

УДК 004.89

Переход от абстрактных к физическим моделям при восходящем проектировании многоагентных систем

Федяев О.И., Жабская Т.Е.

Донецкий национальный технический университет
fedyaev@r5.dgtu.donetsk.ua

Abstract

Fedyayev O., Zhabskaya T. The transition from abstract to physical models under ascending design of the multi-agent systems. In the submitted paper the transition from abstract to physical models under ascending design of the multi-agent systems is considered. For solving of this problem the visual models semantics of JACK environment has been developed and on its base the transformation method of Gaia abstract models to JACK environment physical concepts has been developed.

Введение

Проблематика искусственных агентов, многоагентных систем и интеллектуальных организаций имеет полувековую историю и сформировалась на основе результатов исследований в области параллельного решения задач, распределенных компьютерных систем и распределенного искусственного интеллекта. В настоящее время теория агентов открывает новые перспективные направления в информационных технологиях и искусственном интеллекте. Интеграция достижений в области программной инженерии и искусственном интеллекте привела к появлению новой агентно-ориентированной технологии программирования и возможности построения качественно новых программных моделей. Междисциплинарный характер этого научного направления позволяет успешно применять агентно-ориентированный подход в таких областях, как распределенное решение сложных задач, реинжиниринг предприятий, моделирование распределенных систем, построение виртуальных организаций.

Процесс разработки многоагентных систем является иерархическим и должен связывать модели разных уровней проектирования. Однако, из-за использования различных методологий и инструментариев, оперирующих различными концептами, при реализации этого процесса возникает ряд проблем, связанных с переходом от концептуальных к физическим моделям реализации многоагентной системы.

Задачей данного исследования является разработка методики преобразования абстрактных моделей методологии Gaia [1] в физические концепты инструментария JACK [2]. Инструментальная среда JACK использовалась авторами для создания виртуальной кафедры

университета агентно-ориентированного типа. Кафедра, как объект моделирования, является распределенной системой, субъекты которой выполняют определенную интеллектуальную деятельность (рис.1). Процесс обучения студентов на кафедре университета распределён во времени и пространстве. Для предоставления участникам образовательного процесса возможности автономного и дистанционного выполнения учебно-методических обязанностей поставлена цель создать такую виртуальную кафедру, в которой сохраняются все необходимые для учёбы отношения и устраняются жёсткие пространственно-временные ограничения в виде расписания занятий.

Для разработки многоагентного приложения, автоматизирующего процесс обучения на уровне кафедры, был проведен агентно-ориентированный анализ учебного процесса по методологии Gaia. С помощью этой методологии разработаны концептуальные модели, абстрагирующие деятельность субъектов на кафедре университета [3].

Семантика визуальных моделей среды JACK

Агентно - ориентированные модели, разработанные по методологии Gaia, позволили получить абстрактное описание учебно-методической деятельности субъектов учебного процесса, выполнение которой далее необходимо делегировать их программным агентам.

Полученные модели агентов могут быть реализованы различными способами. Для моделирования ментальных свойств агентов, необходимых для решения задачи обучения, была выбрана BDI-архитектура, основанная на знаниях [4,5]. В качестве инструментария, реализующего эту архитектуру агентов, использовалась агентно-ориентированная среда разработки JACKTM Intelligent Agents (JACK). JACK является

надстройкой Java в виде расширения синтаксиса свойств, связанных с понятием Java конструкциями для программной реализации интеллектуального

