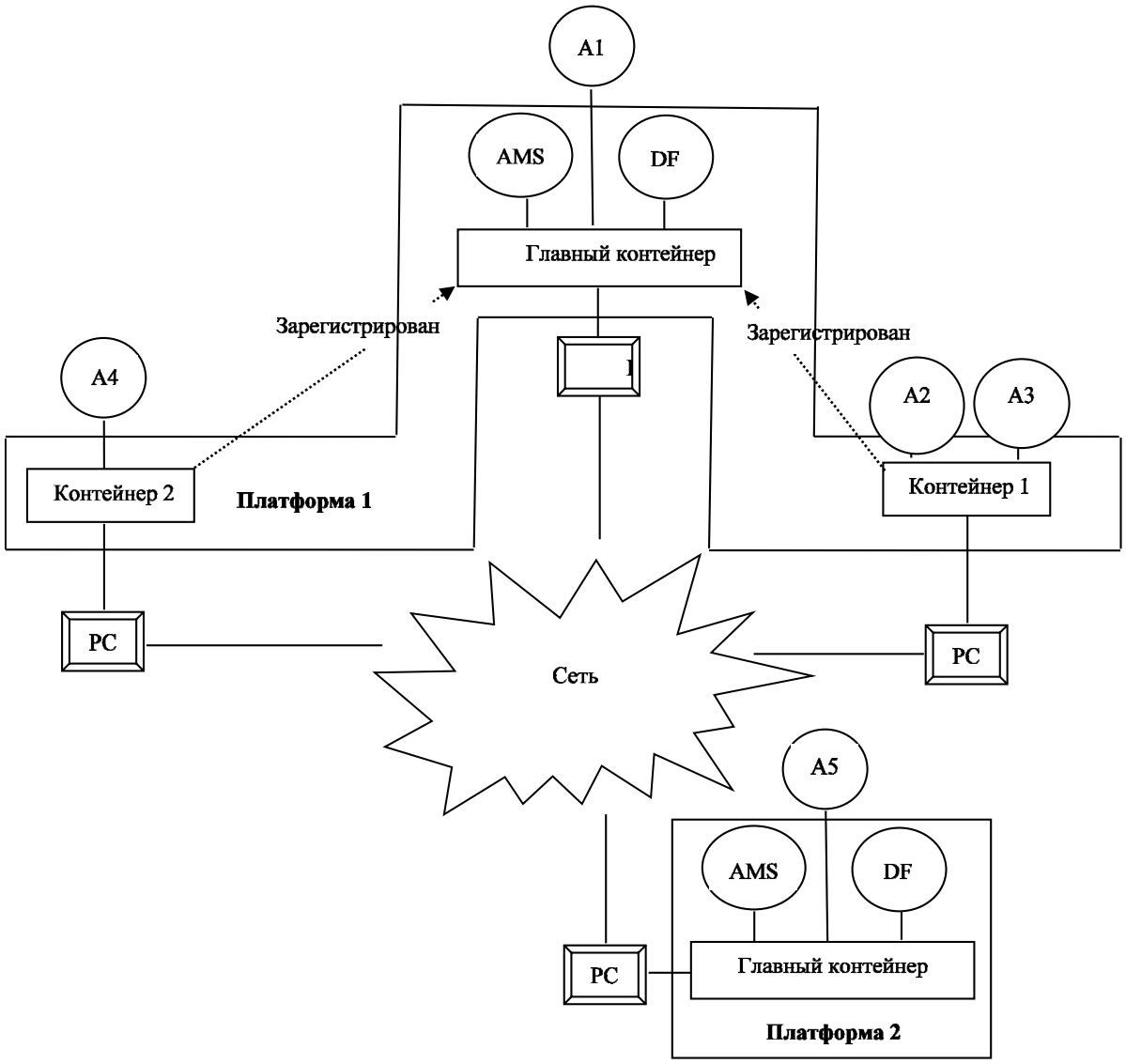


*Рис. 2*

– управляет суждениями, правилами, действиями, а не семантически не связанными объектами;

– описывает ожидаемое состояние, а не процедуру или метод.

