СПбГЭТУ «ЛЭТИ»
Кафедра Вычислительной техники

Магистерская программа
«Семантические технологии и многоагентные системы»

Дисциплина: «Интеллектуальные агенты и многоагентные системы»

Лекция 5

Стандартизация агентных технологий

Содержание лекции

1. Стандартизация агентных технологий
   - Краткая история стандартизации агентных технологий. FIPA
   - Стандарты FIPA. Группы спецификаций:
     - абстрактная архитектура,
     - управление агентами,
     - коммуникации,
     - транспорт агентных сообщений
   - Понятие агентной платформы (АП)
     - Основные компоненты АП:
       - система управления агентами,
       - служба каталогов,
       - служба передачи сообщений.
   - Стандарты приложений агентных систем.

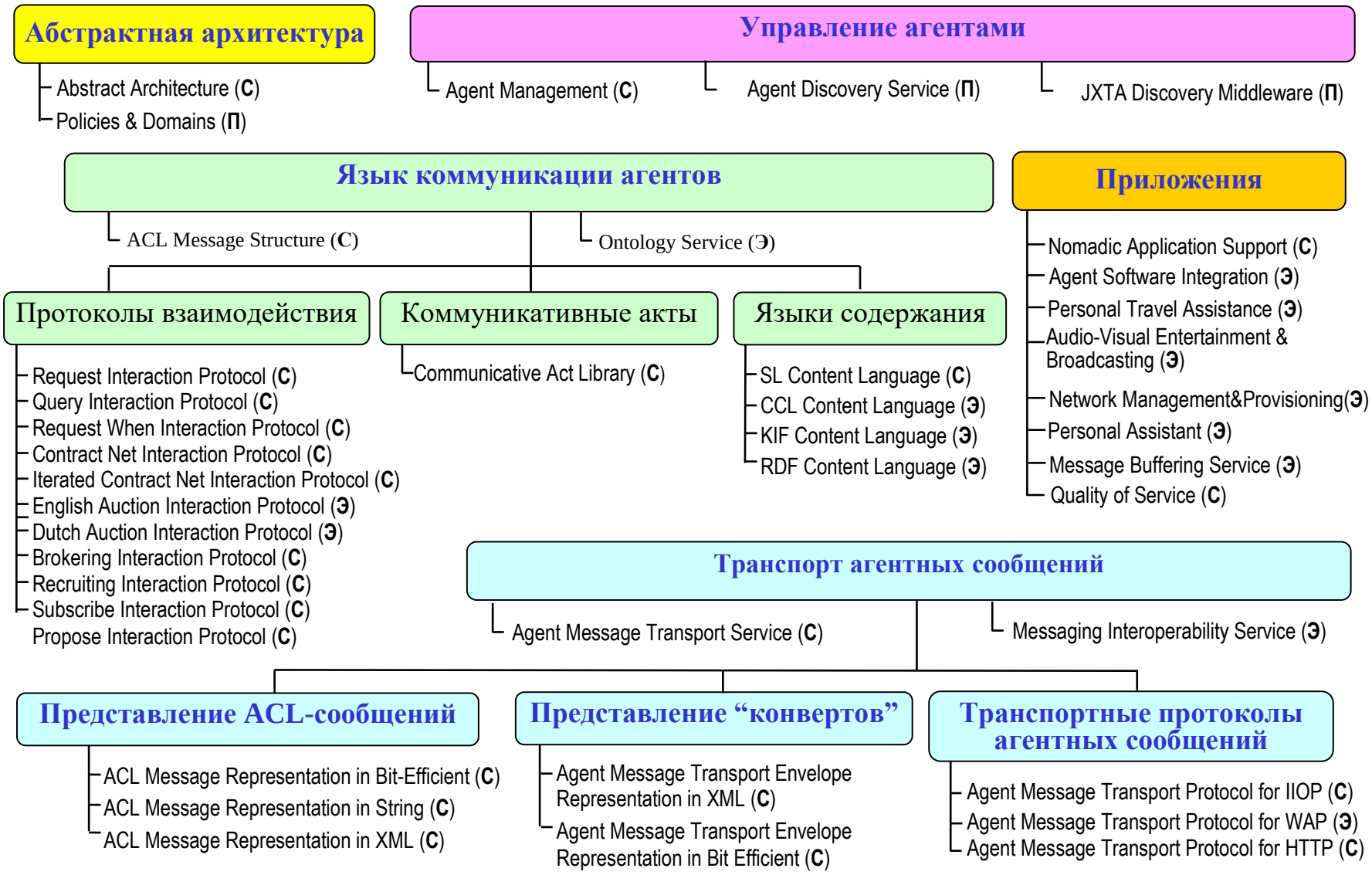
Стандартизация в области ИА

FIPA (Foundation for Intelligent Physical Agents) - международная организация, созданная в 1996 г.

Цель – поддержка продвижения коммерческих приложений технологии ИА путем разработки открытых спецификаций, поддерживающих интероперабельность агентов и агентных сервисов.

Агент, согласно FIPA , «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»

Стандартизация агентных технологий: Состав спецификаций FIPA



Ключевые спецификации FIPA:

00001 – Abstract Architecture Specification

00007 – Content Languages Specification

00008 – SL Content Language Specification

00023 – Agent Management Specification

00024 – Agent Message Transport Specification

00025 – Interaction Protocol Library Specification

00037 – Communicative Act Library Specification

00061 – ACL Message Structure Specification

00063 – Control Agent Specification

00082 – Network Management and Provisioning Specification

00084 – Agent Message Transport Protocol for HTTP Specification

00086 – Ontology Service Specification

00087 – Agent Management Support for Mobility Specification

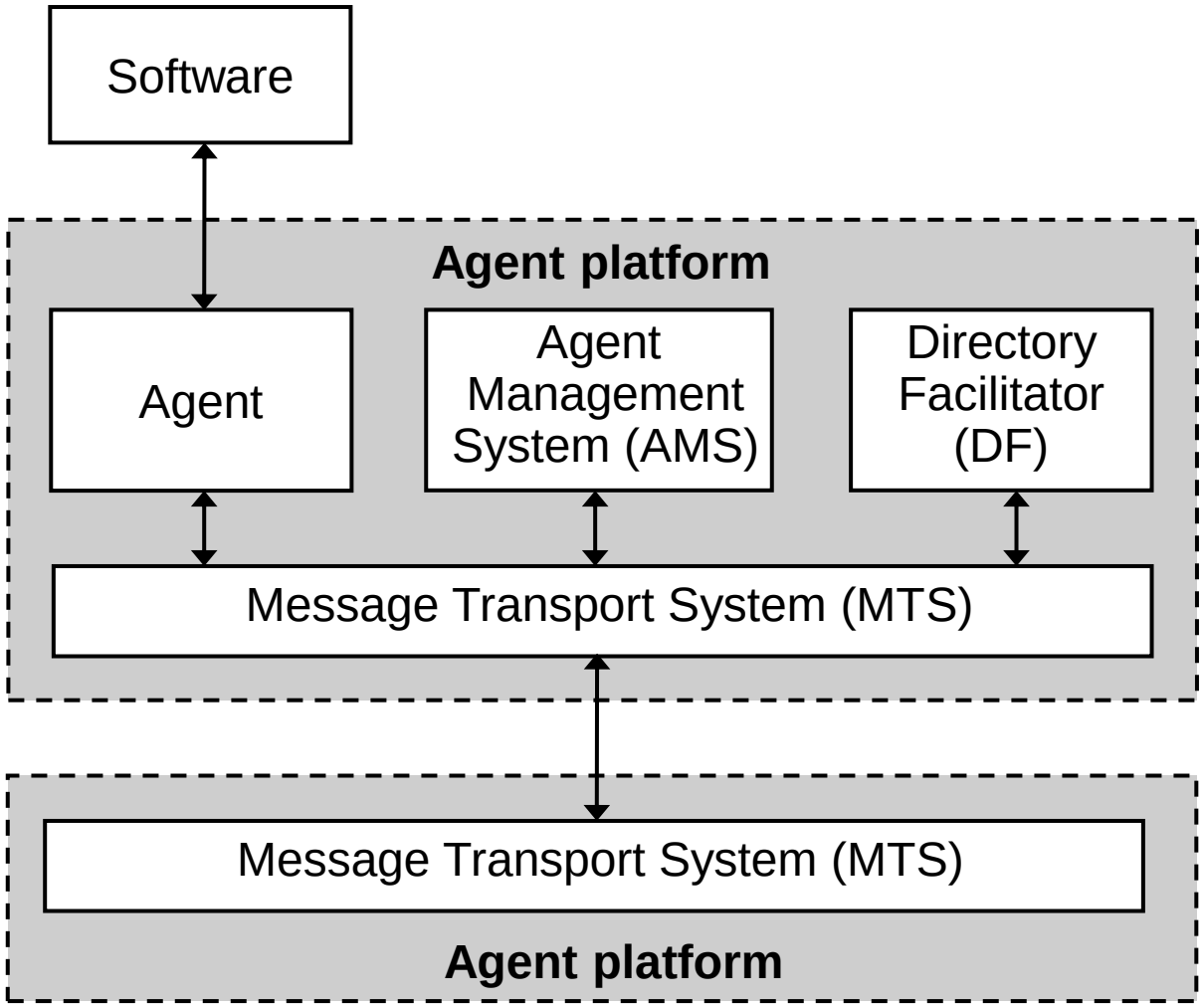
00094 – Quality of Service Specification

....

