УДК 004.434: 004.891 **Информационные технологии**

Слепушко Артём Алексеевич, *магистрант, Московский государственный университет приборостроения и информатики, Россия, Москва*
Корягин Сергей Викторович, *кандидат технических наук, доцент кафедры «Управление и моделирование систем» МГУПИ, «Московский государственный университет приборостроения и информатики», Россия, Москва*

ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЙ ГРАФИЧЕСКИЙ ЯЗЫК ДЛЯ ОПИСАНИЯ МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ

Аннотация: Данная статья посвящена проектированию и разработке проблемно-ориентированного языка описания моделей предметной области. Упомянуты схожие разработки и программные решения, предложена новая графическая модель представления знаний и моделирования. Представлена базовая инструкция формирования моделей на проектируемом языке и описан пример и результаты выполнения программы. Рассмотрены примеры построения пользовательских моделей, иллюстрирующих функциональные возможности предложенного авторами проблемно-ориентированного языка.

Ключевые слова: игровые приложения, многоагентные системы, методы трансляции, проблемно-ориентированные языки программирования, алгоритмы поиска кратчайшего пути, искусственный интеллект.

Annotation: This article is devoted to the design and development of a problem-oriented language for describing domain models. Similar developments and software solutions are mentioned, a new graphical model of knowledge representation and modeling is proposed. The basic instruction for the formation of models in the projected language is presented and the example and results of the program execution are described. Examples of building user models illustrating the functionality of the problem-oriented language proposed by the authors are considered.

Keywords: game applications, multi-agent systems, translation methods, problem-oriented programming languages, shortest path search algorithms, artificial intelligence.

Введение

В основу решения была взята математическая модель чашек петри, декларативный текстовый язык Prologue и инструментальная система «Малый Решатель Проблем», который в свою очередь является упрощенным аналогом экспертных систем [1; 2]. Основным отличием языка является упрощенная для понимания модель представления знаний, язык также сделан с целью популяризации декларативного и графического программирования.

Рассмотренные методы могут служить базой для других программных решений, а также могут быть использованы в образовательных и исследовательских целях.

Формально представленный язык оформляется в графическом редакторе draw.io (app.diagrams.net). Инструмент выбран из-за его доступности, в отличии от аналогов программные компоненты draw.io доступны из сети, их не нужно скачивать, так же не требуется платная лицензия на использование. А исполнителем языка служит расширение для браузеров на хромиуме.

Язык создан для проектирования многогенных систем, которые часто применяются на производстве и в компьютерных играх [3]. С помощью такого инструмента также можно по моделировать цифровой документооборот описывая правила получения тех или иных услуг, а не порядок их получения. Что позволит снизить зависимость бизнес-процессов друг от друга и позволит сосредоточится на скорейшем достижении оказания услуги.

Описание языка

Графический язык состоит из следующих компонент

Объемный прямоугольники – агент, объекты действия, активные объекты, которые выполняют некоторые **действия** и меняю своё общее **состояние,** накапливая или теряя некоторые свойства. Название в объёмном

прямоугольнике служит для обозначения (наименования) объекта действия, участвующего в операциях переходах и может быть любым текстом. Рекомендуется называть объекты существительными. Каждый объект, названный одинаково считается одним и тем же объектом.

Плоский прямоугольник – действие, механизм смены состояния, действия не меняющие состояния не имею смысла и не участвуют в логическом выводе (трансформациях) [4]. Название в плоском прямоугольнике служит для наименования действия и может быть любым текстом. Желательно писать наименования действия как глагол в нормальной форме.

Круг – состояние, бывают двух видов, различаются цветом (зелёный/красный).

При прикреплении состояния к агенту, связанного с действием, накладывается ограничение на его выполнение, рассмотрим пример на рисунке 1.