Формат FIPA ACL сообщений включает следующие поля:

– отправитель сообщения;

– список получателей;

– коммуникативные действия («пожелания»), указывающие цель отправки сообщения:

– REQUEST (запрос), если отправитель желает, чтобы получатель произвел некоторое действие,

– INFORM (информирование), если отправитель желает, чтобы получатель был извещен о некотором факте,

– QUERY_IF, если отправитель желает знать, достигнуто или нет заданного условия,

– CFP (извещение о предложении), PROPOSE (предложение), ACCEPT_PROPOSAL (принятие предложения), REJECT_PROPOSAL (отклонение предложения), если отправитель и получатель ведут переговоры,

– другие типы коммуникативных действий.

Были разработаны программные реализации агентных платформ, каждая из которых имеет свои особенности, достоинства и недостатки. Одной из реализаций стандарта FIPA является проект с открытым кодом (JADE – Java Agents Development Framework) – настраиваемая стандартная среда для разработки распределенных мультиагентных приложений. В JADE интеллектуальный агент может быть определен как вычислительный объект, способный решать прикладные задачи в динамических и открытых средах [10].

Платформа JADE – распределенная, представляет собой набор контейнеров (рис. 2).

Контейнером называется динамическая среда исполнения мультиагентных приложений, в которой содержатся агенты – один или несколько. Набор активных контейнеров называется платформой. Один контейнер всегда главный, остальные связываются с ним и регистрируются в момент запуска, поэтому первым запускаемым контейнером при старте платформы должен быть главный, а остальные контейнеры должны быть «обыкновенными» (т. е. неглавными) и должны заранее «знать», как найти главный контейнер, на котором они будут регистрироваться, – т. е. должны иметь данные о хосте и порте [1].

Главная особенность JADE – это наличие AMS, которая управляет созданием, удалением, деактивацией, возобновлением и перемещением агентов на платформе, а также реализует сервис «белых страниц» для всех агентов, находящихся на ней, который хранит информацию о текущем местонахождении агентов.

В соответствии со стандартом FIPA, платформа системы JADE включает в себя службу каталога DF и канал связи агентов ACC. Служба каталога реализует сервис «желтых страниц» и хранит информацию об агентах, действующих в системе. Канал общения агентов управляет маршрутизацией сообщений между агентами, используя информацию от AMS.

Агент в JADE может взаимодействовать с другими агентами посредством асинхронной передачи FIPA ACL-сообщений с помощью системы доставки сообщений (MTS – Message Transport System). При отправке сообщения указывается ID получателя. Платформа (точнее ACC) сама определяет текущее местонахождение агентов и более подходящий механизм транспортировки. Все получаемые сообщения ставятся в очередь, доступ к которой реализуется агентской платформой. Следовательно, сообщения будут доставлены агенту, даже если он в момент получения запроса находится в состоянии ожидания. Если в процессе взаимодействия один из агентов перемещается, то связь между агентами теряется.

При этом внутри платформы для ускорения работы сообщения пересылаются в виде Java-объектов, а при обмене между платформами (или с другими MAC) – в виде XML-строки. При отправке сообщения оно попадает в «почтовый ящик» (очередь сообщений) агента-адресата, о чем его уведомляют. Сообщение может быть получено из «почтового ящика» агента и обработано любым из его режимов. Для выбора нужного типа сообщения из очереди программист может использовать фильтры. Пример обмена сообщениями показан на рис. 3 [1].

На техническом уровне коммуникация между агентами в JADE происходит за счет передачи сообщений с использованием какого-либо транспортного протокола нижнего уровня (SMTP, TCP/IP, HTTP, IIOP).

Сообщение в JADE реализуется как объект класса jade.lang.acl.ACLMessage, предоставляющего методы get() и set() для установки значений всех полей при получении и отправке сообщений.

Чтобы отправить сообщение другому агенту, необходимо заполнить поля объекта ACLMessage и вызвать метод send() класса Agent. Сразу после доставки сообщений JADE автоматически помещает их в очередь сообщений получателя. Агент может получать сообщения из своей очереди сообщений с помощью метода receive(), который возвращает первое сообщение, находящееся в очереди (и удаляет его оттуда), либо ничего не возвращает, если очередь сообщений пуста, после чего сразу же завершается.