Понятие агентной платформы по FIPA

- *Эталонная модель управления агентами* - множество логических компонентов, каждый из которых предоставляет множество возможностей, которые могут комбинироваться в физических реализациях агентных платформ.
- *Агентная платформа* (АП) предоставляет физическую инфраструктуру, в которой могут быть размещены агенты.
  - АП состоит из компьютера(ов), операционной системы, поддерживающего агентов программного обеспечения, *компонентов управления агентами по FIPA* (DF, AMS и MTS) и агентов.
  - Внутренняя конструкция АП определяется разработчиками агентных систем и не является предметом стандартизации FIPA.
  - FIPA рассматривает только способ реализации коммуникаций между «родными» для данной АП агентами и внешними или динамически регистрирующимися на АП агентами.

Структура агентной платформы по FIPA



Компоненты агентной платформы: Агенты

- *Агент* – основное «действующее лицо» на АП:
  - Сочетает одну или более сервисных возможностей *в унифицированной и интегрированной исполнительной модели*, которая может включать доступ к внешнему программному обеспечению, людям-пользователям и коммуникационные возможности;
  - Должен *иметь* как минимум одного *владельца* (например, на основе принадлежности к организации или человеку-владельцу);
  - Может поддерживать несколько понятий идентичности;
  - *Идентификатор агента* (Agent Identifier - AID) помечает агента так, что его можно *однозначно выделить* внутри Агентного Универсума;
  - *Может регистрироваться с множеством транспортных адресов*, по которым с ним можно контактировать;
  - Может иметь возможности брокера ресурсов для доступа к программному обеспечению;

Компоненты агентной платформы: AMS & DF

- *Agent Management System* (AMS) – *Система Управления Агентами:*
  - *обязательный компонент* агентной платформы;
  - выполняет супервизорное управление доступом к агентной платформе и ее использованием;
  - *на одной агентной* платформе может существовать только *одна AMS*;
  - поддерживает *каталог идентификаторов агентов (AID)*, содержащих (кроме прочего) транспортные адреса агентов, зарегистрированных на данной АП;
  - предоставляет другим агентам сервис «белых страниц»;
  - Каждый агент должен зарегистрироваться у AMS для того, чтобы получить корректный AID.
- *Directory Facilitator* (DF) – *Служба Каталога:*
  - обязательный компонент агентной платформы
  - предоставляет агентам *сервис «желтых страниц»* - агенты могут регистрировать свои сервисы и/или выполнять поиск в службе каталога с целью найти агента, предоставляющего заданные сервисы;
  - на одной агентной платформе может существовать несколько служб каталогов и они могут объединяться в федерации.

Компоненты агентной платформы: MTS & Software

- *Message Transport Service* (MTS) – *Служба Транспортировки Сообщений:*
  - поддерживает заданный по умолчанию метод коммуникации между агентами на различных АП.

- *Software - стороннее (неагентное) программное обеспечение:*
  - совокупность не-агентного исполняемого кода, доступного *через* агента;
  - *не относится к агентной платформе;*
  - агенты могут обращаться к стороннему программному обеспечению, например, чтобы:
    - добавить новые сервисы;
    - приобрести новые коммуникационные протоколы;
    - приобрести новые протоколы/алгоритмы безопасности;
    - приобрести новые протоколы переговоров;
    - обратиться к инструментам поддерживающим миграцию;
    - . . .

Именование агентов - 1

- *Идентификатор агента (AID)* – расширяемая совокупность пар «параметр-значение», определяющих:
  - ***Имя агента*** (обязательный параметр);
  - ***Транспортные адрес(а)*** *агента* (необязательный параметр) – адреса которым можно контактировать с агентом;
  - ***Идентификаторы*** *агентов, предоставляющих* ***сервис разрешения имен*** (необязательный параметр)

```
(agent-identifier
    :name       <символическое имя агента>
   :addresses   <список транспортных адресов агента>
    :resolvers  <список AID сервисов разрешения имен>
)
```

Порядок перечисления адресов и AID в параметрах **`:addresses`** и **`:resolvers`** определяет предпочтение их использования

Именование агентов - 2

- *Транспортный адрес* – (обычно) *специфичен для Протокола Транспортировки Сообщений* (Message Transport Protocol)
- Агент может поддерживать *множество методов коммуникации* - в параметре **`:addresses`** указывается множество значений транспортных адресов
- Сервис разрешения имен предоставляется AMS через функцию поиска.
  - Параметр **`:resolvers`** в AID содержит последовательность AID, по которым AID агента в конечном счете может быть разрешен в транспортный адрес или множество транспортных адресов

Именование агентов - пример

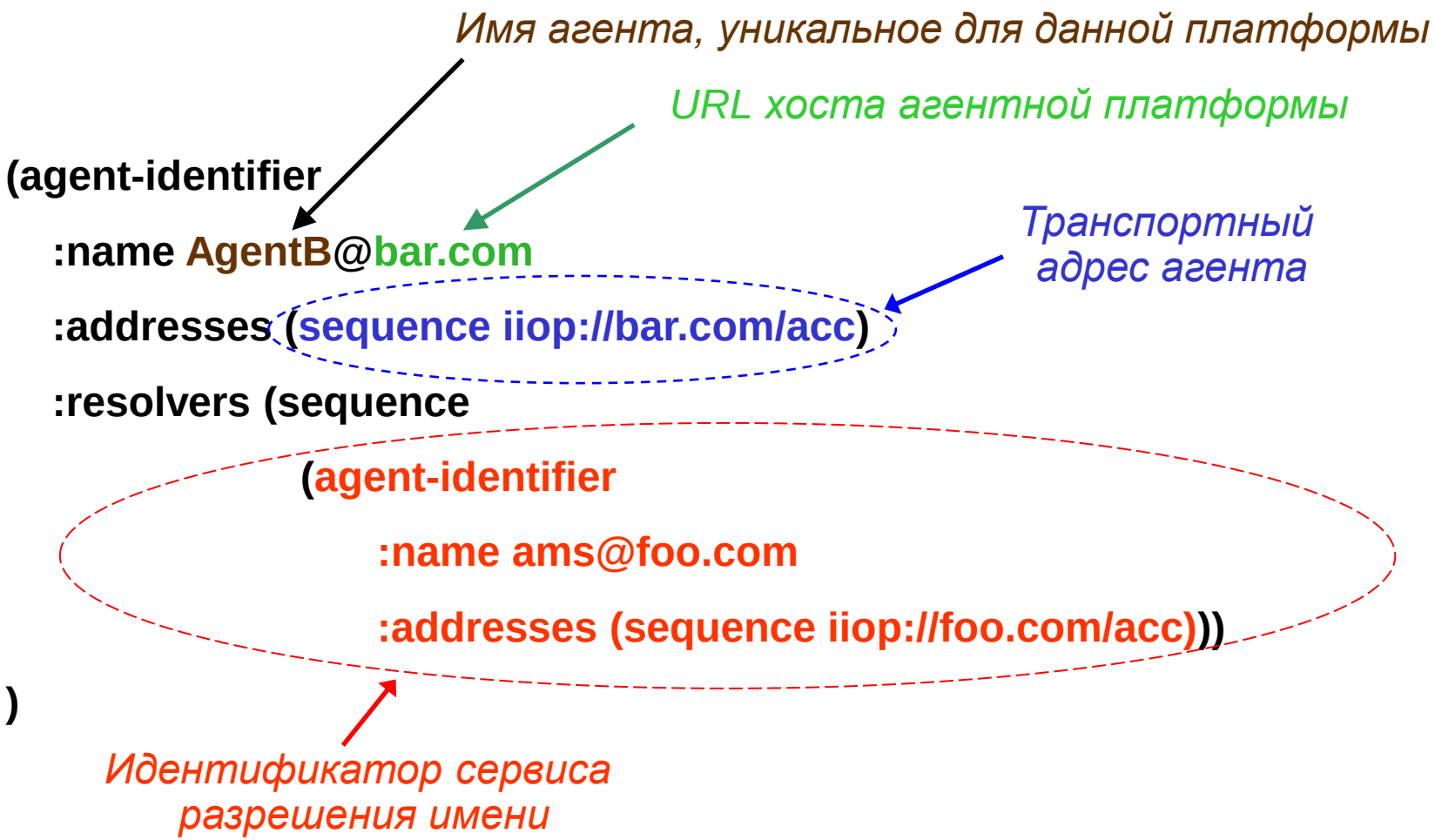
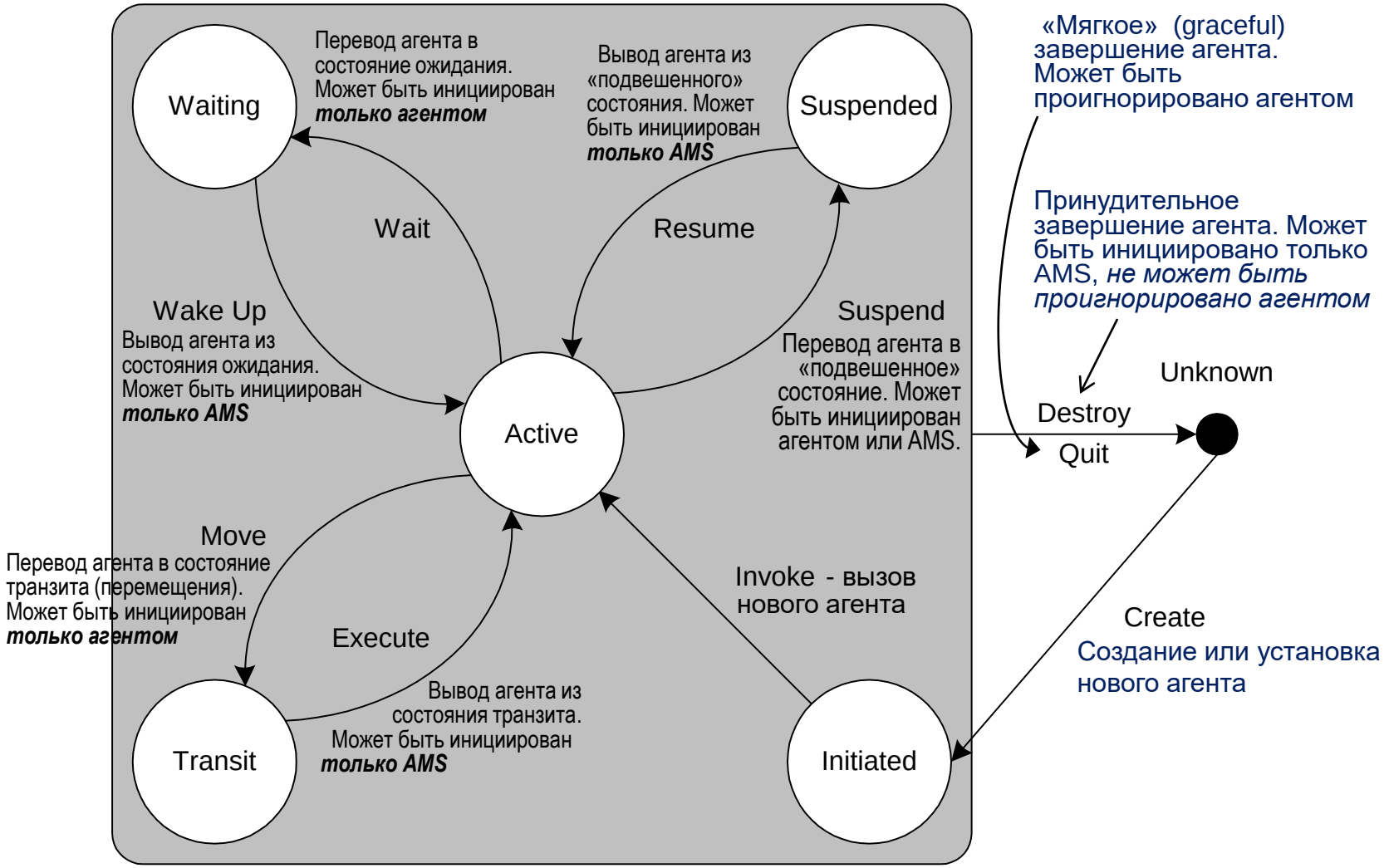


