

Методологии и инструментальные средства проектирования МАС (Конспект)

Расширение практического использования ИА и МАС стимулировало разработку методологий и инструментальных средств проектирования систем данного класса. Методологии проектирования агентно-ориентированных систем (АОС), с одной стороны, базируются на идеях ИИ, а с другой – часто являются прямыми расширениями существующих методологий объектно-ориентированного проектирования (ООП). На рис. 2.3 представлено генеалогическое дерево десяти наиболее известных в настоящее время методологий агентно-ориентированного проектирования (АОП) и их связь с методологиями ООП.

Ряд методологий АОП являются прямыми потомками методов ООП. Так

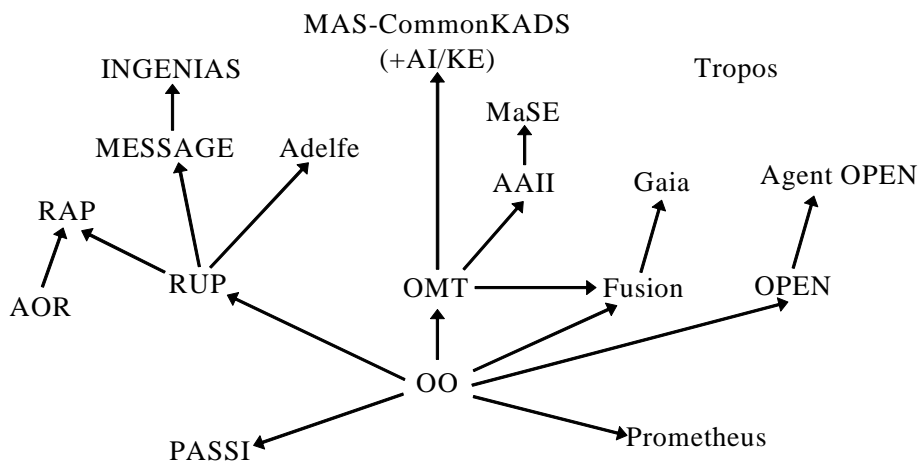


Рис. 2.3. Генеалогическое дерево методологий АОП и ООП

методология MaSE связана с методологией AAI, которая, в свою очередь, создавалась под сильным влиянием методологии ООП OMT.

На проектирование методологии Gaia существенное влияние оказала методология ООП Fusion.

Основой для методологий ADELFE и MESSAGE послужил широко известный унифицированный рациональный процесс RUP. Методология MESSAGE, в свою очередь, явилась основой для методологии INGENIAS. Кроме того, RUP, наряду с AOR, использовался в качестве базы при разработке методологии RAP.

В [92] рассмотрено агентно-ориентированное расширение ООП OPEN, названное Agent OPEN.

Методология MAS-CommonKADS в наибольшей степени основана на ИИ, но связана и с ОО методологией OMT.

Методология Prometheus, хотя и не является прямым потомком конкретной методологии ООП, предполагает использование объектно-ориентированных диаграмм и понятий при согласовании с агентно-ориентированной парадигмой. Методология PASSI использует в качестве основной нотации UML и объединяет идеи ООП и МАС.

Несколько особняком от других стоит методология Tropos.

При анализе методологий АОП целесообразно выделить 4 уровня (аспекта) рассмотрения агентных систем (рис. 2.4).

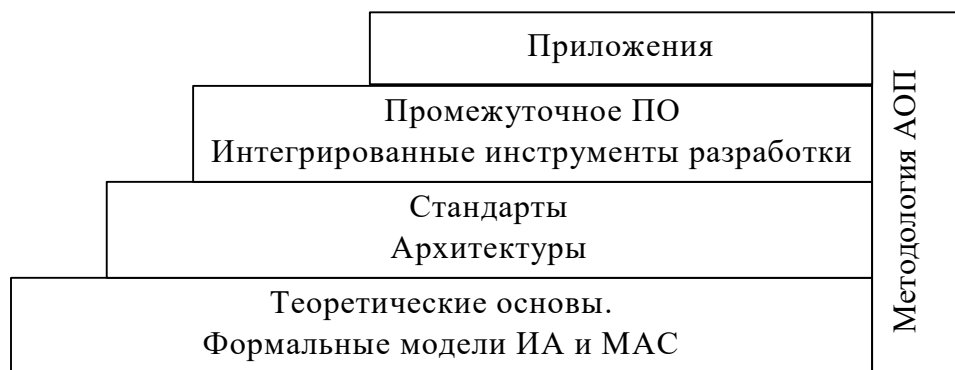


Рис. 2.4. Уровни рассмотрения АОС, связанные с методологией АОП

Фундаментом процесса проектирования всегда является та или иная теория построения АОС, включающая концептуальные и формальные модели ИА и МАС. Для обеспечения интероперабельности агентов необходимы стандарты, которые, в свою очередь, являются основой для создания инфраструктуры (платформ) и инструментальных средств разработки агентных систем. *Приложения* агентных систем разрабатываются с использованием инструментальных средств и исполняются в рамках соответствующей инфраструктуры.

