УДК 004.75

АВТОМАТИЗИРОВАННЫЙ СИНТЕЗ АГЕНТОВ ПРИ СОЗДАНИИ МУЛЬТИАГЕНТНЫХ СИСТЕМ

А.В. Прохоров
Кандидат технических наук, доцент*
Контактный тел. 8 (057) 707-43-02

Е.Н. Владимирская
Аспирант*
*Кафедра информационных управляющих систем
Национальный аэрокосмический университет
им. Н.Е. Жуковского «ХАИ»
г. Харьков, Украина
Контактный тел.:8 (057) 717-26-61
Email: catherine.vladimirskaya@gmail.com

Розроблено метод синтезу агентів розподіленої інтелектуальної системи з використанням онтологічних моделей знань. Запропонована модель взаємодії агентів в мультиагентной системі та з зовнішніми ресурсами з використанням стандартів і технологій Semantic Web. Розроблена і формально описана в термінах дескриптивних логік модель знань агентів

Ключові слова: мультиагентна система, інтелектуальній агент, база знань, онтологія, OWL, SPARQL

Разработан метод синтеза агентов распределенной интеллектуальной системы на основе использования онтологических моделей знаний. Предложена модель взаимодействия агентов в мультиагентной системе и с внешними приложениями с использованием стандартов и технологий Semantic Web. Разработана и формально описана в терминах дескриптивных логик модель знаний агентов

Ключевые слова: мультиагентная система, интеллектуальный агент, база знаний, онтология, OWL, SPARQL

The agent synthesis method for the distributed intelligent system using the ontological knowledge models has been developed. The agent interaction model for the multi-agent system using Semantic Web standards and technologies has been suggested. The agent knowledge model has been developed and formally described using description logics

Keywords: multi-agent system, intelligent agent, knowledge base, Semantic Web, ontology, OWL, SPARQL

1. Введение

В настоящее время возрастает потребность в децентрализации бизнес-приложений с возможностью получения доступа с различных устройств, в то же время Интернет рассматривается в качестве общей платформы для ведения переговоров и взаимодействия в различных областях, в том числе и коммерческой сфере. В области распределенного искусственного интеллекта основное внимание исследователей сосредоточилось на мультиагентных системах (МАС), которые строятся из множества взаимодействующих агентов (зачастую представляющих собой полноценные интеллектуальные системы), совместно решающих поставленную задачу в распределенных средах [1, 2].

Существует множество методологий и средств, применяемых на разных стадиях проектирования и разработки МАС, которые выделяют различные аспекты построения МАС в качестве приоритетов. На данный момент все более актуальными становятся проблемы обеспечения гибкости, динамичности и легкого внесения изменений в разрабатываемую архитектуру, что влечет за собой необходимость автоматизации процессов создания и поддержки МАС. При этом необходимо учитывать сложность распределенной структуры и спонтанный характер процессов, происходящих в системе, согласование методов формирования онтологических моделей, проблемы обмена знаниями на различных уровнях представления и реализации распределенных механизмов использования этих знаний для решения различных задач.

2. Анализ последних исследований и публикаций

Спецификации FIPA[3] и другие современные источники[4,5] описывают, каким образом агенты должны обмениваться информацией, и как они могут использовать онтологии. Онтологическая база знаний[2] является основным элементом программного агента системы, дающим ему возможность принимать решения, планировать действия, взаимодействовать с другими агентами, содержащая модели концептуальных понятий, отношений предметной области и правила для анализа и ситуативной ориентации. В практических задачах часто возникает необходимость интеграции разрабатываемой базы знаний и механизма вывода с существующими в сети знаниями (Semantic Web), в частности на основе языка OWL[6], что позволяет осуществлять обмен данными и их многократное использование в различных приложениях и информационных системах. Подход AgentOWL[7] добавляет агентам механизмы работы с онтологической моделью знаний, основанной на OWL, однако для более эффективного использования требуется расширение данного подхода: добавление методов работы с онтологиями, а также автоматизация некоторых этапов создания МАС.

На данный момент разработано множество методологий [8,9], которые объединяют процессы разработки МАС в единую цепочку. Основной проблемой существующих методологий является отсутствие достаточного количества инструментальных средств, которые поддерживают все стадии разработки МАС и позволяют максимально автоматизировать используемый подход. В литературе описаны некоторые средства и методы для автоматизации процессов разработки МАС, которые используют UML и его расширение AUML[10]. При использовании онтологических моделей знаний ограничением этих подходов является то, что объектно-ориентированная модель не может охватить всю семантику используемых онтологий.