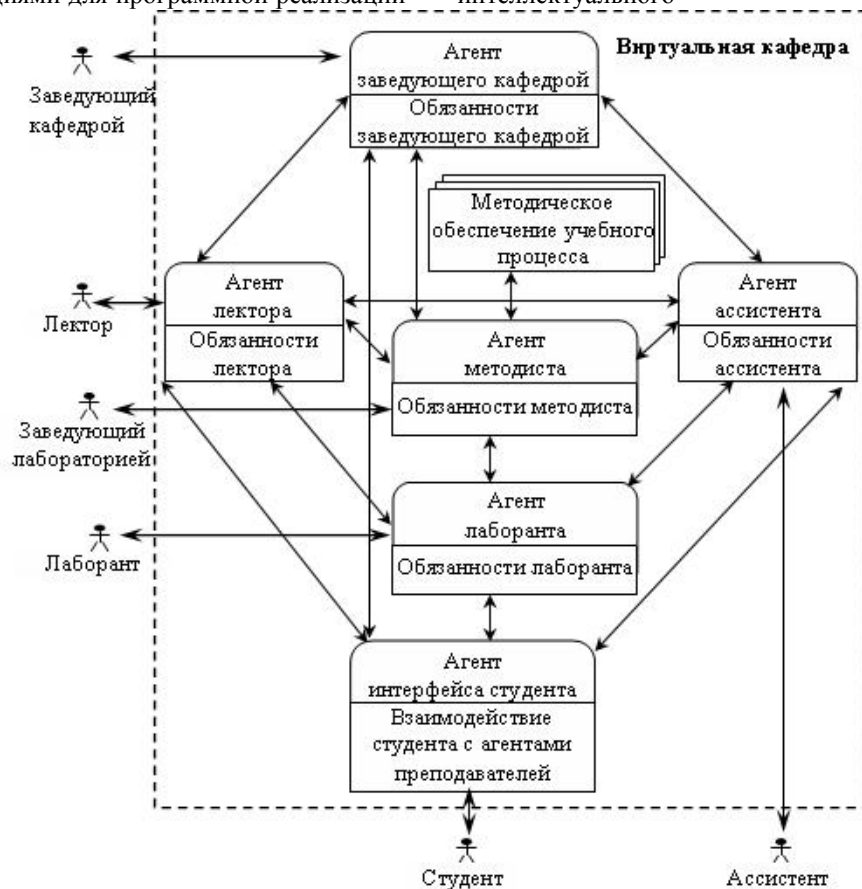


Рисунок 1 – Кафедра университета как распределенный объект моделирования

агента. Следует отметить, что эта среда является одной из наиболее подходящих и гибких технологий для реализации понятия "интеллектуальный агент", благодаря наличию механизма рассуждений для воспроизведения программными агентами делегированной им интеллектуальной деятельности [6].

Агент, имеющий BDI – архитектуру, описывается тремя компонентами:

$$A = (B, D, I),$$

где B – это убеждения агента, которые являются информацией агента о собственном состоянии и состоянии его окружения, и рассматриваются как его информационная компонента;

D – это желания агента в виде информации о его целях, которые рассматриваются как его мотивационная компонента;

I – это намерения агента, которые представляют возможные направления его действий, и являются его рассудительной компонентой.

Для программной реализации убеждений, желаний и намерений агента в языке JACK предусмотрены следующие новые конструкции, расширяющие синтаксис языка Java на уровне классов:

– Agent, определяет интеллектуальных агентов;

- Event, определяет цели агента, в виде событий;
- Plan, описывает намерения агента в отношении достижения цели в виде планов и условия их применимости;
- Beliefset, описывает знания агента;
- Capability, структурирует убеждения, события и планы в кластеры для реализации определенной интеллектуальной способности агента для достижения цели.

Отношения между данными классами агентной системы устанавливаются с помощью деклараций языка JACK, выделяемых предшествующим символом #. Для описания логических рассуждений агента во время выполнения планов введены операторы методов рассуждений, выделяемые предшествующим символом @.

Из специфики BDI-архитектуры следует, что сущность каждого типа агента, описываемая моделями методологии Gaia, должна быть воплощена через реализацию его ментальных свойств – убеждений, желаний и намерений в виде убеждений, событий и планов языка JACK. Другими словами, чтобы воспользоваться инструментальной средой, необходимо мыслить на уровне её понятий.

Таким образом, чтобы правильно перейти от абстрактных моделей агентов к их представлению на уровне визуальных моделей среды JASK, авторами были разработаны спецификации семантики визуальных моделей данной среды.

Семантика визуальной модели агента, представленного формально в среде JASK в виде

$$\text{Agent} = (N, \text{Bel}, \text{PE}, \text{HE}, \text{SE}, \text{PS}),$$

раскрывается следующим образом:

N – имя агента;

Bel – убеждения агента;

PE = {E₁, E₂, ..., E_n} – множество имен событий, создаваемых собственными методами агента (самостоятельно);

HE = {{E₁, E₂, ..., E_n}, HE₁, ...} – множество событий, воспринимаемых агентом для обработки (желания);

SE = {SE₁, SE₂, ..., SE_m} – множество имен событий, передаваемых агентом во внешнюю среду;

PS = {P₁, P₂, ..., P_k} – множество имен планов, определяющих поведение (намерения) агента.

