

Лабораторная работа № 1. Изучение состава и основных функциональных возможностей среды JADE

Содержание работы:

Цель работы.

1. Теоретические сведения.
 - 1.1. Понятие интеллектуального агента и агентной платформы.
 - 1.2. Среда построения мультиагентных систем JADE.
 - 1.2.1. Общая характеристика среды JADE.
 - 1.2.2. Установка и запуск JADE.
 - 1.2.3. Агент удаленного мониторинга - RMA.
 - 1.2.4. Агент-марионетка - Dummy Agent.
 - 1.2.5. Подслушивающий агент – Sniffer.
 - 1.2.6. Оконный интерфейс службы каталогов - DF GUI.
2. Порядок выполнения работы.
3. Содержание отчета.
4. Вопросы для проверки.

Цель работы:

1. Изучить основные понятия в области мультиагентных систем (МАС) и программных платформ для их построения.
2. Получить практические навыки установки и запуска платформы JADE, использования встроенных инструментов JADE, запуска агентов из командной строки и с использованием графического интерфейса пользователя.

1. Теоретические сведения

- 1.1. Понятие интеллектуального агента и агентной платформы.

Общепринятое определение термина «интеллектуальный агент» (ИА) в настоящее время отсутствует. Согласно наиболее распространенному определению под ИА понимается система, обладающая следующими свойствами:

- *автономность* - способность функционировать без вмешательства человека, контролируя свои действия и внутреннее состояние;

- *социальность* (social ability) - способность функционировать в сообществе агентов и обмениваться с ними сообщениями с помощью некоторого языка коммуникаций;

- *реактивность* - способность воспринимать состояние среды и своевременно реагировать на происходящие в ней изменения;

- *внутренняя активность* (pro-activity) - способность проявлять инициативу, т.е. не только реагировать на внешние события, но и генерировать цели и действовать рационально для их достижения.

В 1996 г. была создана международная организация FIPA (Foundation for Intelligent Physical Agents), цель которой – способствовать продвижению коммерческих приложений технологии ИА путем разработки открытых спецификаций, поддерживающих взаимодействие агентов и агентных сервисов. Согласно спецификации FIPA агент «обладает способностью предоставлять в рамках унифицированной и интегрированной исполнительной модели один или более сервисов, которые могут включать доступ к внешнему программному обеспечению, пользователям и коммуникационным возможностям».

Для обозначения инфраструктуры, в которую могут быть помещены программные агенты, используется понятия *агентной платформы и агентного контейнера*. Агент в среде JADE запускается в контейнере, который является частью платформы, где этот агент должен быть зарегистрирован. Одна платформа может содержать несколько контейнеров один, из которых должен быть главным. Структура агентных платформ показана на рис. 1.

Согласно спецификации FIPA агентная платформа включает три основных компонента: коммуникационный канал, систему управления агентами и службу каталогов.

Коммуникационный канал агентов (Agent Communication Channel - ACC), называемый также системой транспортировки сообщений, - программный компонент, управляющий всеми обменами сообщениями как внутри данной платформы, так и с удаленными платформами.

Система управления агентами (Agent Management System - AMS) – системный агент, управляющий доступом к агентной платформе и ее использованием. AMS обеспечивает два базовых сервиса платформы: сервис

«белых страниц» (поиск агентов по именам) и сервис жизненного цикла агентов. Для выполнения своих функций AMS поддерживает каталог идентификаторов агентов (AID) и базу данных состояний агентов. Каждый агент должен зарегистрироваться у AMS, чтобы получить свой уникальный AID. На каждой платформе может существовать только один AMS.