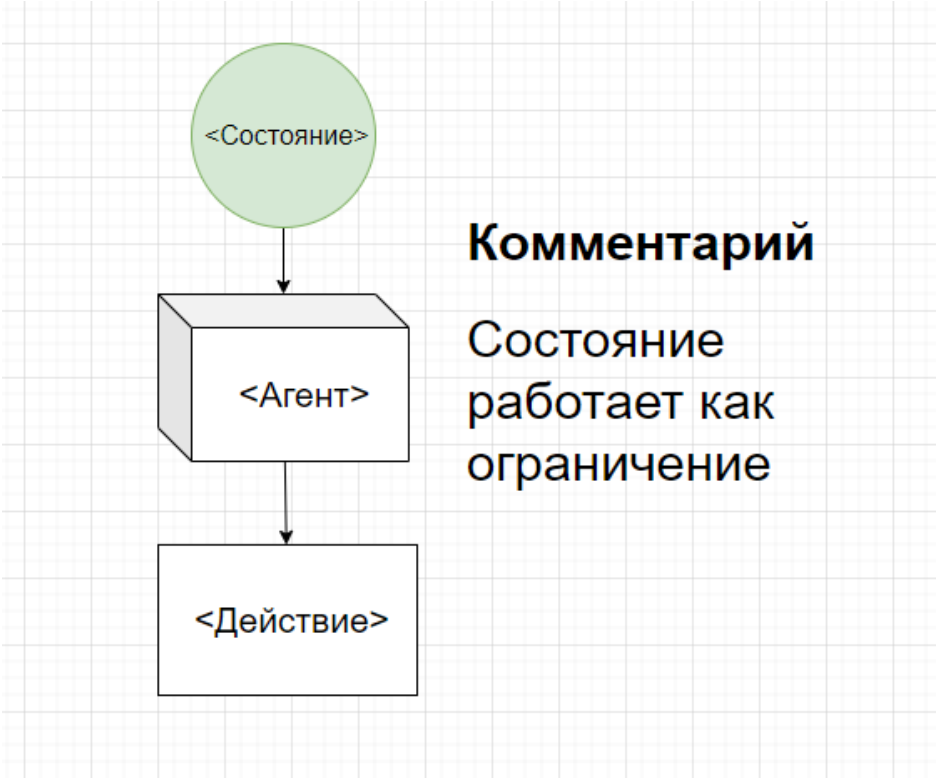

Рисунок 1 – композиция состояния, агента и действия

Такая комбинация означает что для того, чтобы “Агент 1” выполнил “Действие” требуется некоторое “Состояние”. Таким образом получается простейшая условная конструкция, накладывающая ограничение на выполнение действия.

Количество требуемых состояний – не ограничено, так для формирование логического “И” к агенту 1 дополняется крепится дополнительное состояние.

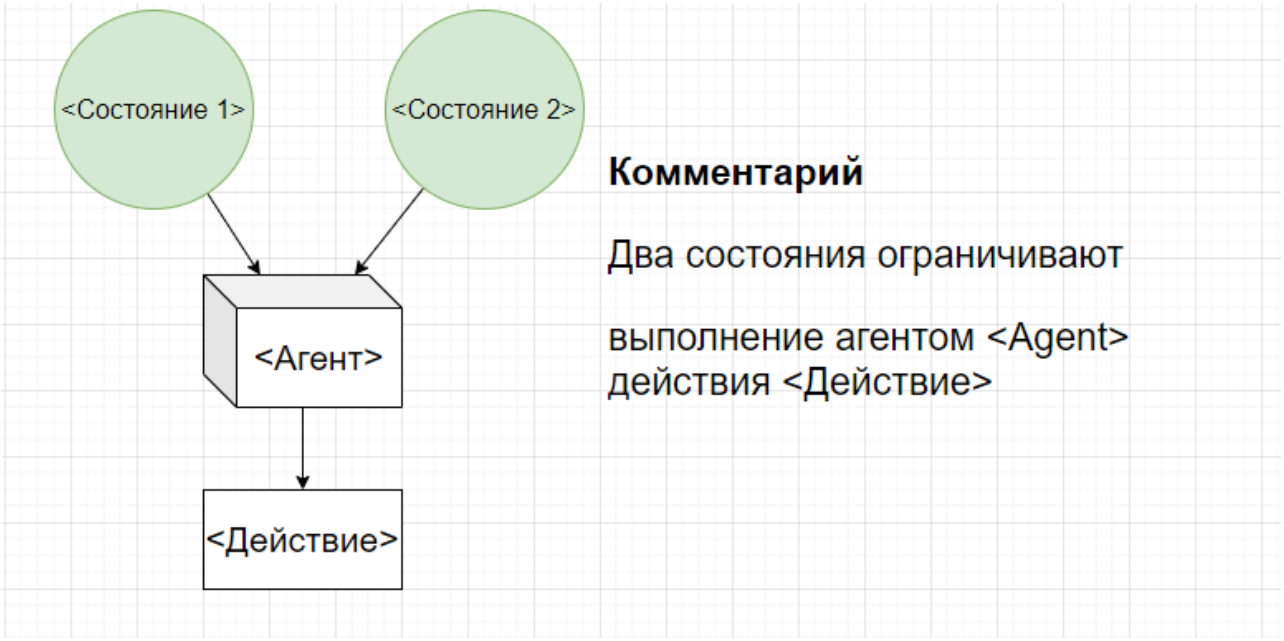

Рисунок 2 – композиция двух ограничивающих состояний

Для формирования логического ИЛИ требуется промежуточное состояние на полотне, обозначающее то или требуемое состояние или связь состояние 1 и 2 стрелкой.