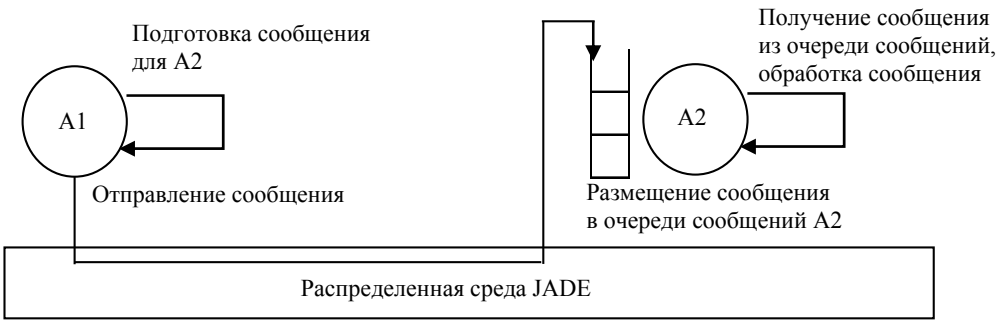


*Рис. 3*

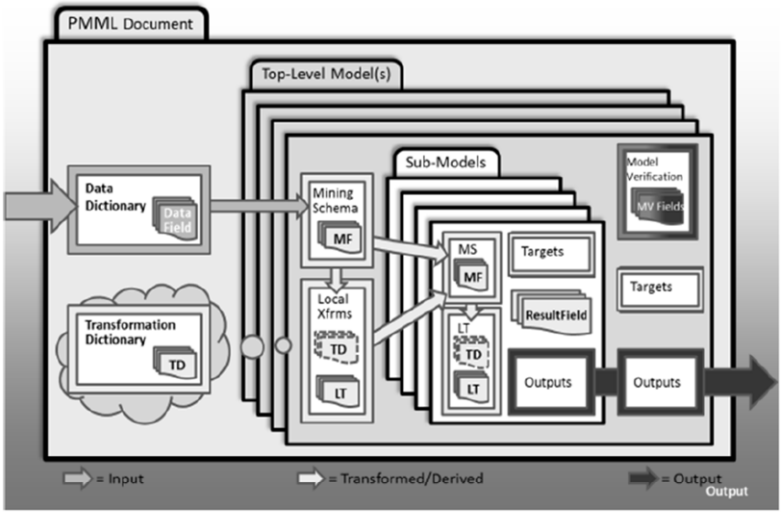


*Рис. 4*

Учитывая циклическое поведение агента и неблокирующий метод receive(), агент по имени agenteReceptor будет постоянно активировать и деактивировать заданное поведение, пока не поступит сообщение. Процесс может потреблять большое количество ресурсов процессора – чтобы избежать этого, метод action() поведения OfferRequestsServer следует выполнять лишь тогда, когда появится новое сообщение. Рекомендуется использовать метод block(). Тогда поведение, из которого он вызывается, удаляется из очереди активных поведений и помещается в очередь заблокированных. Как только сообщение поступает в очередь агента, все заблокированные поведения вновь вводятся в набор действий соответствующего агента и снова становятся активными.

**PMML.** Этот язык может использоваться для определения моделей и ансамблей моделей, построенных в результате интеллектуального анализа данных. Он предоставляет гибкий механизм определения схемы для них и поддерживает их выбор и усреднение при помощи нескольких моделей. PMML оказался полезным для приложений, требующих ансамблевого, многораздельного и распределенного обучения. Кроме того, таким образом облегчается перемещение прогностических моделей между приложениями и системами (рис. 4) [11].

PMML основан на XML и стал де-факто стандартом для представления не только результатов анализа, но также предварительной и последующей обработки данных. При этом он позволяет обмениваться моделями между различными инструментами и средами, в основном избегая проблем, связанных с несовместимостями.

**Передача PMML-модели между агентами в ACL-сообщении.** Языки взаимодействия агентов ACL и описания моделей PMML основаны на языке разметки XML, что позволяет их интегрировать. При этом язык ACL будет инкапсулировать в себе PMML-модели, построенные при анализе данных.