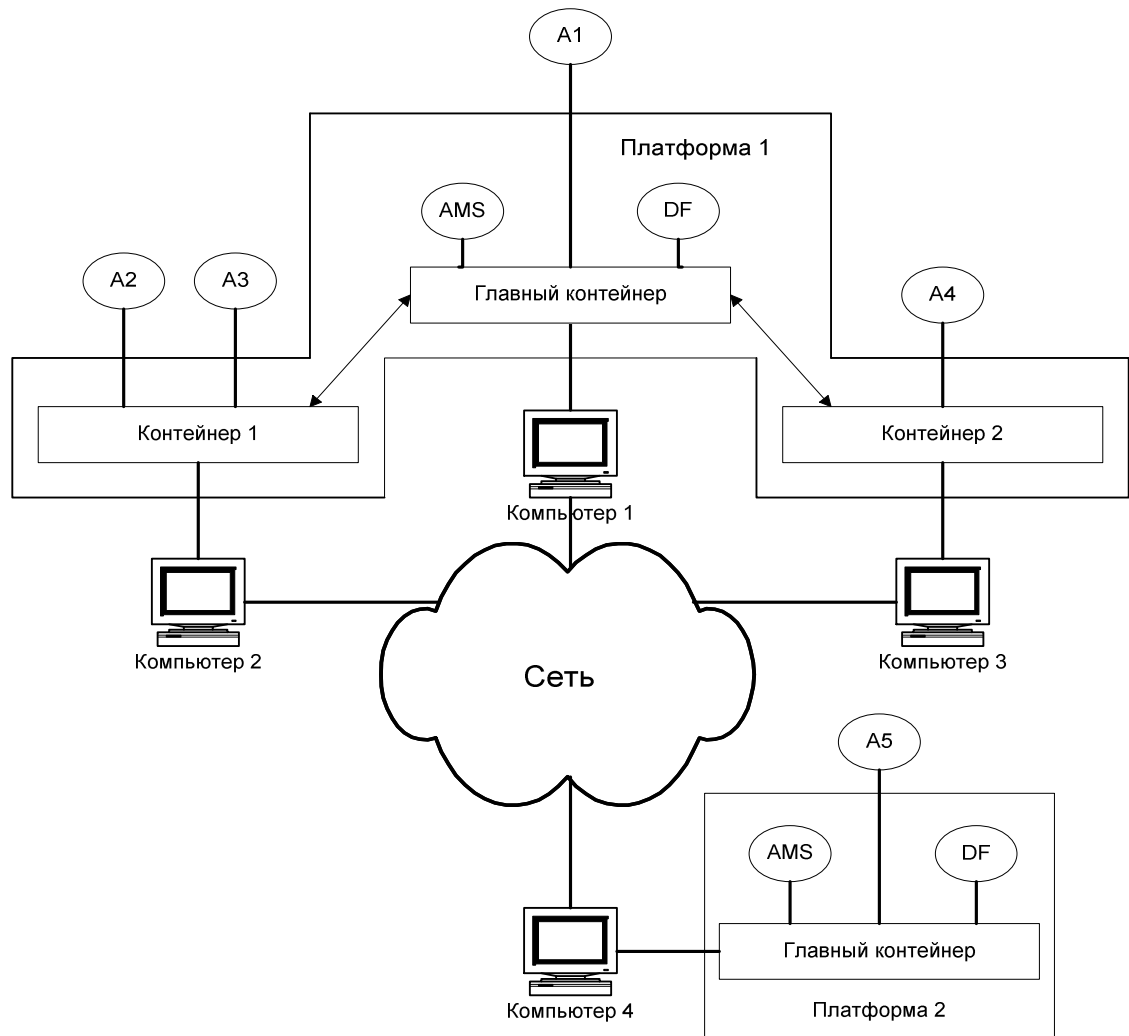


Рис. 1 - Платформы и контейнеры в JADE

Служба каталогов DF (Directory Facilitator) - агент, обеспечивающий по умолчанию сервис «желтых страниц» (поиск агентов по их сервисам) платформы (запрос сервисов у DF будет более подробно рассмотрен с лабораторной №7).

1.2. Среда построения мультиагентных систем JADE.

1.2.1. *Общая характеристика среды JADE.* В данном цикле лабораторных работ для построения ИА и мультиагентных систем используется программная среда JADE (Java Agent DEvelopment Framework), реализованная на языке Java.

JADE предоставляет набор системных сервисов агентной платформы соответствующих спецификациям FIPA: систему управления агентами, службу каталогов и систему транспортировки сообщений. Общение агентов осуществляется путем посылки сообщений на языке FIPA ACL.