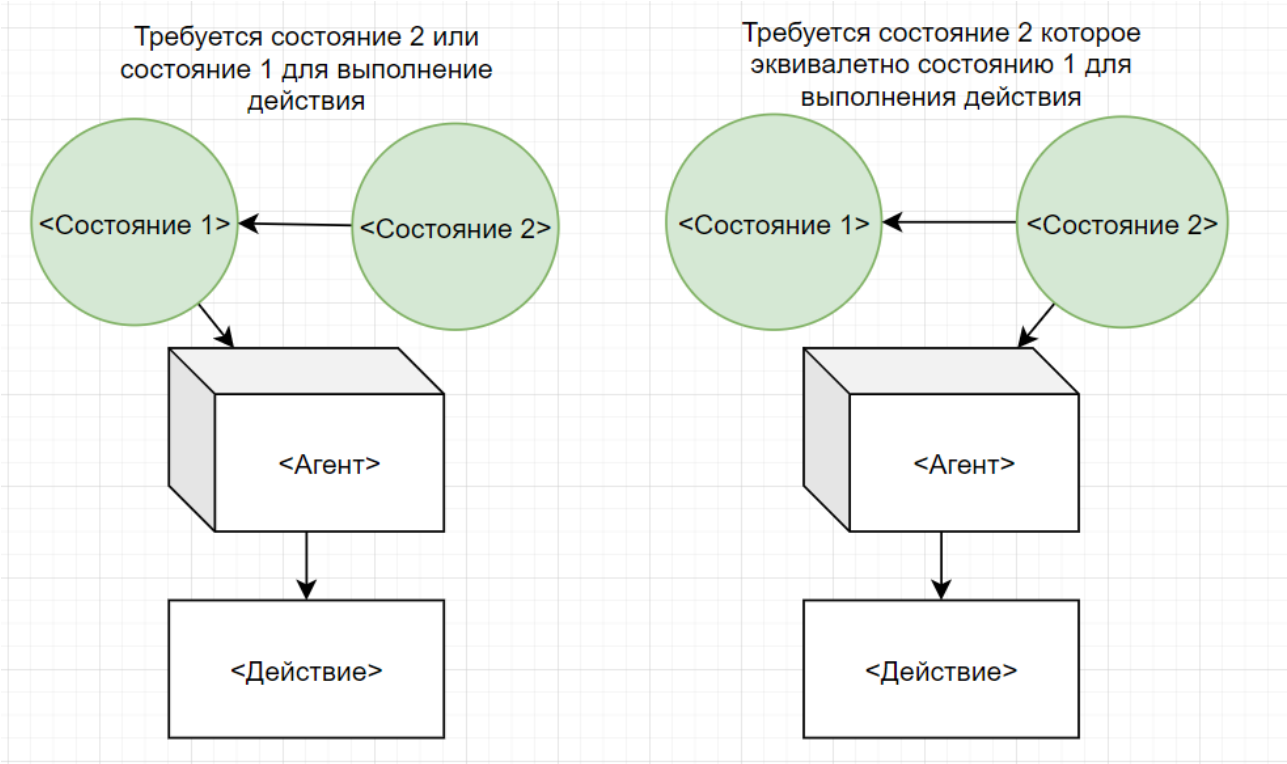

Рисунок 3 – транзитивность состояний

Если агенту присвоено активное состояние значит оно требуется для действия, если же оно пассивное, то агент сможет выполнить действие если у него такого состояния не имеется. Для действий так же состояния могут быть включающие – наделяющие состоянием агента, или исключающие – деактивирующие состояние агента.

