

УДК 519.7, 519.17

АНАЛИЗ ЯЗЫКОВ ЗАДАНИЯ ПРОТОКОЛОВ ВЗАИМОДЕЙСТВИЯ В ОТКРЫТЫХ АГЕНТСКИХ СРЕДАХ



Д.И. Черемисинов

*Ведущий научный сотрудник ОИПИ НАНБ
кандидат технических наук, доцент*



Л.Д. Черемисинова

*Главный научный сотрудник ОИПИ НАНБ
доктор технических наук, профессор*

*Объединений институт проблем информатики Национальной академии наук Беларуси,
Республика Беларусь*

E-mail: {cher, cld}@newman.bas-net.by

Д. И. Черемисинов

Окончил Томский государственный университет, кандидат технических наук, доцент. Работает в ОИПИ НАН Беларуси в должности ведущего научного сотрудника и Белорусском государственном университете информатики и радиоэлектроники в должности доцента.

Круг научных интересов: программирование, логическое проектирование и тестирование дискретных систем управления, реализация параллельных алгоритмов управления.

Л. Д. Черемисинова

Окончила Томский государственный университет, доктор технических наук, профессор. Работает в ОИПИ НАН Беларуси в должности главного научного сотрудника и Белорусском государственном университете информатики и радиоэлектроники в должности профессора. Круг научных интересов: дискретная математика, логическое проектирование и тестирование дискретных систем управления, реализация параллельных алгоритмов управления.

Аннотация. Объектом исследования являются мультиагентная система, рассматриваемая как композиция агентов, взаимодействующих асинхронно и параллельно и связанных каналами передачи данных. Рассматриваются модели поведения и логической структуры взаимодействия агентов в мультиагентной (распределенной) системе логического типа, а также методология использования автоматных языков (конечных, обобщенных и параллельных автоматов) и языков описания протоколов с «истинным» параллелизмом (сетей Петри, алгоритмов логического управления) в качестве языков описания протоколов взаимодействия в мультиагентных системах. Проведен сравнительный анализ возможностей этих языков для представления последовательностей состояний диалога, приема и отправки сообщений, а также для описания поведения всей мультиагентной системы как единого целого.

Ключевые слова: мультиагентная система, протокол взаимодействия, автоматная модель, сети Петри, параллельный алгоритм управления.

Введение.

В последние десятилетия возрастает число компьютерных систем, разрабатываемых на базе моделей, использующих концепцию агентов и мультиагентных систем (МАС). Первые приложения МАС появились в середине 1980-х годов, в настоящее время они применяются во многих предметных областях, моделируемых распределенными децентрализованными системами, например, при управлении производственными процессами, воздушным сообщением, информационными потоками, деятельностью предприятий. Технология программирования на основе использования

взаимодействующих агентов считается наиболее перспективным инструментом современного программирования и проектирования систем управления.

История развития теории агентов начиналась с задачи моделирования свойств живых систем. Она восходит к работам У. Питтса и У. Мак-Каллока по формальным нейронам, Дж.фон Неймана по самовоспроизводящимся автоматам, А.Н. Колмогорова по теории сложности, У. фон Форстера и И. Пригожина по теории самоорганизации, У.Р. Эшби по моделям гомеостазиса, Г. Уолтера по реактивным роботам, Дж. Холланда по генетическим алгоритмам. Использование технологии агентов упрощает процесс разработки сложных распределенных систем. Первые приложения МАС появились в середине 1980-х годов, в настоящее время они применяются во многих предметных областях, моделируемых распределенными децентрализованными системами. Наиболее известной промышленной системой, построенной на основе концепции агентов и предназначенной для управления процессом производства изделий на предприятии, является ARCHON [1]. Эта система использовалась на практике для управления транспортировкой электроэнергии в Ирландии.

Мультиагентные системы (от англ. multi-agent systems) состоят из множества искусственных агентов, которые работают вместе, чтобы достигнуть некоторых целей [1, 2, 3]. Агент представляет собой открытую систему, помещенную в некоторую среду, причем агенты обладают собственным поведением, удовлетворяющим определенным правилам. Примерами искусственных агентов служат роботы.

Центральным понятием мультиагентной системы является взаимодействие, оно предполагает обмен сообщениями между агентами и связано со способом, которым агенты обмениваются информацией. Решение этой задачи осуществляется посредством сотрудничества агентов и достигается через координацию их работы. Цель взаимодействия состоит в том, чтобы осуществить синхронизацию агентов для обмена информацией.

Разработка МАС является сложной задачей, так как требует реализовать не только алгоритмы поведения агентов, но и синхронизацию их взаимодействия. При этом приходится решать не только проблемы, возникающие при проектировании традиционных распределенных систем с параллелизмом поведения, но и дополнительные проблемы, возникающие из-за необходимости учета гибких и сложных взаимодействий между автономными компонентами, решающими плохо формализуемые задачи.