В работах [11] и [12] предлагаются основанные на OWL-онтологиях подходы проектирования и разработки МАС с элементами автоматизации некоторых этапов.

Эти работы и разработанное программное обеспечение тесно связаны с предметной областью – медицинскими проектами, и не могут быть использованы для более общих случаев проектирования и разработки МАС.

Таким образом, актуальной задачей в настоящее время является разработка комплекса моделей и информационной технологии создания распределенных интеллектуальных систем, на основе мультиагентного подхода и онтологических моделей знаний, обеспечивающих автоматизированный синтез агентов и эффективное использование онтологий для принятия решений. В настоящей работе предлагаются модели для построения мультиагентной системы и синтеза агентов для платформы JADE[13] с использованием стандартов Semantic Web.

Мультиагентная система предполагается для использования в сфере Интернет-приложений, где агенты взаимодействуют с различными ресурсами сети, такими как веб-службы, приложения Semantic Web и т.д.

3. Основные особенности разработки МАС и синтеза агентов на основе OWL-онтологий

Предлагаемый подход автоматизации создания МАС включает в себя синтез агентов и добавление им механизмов взаимодействия, основанного на онтологиях. Онтология представляет собой таксономию понятий, расширенную некоторыми правилами-аксиомами. Эти правила специфицируются на некотором языке представления знаний, а агенты могут использовать эти знания для логического вывода. Для описания знаний выбран язык OWL, базовым формализмом для которого являются дескриптивные логики [6, 14].

Рассмотрим основные этапы предлагаемого подхода. Этап **анализа** включает в себя: анализ требований к системе и описания предметной области, построение онтологических моделей, описывающих основные аспекты предметной области и взаимодействия агентов, а именно:

- онтология вариантов использования, которая содержит информацию о пользователях системы, и предоставляемых им возможностям;
- онтология предметной области.

Для автоматизации следующих этапов предлагается использование разработанных системных онтологий, которые являются повторно используемыми для различных предметных областей и проектов. Для удобства и обеспечения гибкости внесения изменений предлагается логическое разделение системных онтологий.

Можно выделить следующие основные системные онтологии для построения МАС:

- модель знаний агента;
- модель взаимодействия агентов в МАС и с внешними ресурсами;
- онтология платформы JADE, которая описывает элементы, специфичные для платформы построения агентов, а также элементы методологии построения МАС;
- виртуальная онтология, которая связывает все другие онтологические модели с использованием механизмов отображения онтологий и представляет собой расширения модели знаний агентов элементами и связями других онтологий, а также служит основой для синтеза агентов.

Первые две модели подробно описаны в данной работе. Они могут обновляться, однако изменения в них вносятся нечасто, по сравнению с изменениями в остальных моделях. Основные изменения в процессе функционирования системы отражаются в онтологической модели, которая включает онтологию предметной области и онтологию вариантов использования.

После изменения этих моделей происходит изменение виртуальной онтологии и генерация новой версии системы. Онтологию платформы Jade в случае необходимости можно заменить моделью другой платформы.

Основные связи между используемыми онтологическими моделями показаны на рис.1.