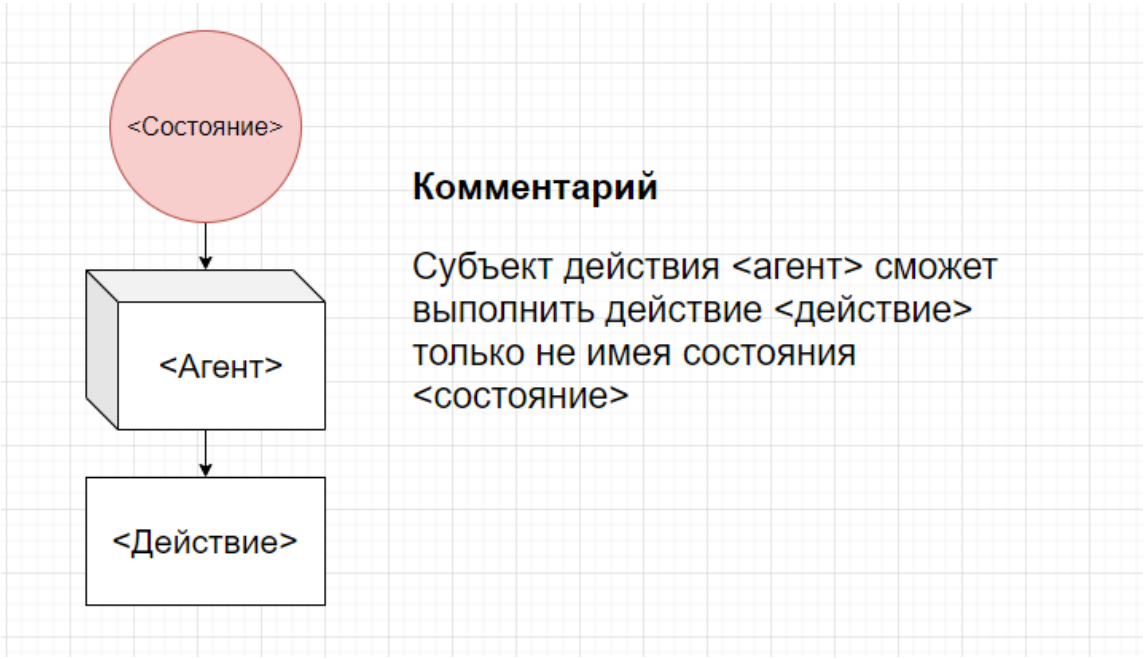

Рисунок 4 – исключающее состояние

Стрелки – связи в модели этого графического языка положение фигур не имеет значения важны только направленность связей. Стрелки выступают в качестве инструмента связывания агентов, действий и состояний.

- Состояние, связанное с агентом накладывает ограничение на выполнение действия.
- Агент связанный с действием показывает субъекта выполняемого действия

Также для работы с состояниями существуют связи:

- Действие, связанное с агентом состоянием агент добавляет состояние агенту.
- Действие, связанное с пассивным состоянием связанное агентом накладывает даёт пассивное состояние агенту.

Рисунок 5 – пример смены состояния

В текущем примере описан агент, находящийся в любом состоянии, и при выполнении действия получающий обновление состояния.

Пример работы

Программирование на языке происходит следующим образом: описывается доменная модель предметной области и её правила, определяется достигаемое состояние – цель и запускается её достижение из определенного состояния, которое можно задать действием без состояния [5]. В качестве

примера предлагается рассмотреть рисунок 6 и запустим приложение и с цель “Поесть”. Логическим выводом такого приложения будет следующий текст что и является формальным выходным языком:

1. Проснуться [+Может спать, +Деньги, В Маске].
2. Выйти на улицу [Может спать, Деньги, В Маске, + На улице].
3. Зайти в магазин [+В Магазине, может спать, Деньги, В маске, -На улице].
4. Купить хлеб [+Хлеб, может спать, -Деньги, В маске].
5. Поесть [-Хлеб, может спать, В Маске].