Рассмотрим ряд наиболее известных методологий АОП.

Методология MaSE (Multiagent System Engineering) является развитием объектно-ориентированного подхода. В отличие от объектов, взаимодействие которых сводится к вызовам методов друг у друга, агенты взаимодействуют посредством высокоуровневой коммуникации с использованием речевых актов.

MaSE определяет последовательный процесс проектирования агентных систем с описанием порождаемых на каждом шаге артефактов (рис. 2.5).

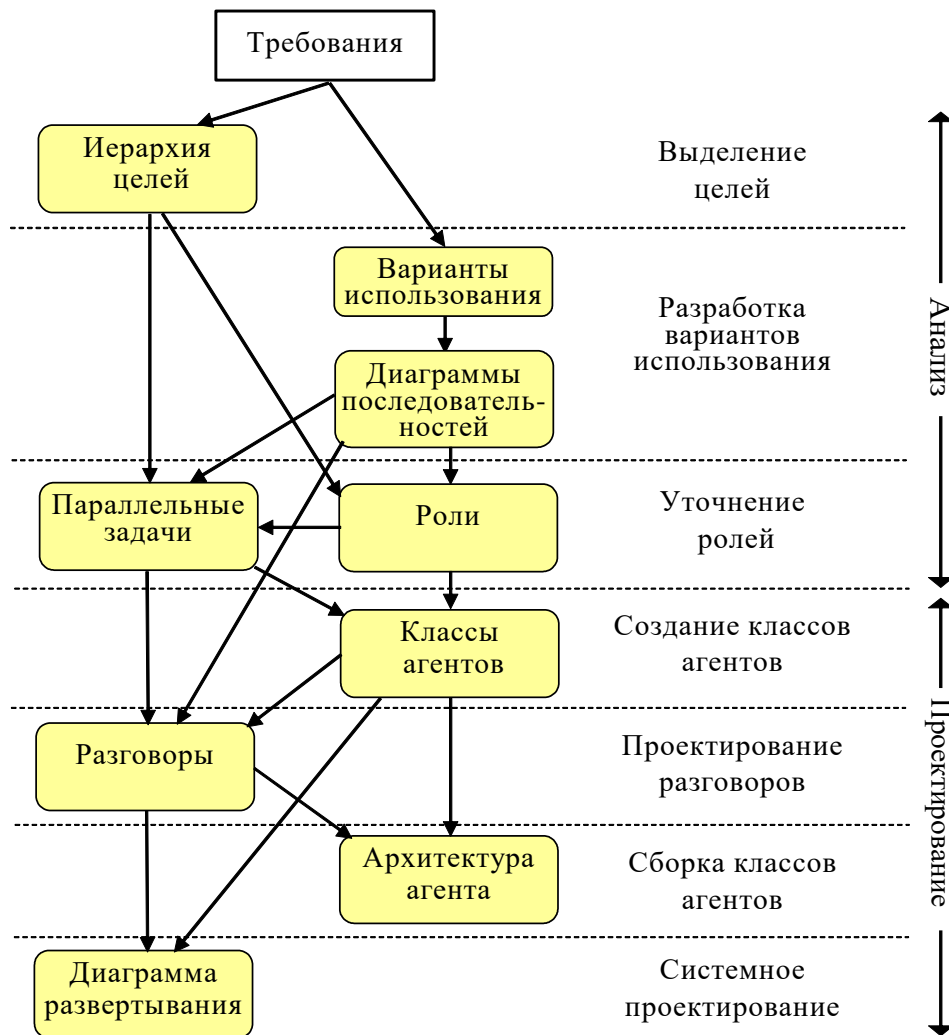


Рис. 2.5. Методология MaSE

На первом этапе специфицируются требования к системе. На этапе анализа порождаются: иерархия целей, диаграммы вариантов использования и описание ролей. Описания этапа проектирования включают: классы агентов, описание взаимодействий агентов, описание всей совокупности агентов и диаграмму развертывания.