Если абстрагироваться от назначения агентов, то единственной целью взаимодействия является синхронизация поведения взаимодействующих агентов, так как ненадлежащая синхронизация при взаимодействии агентов полностью разрушает целесообразность совместного поведения мультиагентной системы. Достижение синхронизации требует специальной организации взаимодействующих процессов. Центральную роль в организации мультиагентной системы играют протоколы.

В докладе предлагаются модели поведения и логической структуры взаимодействия агентов в мультиагентных (распределенных) системах логического типа, а также технология использования автоматных языков (конечных, секвенциальных и параллельных автоматов) и языков логического управления в качестве языков описания протоколов взаимодействия в мультиагентных системах. Показано, в какой мере описание поведения агентов на этих языках позволяет представлять последовательности состояний диалога, приема и отправки сообщений между агентами, а также представление поведения всей мультиагентной системы как единого целого.

Агенты.

Несмотря на широкую распространенность термина «агент» на сегодняшний день нет единого и общепринятого определения этому термину. Большинство специалистов в области МАС придерживаются определения, данного Дженнингсом Вулдриджем [4]: агентом является объект для решения некоторой задачи (реализованный аппаратными или программными средствами или их комбинацией), который обладает следующими свойствами:

- автономностью (должен выполнять свои задачи и контролировать свое состояние без вмешательства извне);
- способностью взаимодействия (с другими агентами или человеком для решения своих задач);
- реактивностью (реагировать на изменения окружающей среды);
- направленностью действий на достижение некоторой цели.

В дополнение к этим свойствам иногда добавляют такие свойства как мобильность, рациональность, правдивость, интеллектуальность и т.д.

Но чаще под агентом понимают открытую систему, помещенную в некоторую среду и обладающую собственными состояниями и поведением, которое удовлетворяет некоторым определенным правилам. Примерами искусственных агентов служат роботы.

Каждый агент в МАС имеет неполную информацию относительно целой системы и восприятие им среды зависит от его способностей и реализуемого алгоритма поведения. Для того чтобы моделировать и управлять «интеллектуальным» поведением агентов предлагались системы, в которых агенты имеют одинаковую «архитектуру». Обычно предполагается, что каждый агент имеет реактивное поведение, т.е. когда агент в состоянии реагировать на внешние стимулы в соответствии с предопределенной схемой; в этом случае способ выработки реакции безотносителен к контексту (состоянию системы в целом).

Поведение каждого агента системы может быть задано некоторой последовательной программой и представлено последовательностью трех типов событий: внутренних (его состояние), отправки сообщения и получения сообщения. Внутренние события затрагивают процесс только того агента, в котором они происходят, и эти события в процессе линейно упорядочиваются в соответствии с порядком возникновения. События отправки и получения описывают поток информации между агентами. Коммуникация между процессами задается отправкой сообщения в процессе одного агента и приемом этого сообщения в процессе другого агента. Из физических соображений задержка между отправкой и приемом не может отсутствовать. Задержка при принятии посланного сообщения конечна, но непредсказуема.

Мультиагентные системы.

При решении сложных, реалистических задач приходится использовать модель не отдельного агента, а мультиагентной системы. На сегодня нет одного общепринятого определения МАС. Большинство исследователей определяют МАС как вычислительную систему, которая состоит из множества агентов, которые работают совместно, чтобы достигнуть некоторых целей. МАС наделяют следующими основными характеристиками:

- управление в системе децентрализовано;
- данные децентрализованы;
- вычисления асинхронны;
- каждый агент решает только часть общей задачи.

Взаимодействие агентов в МАС дает им возможность координировать работу для достижения своих целей. Формализация понятия взаимодействия как способа передать сообщение от одного отправителя нескольким получателям через среду передачи отталкивается от работы Шэннона по теории связи [5]. Дальнейшая формализация этого понятия использует теорию речевых актов [6], в которой коммуникация агентов рассматривается как поведение, так как определенные виды предложений естественного языка имеют характер действий (называемых актами речи) и подразумевают «рациональный эффект». Акты речи составляют основу для языков коммуникации агентов (языки KQML, FIPA-ACL [7, 8, 9]), определяющих наборы допустимых актов речи и связанной с ними семантики.