Более эффективно определять агента через его способности, тогда:

$$\text{Agent} = (N, \text{Bel}, \text{PE}, \text{Cap}),$$

где Cap = {C₁, C₂, ..., C_n} – множество имен способностей, которыми обладает агент для достижения поставленных целей.

Семантика визуальной модели воспринимаемого события, формально представляемая в среде JASK в виде

$$\text{Event} = (N, \text{Pw}, \text{Pa}, \text{EvT}),$$

имеет следующее внутреннее содержание:

N – имя воспринимаемого события;

Pw – метод автоматического порождения новых убеждений при соответствии собственных убеждений агента некоторым условиям;

Pa – метод явного восприятия агентом события внешнего мира и формирования новых убеждений, связанных с данным событием, в результате которого агент получает цель для достижения;

EvT – определение стратегии достижения цели.

Семантика визуальной модели плана для описания рассуждений агента формально в среде JASK представляется в виде

$$\text{Plan} = (N, \text{Ev}, \text{Sev}, \text{MP}, \text{MF}, \text{Rel}, \text{C}, \text{B})$$

и заключается в следующем:

N – имя плана;

Ev – имя события, для обработки которого предназначен данный план;

Sev – множество имен событий, отправляемых агентом во внешнюю среду при выполнении плана;

MP, MF – завершающие действия, выполняемые соответственно при успешном /неуспешном выполнении плана;

Rel – метод определения применимости плана для

обработки события на основе истинности логического условия, зависящего от результата сопоставления убеждений агента о данном событии;

C – метод определения применимости плана для обработки события на основе истинности логического условия, зависящего от результата сопоставления собственных убеждений агента, при условии, что выполнено логическое условие метода Rel;

B – основной метод рассуждений агента, выполняемый в случае применимости плана.

Выделенные наиболее важные аспекты семантики визуальных моделей среды JASK гарантируют качественное их составление на данном уровне проектирования MAC.

Методика отображения абстрактных моделей Gaia в концепты среды JASK

Разработанная семантика визуальных моделей агента и его ментальных составляющих позволяют качественно создать его визуальное представление с помощью набора графических примитивов инструментария JASK. Процесс создания визуальной модели агента является основополагающим, именно на этом этапе разработки реализуется идея создания компьютерной системы на новом уровне абстракции, и поэтому на этом этапе стоит остановиться подробнее. Чтобы успешно изобразить агента с помощью графических средств инструментария, необходимо создать все составляющие его элементы. Например, как определить, какие события он будет создавать самостоятельно (элементы множества PE), воспринимать для обработки (элементы множества HE) или передавать в окружающую среду (элементы множества SE)? На стадии агентно-ориентированного анализа и проектирования мы имели дело со следующими моделями: моделью ролей для описания должностных обязанностей всех ролей в виде активностей, протоколов, полномочий, обязательств; моделью взаимодействий для описания основных видов и назначения общения между ролями в виде протоколов; моделью агентов для определения типов агентов; моделью функционирования для определения действий агентов и моделью связей для отражения возможных коммуникативных связей между агентами. Поэтому возникает необходимость отображения концептуальных моделей Gaia в визуальные модели среды JASK, чтобы получить представление агента в виде событий, планов, убеждений (рис. 2).

Для успешного решения данной задачи следует придерживаться следующей методики отображения.