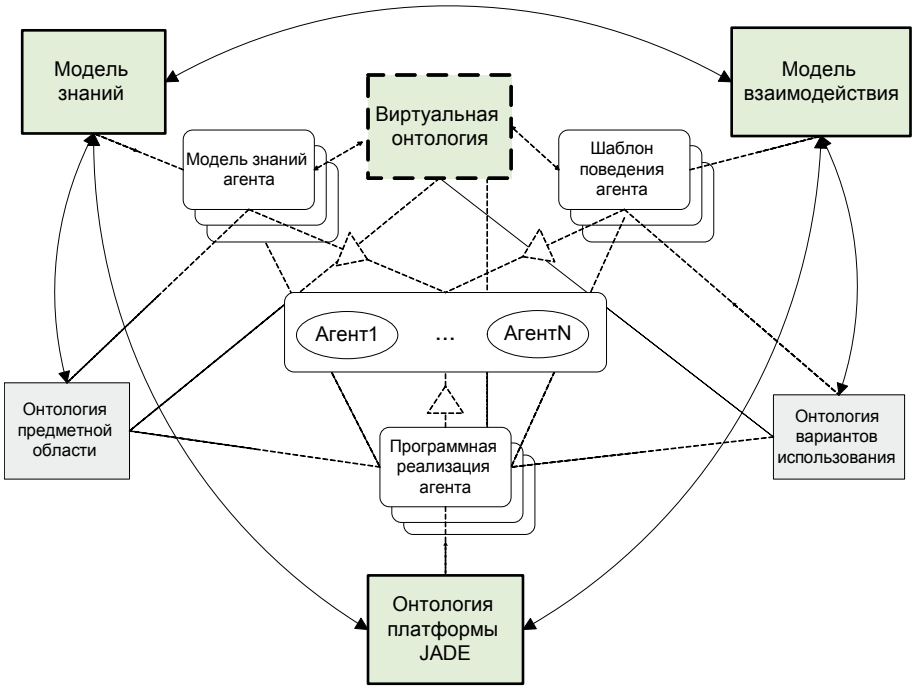

Рис. 1. Онтологические модели для синтеза агентов

На этапе **проектирования** осуществляется автоматизированное установление отображений между элементами модели знаний и онтологии предметной области, а также между элементами модели взаимодействия и онтологии вариантов использования. В случае возникновение проблем или недостатка информации производится уточнение отображений (вручную). Основным результатом этапа проектирования является виртуальная онтология. Для построения виртуальной онтологии разработан также набор правил, описанных с использованием языка правил SWRL (A Semantic Web Rule Language) [15]. SWRL позволяет использовать дизъюнкты Хорна (Horn-like rules) для явного указания способа вывода новых фактов из RDF-утверждений. Используя разработанные правила, виртуальная онтология пополняется новыми фактами.

На этапе **реализации** производится синтез агентов. Из полученной виртуальной онтологии извлекается информация двух типов: для генерации кода агентов, а также для генерации и компоновки ресурсов, касающихся модели знаний. Основными элементами полученного кода агентов для платформы Jade являются: классы агентов (jade.core.Agent), классы шаблонов сообщений (jade.lang.acl.MessageTemplate) и классы поведений агентов (jade.core.behaviours.Behaviour).

Агентам при генерации добавляются методы работы с построенной моделью знаний, а также методы обработки и отправки сообщений в формате OWL и SPARQL с помощью библиотеки AgentOWL и Jade API для работы с онтологиями. Сгенерированные агенты также обладают способностью вывода новых фактов из онтологической модели при помощи предопределенных методов. Точность и уровень детализации сгенерированного кода пропорциональна уровню детализации онтологической модели предметной области. Таким образом, генерируется совокупность агентов для МАС, которая служит основой для окончательной реализации системы.

4. Особенности взаимодействия агентов в мультиагентной системе и с внешними приложениями с использованием стандартов OWL, XSLT, XML, SPARQL

Основные аспекты взаимодействия агентов в МАС и с внешними приложениями определяются исходя из предложенной сферы использования МАС.

Взаимодействие между агентами предполагает использование информации в формате RDF/OWL. Основные принципы взаимодействия (протоколы взаимодействия, шаблоны сообщений и т.д.) определены согласно спецификации FIPA Ontology Service Specification[16].

В сети агенты могут получать данные или обмениваться информацией со следующими ресурсами (рис. 2):

