

УДК 004.896

## **ТРАНСФОРМАЦИЯ МОДЕЛЕЙ МЕТОДОЛОГИИ GAIA В КОНЦЕПТЫ ИНСТРУМЕНТАРИЯ ZEUS ПРИ МНОГОАГЕНТНОМ МОДЕЛИРОВАНИИ ПРОЦЕССА ПРОИЗВОДСТВА ПРОМЫШЛЕННЫХ СМЕСЕЙ**

*Зудикова Ю.В., Федяев О.И.*

*Донецкий национальный технический университет  
Кафедра прикладной математики и информатики  
E-mail: julez88@mail.ru*

### **Аннотация**

*Зудикова Ю.В., Федяев О.И. Трансформация моделей методологии Gaia в концепты инструментария Zeus при многоагентном моделировании процесса производства промышленных смесей. В статье рассматривается вопрос перехода от моделей методологии агентно-ориентированного анализа Gaia к концептам инструментария Zeus при создании многоагентной системы процесса производства промышленных смесей. Составлены модели Gaia на примере взаимодействия двух агентов разрабатываемой многоагентной системы. Создано описание агентов в терминах концептов Zeus.*

### **Общая постановка проблемы**

Процесс производства промышленных смесей как объект моделирования и управления является распределенной системой, в которой взаимодействуют отдельно работающие устройства и персонал. Теория многоагентных систем позволяет строить качественно новые модели таких сложных объектов в виде интеллектуальных организаций, представленных автономными искусственными агентами. На сегодняшний день предложены методологии и инструментальные среды для анализа и разработки многоагентных систем.

Методологии агентно-ориентированного анализа применяются на стадиях анализа и проектирования многоагентных систем. Путем абстрагирования создаются концептуальные модели и методы многоагентных систем. Выделяют четыре класса методологий: базирующиеся на объектно-ориентированных методах и технологиях с использованием соответствующих расширений (AUML), использующие традиционные методы инженерии знаний (MAS-CommonKADS), основанные на организационно-ориентированных представлениях (Gaia), комбинирующие в различной степени методы трех первых классов.

Инструментальные среды используются на стадиях реализации и тестирования многоагентных систем. Они используют свои модели, соответствующие этому уровню детализации системы. Можно выделить два основных класса инструментальных сред: фреймворки (JADE) и среды разработки (Zeus, Agent Builder).

Таким образом, процесс разработки многоагентной системы является иерархическим и должен связывать модели разных уровней проектирования. Однако из-за использования различных методологий и инструментариев, оперирующих разными концептами, этот процесс становится проблематичным.

### **Постановка задач исследования**

Методология Gaia и инструментальная среда Zeus, применяемые для моделирования процесса производства промышленных смесей, оперируют схожими абстрактными концептами и ориентированы на создание многоагентных систем с распределенным искусственным интеллектом. Задачей исследования является разработка методики трансформации моделей методологии Gaia в концепты инструментария Zeus, представляемые в структурной и алгебраической форме.

**Решение задачи и результаты исследований**

На начальном этапе разработки многоагентной системы проводится агентно-ориентированный анализ предметной области. Результатами такого анализа являются абстракции и концептуальные модели многоагентной системы, которые используются для дальнейшей конкретизации на уровне инструментальной среды.

В рамках организационно-ориентированной методологии Gaia [1] многоагентная система проектируется в терминах поведений и структур человеческих организаций и представляет собой совокупность взаимодействующих ролей. Gaia предусматривает создание следующих моделей: ролей, взаимодействий, агентов, услуг и связей (рис. 1).

Инструментальная среда Zeus оперирует такими базовыми концептами, как агент, факт, цель и задача. На их основе она строит следующие модели: онтологии, программных агентов, задач, координации и организации (рис. 1).