Множество взаимосвязанных сообщений образует переговоры, в которых агенты играют различные роли в зависимости от их индивидуальных или общих целей. Переговоры в мультиагентных системах основаны на протоколах взаимодействия, которые

определяют все возможные течения переговоров. В большинстве моделей мультиагентных систем поведение агентов описывается в терминах предположений, желаний и намерений (beliefs, desires and intentions – BDI) [9], то есть на основе понятий из социологии и гуманитарных наук, а коммуникация задается в терминах протоколов, которые не имеют прямой связи с первыми. Одна из проблем в связи таким подходом состоит в том, что чрезвычайно трудно разрабатывать и моделировать коммуникацию между агентами. Эта проблема возникает вследствие отсутствия модели агента, объединяющей аспекты внутреннего состояния и коммуникации. Главная причина отсутствия общей модели состоит в том, что отсутствует общее концептуальное основание, объединяющее все абстракции, связанные с агентами.

Для преодоления этих методологических трудностей используются такие языки представления теории агентов, в основе которых лежит некий формализм, задающий семантику языка программирования агентов. Хотя в литературе предлагаются все новые языки программирования агентов, но немногие из них полностью понятны с семантической точки зрения.

В целом мультиагентная система не имеет централизованного управления, но результаты распределенных вычислений в виде асинхронного поведения составляющих агентов или объектов обеспечивают достижение поставленной перед системой цели. Для того чтобы моделировать и управлять «интеллектуальным» поведением агентов предлагались системы, в которых агенты имеют одинаковую «архитектуру».

Проблема координации в мультиагентных системах.

Координация в МАС обеспечивает непротиворечивость и согласованность действий агентов. Существует множество подходов к обеспечению координации действий в МАС. Самый простой из них – организационное структурирование, предполагающий классические структуры типа доски объявлений, master-slave или клиент-серверные архитектуры. Самый сложный метод обеспечения координации – организация переговоров, которые представляют собой последовательности взаимосвязанных сообщений. Переговоры в МАС задаются двумя компонентами: сообщениями и протоколами. Первые описываются на языках коммуникаций (общений), вторые на языках описания протоколов. В переговорах агенты играют разные роли в зависимости от их индивидуальных или общих целей их взаимодействия.

Задание переговоров в мультиагентных системах определяется двумя компонентами: сообщениями и протоколами. Сообщения описываются на языках коммуникаций, таких как KQML (Knowledge Query и Manipulation Language) [7, 8], FIPA-ACL (Foundation for Intelligent Physical Agents - Agent Communication Language) [9] и других [10]. На этих языках коммуникаций описывается специфический для данной области словарь (онтология [11]) и сообщения, которыми могут обмениваться агенты. Синтаксисы описаний на языках KQML и FIPA-ACL задается форматом обмена данными, а семантика задается онтологией. Описания обмена сообщениями на этих языках схожи.

Переговоры в мультиагентных системах основаны на протоколах взаимодействия, которые определяют все возможные течения переговоров. В большинстве моделей мультиагентных систем автономное поведение агентов описывается с использованием приведенных формализмов высокого уровня абстрактности, а коммуникация задается в понятиях, близких к реализации. Разница в уровнях описания не позволяет моделировать коммуникацию между агентами на том уровне, в котором описано их автономное поведение. Эта проблема возникает вследствие отсутствия модели агента, объединяющей аспекты внутреннего состояния и коммуникации.

В качестве концептуальной основы задания коммуникации агентов предлагается способ описания условий синхронизации как предположений о свойствах событий посылки и приема сообщений, не связанных с механизмом их реализации. Так как описания автономного поведения агентов могут интерпретироваться как логические формулы, то

описание коммуникации тоже должно интерпретироваться на уровне логики для обеспечения возможности целостного моделирования поведения мультиагентной системы.

Отображение параллелизма и асинхронности в описании поведения агентов.

В известных моделях описания поведения мультиагентных систем физический параллелизм и асинхронность функционирования агентов отображается по-разному. В литературе по MAS сформировалось мнение о трудности анализа систем с учетом параллелизма. Это мнение в немалой степени объясняется тем, что существуют различия в понимании этого явления. В работе [12] приводится анализ и классификация моделей распределенных систем (к каковым относятся и MAS) по трем параметрам: типу модели (поведенческая или структурная), типу параллелизма (чередование или истинный параллелизм), модели времени (линейная или ветвящаяся). В работе [13] анализируются понятия параллелизма и асинхронности – ключевые понятия, лежащие в основе современных представлений о проблемах проектирования MAS. Физические понятия параллелизма и асинхронности имеют много аспектов, не все из которых целесообразно отражать в математической модели.

Стремление к минимальности средств описания поведения MAS, а также мнение о сложности понимания описаний, в которых присутствует параллелизм, обуславливают применение моделей и языков описания протоколов, в которых понятия параллелизма и асинхронности не применяются. Однако модели, использующие параллелизм, позволяют выявить такие ошибки в поведении MAS, которые возникают вследствие параллельности функционирования агентов MAS (например, тупики и ловушки), и научиться избегать их.