Диаграмма жизненного цикла агента



Move и **Execute** используются только мобильными агентами

Управление доставкой сообщений

Ответственность AMS в интересах AP в части доставки сообщений в каждом состоянии жизненного цикла агента:

- **Active** – MTS доставляет сообщения агенту в штатном режиме.
- **Initiated/Waiting/Suspended** – MTS либо буферирует сообщения до тех пор пока агент не вернется в активное состояние или пересылает сообщения к новому местоположению (если для агента установлен форвардинг).
- **Transit** – MTS либо буферирует сообщения до тех пор пока агент не станет активен (т.е. функция move на исходной AP завершается неудачей или агент успешно стартовал на целевой AP) или «форвардит» сообщения к новому местоположению (если для агента установлен форвардинг).
  - Только мобильные агенты могут переходить в состояние **Transit**. Это гарантирует, что стационарный агент выполняет все свои функции на узле где он был запущен.
- **Unknown** – MTS либо буферирует сообщения или отвергает их, в зависимости от политики MTS и требований транспорта сообщений

Стандарты ИА: Абстрактная архитектура FIPA
(FIPA Abstract Architecture)

Основной фокус – поддержка *обмена семантически значимыми сообщениями между агентами*, которые могут использовать различный транспорт сообщений, различные Agent Communication Languages, или различные языки содержания. Это требует различных точек интероперабельности.

Спектр вопросов архитектуры включает:

- Модель сервисов и обнаружения сервисов доступных агентам и другим сервисам;
- Интероперабельность транспорта сообщений;
- Поддержка различных форм представления ACL (Agent Communication Languages);
- Поддержка различных форм языка содержания (content language);
- Поддержка представления нескольких (multiple) служб каталогов (directory services).

Цель - Обеспечить возможность создания реализаций, способных взаимодействовать при варьировании этих атрибутов

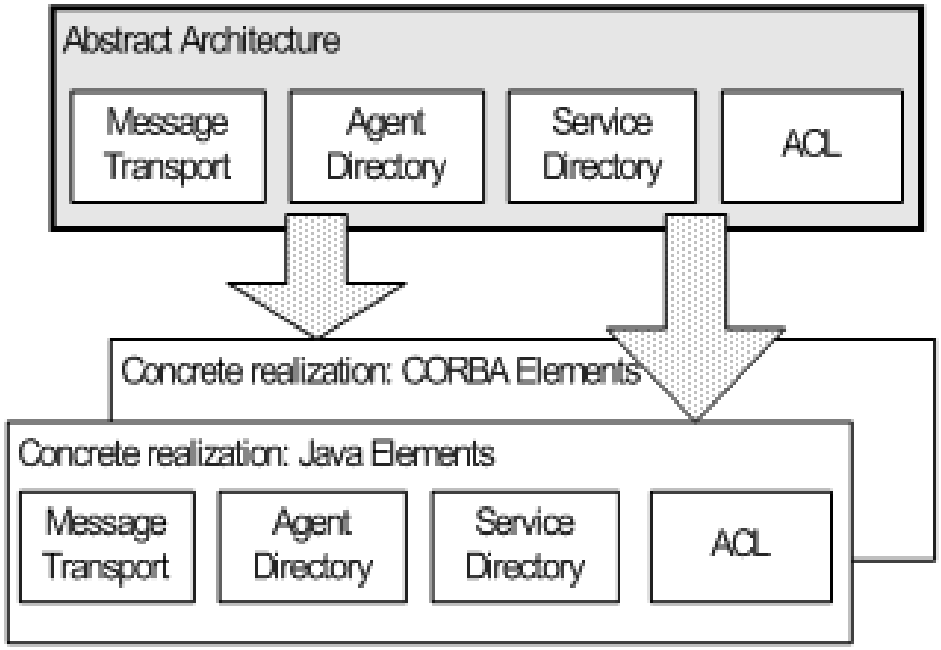
Абстрактная архитектура FIPA

Вопросы, ***не включенные*** в абстрактную архитектуру:

- Жизненный цикл агента и управление им;
- Мобильность агента;
- Предметные области (Domains);
- Политики общения (Conversational policy);
- Идентичность агентов;

- FIPA Abstract Architecture:
  - не предназначена для непосредственной реализации;
  - является основой для разработки конкретных архитектурных спецификаций
  - спецификации детально описывают как построить агентную систему (включая агентов и сервисы, на которые они опираются) в терминах конкретных программных артефактов (языков программирования, API, сетевых протоколов, служб ОС и т.д.)

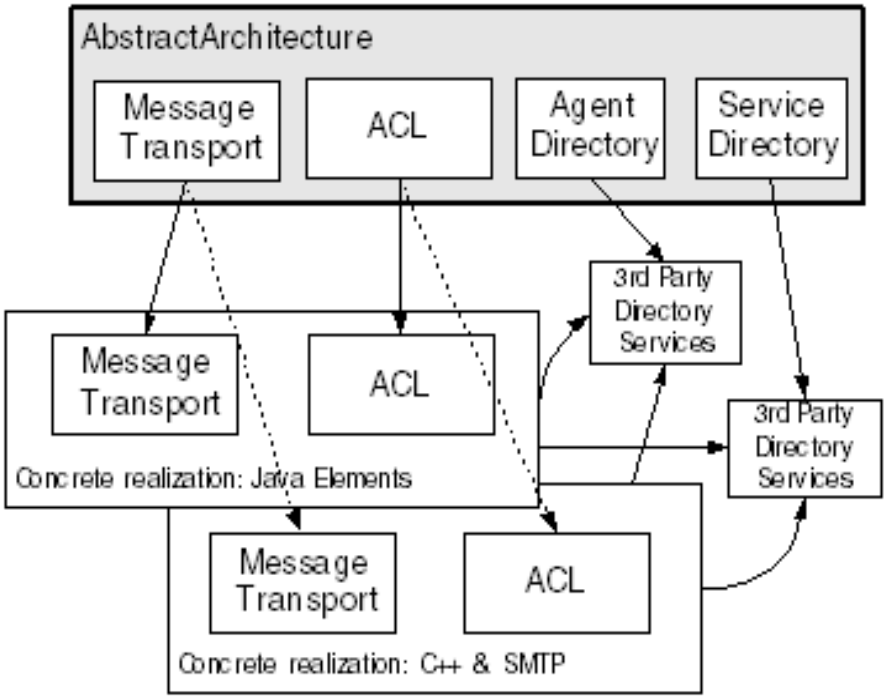
Отображение Абстрактной Архитектуры FIPA в конкретные реализации



- *FIPA-совместимая* конкретная архитектурная *спецификация* должна:
  - включать *механизмы для регистрации* и *обнаружения* (discovery) *агентов* и *передачи межагентных сообщений*;
  - сервисы должны быть явно описаны в терминах соответствующих элементов FIPA Abstract Architecture;
- Реализация элемента - определение элемента абстрактной архитектуры в терминах конкретной архитектуры
  - конкретная архитектура реализует все части абстрактной;

- Проектировщик конкретной архитектуры имеет значительную свободу в выборе способа реализации абстрактных элементов
  - Если конкретная архитектура обеспечивает только одно кодирование для сообщений или только один транспортный протокол, реализация может упростить программный код системы.
  - Вместе с тем, реализация может включать дополнительные опции или функции, которые требуются разработчикам для работы как с абстрактными, так и для конкретных платформ элементов. Т.о. существование абстрактной архитектуры не запрещает введения элементов, полезных для построения хорошей агентной системы, а только *устанавливает минимальные необходимые элементы*.