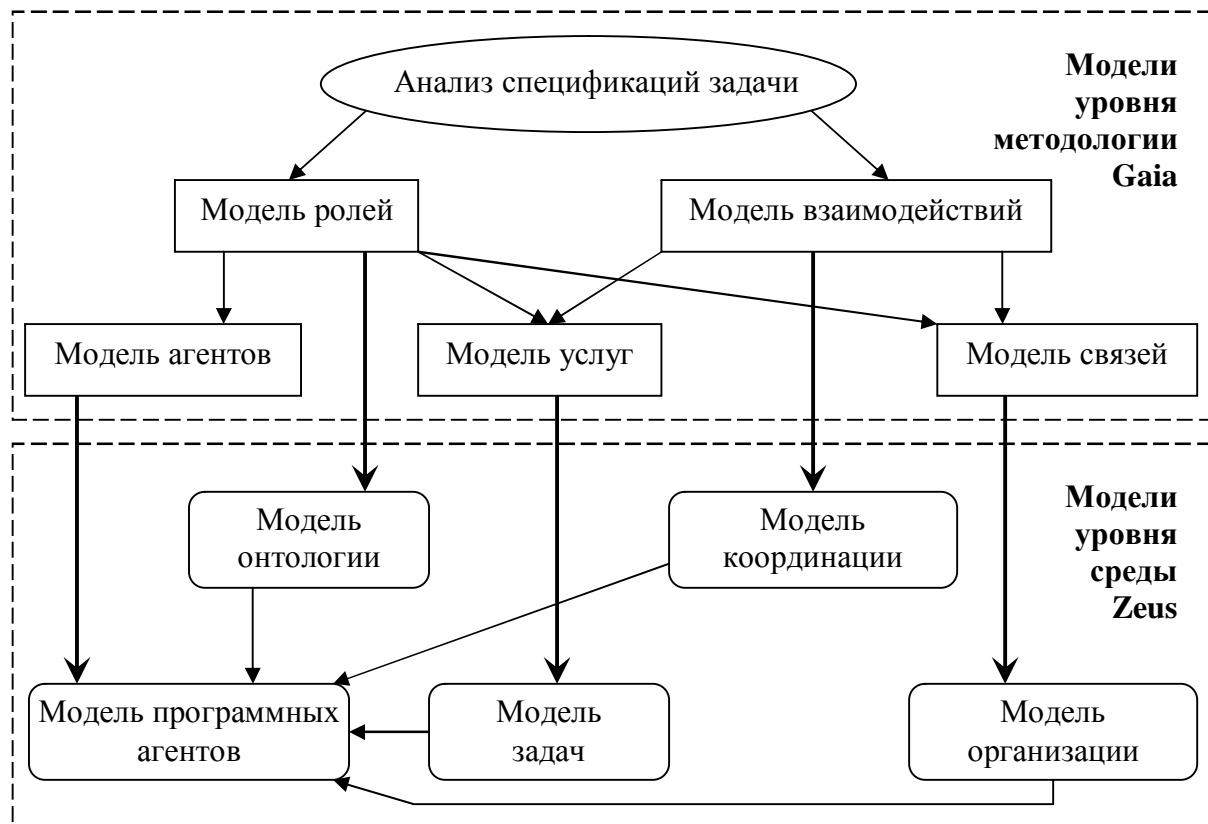


Рисунок 1. Взаимосвязь моделей методологии Gaia и инструментальной среды Zeus

Онтология Zeus представляется декларативным набором знаний, который отражает существенные понятия предметной области и их характеристики (атрибуты и значения). Онтология Ont задается как множество фактов  $f$ , представленных в виде иерархической структуры:

$$Ont = \{f_1, f_2, \dots, f_n\}. \quad (1)$$

Каждый факт  $f$  характеризует одно понятие предметной области и задается в общем случае множеством атрибутов или четверок вида:

$$f = \{(AN, T, R, DV)_1, (AN, T, R, DV)_2, \dots, (AN, T, R, DV)_m\}, \quad (2)$$

где AN – название атрибута; T – тип атрибута; R – ограничение на значение атрибута; DV – значение атрибута по умолчанию. Фактам модели онтологии Zeus соответствуют полномочия (информационные ресурсы) модели ролей Gaia.

Понятие агента  $A$  в Zeus является ключевым и описывается на трех уровнях: определения  $D$ , организации  $O$  и координации  $C$  [2]:

$$A = (D, O, C), \quad D = (R, G, T), \quad O = (AD, R, NA), \quad C = (CP, IS). \quad (3)$$

На уровне определения  $D$  (рис. 2) агент рассматривается как автономная сущность, способная к рассуждениям. Агент задается множеством ресурсов (знаний)  $R$ , множеством целей  $G$  и множеством задач  $T$ .

В Zeus предусмотрено три вида задач: простая (неделимая) задача, группа простых задач, выполняющихся в некоторой последовательности, и база правил. Простая задача  $t$  задается следующим набором элементов:

$$t = (N, T, P, E, CP, CE, C, D), \quad (4)$$

где  $N$  – название задачи;  $T$  – тип задачи;  $P$  – входные факты (предусловия);  $E$  – выходные факты (эффекты);  $CP$  – ограничения на значения входных фактов;  $CE$  – ограничения на значения выходных фактов;  $C$  – стоимость задачи;  $D$  – продолжительность выполнения задачи. База правил задается множеством правил  $rl$ :

$$rl = (P, A), \quad (5)$$

где  $P$  – условия на основе входных фактов;  $A$  – действия, вытекающие из условий. Задачи и правила агента формируются на основе модели услуг Gaia, для которой в среде Zeus вводится дополнительный атрибут – тип услуги, которому соответствует тип задачи.