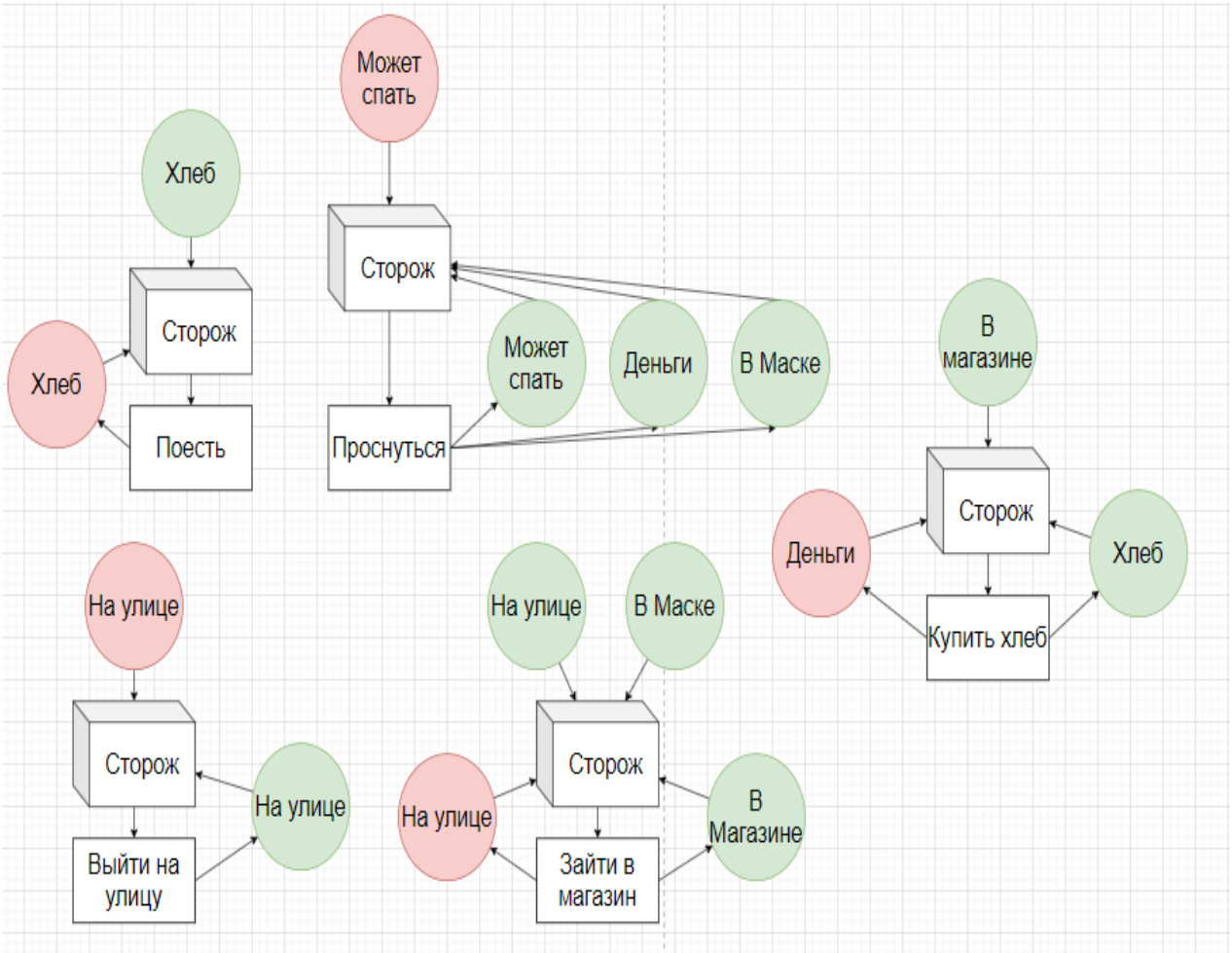

Рисунок 6 – пример программы на графическом языке

Вывод

В статье были рассмотрены примеры построения моделей на описываемом графическом языке построения моделей предметной области, продемонстрирован получаемый текст на естественном языке.

По результатам проведенной работы был спроектирован проблемно-ориентированный язык программирования [6; 7].

В реализованном приложении используются принципы поиска кратчайшего пути и логический вывод.

Библиографический список:

1. Остроух, А. В. Интеллектуальные информационные системы и технологии: монография / А. В. Остроух, А. Б. Николаев. — Санкт-Петербург: Лань, 2019. — 308 с. — ISBN 978-5-8114-3409-1. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/115518 (дата обращения: 18.11.2021). — Режим доступа: для авториз. пользователей.
2. Смольянинова, В.А., 2011. Методическое и программное обеспечение процессов выявления и представления знаний в интеллектуальных системах поддержки принятия решений, Диссертация, МИРЭА, Москва.
3. Корягин С.В. Перекодировщик языков непрерывного моделирования. М., Промышленные АСУ и контроллеры. 2013. № 10, С. 52 – 56.
4. Ахо А., Сети Р., Ульман Д. Компиляторы: принципы, технология и инструменты: пер. с англ. –М., 2011. – 768 с., илл.
5. В.Ш. Кауфман, «Языки программирования. Концепции и принципы», 2011 г. ДМК Пресс, 2-е издание.
6. А.М. Караева, «Проблемно-ориентированные языки в программировании», 2017 г. Международная научно-практическая конференция «Научные исследования: векторы развития».
7. В.Л. Рвачов, А.Н. Шевченко, «Проблемно-ориентированные языки и системы для инженерных расчетов», 1988 г. Издательство «Тэхника».