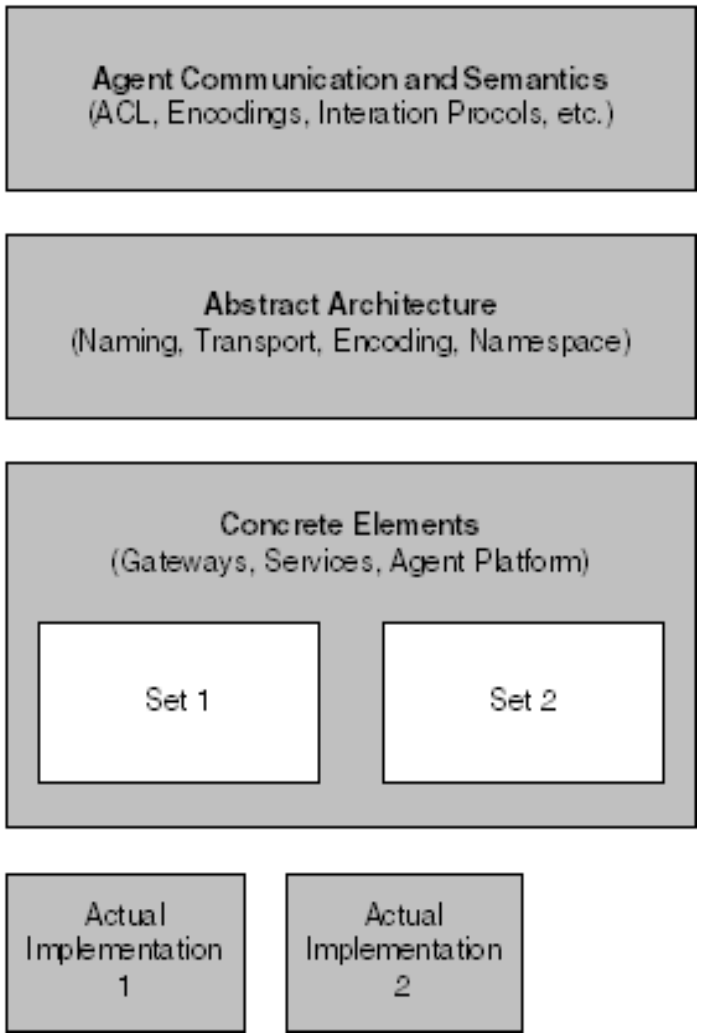
Конкретная реализация с использованием разделяемых элементов



- FIPA Abstract Architecture описывает также необязательные (optional) элементы:
  - Элемент, необязательный *на абстрактном уровне*, может быть *обязательным в конкретной реализации*. То есть, реализация может потребовать существования сущности, необязательной на абстрактном уровне (например, *службы транспорта сообщений*), и дальнейшего уточнения функций и интерфейсов, которые элемент должен иметь в данной реализации.
- Возможна реализация всей архитектуры или только одного элемента
  - Например, может быть создан ряд конкретных спецификаций, описывающих, как представить архитектуру в терминах конкретного языка программирования, используя основанный на сокетах транспорт сообщений. Сообщения обрабатываются как объекты этого языка

- С другой стороны, отдельный конкретно определенный элемент может быть затем использован в ряде различных систем.
  - Например, если была создана конкретная спецификация для *службы-каталога-агентов* (agent-directory-service) на основе реализации LDAP, такой конкретный элемент может использоваться в ряде различных агентных систем.

Отношение между элементами Абстрактной и Конкретных Архитектур



- Диаграмма показывает иерархические отношения между абстракциями, определенными в стандарте и элементами конкретных экземпляров
- Общий подход FIPA Abstract Architecture базируется на ООП, включает использование шаблонов проектирования и моделирования с использованием UML
- Элементы архитектуры следует рассматривать как *множество абстрактных объектных классов*, которые могут быть основой высокоуровневого проектирования конкретных реализаций.
  - хотя архитектура явно избегает любых конкретных моделей композиции элементов, ее естественное представление – множество объектных классов, образующих агентную платформу, поддерживающую агентов и сервисы.

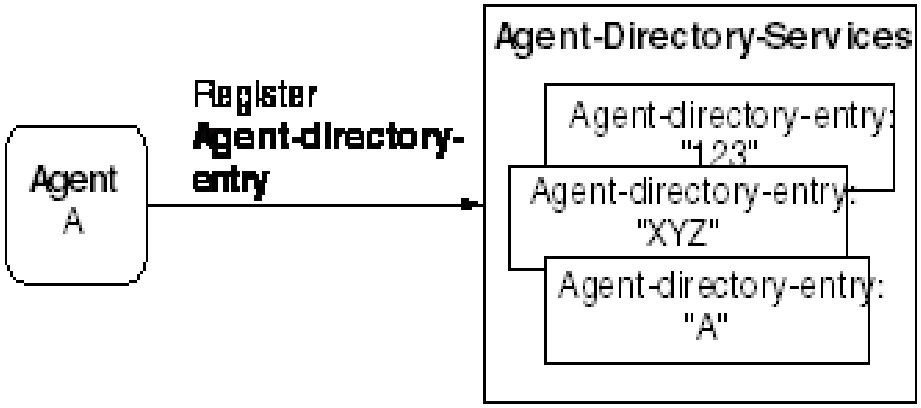
Регистрация агента в Службе Каталога (Agent-Directory-Service)

- Если **`Агент А`** желает объявить (прорекламировать) себя как поставщика определенной услуги, он должен:

1. Привязаться (binds) к одному или более транспортам.
   - В некоторых реализациях он будет делегировать эту задачу службе транспорта сообщений (message-transport-service), в других он может сам работать на более детальном уровне, например, контактировать с ORB, регистрироваться в реестре RMI или установить себя слушателем очереди сообщений.

   В результате этих действий агент становится адресуемым посредством одного или более транспортов.

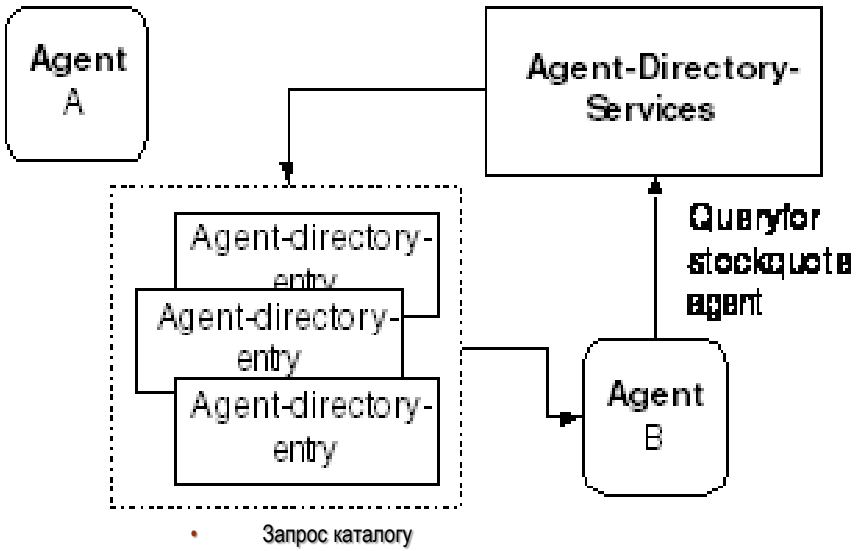
2. Объявить (прорекламировать) свое присутствие. Он выполняет это, формируя *запись агента в каталоге* (agent-directory-entry) и регистрируя ее в *агентной службе каталога* (agent-directory-service)



- *Запись агента в каталоге* включает:
  - имя (**agent-name**);
  - местоположение (**agent-locator**);
  - дополнительные атрибуты, описывающие службу;