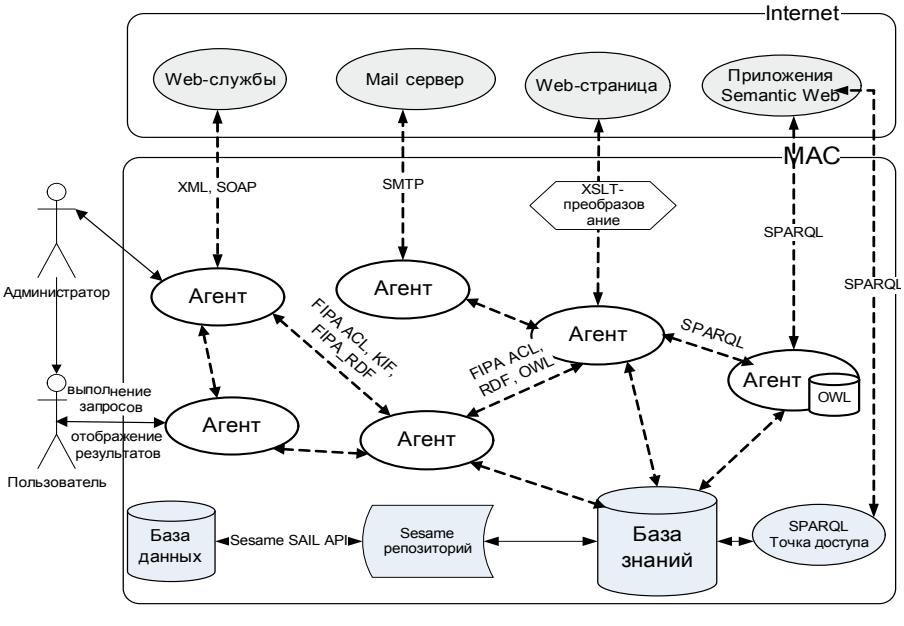

Рис. 2. Схема взаимодействия агентов внутри МАС и с внешними приложениями

- приложения Semantic Web: для взаимодействия используется SPARQL – язык запросов к RDF, стандарт W3C;

- веб-службы: для обмена информацией используется XML-основанный протокол SOAP;

- страницы html: для извлечения данных осуществляется XSLT- преобразование, для отправки данных (заполнение форм, выполнения POST-запросов) используется библиотека apache-httpclient.

Каждый агент отвечает за определенную область знаний, и располагает собственной локальной базой знаний в OWL-представлении (рис.3). В случае если агенту необходимы знания из другой области, производится обращение в общую базу знаний MAC, которая представляет собой наполненную экземплярами виртуальную онтологию, или к другому агенту.

Рис. 3. Специфика формирования базы знаний MAC

Общая база знаний в OWL-представлении отображается в БД с использованием Sesame Framework. Sesame обеспечивает механизмы сохранения знаний, выполнения запросов и логического вывода на знаниях с использованием модели RDF (Resource Description Framework) [17].

Для обеспечения связи MAC с внешними приложениями Semantic Web используется SPARQL язык запросов и SPARQL протокол. Построения внешних запросов разработано с использованием ARQ из библиотеки Jena[18].

Для повышениями уровня совместимости с внешними приложениями разработана SPARQL точка доступа.

5. Формальное описание модели знаний агентов и модели взаимодействия агентов с использованием дескриптивных логик

Важным элементом автоматизированного синтеза агентов является онтология модели знаний, учитывающая особенности среды функционирования агентов и методологии построения MAC (рис.4). Предлагаемая модель знаний расширяет подход, описанный в проекте AgentOWL, при этом также используются элементы онтологической модели платформы JADE.

Для описания модели знаний агентов выбран язык OWL DL , который очень близок к дескриптивной логике $SHOLN^{D}$.

База знаний системы описывается следующим образом:

$$KB = TBox\ T + ABox\ A \quad (1)$$

TBox описывает терминологию (словарь предметной области), а ABox содержит утверждения в терминах данного словаря. Опишем TBox утверждения с использованием синтаксиса дескриптивных логик. Пространство имен agentKM представляет элементы модели знаний агента (рис. 4).

Все внешние и внутренние ресурсы среды функционирования агентов описываются классом Resource. Класс Actor представляет собой любую сущность, которая взаимодействует с системой и использует ее возможности для достижения целей или решения поставленных задач.

$$agentKM{:}Actor \subseteq agentKM{:}Resource \quad (2)$$

Класс Agent является подклассом Actor и обозначает множество агентов общей среды.

$$agentKM{:}Agent \subseteq agentKM{:}Actor,\quad agentKM{:}User \subseteq agentKM{:}Actor \quad (3)$$

Понятие агента можно определить как сложный класс, связанный с другими классами определенными свойствами:

$$agentKM{:}Agent \equiv \exists agentKM{:}plays.agentKM{:}Role \cap \exists agentKM{:}holds.agentKM{:}Responsibility \cap \exists agentKM{:}grants.agentKM{:}Service \quad (4)$$

Класс Context объединяет в себе основные классы онтологии:

$$agentKM{:}Context \supset agentKM{:}Resource \cup agentKM{:}Action \cup agentKM{:}Domain \cup agentKM{:}Task \cup agentKM{:}Device \cup agentKM{:}UseCase \quad (5)$$

Учитывая основные особенности взаимодействия агентов, построена онтологическая модель взаимодействия агентов с использованием языка Web-онтологий OWL. Пространство имен agentIM представляет элементы модели взаимодействия агентов. Модель включает возможные типы взаимодействий (между агентами, с внешними ресурсами и с пользователями), а также объекты, параметры и отношения, необходимые для организации взаимодействий (рис. 5).