На уровне организации  $O$  (рис. 2) агент рассматривается в рамках его отношений с другими агентами. Для агента определяется множество «знакомых» агентов  $AD$ , множество отношений  $R$ , в которых агент состоит с другими агентами, и множество фактов  $NA$ , принадлежащих «знакомым» агентам, о которых знает конфигурируемый агент. Множество «знакомых» агентов  $AD$  формируется из модели организации Zeus и модели связей Gaia, а множество известных фактов  $NA$  – с помощью модели взаимодействий Gaia.

На уровне координации  $C$  (рис. 2) агент рассматривается как социальная сущность, взаимодействующая с другими агентами сообщества в рамках протоколов координации  $CP$  и стратегий взаимодействия  $IS$ , получаемых путем анализа модели взаимодействий Gaia.

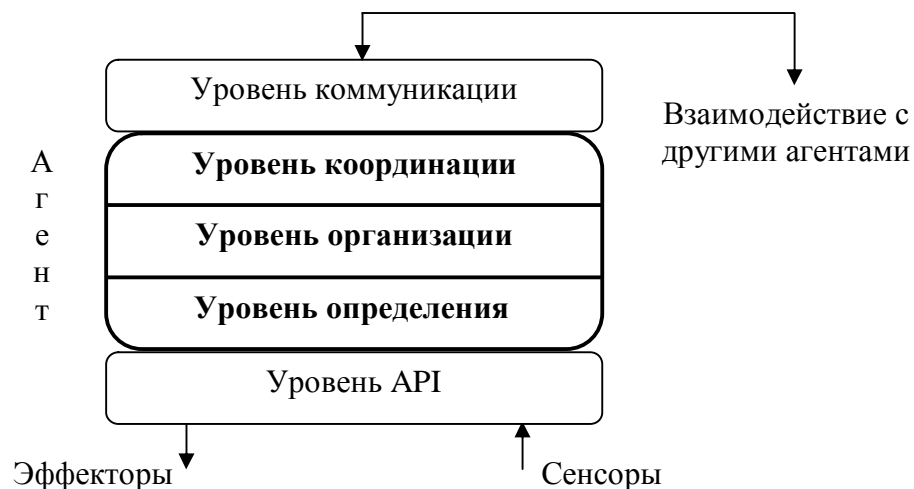


Рисунок 2. Концептуальная структура агента Zeus

Уровень коммуникации (рис. 2) определяет язык и протокол передачи сообщений агентами в рамках их взаимодействия. Уровень API (рис. 2) обеспечивает связь агента с внешними программами, предоставляющими ему ресурсы и/или реализующими его способности.

В качестве примера рассмотрено взаимодействие двух агентов разрабатываемой многоагентной системы [1]. Агент Кладовщик выдает порцию сырья со склада, а агент Кран-балка принимает и транспортирует сырье к месту назначения. Упрощенные в этом контексте модели методологии Gaia представлены на рис. 3, 4, 5.

Схема роли Кладовщик	Схема роли Кран-балка
<b>Описание роли:</b> обслуживает склад сырья	<b>Описание роли:</b> доставляет сырье к бункеру
<b>Протоколы и Активность:</b> Выдать порцию сырья по плану выдачи <u>Изменить объем сырья с учетом выдачи</u> Сообщить об отсутствии сырья	<b>Протоколы и Активность:</b> <u>Подъехать к складу сырья</u> Получить сырье Определить бункер приема сырья <u>Переместиться к бункеру приема</u>
<b>Полномочия:</b> <b>reads</b> вместительность склада <b>reads</b> план выдачи сырья <b>changes</b> типы сырья <b>changes</b> количество сырья каждого типа <b>changes</b> уровень наполненности склада	<b>Полномочия:</b> <b>changes</b> местоположение <b>changes</b> конечный пункт <b>changes</b> состояние загрузки (с грузом/без груза) <b>changes</b> состояние исправности
<b>Обязательства:</b> <b>жизнеспособности:</b> КЛАДОВЩИК = ВЫДАЧА СЫРЬЯ ВЫДАЧА СЫРЬЯ = (Выдать порцию сырья по плану выдачи. <u>Изменить объем сырья с учетом выдачи</u> )   Сообщить об отсутствии сырья <b>условия безопасности:</b> $0 \leq \text{уровень наполненности} \leq \text{вместительность склада}$	<b>Обязательства:</b> <b>жизнеспособности:</b> КРАНБАЛКА_С = ДОСТАВКА СЫРЬЯ ДОСТАВКА СЫРЬЯ = <u>Подъехать к складу сырья</u> . Получить сырье. Определить бункер приема сырья. <u>Переместиться к бункеру приема</u> <b>условия безопасности:</b> состояние исправности = «исправно»