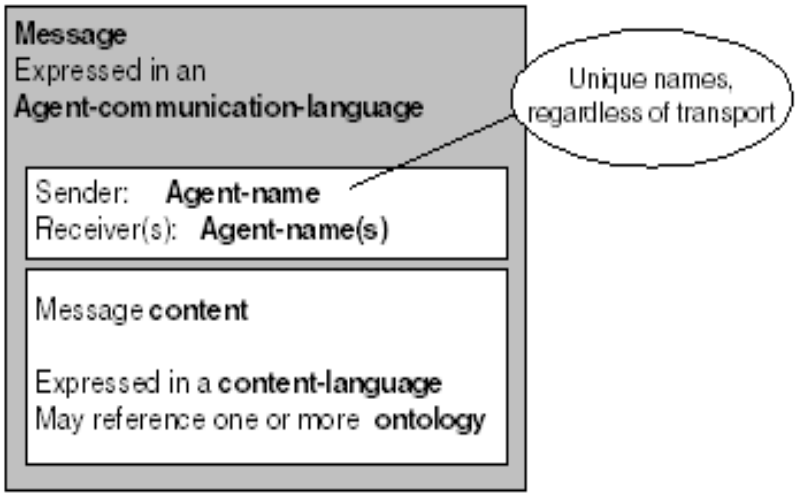
Обнаружение (Discovering) Агента

- Агенты могут использовать *службу каталога агентов* для поиска других агентов, с которыми можно общаться;
- Если агенту B (рис.) нужны данные о котировках акций, он может искать агента, который объявил об использовании соответствующей онтологии (StockQuote).
- Это включает поиск *записи агентного каталога* (agent-directory-entry), которая включает в себя пару *ключ-значение* {ontology, {com, dowjones, ontology, stockquote}}.
- Если это удастся он получит *запись агентного каталога* (agent-directory-entry) для агента A.
- Он может также получить другие *записи агентного каталога* об агентах, которые поддерживают эту онтологию



- Агент B может затем изучить возвращенные *записи агентного каталога*, чтобы определить, какой агент наилучшим образом соответствует его потребностям
- *Записи агентного каталога* включают имя агента, местоположение агента (**agent-locator**), содержащее информацию о том, как общаться с агентом, и другие дополнительные атрибуты

Сообщения агентов



• В FIPA-совместимых агентных системах агенты общаются друг с другом путем посылки сообщений.

•Три фундаментальных аспекта обмена сообщениями между агентами:

- структура сообщения
- представление сообщения
- транспорт сообщения

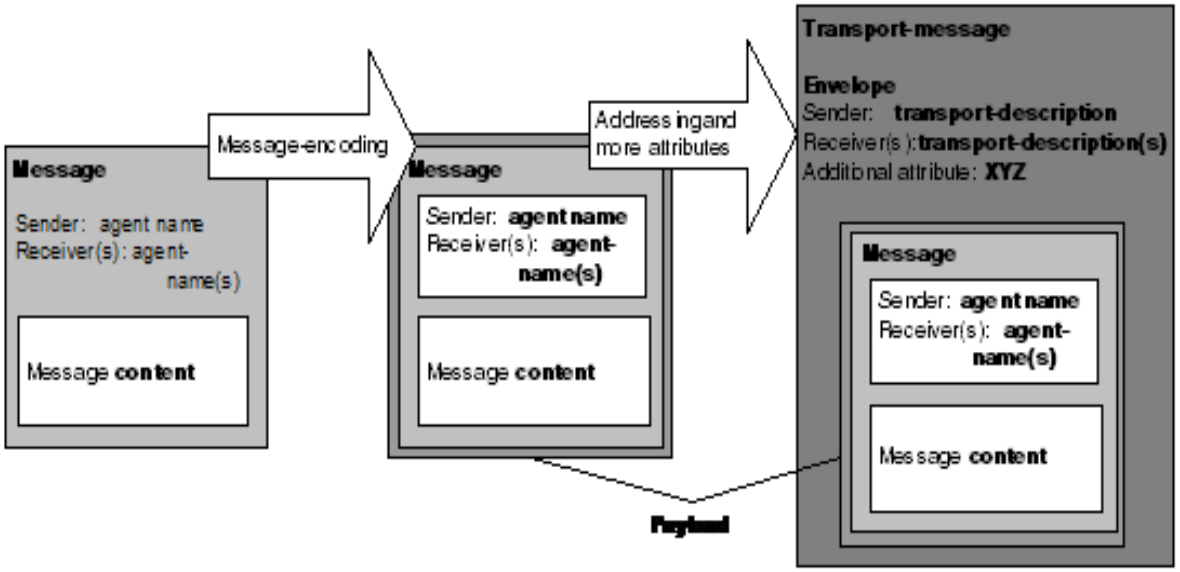
• Сообщение записывается в языке общения агентов (agent-communication-language), таком как FIPA ACL.

• Структура сообщения представляет множество пар «ключ-значение» (**key-value-tuple**).

• Содержание (content) сообщения выражается в языке содержания (content-language), таком как KIF или SL.

• Выражения **содержания** могут базироваться на онтологиях, на которые имеется ссылка в поле **ontology**. Сообщение также содержит имена **sender** и **receiver**, представленные **agent-names. Agent-names** – уникальное имя, идентифицирующее агента. Каждое сообщение имеет одного отправителя и ноль или более получателей. Случай с нулем получателей соответствует широковещательному режиму сообщений (как в ad-hoc беспроводных сетях).

Транспорт сообщений: Преобразование исходного сообщения в транспортное

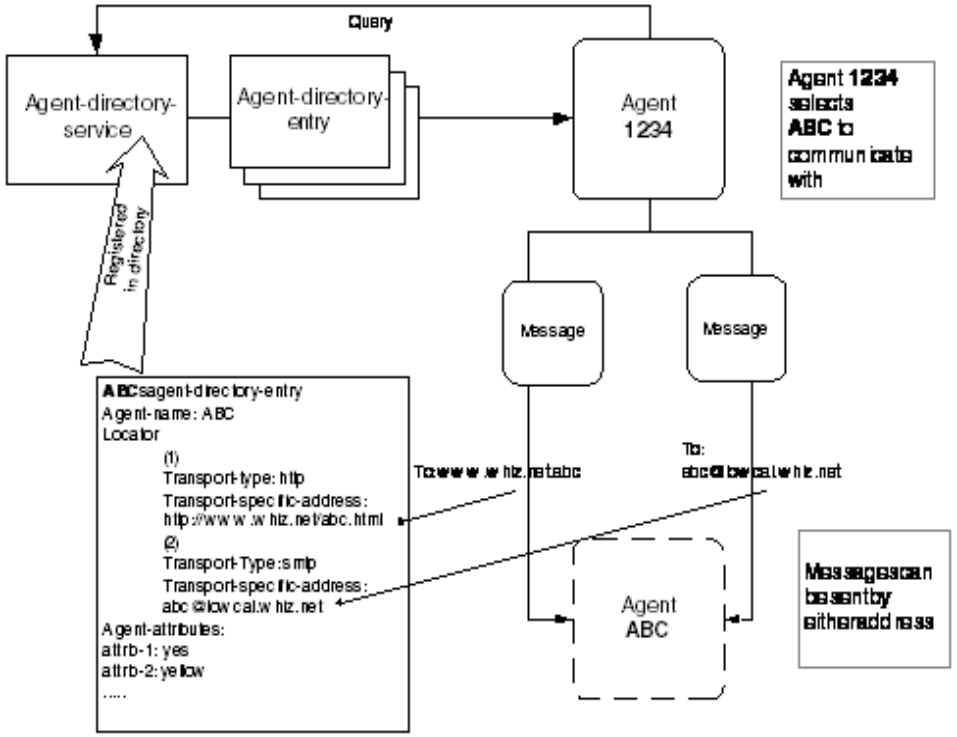


Транспортное сообщение =
Полезная нагрузка + Конверт

1. Сообщение кодируется в полезную нагрузку (**payload**), пригодную для выбранного транспорта сообщений (**message-transport**).
   - Полезная нагрузка (**payload**) *ничего не добавляет* к сообщению, а *только кодирует* его в другое представление. Например, если сообщение предполагается отправить по каналу с низкой пропускной способностью (например, беспроводному), может использоваться более эффективное (компактное) представление, вместо строкового.

2. Создается **конверт** (**envelope**), содержащий описание транспорта (**transport-description**) отправителя и получателя (в соответствии с выбранным транспортом). Описание транспорта содержит информацию: посредством какого транспорта посылать сообщение, по какому адресу, детали о том как использовать транспорт
   - в конверт могут включаться дополнительные данные: представление кодирования (**encoding-representation**), безопасность данных и другие специфичные для реализации данные, которые должны быть «видны» транспорту или получателю.