Протоколы взаимодействия агентов.

Центральную роль в организации MAS играют протоколы, определяющие правила взаимодействия между двумя и более агентами, вступающими во взаимодействие. Протокол – это набор правил, которым соответствует взаимодействие, имеющее место при координации работы нескольких агентов. Протокол для координации взаимодействия агентов в мультиагентной системе имеет следующие особенности:

- протокол определяет все возможные допустимые течения переговоров, т.е. все возможные допустимые последовательности обмена сообщениями;
- протокол сокращает пространство поиска возможных решений агента;
- протокол ограничивает диапазон ответов агентов на сообщения, возможные в данном состоянии (в данной ситуации), позволяя сократить пространство поиска возможных решений;
- протокол не зависит от языка общения агентов;
- протокол отображает причинно-следственные и временные отношения между актами взаимодействия агентов.

Анализ причинно-следственных зависимостей позволяет установить возможные последовательности возникновения событий при функционировании протокола. Это дает возможность выявить, реализуются ли при выполнении протокола желательные с точки зрения разработчика протокола события, и обнаружить ошибки, вызывающие нежелательные события.

Формальные модели протоколов изучались в рамках теории распределенных вычислений. Фундаментальным признаком, по которому отличаются эти модели, является степень синхронизации поведения участников взаимодействия. Если абстрагироваться от назначения агентов, то единственной целью взаимодействия является синхронизация поведения взаимодействующих агентов. Ненадлежащая синхронизация при взаимодействии агентов полностью разрушает целесообразность совместной работы агентов в рамках мультиагентной системы. Достижение синхронизации требует специальной организации взаимодействующих процессов.

Когда агенты вовлечены во взаимодействие, где параллелизм не допустим, протоколы традиционно задаются детерминированными конечными автоматами и диаграммами состояний (statecharts). Самым простым из других представлений протокола является диаграмма потока сообщений, такая как используется в стандарте FIPA [10].

Для более сложных протоколов лучшим представлением являются диаграммы взаимодействия таких языков, как универсальный язык моделирования UML (Unified Modelling Language) [14], AUML [15], цветные сети Петри (Colored Petri Nets) [16, 17] и языки описания алгоритмов логического управления [18, 19].

Представление протоколов на языках UML и AUML

Одним из наиболее популярных в настоящее время графических языков проектирования является язык UML (Unified Modelling Language), который де-факто является стандартом для описания систем программного обеспечения. Модель программной системы на UML описывается интегрированным когерентным набором UML-диаграмм [14]. Эти диаграммы описывают разные аспекты системы в целом и отдельных ее компонент: поведение, структуру, взаимодействие, временные характеристики и т.д.

В литературе описаны попытки описания мультиагентных систем на языке UML, это, однако, оказалось непростым делом, поэтому был предложено расширение языка UML для описания протоколов для MAS – язык AUML (Agent UML). В язык AUML введены новые диаграммы, ориентированные на представление асинхронного обмена сообщениями между агентами.

Аналитические обзоры отмечают, что AUML имеет два основных недостатка: он определен недостаточно точно и формально и потому отсутствуют описание формальной семантики и реализации этого языка. Отсутствие формальной семантики AUML ведет к возможным несоответствиям в описаниях диаграмм, а значит и в поведении MAS.

Представление протоколов конечными автоматами.

Детерминированный конечный автомат широко используется в качестве языка задания протоколов MAS [20] и задает простой, но достаточно широкий класс протоколов взаимодействия. В этом формализме предполагается выполнение следующих ограничений на вид взаимодействий агентов системы: как правило, рассматриваются диалоги, т. е. системы, состоящие только из двух агентов, которые последовательно чередуют шаги диалога (передачу и прием сообщений). Эти ограничения (особенно исключение параллелизма) позволяют ограничиться специфическим классом протоколов, представимых посредством хорошо изученных конечных автоматов.

Конечный автомат задается 1) множеством состояний (включая начальное состояние и набор заключительных состояний); 2) входным и выходным алфавитами; и 3) функцией переходов и выходов, которая отображает пару: состояние и символ входного алфавита в следующее состояние и символ выходного алфавита. В контексте протоколов взаимодействия символы входного алфавита – это шаги диалога, и состояния – возможные стадии взаимодействия.

Протокол, заданный конечным автоматом, определяет класс правильно построенных диалогов, где каждый шаг диалога – это допустимое продолжение взаимодействия, которое имело место к этому моменту. Если протокол задан автоматом с функцией перехода φ , то шаг диалога P допустим в состоянии s , если и только если существует состояние s^* , такое что $s^* = \varphi(s, P)$. Однако для описания систем с большим числом агентов, систем с параллелизмом применение этого языка обременительно.