Процесс проектирования является итеративным, уточнение деталей на любом этапе влечет уточнение на всех последующих этапах.

Цель *этапа анализа* – идентификация задач, определяющих в совокупности системные требования, и ролей, которым назначается выполнение задач. В методологии MaSE этот этап состоит из трех шагов: *определения целей, вариантов использования и ролей.*

Цель в MaSE понимается как абстракция, идентифицирующая совокупность функциональных требований. Помимо выделения целей на данном этапе выполняется их иерархическое упорядочение на основе отношения *цель-подцель*.

Варианты использования описывают функциональное назначение системы. Это описание выполняется в двух плоскостях: описание структуры (иерархии) вариантов использования и описание для каждого варианта использования его динамики в форме диаграммы последовательностей. Диаграммы последовательностей отражают роли и сообщения, которыми обмениваются агенты при взаимодействиях.

Уточнение ролей предполагает функциональную декомпозицию системы путем выделения ролей и ассоциированных с ними задач. С каждой ролью связываются одна или более целей из числа выделенных на первом шаге, а также реализующие каждую цель задачи.

Цель *этапа проектирования* – определить, каким образом система должна быть реализована и как она должна достигать своих целей. Этот этап в MaSE состоит из следующих шагов: *создание классов агентов, проектирование разговоров, сборку классов агентов и системное проектирование*.

При создании классов агентов определяется общая архитектура системы в терминах агентов и диалогов между ними. Классы агентов создаются для ролей, определенных на этапе анализа. Каждый агент ассоциируется, как минимум, с одной ролью.

При проектировании разговоров определяются протоколы взаимодействия между парами агентов, причем каждое взаимодействие отражается парой диаграмм классов. Первая диаграмма отражает поведение инициатора диалога, вторая – реакцию отвечающего.

Сборка классов агентов предполагает моделирование внутренней архитектуры агентов.

На шаге системного проектирования определяется физическая архитектура системы.

Методология MaSE поддерживается библиотекой agentMOM, которая может использоваться для генерации кода. Кроме того, в инструментальном плане MaSE реализована в продукте Agent Tool. Этот продукт предоставляет возможности для построения и анализа диаграмм MaSE, автоматического преобразования моделей и генерации кода заглушек.

Методология *PASSI* (Process for Agent Societies Specification and Implementation) объединяет концепции объектно-ориентированного

проектирования и подходы ИИ посредством нотации UML [200, 81]. Эта методология регламентирует создание пяти моделей в рамках трех пространств проектирования и включает 12 шагов построения МАС. Пространства проектирования являются (рис. 2.6):

- пространство задачи (Problem Domain);
- пространство решений (Solution Domain);
- пространство агентов (Agency Domain).