Агентная платформа может быть распределена между несколькими хостами. При этом на каждом хосте выполняется только одно Java-приложение (и, следовательно, только одна виртуальная Java-машина - JVM). Каждая JVM обеспечивает им среду исполнения агентов, причем на одном хосте могут параллельно исполняться несколько агентов. В базовом варианте каждый агент выполняется в собственном потоке (thread), однако агенту часто необходимо выполнять параллельные задачи. Поэтому помимо многопоточности, предоставляемой самим языком Java, JADE также поддерживает планирование поведения (задач) агента.

Агентная платформа предоставляет графический интерфейс пользователя (GUI) для удаленного управления и мониторинга состояния агентов, позволяя создавать, запускать исполнение, останавливать и перезапускать агентов.

1.2.2. *Инсталляция и запуск платформы JADE.* Для установки среды JADE на локальном компьютере необходимо создать каталог (например, jade) и распаковать в него архивный файл(ы). Необходимо также, чтобы на данном компьютере была установлена среда JDK не ниже 1.2.2. Для конфигурирования среды и запуска JADE удобно использовать bat-файл (например, start_jade.bat), выполняющий установку переменных среды path и classpath. Пример такого файла приведен ниже (предполагается, что каталоги jade и jdk1.4, содержащий среду Java, находятся на диске C).

```
// Файл start_jade.bat
set PATH=%PATH%;c:\jdk1.4\bin;c:\jdk1.4\jre\bin
//Задание пути к исполняемым файлам JVM
set CLASSPATH=%CLASSPATH%;c:\jdk1.4\lib\;c:\jdk1.4\jre\lib\
//Задание пути к классам JDK
set CLASSPATH = %CLASSPATH%; .; c:\jade\lib\jade.jar;
c:\jade\lib\jadeTools.jar; c:\jade\lib\ Base64.jar;
c:\jade\lib\iiop.jar
//Задание пути к классам JADE
java jade.Boot -gui
// Запуск среды JADE
```

Примечание. В именах Java-классов регистр символов имеет значение, поэтому, например, “jade.Boot” и “jade.boot” являются разными именами.

При запуске файла start_jade.bat должны появиться два окна: консоль командной строки, в которой выводятся сообщения Java и JADE, и графическая консоль Агента Удаленного Мониторинга (Remote Monitoring Agent - RMA) JADE. На необходимость активизации последней указывает опция “-gui” в команде запуска JADE.

Для того, чтобы настроить JBuilder для работы с JADE необходимо:

1. Настроить библиотеки JBuilder для чего выбрать пункт меню Tools->Configure Libraries...

- 1.1. Создать новую библиотеку, нажав кнопку New...

- 1.2. Добавить список библиотек для JBuilder, нажав кнопку Add..., указав путь к файлам JADE (jade.jar, jadeTools.jar, Base64.jar, iior.jar).

2. Создать новый проект, указав в Required Libraries созданную запись о JADE (на втором шаге мастера создания проекта).

3. Создать агента наследовав его от класс jade.core.agent. Написать его бизнес-логику (рассмотрено в последующих лабораторных работах).

4. Для того, чтобы запустить созданного агента или графическую среду JADE нужно настроить проект.

- 4.1. В меню Project -> Project Properties на вкладке Run создать новую конфигурацию загрузки нажав кнопку New...

- 4.2. В окне Runtime configuration Properties указать тип приложения Application, с качестве главного класса (Main class) указать jade.Boot. Указать параметры в поле Application parameters: имя и тип запускаемого агента (например: theDummy;jade.tools.DummyAgent.DummyAgent).

Примечание. Для нормальной работы платформы JADE не забываете прерывать процесс jade.Boot после окончания работы с JADE (Два процесса jade.Boot вызываю исключительную ситуацию).

- 1.2.3. *Агент Удаленного Мониторинга RMA.* RMA обеспечивает управление жизненным циклом агентной платформы и всех зарегистрированных на ней агентов, как на локальном, так и на удаленных хостах. На одной платформе может быть запущено несколько RMA, при этом каждый экземпляр имеет свое локальное

имя. Однако в каждом агентном контейнере может выполняться только один RMA. Для управления платформой и отдельными агентами в меню RMA имеются следующие команды.

Команды меню **File**:

Close RMA Agent – завершить работу агента RMA.