Представление протоколов параллельными автоматами.

Формализм параллельного автомата, предложенного А.Д. Закревским [21], позволяет задавать протоколы взаимодействия агентов достаточно широкого класса. Основные отличия и достоинства языка параллельных автоматов по сравнению с языком конечных автоматов состоят в том, что он: 1) позволяет в явном виде отображать параллелизм

функционирования отдельных компонентов МАС; 2) не накладывает ограничения на число агентов; 3) позволяет единообразно описывать внешнюю среду МАС и взаимодействие агентов. Параллельный автомат является расширением последовательного конечного автомата. Принципиальным отличием параллельного автомата от последовательного является то, что каждое его состояние, называемое в этой модели «полным», может быть представлено в виде совокупности некоторых взаимно совместимых «частичных» состояний. Соответственно параллельный автомат может одновременно находиться в нескольких частичных состояниях, называемых потому параллельными.

Параллельный автомат задается 1) множеством S частичных состояний; 2) входным и выходным алфавитами X и Y , которые состоят из входных и соответственно выходных булевых переменных; 3) обобщенными переходами $(X_{kl}, S_k) \rightarrow (S_l, Y_{kl})$ автомата, реализуемыми независимо друг от друга (параллельно другим, допустимым в текущем состоянии автомата). Начальное и заключительное состояния параллельного автомата представляют собой в общем виде подмножества S_i частичных состояний.

Обобщенный переход $(X_{kl}, S_k) \rightarrow (S_l, Y_{kl})$ задает переход параллельного автомата из подмножества S_k в подмножество S_l частичных состояний. Каждому обобщенному переходу соответствует условие перехода X_{kl} – совокупность значений входных булевых переменных и совокупность Y_{kl} значений, изменяемых в результате перехода выходных булевых переменных. X_{kl} и Y_{kl} представляются в виде элементарных конъюнкций булевых переменных. В некоторых фрагментах могут отсутствовать X_{kl} и/или Y_{kl} , что интерпретируется как $X_{kl} = 1$ и/или $Y_{kl} = 1$.

МАС рассматривается как система из параллельно и асинхронно работающих агентов. В контексте протоколов взаимодействия агентов частичные состояния могут отображать поведение разных агентов, а протокол функционирования каждого i -го агента определен на своем подмножестве множества частичных состояний. Синхронизация действий агентов может быть достигнута в рассматриваемой модели введением обобщенных переходов $(X_{kl}, S_k) \rightarrow (S_l, Y_{kl})$, в которых одно из множеств S_k или S_l , и оба, имеют мощность более, чем 1. Причем, если $|S_k| > 1$, означает в нашем случае синхронизацию состояний агентов: только в том случае, когда $|S_k|$ агентов будут находиться в состояниях из S_k , необходимо выполнить действия из Y_{kl} . $|S_l| > 1$ означает перевод $|S_l|$ агентов в состояния из S_l .

Полное описание протокола функционирования мультиагентной системы, стоящей из m агентов, может быть получено следующим образом. Сначала записывается протокол автономного функционирования каждого i -го агента на языке конечных автоматов, затем синхронизация действий агентов отображается в виде протокола на языке параллельных автоматов.

Представление протоколов обобщенными конечными автоматами.

Обобщенный конечный автомат [22] можно считать идеологическим предшественником параллельного автомата. Базовыми понятиями обобщенного конечного автомата являются переход, переменная и ее значение. Множество переменных определяет пространство состояний обобщенного автомата, а отдельное его состояние определяется значениями всех его переменных. Обычно одна из переменных выделена и называется основным состоянием, другие переменные называются контекстными.

Формализм обобщенного автомата ориентирован на описание функций компонентов распределенной системы – протокольных объектов и среды взаимодействия. Вся система в целом требует для своего описания модели сети взаимодействующих автоматов. Включение в модель автомата переменных позволяет представить группы переходов конечного автомата командами абстрактной машины в «интервальном» виде.

Но в отличие от модели параллельного автомата, понятие параллелизма в формализме обобщенного автомата, так же как и в модели конечного автомата, в явном виде не используется.

Представление протоколов на языке цветных сетей Петри

Модель системы в виде цветных сетей Петри позволяет описывать естественным образом параллелизм, присущий мультиагентным системам. Сеть Петри представляет множество состояний, в которых может находиться система, и переходы между этими состояниями. Формализм цветных сетей Петри обеспечивает математический базис для описания, реализации и анализа распределенных и параллельных систем, может выражать взаимодействия в графической форме и обладает строгой семантикой, что позволяет автоматизировать формальный анализ и преобразования описаний [16]. Используя цветные сети Петри, протокол взаимодействия агентов может быть задан сетью компонентов, задающей структуру протокола.