При создании модели агента в JACK, все агентные типы для MAC обучения берутся из модели агентов Gaia. Для каждого агентного типа

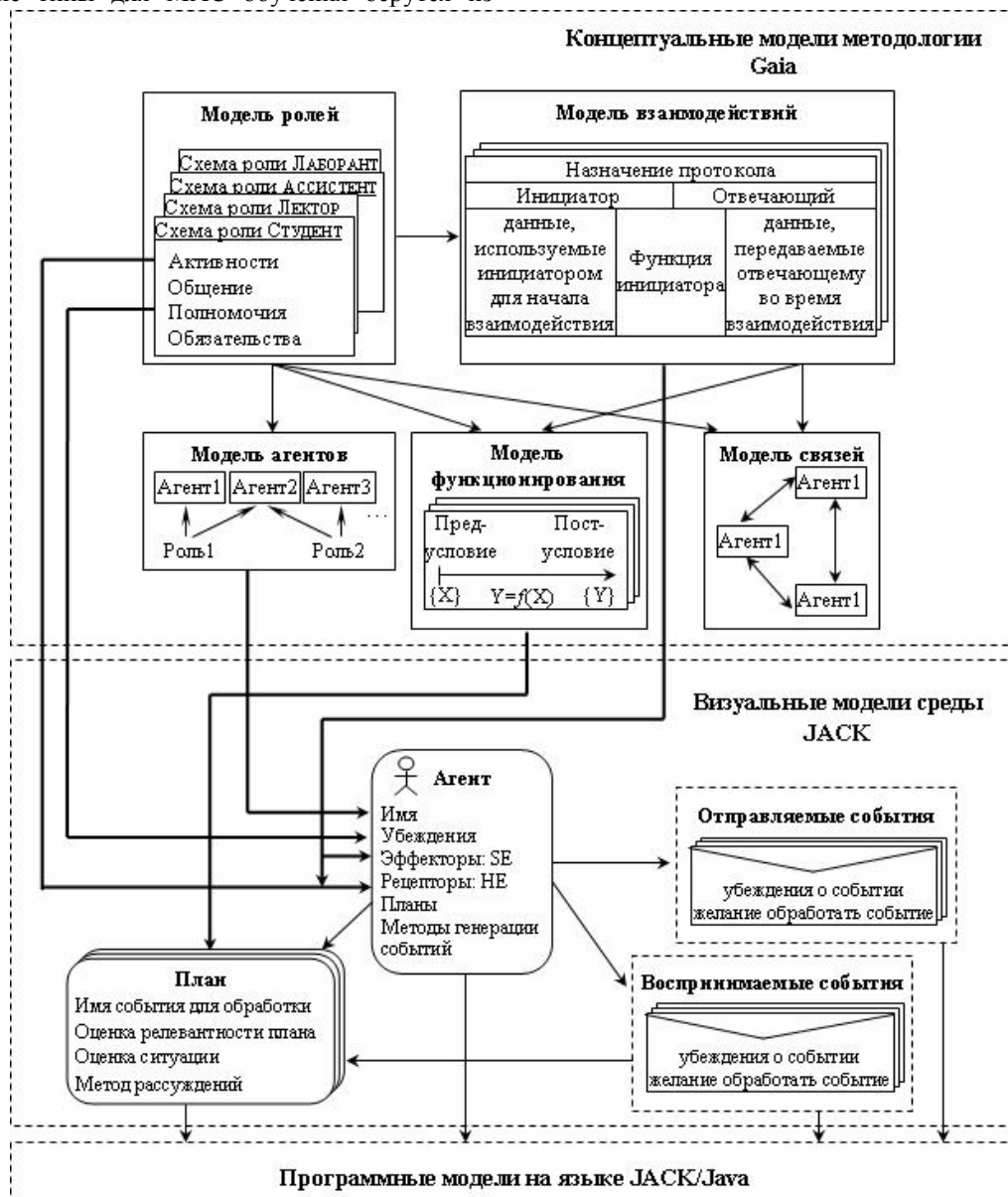


Рисунок 2 – Связь абстрактных моделей методологии Gaia с агентными моделями языка JACK

с помощью базового графического примитива визуально создается агент со своим именем.

При создании модели воспринимаемых и передаваемых событий агента в JACK определяются множества событий HE, PE, SE. Определение событий является ключевым вопросом для определения агента в связи с тем, что его деятельность зависит от их возникновения, если события не происходят - агент бездействует. Если событие произошло, то у агента появляется желание, воспринимаемое как цель, обработать это событие и новые убеждения об этом событии, которые играют определяющую роль при выборе агентом намерений в отношении его ответных действий.