Exit this Container – закрыть контейнер, в котором “живет” RMA, “убивая” его и всех других агентов, “живущих” в этом контейнере. Если контейнер является Главным Контейнером Агентной Платформы (Agent Platform Main-Container), то завершается работа всей платформы.

Shut down Agent Platform - завершить работу всей платформы, включая все контейнеры и “живущих” в них агентов.

Команды меню **Actions**:

Start New Agent – создать нового агента. Пользователь должен ввести имя нового агента и имя Java-класса, экземпляром которого он является. Если при этом выбран контейнер, агент создается и запускается в этом контейнере. В противном случае пользователь должен указать имя контейнера, в который он хочет поместить агента. Если контейнер не указан, агент запускается в главном контейнере платформы (Agent Platform Main-Container).

Kill – уничтожить всех выделенных агентов и контейнер. Если выбран Главный Контейнер Платформы, завершается работа всей платформы.

Suspend – приостановит работу выделенных агентов.

Resume – восстановить работу выделенных агентов. Выделенные агенты переводятся в активное состояние (AP_ACTIVE).

Send Message – послать сообщение агенту. Активизируется специальное диалоговое окно, в котором пользователь может подготовить и отослать сообщение указанному агенту на языке ACL.

Migrate Agent – переместить агента. Активизируется специальное диалоговое окно, в котором пользователь должен указать контейнер платформы, в который необходимо мигрировать выбранному агенту. Не все агенты способны мигрировать, если в их реализации не поддерживается сериализация.

Clone Agent – клонировать агента. Активизируется диалоговое окно, в котором пользователь должен указать имя нового агента и контейнер, в котором он должен запускаться.

Меню **Tools** содержит команды для запуска инструментов среды JADE (Sniffer, Dummy Agent и DF GUI), поддерживающих разработку и тестирование прикладных агентов на платформе JADE.

Команды меню **Actions** и **Tools** дублируются кнопками панели инструментов.

Левая часть клиентской области окна RMA используется для отображения существующих в данный момент на платформе контейнеров и агентов. При двойном щелчке на строке JADE раскрывается список контейнеров (в начальный момент он содержит только Main-Container), а при двойном щелчке на имени контейнера раскрывается список агентов, содержащихся в данном контейнере. При выделении имени контейнера или агента в нижнем окне отображается его полное имя. При щелчке правой кнопкой мыши на выделенном агенте раскрывается меню, содержащее основные команды управления агентом.

Для запуска с помощью RMA системных инструментов Dummy Agent, Sniffer и DF GUI можно использовать команды меню Tools или соответствующие кнопки панели инструментов. При этом сначала необходимо выделить в левом окне Main-Container (или любой другой контейнер, в который пользователь хочет поместить агента), а затем выбрать нужный пункт меню Tools или кнопку на панели инструментов. Вместе с тем, эти инструменты, являясь экземплярами Java-классов, могут быть запущены и из командной строки. Например, для запуска DummyAgent необходимо использовать команду:

```
java jade.Boot theDummy:jade.tools.DummyAgent.DummyAgent
```

Для запуска пользовательских агентов также сначала необходимо выделить Main-Container (или другой предварительно созданный контейнер, в который пользователь хочет поместить агента), а затем выбрать команду “Start New Agent” в меню Actions или воспользоваться кнопкой на панели инструментов. При этом раскроется окно, в котором надо ввести имя агента и содержащий его Java-класс. Поле “Arguments” заполнять необязательно. Имя агента можно выбрать произвольно, оно будет использоваться при обращении к нему на данной платформе.

1.2.4. *Агент-марионетка* (Dummy Agent - DA) предоставляет пользователю возможность осуществлять через него произвольное общение с другими агентами и, таким образом может использоваться как средство мониторинга и отладки взаимодействий агентов. Графический интерфейс агента DA служит формой для подготовки сообщений на языке ACL и их отправки другим агентам. Кроме того, DA поддерживает список всех посланных и принятых сообщений, снабженных временными метками. Этот список доступен пользователю для просмотра и редактирования, а также может быть сохранен на диске для последующего анализа разговоров агентов. На агентной платформе может быть запущено любое число экземпляров DA. Возможности, предоставляемые агентом DA, реализованы через команды двух меню: Current Message и Queued Message.