Отправка сообщений с использованием различного транспорта



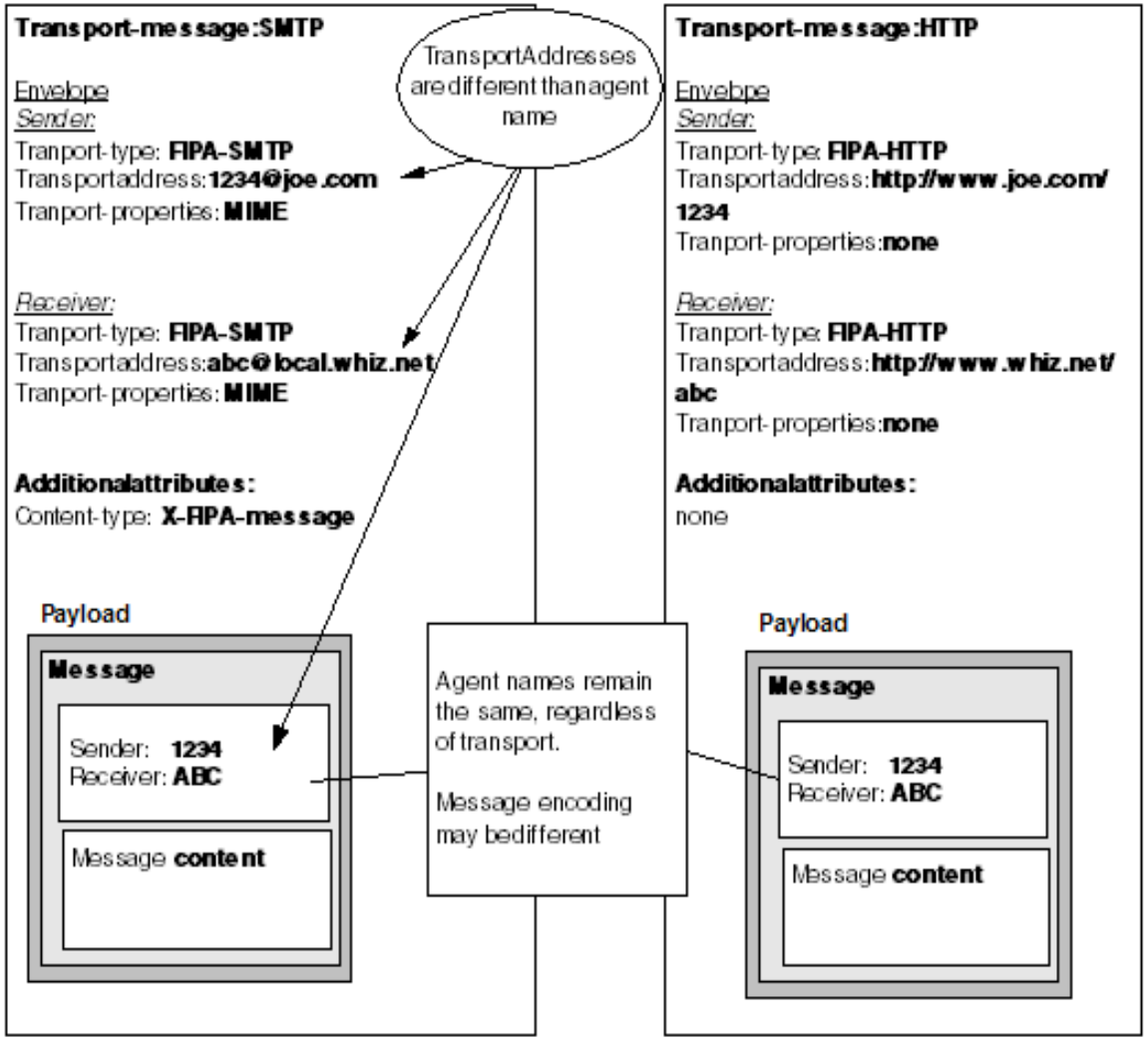
- Каждый агент:
  - имеет уникальное и неизменное имя (**agent-name**) и одно или более описание транспорта (**transport-descriptions**), используемых другими агентами для посылки транспортных сообщений (transport-message);
  - каждое *описание транспорта* связано с конкретной формой транспорта сообщений, такой как IIOP, SMTP или HTTP. Транспорт – механизм для передачи сообщений. Транспортное сообщение – сообщение, посылаемое одним агентом другому в формате (или кодировке) соответствующем используемому транспорту. Множество transport-descriptions может добавляться в agent-locator.
- Например, агент с agent-name “ABC” может быт адресуем посредством двух транспортов: HTTP и SMTP. Т. о. агент имеет два transport-descriptions, содержащихся в agent-locator.

Описания транспорта:
Запись в каталоге (Directory entry) для агента ABC
Agent-name: ABC
Agent Locator:

Transport-type	**Transport-specific-address**	**Transport-specific-property**
HTTP	http://www.*whiz.net/abc*	(none)
SMTP	Abc@lowcal.*whiz.net*	(none)

Agent-attributes:
Attrib-1: yes
Attrib-2: yellow
Language: French, German, English
Preferred negotiation: contract-net

Два различных транспортных сообщения одному агенту

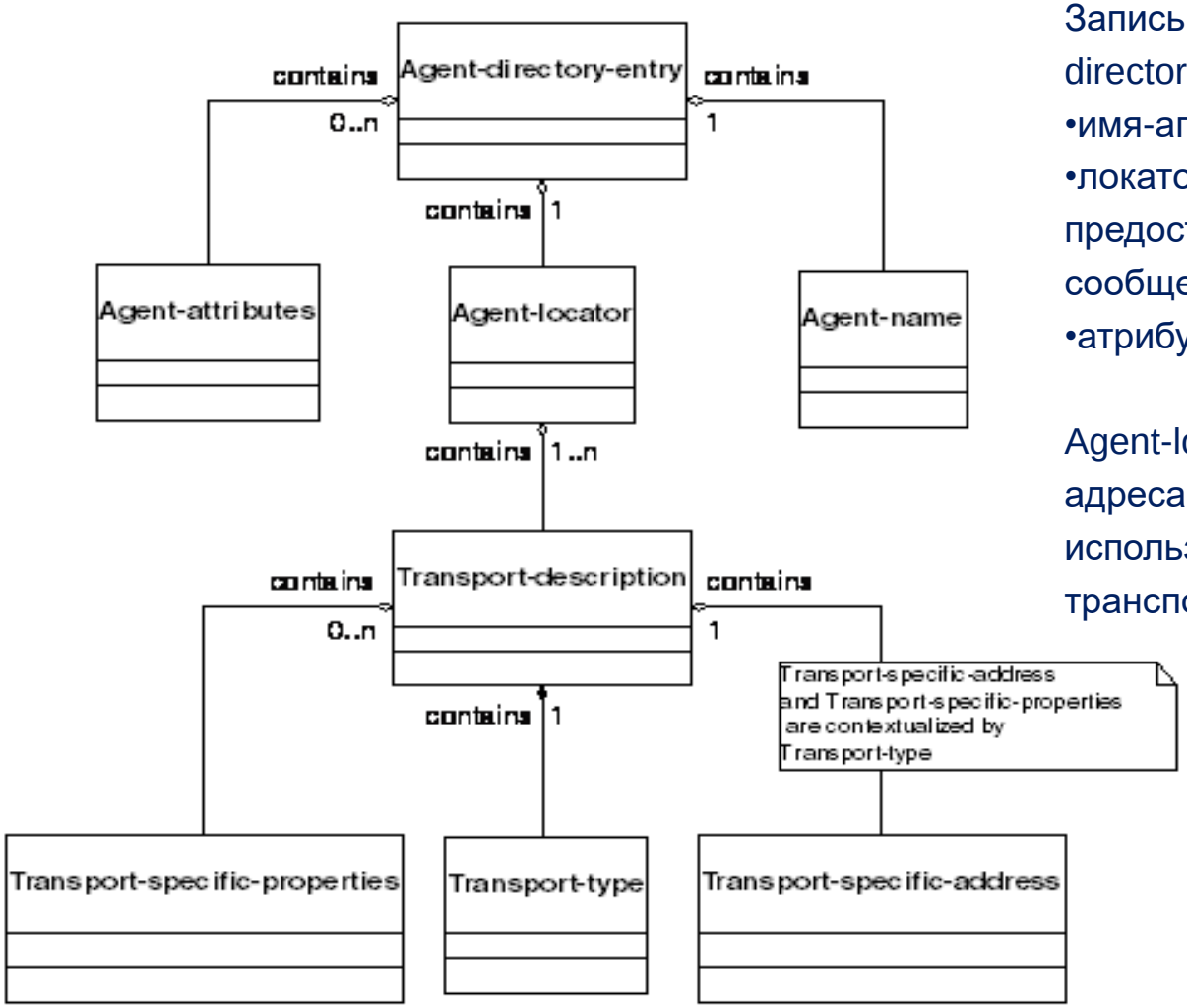


- *Описание транспорта* (transport-description) отличается в зависимости от транспорта, который планируется использовать

- Аналогично может отличаться *кодирование сообщения полезной нагрузки*

- Однако, ***имена агентов остаются неизменными*** для двух представлений сообщений

Диаграмма отношений между записями в каталоге агентов



Запись агента в каталоге (agent-directory-entry) содержит:

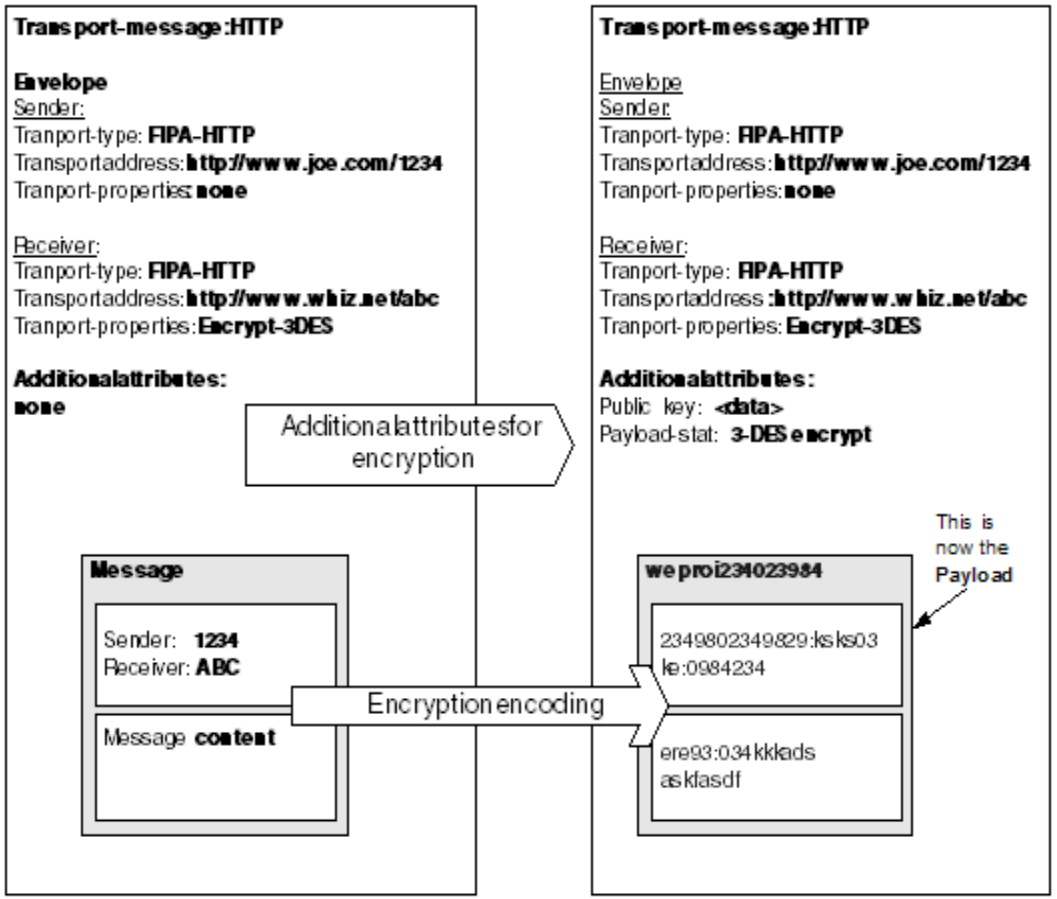
•имя-агента (Agent-name);

•локатор агента (Agent-locator) – предоставляет способы адресации сообщений агенту;

•атрибуты-агента.

Agent-locator предоставляет способы адресации сообщений агенту и используется для изменения транспортных запросов

Обеспечение валидности и шифрование сообщений



- Проблема безопасности в агентных системах имеет много аспектов.
- В FIPA Abstract Architecture рассматривается простая форма безопасности:
- *валидирование* сообщения и кодирование сообщения;
- при *валидировании* сообщения посылаются так, что любая их модификация во время передачи является идентифицируемой;
- при *шифровании* сообщение посылается в зашифрованной форме, не авторизованные сущности не могут «понять» содержание сообщения.

- В FIPA Abstract Architecture эти функции реализованы через кодирование представления и использование дополнительных атрибутов конверта.
  - Например, в процессе кодирования полезной нагрузки, одним из вариантов кодирования может быть зашифрованное представление, с использованием открытого ключа и предпочтительный алгоритм шифрования. Для указания этих характеристик в конверт добавляются дополнительные параметры

Формат сообщения на языке FIPA-ACL

```
( <Тип сообщения>
    :sender              // Отправитель сообщения
    :receiver            // Получатель(и) сообщения
    :content             // Содержание сообщения
    :reply-with          // Метка исходящего сообщения
    :in-reply-to         // Ссылка на входящее сообщение
    :replyBy             // Лимит времени на ответ
    :language            // Язык сообщения
    :ontology            // Онтология
    :protocol            // Используемый протокол общения
    :conversation-id     // Идентификатор разговора
)
```

Типы коммуникативных актов FIPA-ACL: **Inform** и **Request**

- **Inform** и **Request** - два базовых коммуникативных акта (performatives) в FIPA ACL.
  - все остальные типы коммуникативных актов (КА) являются *макро*определениями, определенными в терминах этих двух КА

- Значение “Inform” и “Request” определяется через:
  - *предусловия*, которые должны быть истинны для того, чтобы КА достигал цели;
  - *рациональный эффект* (“rational effect”) – чего надеется достичь отправитель сообщения.

FIPA-ACL: Семантика коммуникативного акта **inform**

Для КА “inform”:

- Содержанием является ***утверждение*** (*statement*)
- Предусловие состоит в том, что отправитель:
  - Считает содержание истинным;
  - Имеет намерение, чтобы получатель верил (believe) содержанию;
  - Пока не убежден в том, что получатель осведомлен о том истинно содержание или нет;

Формальная семантика КА **inform**

Агент *i* информирует агента *j* о содержании ϕ :

$< i, inform(j, \phi)>$

FP: $B_i (\phi) \wedge \neg B_i (B_i f_j (\phi) \vee U_i f_j (\phi))$

RE: $B_j (\phi)$

FP (*feasibility preconditions*) – предусловия выполнимости, описывают необходимые условия для отправителя КА.

RE (*rational effect*) – предусловие выполнимости

Пример сообщения **inform** на языке FIPA-ACL

```
(inform
   :sender agent1
   :receiver agent5
   :content (price good200 150)
   :language sl
   :ontology hpl-auction
)
```

FIPA-ACL: Семантика коммуникативного акта **request**

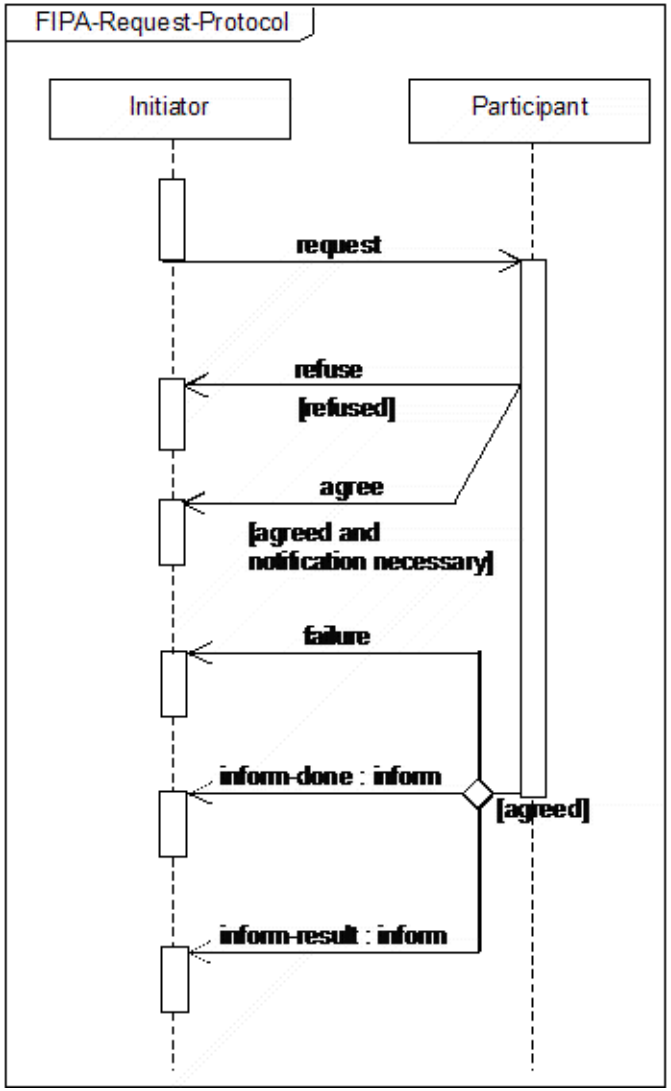
Для КА “request” :

- Содержанием является ***действие*** (*action*);
- Предусловие состоит в том, что *отправитель*:
  - Имеет намерение, чтобы действие в содержании было выполнено;
  - Считает, что получатель способен выполнить это действие;
  - Не убежден, что получатель уже намеревается выполнить это действие;