Рисунок 3. Схемы ролей Кладовщик и Кран-балка

Выдать порцию сырья по плану выдачи		План выдачи сырья Количество сырья по типам
Кладовщик	Кран-балка	
Выдать очередную порцию сырья со склада по плану выдачи		Порция сырья
↓		
Получить сырье		Порция сырья Подтверждение получения порции сырья
Кран-балка	Кладовщик	
Получить очередную порцию сырья со склада		

Рисунок 4. Протокол выдачи сырья со склада

Услуга	Входные данные	Выходные данные	Предусловия	Постусловия
Для роли Кладовщик				
Выдать сырье	Наименования и количество сырья	Порция сырья	количество сырья > 0	Порция сырья > 0 (выдана)
Для роли Кран-балка				
Принять сырье	Порция сырья, Состояние загрузки	Сообщение о получении сырья	состояние загрузки = 0	состояние загрузки = 1 (с грузом)

Рисунок 5. Модель услуг Gaia для агентов Кладовщик и Кран-балка

На основе созданных моделей Gaia и установленных межмодельных связей проведена спецификация структуры программных агентов по правилам, принятым в инструментальной среде Zeus. Примеры моделей онтологии и агентов показаны на рис. 6, 7.

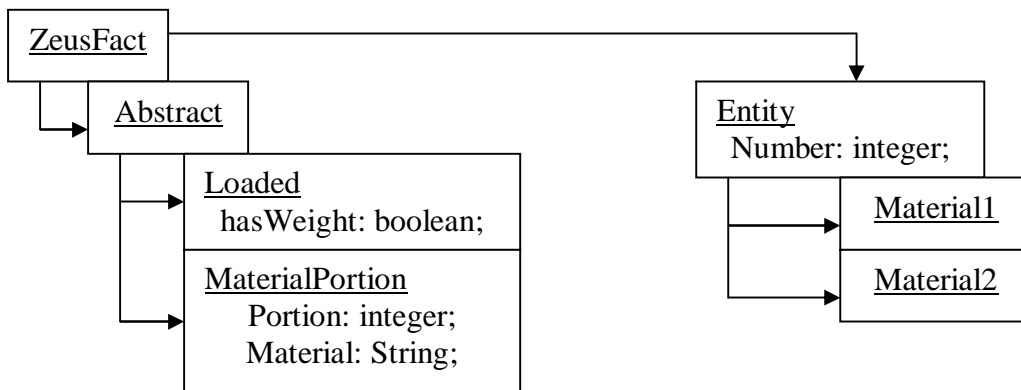


Рисунок 6. Модель онтологии Zeus

Агент Kladovshik	Агент Kran-balka
Уровень определения	
R = {(Material1 (Number = 100)), (Material2 (Number = 80))}	R = {(Loaded (hasWeight = false))}
T = {(GiveMaterial: (P: Material1, Material2) (E: MaterialPortion (Portion = 1)) (C: 0) (D: 1))}	T = {(LoadMaterial: (P: MaterialPortion (Portion = 1)) (E: Loaded (hasWeight = true)) (C: 0) (D: 1))}
Уровень организации	
AD = {Kran-balka}, R = {peer}	AD = {Kladovshik}, R = {peer}
Уровень координации	
CP = {Respondent: FIPA-Contract-Net-Contractor}, IS = {Default}	CP = {Initiator: FIPA-Contract-Net-Manager}, IS = {Default}

Рисунок 7. Модель агентов Zeus

По составленным спецификациям онтологии, структуры и поведения агентов в инструментальной среде Zeus был сформирован программный код системы на языке Java.

### Выводы

В результате исследования предложена методика трансформации моделей методологии агентно-ориентированного анализа Gaia в концепты инструментальной среды Zeus. Методика позволяет систематизировать проектирование и автоматизировать генерацию программного каркаса многоагентной системы.

### Список литературы

1. Зудикова Ю.В., Федяев О.И. Разработка многоагентной модели процесса производства промышленных смесей [Текст] / Ю.В. Зудикова, О.И. Федяев // Информатика та комп'ютерні технології / Матеріали V міжнародної науково-технічної конференції студентів, аспірантів та молодих науковців - 24-26 листопада 2009 р., Донецьк, ДонНТУ. - 2009, с. 261-264.
2. Nyacinth S. Nwana, Divine T. Ndumu, Lyndon C. Lee. ZEUS: An advanced tool-kit for engineering distributed multi-agent systems [Electronic resource] / Интернет-ресурс. – Режим доступа: <http://www.agent.ai/doc/upload/200302/nwan98.pdf>