Взаимодействия описываются классом agentIM: Interaction, который включает в себя:

- agentKM:Responsibility - описание последовательности действий, которые может осуществлять агент в ответ на внешние воздействия или выполняя поставленную задачу;

- agentKM:InteractionProtocol – протокол, по которому осуществляется обмен сообщениями между агентами; преимущественно используются протоколы FIPA;

- agentKM:Role – роль рассматриваемого агента во взаимодействии: может быть Initiator или Responder.

$$agentIM{:}Interaction \subseteq \exists agentIM{:}include.agentKM{:}Responsibility \cap \exists agentIM{:}include.agentKM{:}Role \cap \exists agentIM{:}include.\ agentIM{:}InteractionProtocol \cap \exists include.\neg agentKM{:}Role \quad (6)$$

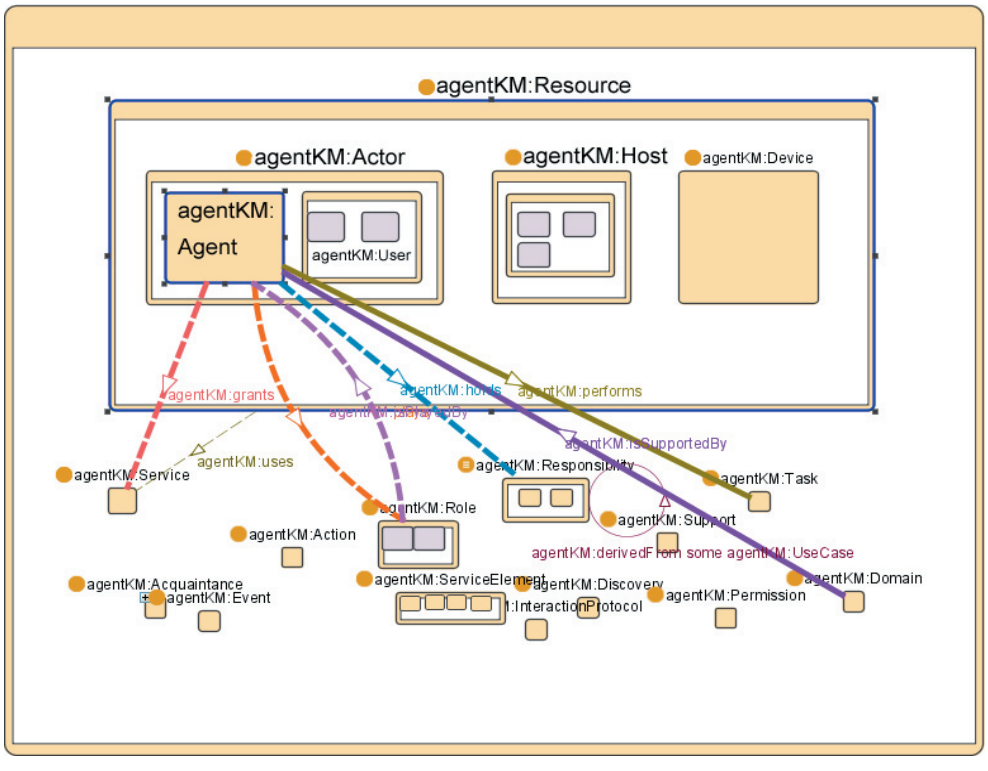

Рис. 4. Основные элементы онтологической модели знаний агента

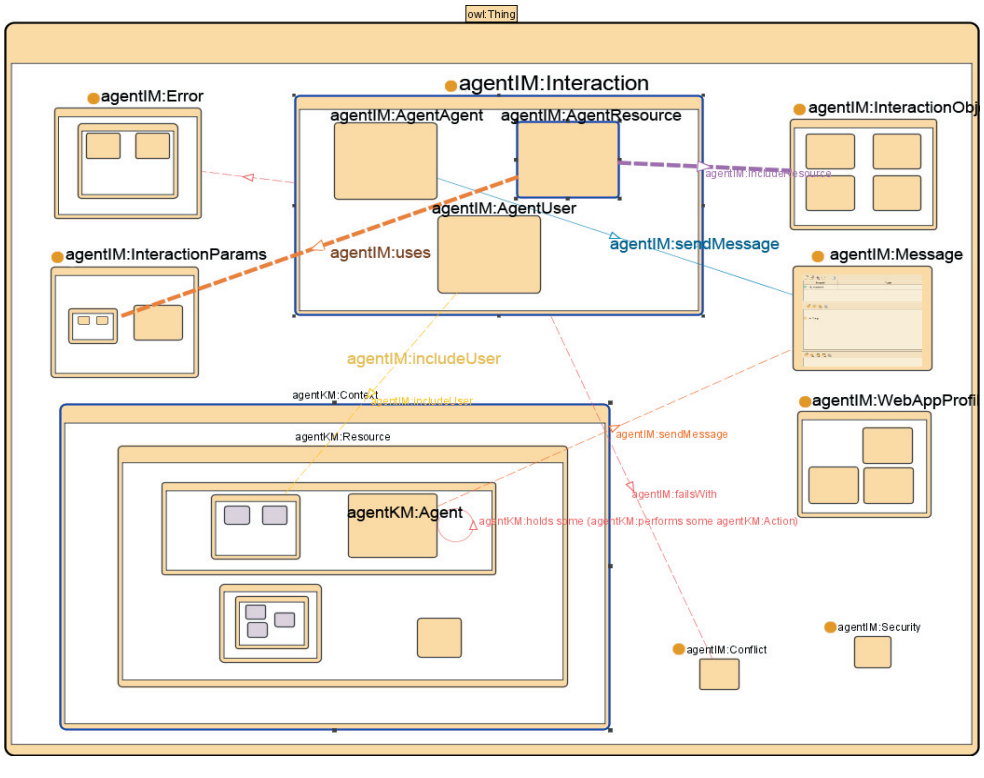

Рис. 5. Основные элементы онтологической модели взаимодействия агентов

6. Примеры аксиом и правил, используемых для синтеза агентов

Следующие аксиомы виртуальной онтологии описывают связи элементов, описывающих предметную область с элементами платформы JADE для автоматизированного построения интеллектуальных агентов:

$$\begin{aligned} &\texttt{jade:JadeAgent} \subseteq \texttt{jade:JadeClass} \\ &\texttt{jade:JadeAgent} \subseteq \texttt{agentKM:buildFrom.agentKM:Device} \cap \\ &\texttt{agentKM:buildFrom.agentKM:User} \cap \texttt{agentKM:buildFrom.agentKM:Host} \\ &\texttt{jade:JadeBehaviour} \subseteq \texttt{agentKM:buildFrom.agentKM:Responsibility} \end{aligned} \quad (7)$$