Команды меню **Current Message**:

Reset – очистить поля сообщения в окне DA.

Send – отослать подготовленное сообщение указанному агенту-получателю.

Open – загрузить ACL-сообщение из файла.

Save – сохранить подготовленное ACL-сообщение в файле.

Команды меню **Queued Message**:

Open Queue – прочитать очередь отправленных/полученных сообщений из файла.

Save Queue – сохранить очередь отправленных/полученных сообщений в файле.

Set as current – сделать выбранное ACL-сообщение текущим.

Reply – для выбранного сообщения подготовить ответное сообщение.

View – просмотреть выбранное ACL-сообщение.

Delete – удалить выбранное ACL-сообщение.

После запуска агента DA открывается окно Dummy Agent, в левой части которого содержатся поля для формирования посылаемого сообщения на языке общения агентов. Синтаксис и семантика языка общения агентов будут подробно изучены в последующих работах. В правой части окна отображается список сообщений, посылаемых и получаемых DA. Выделив в этом списке сообщение и выбрав пункт меню View (или соответствующую кнопку панели инструментов) можно просмотреть содержание данного сообщения.

1.2.5. *Подслушивающий агент* (Sniffer) предоставляет пользователю возможность отслеживать (подслушивать) сообщения, которыми обмениваются другие агенты. Когда пользователю нужно “подслушать” некоторого агента или группу агентов, каждое сообщение направляемое к (от) этому агенту(группе) будет отслеживаться и отображаться в графическом окне агента Sniffer. Пользователь может просмотреть любое сообщение и сохранить его в файле, либо сохранить все отслеженные сообщения и перезагрузить их затем из одного файла для дальнейшего анализа.

Команды меню **Actions**:

Do sniff this agent(s) – включить «подслушивание» выделенного(ых) агента(ов).

Do not sniff this agent(s) - выключить «подслушивание» выделенного(ых) агента(ов).

Show only agent(s) – показывать только выделенных агентов.

Clear canvas – очистить видимое поле.

Open snapshot file – открыть файл «моментального снимка»

Save snapshot file – сохранить «моментальный снимок» в файл

Write message list file – сохранить текстовый файл с протоколом общения агентов.

Для запуска Sniffer с помощью RMA используется команда меню Tools или кнопки панели инструментов. Запуск агента из командной строки может быть выполнен командой:

```
java jade.Boot sniffer:jade.tools.sniffer.Sniffer
```

После запуска агента Sniffer открывается окно Sniffer Agent, в левой части которого отображается Main-Container, содержащий активизированных агентов. Для включения «подслушивания» некоторого агента необходимо щелкнуть на нем правой кнопкой мыши и в появившемся меню выбрать пункт “Do sniff this agent(s)”. (Это же действие можно выполнить, используя соответствующий пункт меню или панель инструментов.) При этом в правом окне появится красный прямоугольник, обозначающий подслушиваемого агента, и стрелки, обозначающие сообщения. Щелкнув правой кнопкой на стрелке, можно посмотреть текст переданного сообщения.

1.2.6. *Оконный интерфейс службы каталогов (DF GUI)*. DF GUI может быть запущен из меню Tools RMA. Это действие реализуется путем отправки ACL-сообщения службе каталогов (DF) с просьбой показать его GUI. Следовательно, GUI может быть показан только на том хосте, где исполняется платформа (главный контейнер). Используя GUI пользователь может взаимодействовать со службой каталогов: просматривать описания зарегистрированных агентов, регистрировать и отменять регистрацию агентов, модифицировать описание зарегистрированных агентов, а также выполнять поиск описаний агентов. Кроме того, GUI позволяет объединять службы каталогов в федерации и создавать сложные сети областей и подобластей “желтых страниц”. Любой включенный в федерацию DF (включая расположенные на удаленных, построенных не на основе JADE платформах) может управляться через этот же GUI и те же самые базовые операции (просмотр, регистрация, отмена регистрации, модификация, поиск) могут выполняться на удаленной DF.

Команды меню **General**:

Exit DF – прекратить работу агента DF.

Close GUI – закрыть оконный интерфейс агента DF (без прекращения его работы).