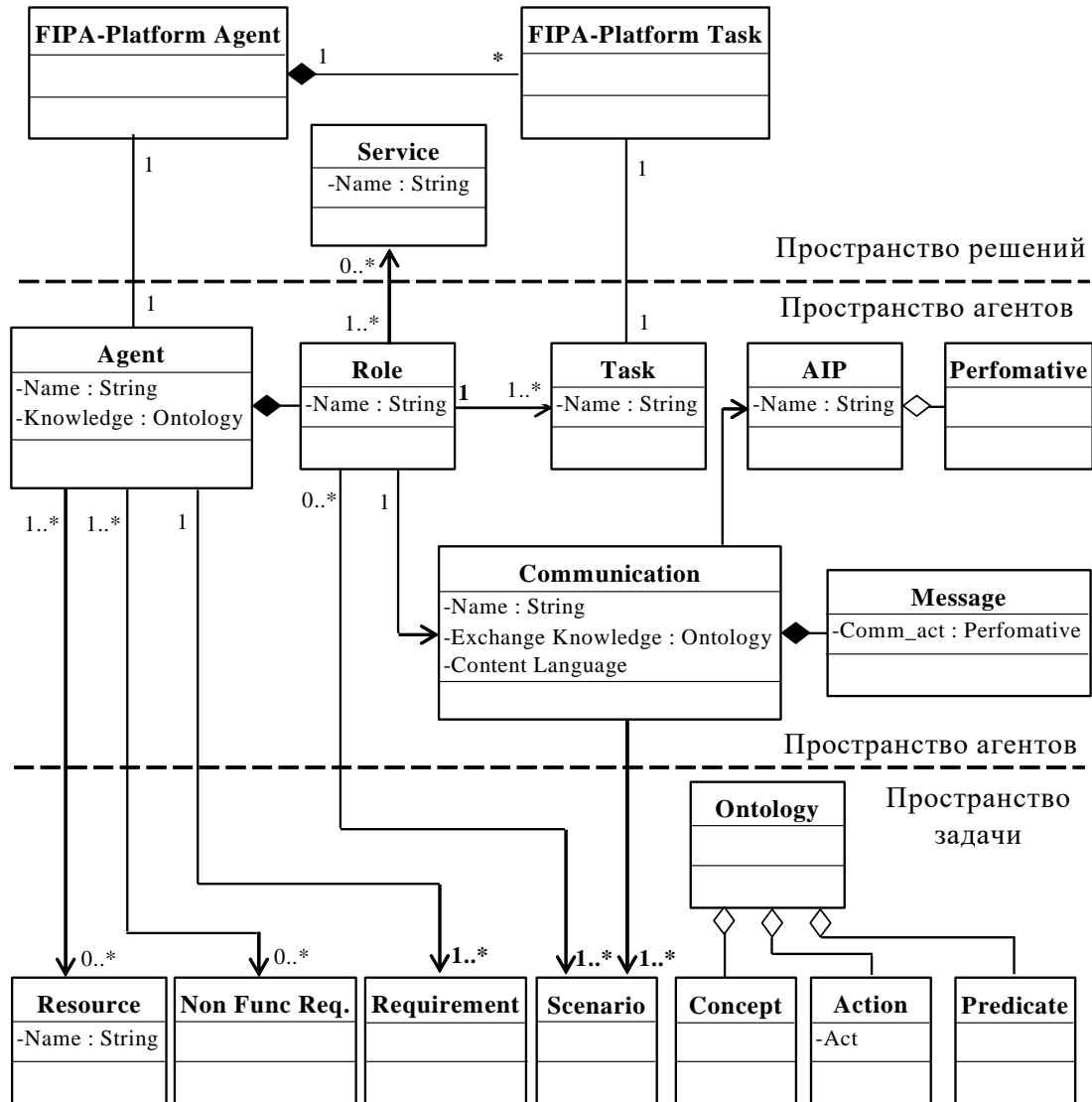


Рис. 2.6. Пространства проектирования в методологии PASSI

Ключевым для определения агентного подхода к проектированию является пространство агентов, которое используется для определения структуры решения задачи в терминах ролей, коммуникаций и агентов.

Пространство задачи описывает проблему в терминах требований, онтологий, ресурсов и сценариев. Сценарии описывают последовательности взаимодействий системы с пользователями. Онтологии определяют понятия, действия и отношения, существующие в проблемной области.

Пространство агентов содержит элементы агентного подхода (агент, роль, задача) и определяет в этих терминах схему решения. Вопросы реализации элементов на данном уровне не рассматриваются. На последующих этапах все элементы уровня агентов транслируются в элементы уровня решения.

Пространство решений описывает реализуемые сущности в терминах FIPA-совместимой платформы реализации решения. Необходимыми элементами данного уровня являются: агент (FIPA-Platform Agent), задача (FIPA-Platform Task) и сервис (Service), описывающий набор реализуемых агентом функций.

Пять моделей, регламентируемых методологией, позволяют на основе системных требований определить структуру, состав и функциональность МАС. Эти модели включают:

- модель системных требований (System Requirements Model);
- модель сообщества агентов (Agent Society model);
- модель реализации агентов (Agent Implementation Model);
- модель кода (Code Model);
- модель развертывания (Deployment Model).

Модель системных требований в методологии PASSI является высокоуровневым описанием требований в терминах агентов и их задач. Она отражает четыре аспекта целевой агентной системы: системные требования; функциональность агентов; роли, назначаемые агентам, и задачи, выполняемые агентами в соответствии с ролями.

Системные требования формулируются в виде диаграмм вариантов использования. Функциональность агентов задается совокупностью вариантов использования, сгруппированных по пакетам. Каждый пакет соответствует агенту и снабжается стереотипом, указывающим на эту связь. Для задания ролей агентов применительно к описанным сценариям используются диаграммы последовательностей, формируемые на основе диаграмм использования.