Далее представлены SWRL-правила, расширяющие возможности вывода новых фактов виртуальной онтологии.

Построение агента на основании более чем одного базового класса онтологии может осуществляться по следующему правилу:

```
agentKM:Host(?x) ∩  owl:SubClassOf:?x(?y)  ⇒
agentKM:JadeAgent(?a)∩ jade:buildFrom(?a,(?x ?y)); (8)
```

Правило 9 описывает, каким образом создаются элементы класса Responsibility, которые представляют поведение агентов, а также определяет параметры, которые используются в классах Behaviour.

```
owl:ObjectProperty(?p)   ∩agentKM:User(?x)∩
tbox:isInDomainOf(?x,?p)  ∩  tbox:isInRangeOf(?y, ?p) ⇒  (9)
agentKM:Responsibility(?p)∩ agentKM:hasParameter(?p, ?y)
```

Взаимодействия подразделяются на взаимодействия между агентами, с внешними ресурсами и с пользователями.

Все агенты, созданные из подклассов, взаимодействуют с агентом, созданным из общего родительского класса. Некоторые аспекты взаимодействий и обработки сообщений описывается следующим правилом:

```
agentKM:UserAgent(?x)∩agentKM:HostAgent(?y)∩
agentIM:interaction(?x,?y)∩
agentKM:grants(?x,someService)⇒                    (10)
agentIM:sendMessage(?x,?y)∩
agentIM:messageTemplate(?x,aGrantService)∩
agentIM:messageTemplate(?y,aRequestService)
```

Подобные взаимодействия дополняются описанием сообщений. Для создания сообщений используются элементы онтологии предметной области. Агенты понимают сообщения, созданные из элементов онтологии предметной области, и по запросу могут выполнять действия, описанные в поведении агентов.