В цветных сетях Петри состояния взаимодействия агентов представляются местами, передаваемые при взаимодействии данные – символами, значения которых указываются их цветами. Последовательность взаимодействий задают переходы и связанные с ними дуги. Переход допустим, если все его входные места имеют символы и цвета этих символов удовлетворяют ограничениям, которые определены на дугах. Если переход допустим, то он может быть запущен, а определяемые им действия выполнены. После выполнения перехода, состояние (маркировка) сети изменяется, и работа протокола заканчивается, когда нет допустимых или запущенных переходов.

Имеется множество работ по использованию обычных [17] или цветных сетей Петри для представления протоколов взаимодействия агентов. Однако понятие выполнения действия агентом в сети Петри не имеет явного представления [23], каждую роль агента нужно задавать отдельной сетью, некоторые ситуации в поведении агентов не могут быть выражены стандартной сетью Петри. Все это значительно усложняет проектирование всего протокола в целом.

Представление протоколов на языке параллельных алгоритмов логического управления.

В случае мультиагентных систем, для которых характерно сложное взаимодействием, асинхронность и внутренний параллелизм, наиболее удобно использовать для их описания язык описания параллельных алгоритмов логического управления ПРАЛУ [21]. В основе этого языка лежат сети Петри (расширенные сети свободного выбора). Формальный язык ПРАЛУ объединяет возможности моделей «если-то» с сетями Петри. Язык поддерживает иерархическое описание алгоритмов, которое является особенно важным в случае сложных систем, обладает специальными средствами для представления последовательности состояний диалога, приема и отправки сообщений, для синхронизации параллельно протекающих процессов. Частным случаем языка описания параллельных алгоритмов логического управления ПРАЛУ является параллельный автомат.

Алгоритм управления представляется неупорядоченной совокупностью цепочек « $\mu_i: l_i \rightarrow v_i$ », каждая из которых реализует некоторый линейный алгоритм l_i , составленный из операций языка; μ_i и v_i – начальная и конечная метки, которыми служат непустые подмножества совместимых («параллельных») частичных состояний системы.

Основными операциями языка являются операции ожидания и действия. Операция ожидания « $-r_i$ » сводится к ожиданию наступления некоторого события r_i , представленного конъюнкцией логических переменных, операция действия « $\rightarrow A_i$ » приводит к наступлению события, представленного также конъюнкцией логических переменных, в описываемом объекте (каким-то изменением его состояния). Наряду с логическими переменными в ПРАЛУ используются и арифметические переменные, операции: задержки процессов; счета событий и «гашения» некоторых из параллельно выполняемых процессов.

Порядок выполнения цепочек алгоритма управления в процессе его реализации определяется множеством N запуска – множеством частичных состояний, в которых система находится в текущий момент времени. В процессе реализации алгоритма

управления цепочки, метки которых входят в N , запускаются независимо друг от друга. Если в некоторый момент времени для некоторой цепочки « $\mu_i: l_i \rightarrow v_i$ » выполняется условие $\mu_i \subseteq N_t$ и реализуется событие p_i , с ожидания которого начинается цепочка l_i , то она запускается. При этом N_t заменяется на $N_t \setminus \mu_i$, а после завершения цепочки N_t становится равным $(N_t \setminus \mu_i) \cup v_i$. Синтаксически параллельный алгоритм характеризуется наличием меток $|\mu_i| > 1$ (слияние процессов), $|v_i| > 1$ (разветвление).

В работе [19] предложена методология программирования агентов мультиагентных систем, в основе которой лежит язык ПРАЛУ описания параллельных алгоритмов логического управления. В основе методологии лежит двухблочная архитектура организации описания и реализации МАС, состоящая из блока синхронизации и функционального блока. Блок синхронизации координирует совместное выполнение параллельных процессов агентских программ, т.е. управляет поведением агентов. Функциональная часть управляет данными и выполняет требуемые программой вычисления, связанные со сложными информационными структурами логики предикатов или модальной логики. Поведение блока синхронизации описывается на языке ПРАЛУ, функциональный блок – на процедурном языке.

В описании алгоритма управления (в блоке синхронизации) каждому предикату соответствует своя логическая переменная, установка единичного значения которой запускает процесс вычисления предиката. Такой подход позволяет отделить разработку синхронизирующей части программы агента от разработки функционального блока. При проектировании мультиагентной системы по предлагаемой методологии программная реализация функциональной части может быть выполнена позднее, позволив сконцентрировать основные усилия на разработке и отладке блока синхронизации, реализуемого протоколом на ПРАЛУ. Это может значительно упростить проверку логической непротиворечивости поведения агентов.