В разных сценариях агент может играть разные роли. Задачи агентов специфицируются в терминах состояний и переходов на диаграммах состояний-переходов. Для каждого агента создается своя диаграмма

состояний-переходов, полностью описывающая его поведение. Каждый переход в диаграмме отражает задачу (метод) агента.

Модель сообщества агентов описывает взаимосвязи агентов в МАС. Ее построение включает пять шагов, в процессе которых определяются:

- онтологии системного уровня;
- онтологии межагентных взаимодействий;
- сервисы, предоставляемые агентами;
- коммуникации агентов (в терминах онтологий, протокола взаимодействия агентов и языка описания содержания);
- протоколы взаимодействия агентов.

Онтология системного уровня описывается в терминах диаграмм классов с использованием классов предопределенных стереотипов: *concept*, *predicate*, *action*. Стереотип определяет семантику сущности: концепты (*concepts*) определяют онтологические понятия, предикаты (*predicates*) определяют свойства и взаимосвязи объектов, а действия (*actions*) определяют способы использования понятия. Описание онтологии может быть преобразовано в RDF формат, что обеспечивает возможность обмена такими описаниями.

Онтологии межагентных взаимодействий также описываются в терминах диаграмм классов, но с использованием иных предопределенных стереотипов: *agent*, *communication*. При этом сущность со стереотипом *agent* определяет агента и знания, необходимые ему для коммуникаций (они задаются атрибутами класса). Взаимодействие описывается классом со стереотипом *communication*. Такое описание связывается с направленной ассоциативной связью пары агентов. Оно включает в себя ссылку на сущность-действие из онтологии системного уровня, язык и протокол взаимодействия.

Коммуникации агентов описываются в терминах диаграмм классов, в которых ранее выделенные и представленные соответствующими классами роли группируются в пакеты, описывающие агентов. Диаграмма детализирует взаимодействия между агентами, определяя роли в тех или иных типах взаимодействий. Кроме того, диаграмма специфицирует процесс смены ролей агента.

По умолчанию предполагается использование FIPA-совместимых протоколов взаимодействия агентов. При необходимости описания нового протокола используется язык AUML.

Модель реализации агентов описывает архитектуру решения в терминах классов и методов. Она строится за четыре шага, в процессе которых специфицируются:

- структура сообщества агентов (МАС);
- структура отдельных агентов;
- поведение агентов на уровне сообщества;
- поведение агентов на уровне отдельных агентов.

Структура уровня сообщества агентов описывается в терминах диаграммы классов, при этом строится одна диаграмма для всех агентов. Кроме того, для каждого агента в отдельности строится диаграмма, описывающая его структуру. Совокупность двух описаний полностью определяет структуру решения на уровне классов.

Поведение агентов описывается в терминах диаграмм активностей, на которых отражаются шаги, выполняемые агентами при исполнении конкретных методов; сообщения, посылаемые другим агентам, и действия, вызываемые данными сообщениями.

Конечный компилируемый код системы строится с использованием шаблонов кода и проектных спецификаций. Фактически, *модель кода* строится в два этапа, включающих выбор шаблонов и создание кода.

Использование шаблонов позволяет ускорить разработку и сократить число ошибок в системе. Методология PASSI регламентирует шаблоны следующих функциональных категорий:

- шаблон мобильности – описывает особенности агента, необходимые для обеспечения миграции с одной платформы на другую;
- шаблон коммуникаций – описывает особенности агентов, обеспечивающие возможность общаться с использованием стандартных протоколов;
- шаблон уточнения (elaboration) – описывает особенности агентов, обеспечивающие определенные релевантные операции работы с данными;
- шаблон доступа к локальным ресурсам – описывает особенности работы со стандартными источниками данных.

Структурно шаблоны разделяются на четыре категории:

- шаблон действия (action pattern) содержит методы классов агентов (agent class) или задач (task class);
- шаблон поведения (behavior pattern) – модель поведения агента, представленная специфичным, зависящим от конкретной агентной платформы классом;

- шаблон компонентов (component pattern) содержит прототип агента вместе с его моделями поведения;
- шаблон сервиса описывает модель взаимодействия между агентами и представляет собой агрегацию шаблонов компонентов.

Модель развертывания. Модель развертывания описывает распределение программных частей системы по вычислительным (аппаратным) компонентам сети и информационные потоки между агентами. Данная модель строится за один шаг и представляется в виде диаграммы развертывания.