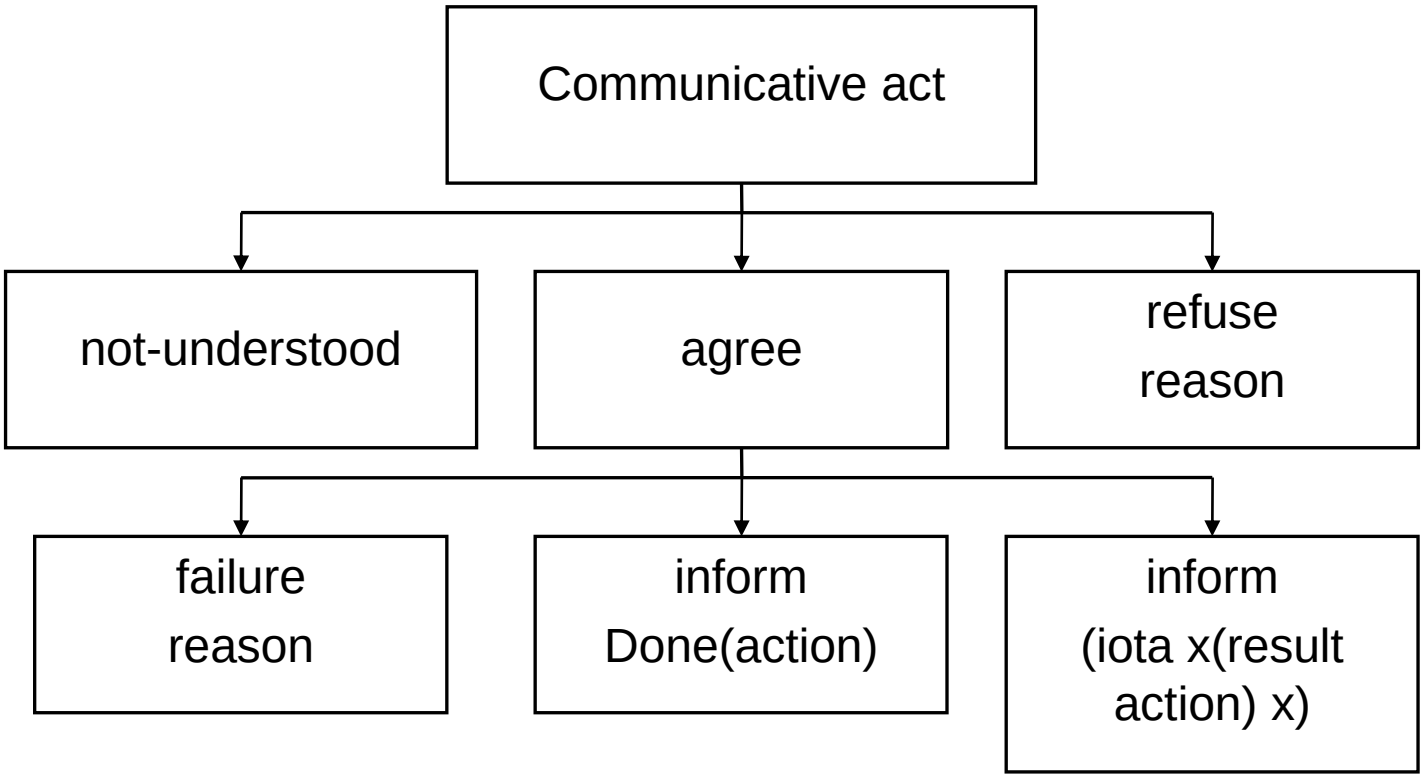
Пример сообщения **request** на языке FIPA-ACL

```
(REQUEST
    :sender (agent-identifier :name SendAg@host:1099/JADE)
    :receiver (set (agent-identifier :name RecvAg@host:1099/JADE))
    :content "((action
                (agent-identifier :name RecvAg@host:1099/JADE)
                (REGISTER :student (STUDENT :name Ivanov
                                            :groupnumber \"8307\")
                                  :course (COURSE :name \"MAS Course\"
                                  :instructor (INSTRUCTOR :name Panteleev
                                                          :dept CS)))))"
    :language  fipa-sl0
    :ontology  eLearning
)
```

Протокол общения агентов **FIPA-Request**



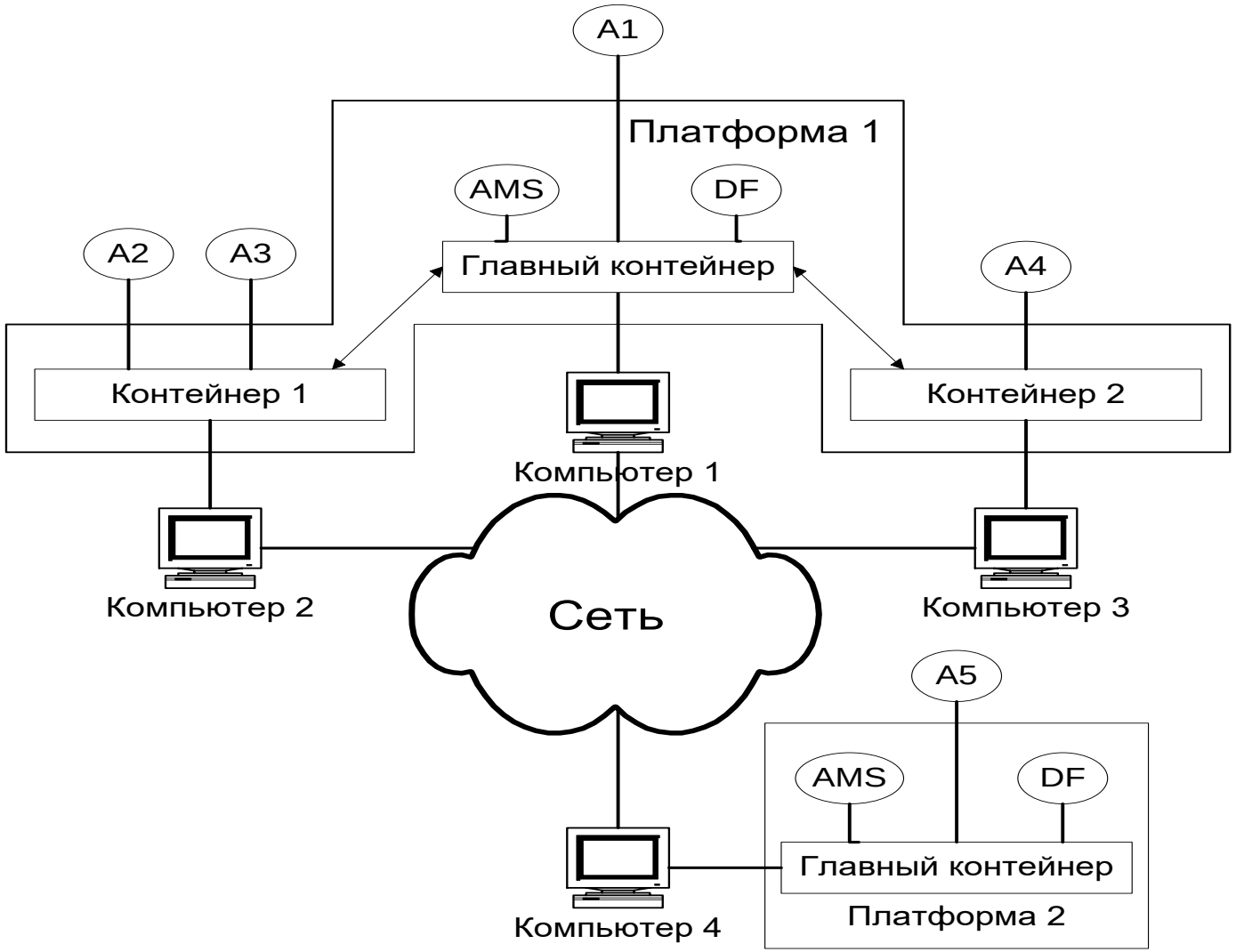
Общая структура протокола взаимодействия FIPA



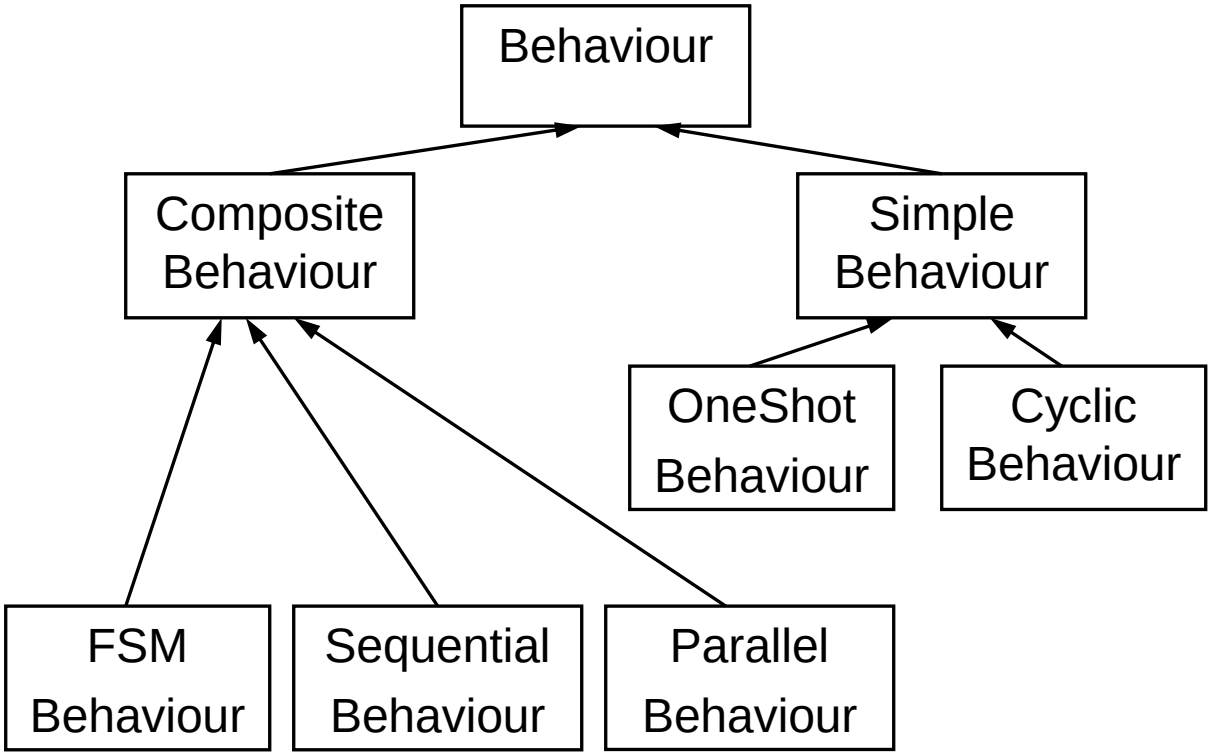
FIPA-совместимые агентные платформы:

- JADE – (http://sharon.cselt.it/projects/jade)
- FIPA-OS – (www.nortelnetworks.com/fipa-os)
- Zeus – (http://innovate.bt.com/projects/agents.htm)
- . . .

Платформы и контейнеры в JADE



Иерархия классов Behaviour



Иерархия классов Behaviour