Для случая сложных распределенных систем, которым присущ параллелизм работы и взаимодействия агентов (системы с истинным параллелизмом), язык ПРАЛУ как язык описания поведения МАС, имеет очевидные преимущества перед цветными сетями Петри – другим мощным механизмом задания протоколов взаимодействия в МАС. ПРАЛУ как язык программирования агентов имеет то преимущество, что он имеет в качестве семантики логический формализм и допускает простую реализацию. Компактность представления программ и простота синтаксиса являются факторами, значительно упрощающими реализацию языка. Показано [24], что описание поведения агентов на языке ПРАЛУ позволяет моделировать поведение МАС целиком.

Заключение.

В практическом программировании агент – это оформленная в оболочку компьютерная система, которая расположена в некоторой окружающей среде и предназначена для гибких, автономных действий в этой среде с целью достижения заданных целей. Агенты отличаются от обычного программного обеспечения сложностью сценариев взаимодействия и коммуникации. Хотя эти свойства агентов делают мультиагентные системы очень естественным и эффективным подходом для решения широкого круга проблем разработки программных систем, сложность сценариев коммуникации и поведения системы часто ведет к большим затратам при предсказании и планировании их поведения.

Литература по разработке MAS содержит большое количество описаний языков, ориентированных на программирование агентов. Некоторые из них представляют собой оригинальные языки, непосредственно кодирующие некоторую теорию агентов, в то время как другие расширяют существующие языки, для учета особенностей этой новой парадигмы программирования. Использование всех этих языков вместо стандартных

языков программирования оказывается полезным, когда разрабатываемая программная система выглядит как мультиагентная.

Промышленные системы программирования агентов не связаны с каким-то одним определенным языком программирования. Вместо этого они ориентированы на обеспечение общих методов для характерных задач типа коммуникации агентов и координации их совместной работы. Такие системы программирования агентов имеют интегрированную среду разработки (IDE), предназначенную для повышения производительности труда программистов посредством автоматизации рутинных задач кодирования программы. Как правило, они обеспечивают такие функциональные возможности как управление проектом, создание и редактирование исходных файлов, переструктурирование описания функционирования, управление компиляцией и запуском программ, а также отладку.

Чтобы справиться с проблемой сложности MAC и с целью помощи в структурировании процесса предложено множество методов. Однако даже если при практической разработке MAC следовать рекомендациям этих методов, в фазе реализации встречаются большие трудности частично из-за непроработанности как методологий, так и программных инструментов. Для преодоления этих методологических трудностей лучше подходят языки представления некоторой теории агентов, в основе которой лежит некий формализм и система программирования, задающие семантику языка программирования агентов. Хотя предложен ряд новых языков программирования агентов, все еще немногие из них полностью понятны с семантической точки зрения.

Список литературы

- [1].Lesser, V. Cooperative Multiagent Systems: A Personal View of the State of the Art / V. Lesser // IEEE Trans. Knowledge and Data Engineering. – 1999. – V. 11. – No 1. – P. 133–142, 1999.
- [2].Burmeister, B. Cooperative problem-solving guided by intensions and perception / B. Burmeister, K. Sundermeyer // Proceedings of the Third European Workshop on Modelling Autonomous Agents in a Multi-Agent World. – Elsevier, Amsterdam. – 1992. – P. 77–92.
- [3].Subrahmanian, V.S. Heterogeneous Agent Systems / V.S. Subrahmanian, P. Bonatti, J. Dix et al. – MIT Press, 2000.
- [4].Wooldridge, M. Jennings. Intelligent agents: Theory and practice / M. Wooldridge, N.R. Jennings // The Knowledge Engineering Review. – 1995. – Vol. 10. – No 2.
- [5].Shannon, C.E. A mathematical theory of communication / C.E. Shannon // Bell System Technical Journal. – 1948. – Vol. 27. – P. 379–423, 623–656.
- [6].Серль, Дж. Р. Что такое речевой акт / Дж. Р. Серль // Новое в зарубежной лингвистике. – М. . – 1986. – Вып. 17. – С. 151–169.
- [7].ARPA Knowledge Sharing Initiative. Specification of the KQML agent-communication language. – ARPA Knowledge Sharing Initiative, External Interfaces Working Group, July 1993.
- [8].Finin, T. KQML as an agent communication language / T. Finin, Y. Labrou, J. Mayfield. – Software Agents, AAAI Press/MIT Press, 1997. – P. 291–316.
- [9].Foundation for Intelligent Physical Agents (FIPA). Communicative Act Library Specification, 2002. – <http://www.fipa.org/specs/fipa00037/>.
- [10].FIPA. AgentTalk Reference Manual. NTT Communication Science Laboratories and Ishida Laboratory, Department of Information Science, Kyoto University, 1996.
- [11].Gruber, T.R. A Translation Approach to Portable Ontologies / T.R. Gruber // Knowledge Acquisition. – 1993. – Vol. 5. – No 2.
- [12].Черемисинов, Д.И. Проектирование и анализ параллелизма в процессах и программах / Д.И. Черемисинов. – Минск: Беларус. навука, 2011. – 300 с.
- [13].Sassone, V. A Classification of Models for Concurrency / V. Sassone, M. Nielsen, G. Winskel // Concurrency Theory: proceedings of the 4th Intern. Conference, Hildesheim, Germany, August 23–26, 1993. – P.82–96.
- [14].Booch, G. The Unified Modeling Language User Guide / G. Booch, J. Rumbaugh, I. Jacobson. – Addison Wesley, 1999.
- [15].B., Müller, J.P. Agent UML: A Formalism for Specifying Multiagent Interaction / B. Bauer, J. Odell // Agent-Oriented Software Engineering (ed. by P. Ciancarini and M. Wooldridge). – Springer-Verlag, Berlin, 2001. – P. 91–103.