Пять видов моделей отражают семантику, структуру и динамику проектируемой агентной системы. Последовательность создания моделей в соответствии с методологией PASSI представлена на рис. 2.7.



Рис. 2.7 – Последовательность этапов в методологии PASSI

Процесс проектирования разделяется на два этапа: этап логического проектирования и этап проектирования кода. Этап логического проектирования включает анализ требований, а также построение моделей системных требований, сообщества и реализации. На этапе проектирования кода выполняется анализ шаблонов кода с целью их повторного использования, построение кода и развертывание системы.

Использование в качестве базового языка UML позволило разработчиком PASSI опираться на существующие средства объектно-

ориентированного проектирования. Абстрагирование от конкретного языка реализации обеспечивает универсальность подхода.

Для инструментальной поддержки методологии PASSI разработано средство проектирования систем PASSI ToolKit. Данный инструмент состоит из двух частей:

1. Встраиваемого модуля (Plug-in) для Rational Rose;
2. Приложения для управления репозитарием шаблонов – AgentFactory.

PASSI ToolKit позволяет использовать шаблоны агентов и их действий, а также обеспечивает автоматическую генерацию части диаграмм, предусмотренных методологией, что существенно упрощает разработку модели. В табл. 2.2 перечислены шаги проектирования MAC с использованием PASSI ToolKit.

Методология ADELFE нацелена на проектирование MAC, способных адаптироваться к изменяющемуся окружению, что особенно важно в открытых системах таких, как Internet. Методология базируется на теории адаптивных MAC (AMAC), в соответствии с которой система строится из социально-ориентированных агентов, разрешающих свои задачи посредством взаимодействий.

Таблица 2.2

**Поддержка этапов проектирования АОС в инструментальной среде
PASSI Tool Kit**

Этапы проектирования	Функции, поддерживаемые PASSI Tool Kit
1. Описание предметной области	Создание заготовки проекта PASSI в Rational Rose модели
	Создание нового агента (скелетона) и назначение функций агентам
	Изменение существующих агентов (название и описание)
	Удаление существующего агента
	Добавление или удаление агентам вариантов использования
2. Идентификация агентов	Автоматическая сборка всех диаграмм фазы
	Удаление или изменение существующего агента
3. Идентификация ролей	Описание новой роли агента (автоматическое добавление агента в диаграмму)
4.	Создание задачи и назначение ее агенту

Специфицирование задач	Описание внешнего агента и назначение ему задач
------------------------	---

Продолжение табл. 2.2

Этапы проектирования	Функции, поддерживаемые PASSI Tool Kit
5. Описание онтологии	Автоматическое создание части онтологии взаимодействия
	Описание коммуникативных актов между агентами (установка языка онтологии, названия коммуникативного акта, участников и инициатора взаимодействия)
	Автоматическое создание RDF и XMI файлов с описаниями онтологии предметной области (Domain Ontology)
	Автоматическая генерация каталога типов данных в логическом представлении на основании онтологии предметной области
	Автоматическая генерация кода онтологии для платформ FIPA-OS и Jade
	Автоматическая генерация кода сообщений агентов
6. Описание ролей	Автоматическое создание части диаграмм описания ролей. В диаграмму добавляются: агенты, связанные с ними роли, взаимодействия между ролями и направления смены ролей
	Сопоставление взаимодействий между ролями и взаимодействий между агентами, описанными в онтологии взаимодействия
7. Описание протоколов	Средства поддержки не предусмотрены
8. Определение структуры агентов	Автоматическое создание диаграммы MAC и для каждого агента – диаграммы, отражающей его внутреннюю структуру
	Автоматическое обновление структуры агента в соответствии с добавляемыми задачами
9. Описание поведений агента	Добавление нового потока задач
	Добавление активности в методы существующих задач

10. Конфигурация развертывания	Добавление экземпляров агентов
	Назначение агентов на доступные вычислительные модули

ADELFE является развитием методов объектно-ориентированного проектирования, основанных на рекомендациях RUP, и использует для описания моделей языка UML и AUML.

Данная методология, как и PASSI, обеспечивает полный цикл проектирования агентных систем. Процесс проектирования в соответствии с ADELFE включает три этапа (рис. 2.8). На каждом этапе возможен возврат к предыдущему шагу с целью его уточнения.



Рис. 2.8. Этапы методологии ADELFE

Этап формирования требований является фундаментальным во всех методологиях проектирования программных систем и имеет целью определить разрабатываемую систему в терминах вариантов использования и упорядоченного множества функциональных и нефункциональных требований.