Отдельной задачей является определение необходимости помещения ресурса в модель знаний каждого агента, для чего используются следующие правила:

```
owl:ObjectProperty(?p)  ∩ ?p(someResource, agentKM:Host)  ⇒
agentKM:includeInModel(agentKM:HostAgent, someResource)    (11)
```

```
owl:ObjectProperty(?p)  ∩ ?p(someResource, agentKM:User) ⇒
agentKM:includeInModel(agentKM:UserAgent, someResource)    (12)
```

Данные правила(8-12) также используются в обратном порядке, добавляя агентам знания о соответствии поведений и других программных объектов элементам онтологической модели знаний.

7. Синтез агентов на примере управления контекстной рекламой

Рассмотрим применения предложенного подхода для решения задач управления рекламными кампаниями в Интернет.

Контекстная реклама – вид динамического размещения интернет-рекламы, при котором рекламное объявление соответствует содержанию интернет-страницы, где оно размещается и которую посещает интернет-пользователь.

Основными задачами в рамках автоматизации маркетинговых процессов управления контекстной рекламой являются следующие:

- управление рекламными кампаниями, включающее принятие решений и ситуативную ориентацию;
- сбор статистики, интерактивные предложения по увеличению эффективности рекламных кампаний;
- выбор сетей для показов рекламных объявлений;
- подбор ключевых фраз по текстам объявлений;
- оценка эффективности кампаний и медиапланирование.

На этапе анализа строится онтологическое описание среды.На этапе проектирования, с использованием правил (7-12 и др.) виртуальная онтология наполняется экземплярами (индивидами), построенными на основании онтологии контекстной рекламы.

Исходя из утверждений TBox, и имея следующие ABox утверждения, создаются java-классы для функционирования Jade-агентов (табл.1), а также другие классы, необходимые для программной реализации системы.

Все входные параметры перед передачей в какое-либо поведение проходят проверку на корректность с использованием OWL-аксиом, представляющих ограничения для класса (табл.2).

Используя правила SWRL создаются шаблоны сообщений.

Сгенерированные элементы системы могут быть модифицированы вручную, при этом все изменения согласуются с онтологическими моделями для обеспечения соответствия с последующими версиями. Основная структура МАС, полученная в результате синтеза также может быть расширена более детализированными и усложненными реализациями некоторых методов сгенерированных классов.

Таким образом, предлагаемый подход упрощает процесс создания МАС, значительно сокращает количество программного кода, необходимого для управления основанным на онтологиях взаимодействием между агентами, а также позволяет более эффективно осуществлять поддержку системы и выпуск новых программных версий.

Таблица 1

Пример синтеза классов агентов и поведений агентов

ABox утверждения	Java интерфейсы и классы
agentKM:Host(YandexDirect, GoogleAdwords,Begun); agentIM:InteractionObject (WebService, HTML, Mail)	YandexWebServiceAgent, YandexHTMLAgent, YandexMailAgent; GoogleWebServiceAgent, GoogleHTMLAgent, GoogleMailAgent, BegunWebServiceAgent, BegunHTMLAgent, BegunMailAgent
agentKM:Responsibility (create, update, remove); agentKM:ResponsibilityRange (campaign, advertisement, keyword)	CreateCampaignBehaviour, UpdateCampaignBehaviour, RemoveCampaignBehaviour, аналогично для Advertisement, Keyword

Таблица 2

Пример синтеза шаблонов сообщений и ограничений онтологии

TBox утверждения	Java-код
creates.GoogleCampaign tbox:isInDomainOf(Agency,creates)	aclMessage .addReceiver(GoogleAgent); aclMessage .setContent(createCampaign);
Campaign ∩ ∀ belongsToSite.YandexDirect ∩ ∀ hasBudget.FullBudget ∩ ∀ isAccessibleFrom.HTML ∩ ≥ 1hasAdvertisement .YandexAdvertisement	ConceptSchema cs = new ConceptSchema(«Campaign»); cs.add(”ADVERTISEMENT”, new YandexAdvertisement()); cs.add(”Budget”, new FullBudget());