- [16]. Bai, Quan, A Colored Petri Net Based Approach for Multi-agent Interactions / Quan Bai, Minjie Zhang Khin, Than Win // 2nd Intern. Conf. on Autonomous Robots and Agents, Palmerston North, New Zealand, Dec. 13–15, 2004. – P. 152–157.
- [17]. Nelson, R.A. Casting Petri nets into programs / R.A. Nelson, L.M. Haibt, P.T. Sheridan // IEEE Trans. Software Eng. – 1983. – V. 9. – No 5. – P. 590–602.
- [18]. Cheremisinov, D. Specifying agent interaction protocols with Parallel control algorithms / D. Cheremisinov, L.D. Cheremisinova // Proc. XI-th Intern. Conference «Knowledge-Dialogue-Solution. KDS 2005», June 20-30, 2005, Varna, Bulgaria. – Sofia: FOI-COMMERCE, 2005. – V. 2. – P. 496–503.
- [19]. Черемисинова, Л.Д. Программирование агентов на языке ПРАЛУ / Л.Д. Черемисинова, Д.И. Черемисинов // Автоматика и вычислительная техника. – 2008. – № 4. – С. 14–26.
- [20]. Pitt, J. Protocol-based Semantics for an Agent Communication Language / J. Pitt, A. Mamdani // Proc. of the 16th Intern. Joint Conf. on Artificial Intelligence (IJCAI-1999). – Morgan Kaufmann Publishers. – 1999.
- [21]. Закревский, А.Д. Параллельные алгоритмы логического управления / А.Д. Закревский. – Минск: Ин-т техн. кибернетики НАН Беларуси, 1999.
- [22]. Börger, E. High level system design and analysis using abstract state machines / E. Börger // Current Trends in Applied Formal Methods (FM-Trends 98). Lecture Notes in Computer Science. – Vol. 1641. – Springer Verlag, 1999. – P. 1–43.
- [23]. Paurobally, S. Achieving Common Interaction Protocols in Open Agent Environments / S. Paurobally, J. Cunningham // AAMAS, 2002, Melbourne, Australia.
- [24]. Cheremisinov, D. Programming of Agent-Based Systems / D. Cheremisinov, L.D. Cheremisinova // Intern. Journal «Information Theories and Applications». – 2014. – Vol. 21. – No 2. – P. 103–119.

ANALYSIS OF LANGUAGES FOR SPECIFYING INTERACTION PROTOCOLS IN OPEN AGENT ENVIRONMENTS

D.I. Cheremisinov

*Leading researcher of UIIP of NAS of Belarus,
candidate of technical sciences, associate
professor*

L.D. Cheremisinova

*Principal researcher of UIIP of NAS of Belarus,
doctor of technical sciences, professor*

*United Institute of Informatics Problems of National Academy of Sciences of Belarus,
Republic of Belarus
E-mail: {cher, cld}@newman.bas-net.by*

Abstract. The object of research is multi-agent system considered as composition of agents interacting asynchronously and in parallel and connected by data transmission channels. The behavior models and logical structure of interaction of agents in multi-agent (distributed) system of logical type are considered, as well as the methodology of using automata languages (finite, generalized and parallel automata) and languages for describing protocols with “true” parallelism (Petri nets, logic control algorithms) as languages for describing interaction protocols in multi-agent systems. A comparative analysis of the capabilities of these languages for representing sequences of dialog states, receiving and sending messages, as well as for describing the behavior of the entire multi-agent system as a whole is carried out.

Keywords: multi-agent system, interaction protocol, automaton model, petri net, parallel control algorithm.