Команды меню **Catalogue**:

View – просмотр сервисов, предоставляемых выделенным агентом

Modify – редактирование сервисов, предоставляемых выделенным агентом

Deregister – удалить регистрацию выделенного агента

Register new agent – зарегистрировать нового агента

Search – поиск агента

Команды меню **Super DF**:

Federate – создать федерацию данного DF с другим DF

Пояснения к оконному интерфейсу DF GUI:

Registrations with this DF – агенты, зарегистрированные в данном DF

Search result – результат последнего поиска

DF federation – дерево федераций, в которых участвует данный DF

2. Порядок выполнения работы

2.1. Инсталлировать платформу JADE и запустить ее с активизацией агента RMA.

2.2. В открывшемся окне RMA раскрыть содержимое главного контейнера (JADE→Main-Container) и зафиксировать список агентов, активизированных при запуске платформы.

Поэкспериментировать с командами меню, последовательно выделяя в списке главный контейнер и содержащихся в нем агентов и выполняя над ними различные действия. Зафиксировать и объяснить полученные результаты.

2.3. Запустить с помощью меню Tools RMA агента Sniffer. Зафиксировать изменение списка содержимого Main-Container в окне RMA.

В открывшемся окне Sniffer Agent включить «подслушивание» всех агентов, содержащихся в Main-Container. Просмотреть сообщения, соответствующие всем появившимся в правом окне стрелкам и зафиксировать основные параметры сообщений (отправитель, получатель, тип ACL-сообщения (коммуникативный акт), тип действия в поле content).

2.4. Запустить с помощью RMA агента DummyAgent. Зафиксировать изменение списка содержимого Main-Container в окне RMA. В открывшемся окне DummyAgent изучить команды меню и поля для формирования сообщений на языке общения агентов.

Переключится в окно Sniffer Agent, и зафиксировать появившиеся новые сообщения между агентами. Включить «подслушивание» агента DummyAgent.

Используя окно DummyAgent подготовить тестовое сообщение от DummyAgent к AMS, заполнив поля, как указано ниже:

Sender:	da0@localhost:1099/JADE
Receivers:	ams@localhost:1099/JADE
Communicative act:	Request
Content	((action (agent-identifier :name ams@localhost:1099/JADE) (get-description)))
Language	FIPA-SL0
Ontology	fipa-agent-management
Protocol	fipa-request

Данное сообщение представляет собой запрос списка адресов AMS на данной платформе (действие `get-description`). На протяжении жизненного цикла JADE-агента с ним можно контактировать через различные адреса. При этом профиль платформы и база знаний AMS автоматически поддерживаются согласованными с актуальными (активными на платформе в данный момент времени) протоколами транспорта сообщений MTP. Поэтому любой агент может получить список адресов другого агента, запросив его у AMS. Запись данного сообщения в формате языка общения агентов FIPA ACL имеет вид:

```
(REQUEST
:sender (agent-identifier :name da0@localhost:1099/JADE)
:receiver (set (agent-identifier :name ams@localhost:1099/JADE))
:content ((action
            (agent-identifier :name ams@localhost:1099/JADE)
            (get-description)))
:language FIPA-SL0
:ontology fipa-agent-management
:protocol fipa-request
)
```

Помимо очевидных по назначению полей с указанием отправителя, получателя и собственно содержания, в сообщении также указываются используемый для записи содержания язык, онтология и протокол взаимодействия агентов. Более подробно язык общения агентов будет изучен в лабораторной работе № 3.

Отправить подготовленное сообщение (используя пункт “Send” меню Current message или кнопку панели инструментов). В правом окне DA появится краткая информация о посланном и принятом ответном сообщениях. Для просмотра полного сообщения надо выделить его и выбрать пункт View меню Queued message (или соответствующую кнопку панели инструментов). Зафиксировать ответное сообщение.

Внести в поле `content` изменения, нарушающие синтаксис сообщения, (например, удалить последнюю скобку или букву в команде `get-description`). Отправить сообщения и зафиксировать реакцию системы.

2.5. Используя RMA (команда “Show the DF GUI” меню Tools) активизировать оконный интерфейс службы каталогов (DF GUI). Изучить состав и назначение пунктов меню и панели инструментов DF GUI.