8. Заключение

В работе на примере системы управления контекстной рекламой показаны преимущества интегрирования разработанных онтологических моделей в процессы проектирования, разработки и поддержки МАС. Разработан метод синтеза агентов распределенной интеллектуальной системы на основе использования онтологических моделей знаний, позволяющий автоматизировано создавать новых агентов и менять поведение существующих для обеспечения функционирования системы в условиях изменяющейся среды. Усовершенствована модель взаимодействия агентов в мультиагентной системе и с внешними приложениями с использованием стандартов OWL, SPARQL для обеспечения ведения переговоров и принятия решений. Основные преимущества от применения предложенного подхода можно получить в таких областях как электронное обучение, электронная коммерция, а также в сферах общественного и ассоциативного обслуживания, где важным фактором является распараллеливание задач и взаимодействие агентов для достижения поставленных пользователем целей.

Литература

1. Швецов А.Н., Яковлев С.А. Распределенные интеллектуальные информационные системы. – СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2003. – 318 с.
2. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2003.
3. FIPA Interaction Protocol Specifications, see: http://www.fipa.org/repository/ips.php3.
4. Филатов В. А. Мультиагентные технологии интеграции гетерогенных информационных систем и распределенных баз данных : Дис... д-ра техн. наук: 05.13.06 / Харьковский национальный ун-т радиоэлектроники. — Х., 2004. — 341л. : рис. — Библиогр.: л. 313-336.
5. Federico Bergenti, Agostino Poggi. Exploiting UML in the Design of Multi-Agent Systems. In Proceedings of Engineering Societies in the Agents' World, 2000
6. Майкл К. Смит, Крис Велти, Дебора Л. МакГиннес. OWL, язык веб-онтологий. Руководство. - http://www3.org/TR/2004/REC-owl-guide-20040210/.
7. Michal Laclavik, Marian Babik, Zoltan Balogh, Ladislav Hluchy AgentOWL: Semantic Knowledge Model and Agent Architecturt In Computing and Informatics. Vol. 25, no. 5 (2006), p. 419-437. ISSN 1335-9150, Chapters 1, 4, 5.
8. The Gaia Methodology for Agent-Oriented Analysis and Design (2000) by Michael Wooldridge, Nicholas R. Jennings, David Kinny Journal of Autonomous Agents and Multi-Agent Systems http://www.ecs.soton.ac.uk/~nrj/download-files/jaamas2000.pdf
9. Scott A. DeLoach. The MaSE Methodology. In Methodologies and Software Engineering for Agent Systems. The Agent-Oriented Software Engineering Handbook Series : Multiagent Systems, Artificial Societies, and Simulated Organizations, Vol. 11. Bergenti, Federico; Gleizes, Marie-Pierre; Zambonelli, Franco (Eds.) Kluwer Academic Publishing (available via Springer), August 2004.
10. J. Odell, H. Van Dyke Parunak and B. Bauer, “Representing Agent Interaction Protocols in UML”, in Proceedings of Agents 2000, 2000.
11. Hajnal Á, Pedone G, Varga LZ. Ontology-Driven Agent Code Generation for Home Care in Protégé. 10th International Protégé Conference, pp. 91-93; 2007.
12. Nyulas CI, O'Connor MJ, Tu SW, Buckeridge DL, Okhmatovskaia A, Musen MA. An Ontology-Driven Framework for Deploying JADE Agent Systems. IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'08), 2008.
13. Fabio Bellifemine, Giovanni Caire, Tiziana Trucco (TILAB, formerly CSELT), Giovanni Rimassa (University of Parma) . JADE PROGRAMMER'S GUIDE, 2006.
14. Berners-Lee T., James Hendler, Ora Lassila. “The Semantic Web”. Scientific American Magazine, May 2001.
15. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, W3C Member Submission 21 May 2004. – http://www.w3.org/Submission/SWRL/
16. FIPA Ontology Service Specification http://www.fipa.org/specs/fipa00086/XC00086C.html.
17. Herman Ivan, Swick Ralph, Brickley Dan. Resource Description Framework (RDF) - http://www.w3.org/RDF/.
18. Jena, HP Labs Semantic Web Toolkit. http://jena.sourceforge.net/.