Все желания, которые может иметь агент, в агентном языке JACK определяются множеством

воспринимаемых событий для обработки HE, в котором выделяется два непересекающихся подмножества событий – это множество PE и множество событий, воспринимаемых из внешнего мира. Исходя из того, что все проявления деятельности агента определены активностями и протоколами в модели ролей (рис.2), то предшествующие им события множества PE естественно определить в соответствии с ними. Например, для агента Студент это будут следующие события: “Получить допуск к работе”, “Записаться на изучение дисциплины”, “Выбрать вид занятия”, “Запросить лекцию”, “Слушать лекцию”, “Запросить тестовое задание”, “Выполнять тест”, “Отправить выполненный тест на проверку”, “Запросить задание к лабораторной работе”;

“Выполнить лабораторную работу”, “Отправить отчет на проверку”, “Запросить экзаменационный билет”, “Ответить на экзаменационный билет”, “Отдать экзаменационную работу на проверку” и “Завершить сеанс обучения”. Множество воспринимаемых из внешней среды событий определяется в соответствии с протоколами модели взаимодействий, в которых данный агент является отвечающим. На рисунке 3 показан протокол взаимодействия агентов Студента и Лектора при запросе лекции. Как можно видеть, воспринимаемым событием для агента Студента будет “Получить лекцию”.



Рисунок 3 – Протокол запроса лекции на её изучение

Множество передаваемых во внешнюю среду событий SE определяется в соответствии с протоколами, в которых данный агент является инициатором взаимодействия. Например, в соответствии с протоколом взаимодействия для запроса лекции, показанного на рис. 3, передаваемым событием будет “Передать запрос лекции”.

Таким образом, определяются воспринимаемые и передаваемые события для агента или, другими словами, его рецепторы и эффекторы.

При определении метода создания воспринимаемых агентом событий (элементов множества HE) необходимо выполнить следующее:

- определить параметры для описания каждого события: для событий, инициирующих выполнение активностей, эти параметры определяются владельцем агента, для событий, инициирующих выполнение протоколов взаимодействий, это параметры, на основе которых агент начинает взаимодействие, и извлекаются из соответствующего протокола;
- представить входные данные в виде убеждений агента о воспринимаемом событии посредством метода Pa;
- разработать метод генерации события агентом на основе входных данных для воспринимаемого события.

В результате выполнения вышеописанных действий разработаны модели воспринимаемых и отправляемых событий агента (рис. 2).

Намерения агента относительно его образа действий для достижения поставленной цели определяются множеством планов PS, в котором необходимо определить хотя бы по одному плану для каждого типа событий из множества HE. На визуальном уровне план создается с помощью графического примитива со своим именем. Далее план связывается с предшествующим ему событием (Ev) и с самим агентом. В результате этих действий у агента появляются намерения – замысел, “идеи” действовать определенным образом для достижения поставленной цели в каждой сложившейся ситуации.

Если событие произойдет, то в результате метода Pa агент получит цель, к которой будет стремиться. Каждый план представляет собой заранее намеченную последовательность действий, которую агент может выполнить в имеющей место ситуации. Ситуация определяется путем сопоставления состояния убеждений о воспринятом событии и собственных убеждений агента в методах плана Rel и C соответственно.

При определении множества планов действий агента для обработки каждого воспринимаемого события в зависимости от сложившейся ситуации, следует исходить из того, что весь функциональный аспект агента зафиксирован в модели функционирования уровня Gaia для данного агентного типа. Таким образом, при создании конкретных планов для обработки различных ситуаций, возможных для события, следует использовать описание его функций, разработанное в модели функционирования каждого агентного типа (рис. 2).

При составлении логических условий для методов Rel и C надо исходить из предусловий для выполнения функций данной модели. Показателем успешности достижения цели агентом в результате применения плана (осуществления намерения), определяется в зависимости от выполнения пост условий соответствующей функции.

Если план связан с обработкой протокола взаимодействия, то в нем присутствует множество событий Sev, которое является подмножеством множества SE агента и уже содержится в модели событий агента.

Убеждения каждого агента разрабатываются по моделям ролей и функционирования (рис. 2).

Программные модели на языке JACK

Убеждения агента реализованы в виде Java объектов и конструкции BeliefSet языка JACK. С целью упрощения проекта и возможности повторного использования кода для реализации

функціональних аспектів діяльності агентів використовувалися можливості.