Создать регистрационную запись об агенте DummyAgent, предварительно запустив его. По умолчанию имя созданного агента будет ‘da0’. Заполнить поля согласно следующему рисунку 2 (в поля либо вписывается текст, либо на поле необходимо щелкнуть правой кнопкой мыши и выбрать пункт add – добавить).

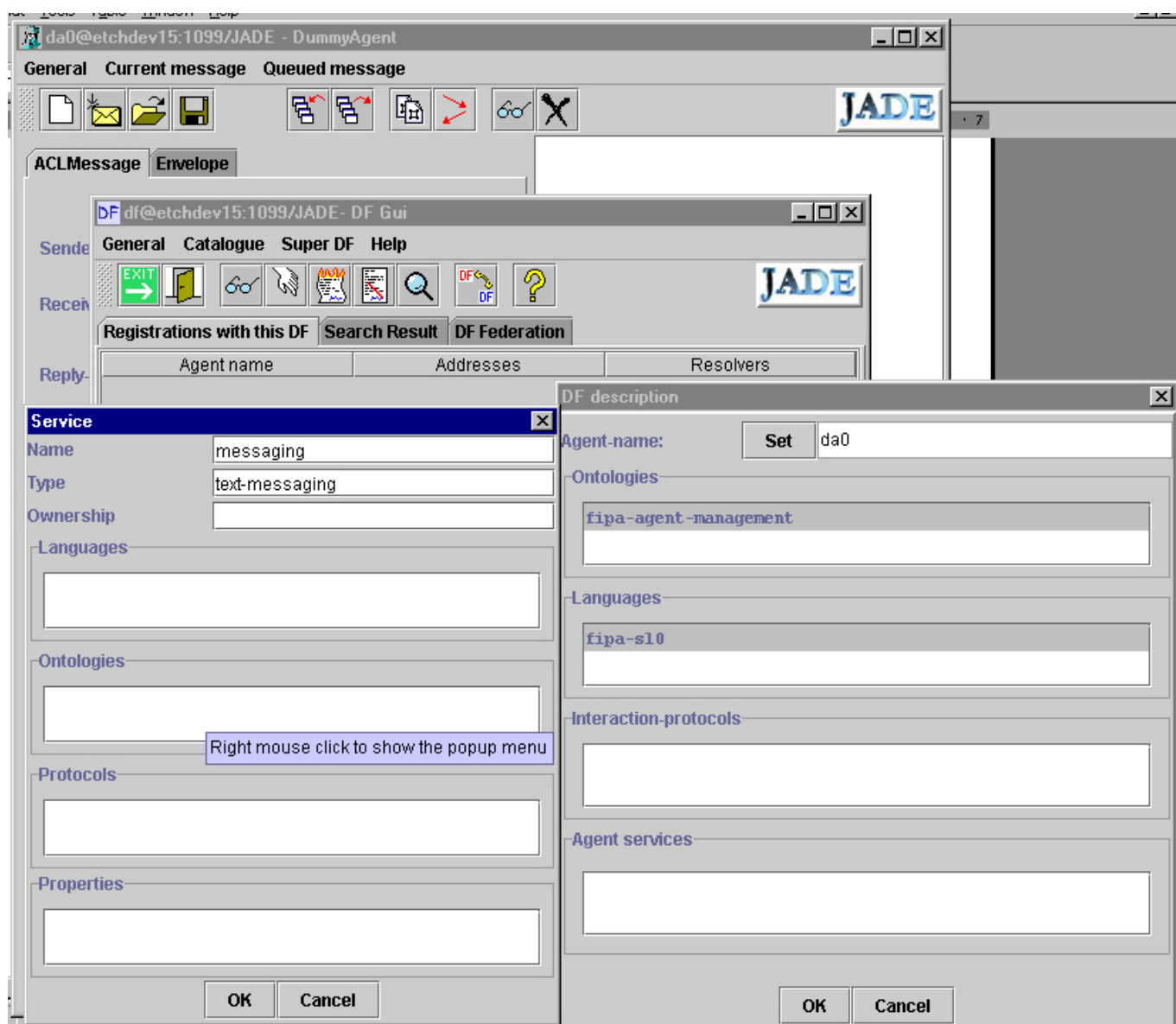


Рис. 2 – Оконный интерфейс DF

Затем вызвать функцию «поиск» и найти регистрацию данного агента, указав в параметрах поиска “max propagation length = 5”, “max results = 5”, “DF description – Agent Services – name – ‘messaging’ ”. При нажатии на ОК в средней

вкладке DF GUI появится регистрационная запись. Зафиксировать имя агента в данной записи. Сделать двойной щелчок мыши на этой записи и зафиксировать данные об агенте.