Моделирование среды включает детальное описание контекста системы, т. е. среды, в которой будут функционировать агенты. В первую очередь определяются взаимодействия системы с окружением в терминах активных или пассивных сущностей и ограничений. Во вторую очередь определяется контекст: описываются потоки данных между сущностями.

Описание ведется в терминах диаграмм взаимодействия и диаграмм последовательностей. Затем выполняется характеристика системы по классификации Рассела. Варианты использования определяются для уточнения функциональности системы, при этом учитываются только активные сущности.

Общей задачей этапа анализа является выделение агентов и определение необходимости применения теории АМАС.

На этапе анализа для каждого варианта использования определяются сценарии и роли, выполняющие в разработанных сценариях последовательности действий. После определения ролей на основе анализа их взаимодействий, мобильности, автономности и личных целей выделяются взаимодействующие агенты. Для каждого агента анализируются возможные ситуации взаимодействия с окружением.

На этапе проектирования проектируются составляющие систему программные модули. Для каждого агента строятся диаграммы классов, после чего строится общая модель, отражающая взаимосвязи между агентами.

На рис. 2.9 представлена метамодель мультиагентной системы, предложенная в методологии Adelfe.

Метамодель МАС в методологии Adelfe связывает агента со следующими сущностями:

- элементы окружения (Element), которые агент может воспринимать и на которые он может воздействовать;
- представление агента (Representation), определяющее способ взаимодействия с окружением;
- набор характеристик агента (Characteristics), включая его умения (Skills) и способности (Aptitudes);
- коммуникации, включая протоколы взаимодействия (AIP);
- правила сотрудничества (Cooperative rules) и описания некооперативных ситуаций.

Этапы реализации и тестирования, следующие за этапом проектирования, полностью наследуются из объектно-ориентированной технологии RUP.

Сравнительный анализ методологий. Подход FIPA Modeling TC.

В табл. представлено сравнение методологий проектирования АОС по критериям степени поддержки и удобства реализации различных шагов

процесса проектирования. Оценка дана по трехбалльной шкале: В – «высокая», С – «средняя», Н – «низкая».

Методологии проектирования АОС продолжают развиваться. Для создания единой метамодел и выработки единой методологии в 2003 г. в рамках FIPA создан специальный технический комитет – FIPA Methodology Technical Committee. За основу подхода была принята мета модель процесса разработки программных продуктов SPEM (Software Process Engineering Metamodel).

Сравнение методологий агентно-ориентированного проектирования

Шаги	Gaia	Tropos	MAS-CommonKADS	Prometheus	PASSI	Adelfe	MaSE	RAP	MESSAGE	INGENIAS
1. Идентификация целей системы		В	В	В			В	В	В	В
2. Идентификация задач и поведения системы	С	В	В	В	В	В	В	В	В	В
3. Специфицирование сценариев вариантов использования			В	В	В	В	В	В		В
4. Идентификация ролей	В				В		В	С	С	С
5. Идентификация классов агентов	В	В	С	В	С	В	В	Н	С	В
6. Концептуализация модели предметной области			С		С				С	
7. Специфицирование знакомств между классами агентов	С	С	В	В	В	С	В	С	С	В
8. Определение протоколов взаимодействия	В	В	В	В	В	С	В		В	В
9. Определение содержания сообщений между агентами		С	В	Н	В	С	В	С	Н	С
10. Специфицирование архитектуры агентов			Н	В	В	В	С		В	
11. Определение ментальных характеристик агента (целей, мнений, планов, обязательств и т. п.)		С	С	В	С	В	С	В	В	В
12. Определение поведенческого интерфейса агента (способностей, услуг, контрактов и т. п.)	В		Н	В	В	С				
13. Специфицирование архитектуры системы (обзор всех компонентов и их связей)			С	В	В	В			В	В
14. Специфицирование организационной структуры, режимов управления, социальных отношений между	В	В	Н						В	В

агентами										
15. Моделирование среды МАС (ресурсов, средств, характеристик)	С	В	С	С		В			Н	В
16. Специфицирование механизмов взаимодействия агент-среда				В		Н				В
17. Специфицирование наследования и агрегирования агентов	В		С			С				
18. Создание экземпляров классов агентов	С		Н	Н			В	Н	Н	
19. Специфицирование развертывания экземпляров агентов					Н		В			