Далее опишем шаги формирования и передачи PMML-модели между агентами в ACL-сообщении:

1. Сгенерировать модель PMML.

2. Сохранить модель PMML в файл .xml.

3. Создать метод чтения xml-файла (модель PMML).

4. На платформе JADE создать агент-отправитель (отправить сообщение) и агент-получатель (получить сообщение).

5. Сформировать FIPA ACL-сообщение, в котором вызвать метод для чтения xml-файла (модель PMML) и передать это значение агенту-отправителю.

6. Передать сформированное FIPA ACL-сообщение, в качестве выходных данных функции.

Огромный потенциал многоагентных систем заключается в их способности взаимодействовать друг с другом на платформах, размещенных на разных узлах. Для их взаимодействия стандрат FIPA предлагает язык общения ACL. Таким образом, агенты, реализованные в системах, поддерживающих стандарт FIPA, могут обмениваться информацией. В многоагентных системах, используемых для интеллектуального анализа дан-

ных, такой информацией являются знания, извлекаемые из данных. Они описываются в виде моделей на языке PMML.

Учитывая то, что оба языка основаны на языке разметки XML, они легко могут быть интегрированы. В статье предложен метод интеграции PMML-моделей в ACL-сообщения системы интеллектуальных агентов JADE, которая реализует стандарт FIPA.

## СПИСОК ЛИТЕРАТУРЫ

1. Симонова Е. В. Использование платформы JADE для разработки мультиагентных приложений. Самара: Изд-во Самарского ун-та, 2017. С. 10–12.

2. Михайличенко В. В. Реализация модели коммуникации программного агента в мультиагентных системах / Запорожский гос. ун-т. Запорожье, 2000. С. 18–23.

3. Cao L., Weiss G., Yu P. S. A Brief Introduction to Agent Mining. Springer, 2012.

4. Peña J. M. Arquitectura distribuida de control para sistemas con capacidades Data Mining. Madrid: Universidad Politécnica de Madrid, 2001. P. 111–113.

5. PMML: An open standard for sharing / A. Guazzelli, M. Zeller, W. C. Lin, G. Williams // The R. J. 2009. Vol. 1, № 1. P. 2–5.

6. Balaji P. G., Srinivasan D. An introduction to multiagent systems. Springer-Verlag Berlin Heidelberg, 2010. P. 3–5.

7. Mateus S. S. Modelo de un Entorno Virtual Inteligente basado en la percepción y el razonamiento de sus elementos con un personaje para la generación de realism. Medellín: Universidad Nacional de Colombia, 2015. P. 6–8.

8. Peula P. J. M. Sistema de coordinación multiagente basado en comportamientos aprendidos. Málaga: Universidad de Málaga, 2015. P. 14.

9. Радченко Г. И., Захаров Е. А. Распределенные объектные технологии. Челябинск: Издательский центр ЮУрГУ, 2013. С. 21.

10. Cepero P. N., Moreno E. M. Transformación del Q-Learning para el Aprendizaje en Agentes JADE // Lámpsakos. 2015. № 14. P. 26–28.

11. The management and mining of multiple predictive models using the Predictive Modeling Markup Languaje (PMML) / R. Grossman, S. Bailey, A. Ramu, B. Malhi, P. Hallstrom, I. Pulleyn, X. Qin // ResearchGate. Chicago: University of Illinois, 2002. P. 4–7.

G. P. Yuleisy, I. I. Kholod
*Saint Petersburg Electrotechnical University*

## INTEROPERABILITY IN MULTIAGENT SYSTEMS DATA MINING

*Describes the basics of agents and multiagent systems. Analyses how communication between multiagent systems using data mining works. An overview of languages that provide communication between agents in distributed environments is presented. The study allows us to determine the most optimal offer that will be used, taking into account that the need for interaction between the system components is becoming increasingly important for solving joint tasks that would be very expensive or even impossible to solve individually. The structure of the platform for creating and interacting agents, the structure of the communication channel, and how message transfer between agents works are illustrated in detail. The author analyzes the architectures for creating and the most frequently used languages for interaction between agents. It details aspects of the PMML model and describes the algorithm for transferring the PMML model between agents in the ACL.*

**Agent, multiagent systems, PMML, FIPA ACL**