По розробленим в середі JACK візуальним моделям агентів сгенерований скелет програмного Java коду для агентів і їх ментальних властивостей. Сгенерований код необхідно доработати в частині реалізації

основних методів розсуджень в планах агента. Тут були закладені функціональні знання, необхідні агенту для обробки подій.

Фізичні компоненти агентів представлені в вигляді класів з певними відносинами між ними (рис. 4).

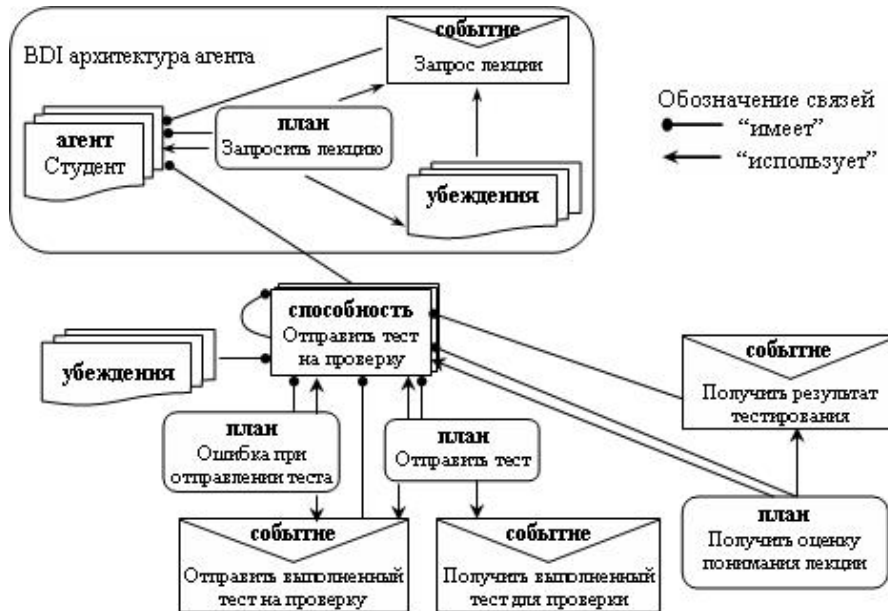


Рисунок 4 – Архитектура программного агента Студента в системе JACK

Заключение

Решена задача преобразования абстрактных моделей высокого уровня, полученных на стадии анализа по методологии Gaia, в модели более низкого уровня, упрощающих реализацию MAS с помощью инструментальной среды JACK. На

основе составленной семантики визуальных моделей среды JACK разработана методика трансформации абстрактных моделей Gaia в физические концепты среды JACK, обеспечивающая качественное выполнение всех этапов разработки многоагентной системы.

Литература:

1. F. Zambonelli, N. R. Jennings, and M. Wooldridge. Developing Multiagent Systems: The Gaia Methodology. In ACM Transactions on Software Engineering Methodology, 12(3):317-370, July 2003.
2. JACK Intelligent Agents. – Mode of access: <http://www.agent-software.com/products/jack/index.html>
3. Федяев О.И., Жабская Т.Е., Грач Е. Г. Многоагентная модель процесса обучения студентов на кафедральном уровне. // Сб. науч. тр. ДонНТУ. Серия: «Проблемы моделирования и автоматизации проектирования динамических систем» (МАП-2006), выпуск 5(116)-Донецк: ДонНТУ, 2006. - С.105-116.
4. David Kinny, Michael Georgeff, Anand Rao A Methodology and Modelling Technique for Systems of BDI Agents. In W. van der Velde and J. Perram, editors, Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent MAAMAW'96, (LNAI Volume 1038). Springer-Verlag: Heidelberg, Germany, 1996.
5. M.P. Georgeff, and A.S. Rao. Formal models and Decision Procedures for Multy-Agent Systems. Technical note 61, Australian Artificial Intelligence Institute, 1995.
6. Жабская Т.Е., Федяев О.И. Анализ инструментальных средств построения агентно-ориентированных систем // Сб. научных трудов Донецкого национального технического университета, серия "Информатика, кибернетика и вычислительная техника", выпуск 10 (153) – Донецк: ДонНТУ, 2009. – С. 54 – 62.

Поступила в редакцию 30.03.2010