Выполнить новый поиск, но теперь не по имени сервиса ‘messaging’, а по имени поддерживаемой онтологии ‘fipa-agent-management’ или по имени поддерживаемого языка ‘fipa-slo’. Поэкспериментировать с другими возможностями поиска.

Регистрация агента DummyAgent в службе каталогов DF

2.6. Запустить агента TalkerAgent, для чего в меню Actions RMA выбрать пункт “Start New Agent” (или кнопку панели инструментов). В открывшемся окне ввести:

Agent Name: TalkerAgent

Class Name: MASLabs.lab1.TalkerAgent

(Агент TalkerAgent может быть также запущен из командной строки или из bat-файла. При этом в переменной classpath необходимо указать путь к файлу содержащему класс, например: c:\JADE2\Labs\TalkerAgent.jar. Полный путь к классу агента выглядит так: maslabs.lab1.TalkerAgent).

Исходный код агента TalkerAgent представлен в приложении А. Агент TalkerAgent получив любое сообщение на языке ACL от другого агента, выводит его на экран в консоли, выводит имя агента пославшего это сообщение и формирует ответное ACL-сообщение.

Запустить (если не запущены) DummyAgent и SnifferAgent. Используя SnifferAgent включить «подслушивание» DummyAgent и TalkerAgent. Используя оконный интерфейс DummyAgent подготовить сообщение к TalkerAgent. Для этого в поле Receivers окна DummyAgent ввести имя агента TalkerAgent (щелчком правой кнопки мыши и выбором команды “add”), а в поле содержания Content - любой текст, например, “Hello”. Другие поля можно не заполнять. Отправить сообщение.

Переключиться в консольное окно JAVA и зафиксировать выведенный текст. TalkerAgent должен ответить на посланное сообщение приветствием. Затем просмотреть диаграмму диалога и содержание сообщений средствами инструментов Sniffer и DummyAgent.

Сохранить протокол обмена сообщениями, для чего в окне Sniffer выбрать пункт “Write message list file” меню Actions, указать путь к каталогу и произвольное имя файла с расширением txt. Просмотреть сохраненный текстовый файл, воспользовавшись любым редактором.

Изменить посылаемые DummyAgent сообщения, выбирая различные типы коммуникативных актов и протоколов. Просмотреть пересылаемые сообщения средствами Sniffer (или DummyAgent) и проанализировать их.

3. Содержание отчета

3.1. Цель работы.

3.2. Определение интеллектуального агента и агентной платформы. Состав агентной платформы, назначение основных компонентов и инструментов среды JADE.

3.3. Сообщения между базовыми компонентами платформы, “подслушанные” агентом Sniffer (п. 2.3 работы).

3.4. Дополнительные сообщения, появившиеся после запуска DummyAgent (п. 2.4). Текст ACL-сообщения DummyAgent с запросом списка адресов AMS и ответное сообщение. Реакция системы на сообщения с некорректным синтаксисом в поле content.

3.5. Протокол обмена сообщениями между DummyAgent и TalkerAgent. Сообщения TalkerAgent`а в консольном окне JAVA.

4. Вопросы для самопроверки

4.1. Что понимается под интеллектуальным агентом ?

4.2. Что понимается под агентной платформой? Каково назначение компонентов платформы: AMS, DF, ACC?

4.3. Что представляет собой среда JADE? Как запустить JADE из командной строки?

4.4. Каковы функции инструментов среды JADE: RMA, DummyAgent, Sniffer и DF GUI? Как можно запустить инструменты среды JADE?

4.5. Какие существуют способы запуска пользовательских агентов на платформе JADE?

4.6. Перечислите поля ACL-сообщения.

4.7. Как можно просмотреть и сохранить диалог между агентами?