На правах рукописи

Набиуллин Олег Ривхатович

Разработка метода проектирования многоагентных имитационных моделей на основе формализма машин абстрактных состояний

Специальность 05.13.18 – «Математическое моделирование, численные методы и комплексы программ»

АВТОРЕФЕРАТ

диссертации на соискание ученой степени

кандидата технических наук

Москва – 2010

Работа выполнена в Нижегородском филиале Государственного университета - Высшей Школы Экономики

Научный руководитель:	кандидат физико-математических наук, профессор Козырев Олег Рамазанович
Официальные оппоненты:	доктор технических наук, доцент, Хранилов Валерий Павлович
	кандидат физико-математических наук, Лейкин Максим Валентинович
Ведущая организация:	Институт Прикладной Физики Российской Академии Наук

Защита состоится “ ” _____ 2010 г. в ___ часов на заседании диссертационного совета Д 212.048.09 при Государственном Университете - Высшей Школе Экономики по адресу: 105679, Москва, ул. Кирпичная, д. 33.

С диссертацией можно ознакомиться в библиотеке ГосударственногоУниверситета – Высшей Школы Экономики по адресу: 105679, Москва, ул. Кирпичная, д. 33.

Автореферат разослан “ ” _____ 2010 г.

Ученый секретарь диссертационного совета доктор технических наук В.А. Фомичев

I. Общая характеристика работы

Настоящая диссертационная работа посвящена разработке новых методов создания и анализа распределенных программных комплексов для многоагентного имитационного моделирования сложных технических и социально-экономических систем. Предлагаемые в этой работе методы позволяют выполнять автоматическую верификацию многоагентных алгоритмов с использованием исполняемых спецификаций, основанных на математическом формализме машин абстрактных состояний.

Актуальность работы

Компьютерное моделирование используется во многих областях науки. Начиная с 90-х годов оно активно применяется для анализа сложных технических и социально-экономических систем. Одной из причин, из-за которых подобные системы сложны в изучении, является тот факт, что для них характерно большое количество нелинейных взаимодействий между элементами. Такие взаимодействия включают в себя передачу знаний и материалов, которые часто влияют на поведение получателей. В системном анализе такие системы называются системами с организованной сложностью. Одним из методов компьютерного моделирования и исследования таких систем является программная реализация многоагентных имитационных моделей. Многоагентная модель состоит из некоторого числа программных объектов – «агентов», взаимодействующих в виртуальной среде.

Многоагентные модели и программные комплексы на их основе сложны в проектировании и разработке. Тем более сложным является анализ безошибочности их работы и соответствия моделируемой системе. Обычно агенты программируются либо на объектно-ориентированном языке программирования, либо с помощью специальной библиотеки – программного каркаса или среды моделирования. В качестве примера таких сред можно назвать Swarm, RePast, Mimosa. Разработка комплексов имитационного моделирования в таких средах требует создания значительного объема программного кода с большим количеством неявных взаимосвязей между программными агентами. Эмпирический анализ безошибочной работы созданного программного обеспечения становится практически невыполнимым.

Многообещающим подходом, позволяющим справиться со сложностью анализа алгоритмов в многоагентных комплексах имитационного моделирования, является разработка, управляемая моделями. В этом подходе используется иерархия моделей и метамоделей, которые на различном уровне абстракции или детализации определяют поведение и структуру программной реализации; при этом также используются алгоритмы анализа и автоматической генерации программного кода по заданным моделям.

Использование моделей для построения и анализа многоагентных алгоритмов имитационного моделирования является одной из актуальных тем научных исследований. Например, работы Альбайрака и Суэрдема посвящены использованию метамоделей для многоагентного имитационного моделирования социальных систем. Некоторые программные комплексы в той или иной степени автоматизируют процесс создания программного обеспечения, с использованием модели, выраженной на специальном языке, в частности проект Mimosa использует для этой цели онтологии. Мюллер также предложил подход к построению многоагентных программных комплексов, основанный на использовании онтологий. Частным случаем моделей являются математические модели, использующие тот или иной математический аппарат. В работах Фомичева В.А. математические модели используются для представления содержания посланий интеллектуальных агентов. В работе Бабкина Э.А. с помощью специализированного языка BRIC на основе сетей Петри была построена модельно-ориентированная среда разработки для многоагентных систем.

Однако одной из серьезных проблем существующих решений, использующих модели, является отсутствие практической возможности выполнять автоматическую или автоматизированную проверку соответствия модели (спецификации) программной реализации многоагентного комплекса. Подобная проверка носит название *верификации*. В существующих решениях также слабо изучены методы автоматической проверки соответствия метамодели свойствам предметной области. Подобная проверка носит название *валидации*.

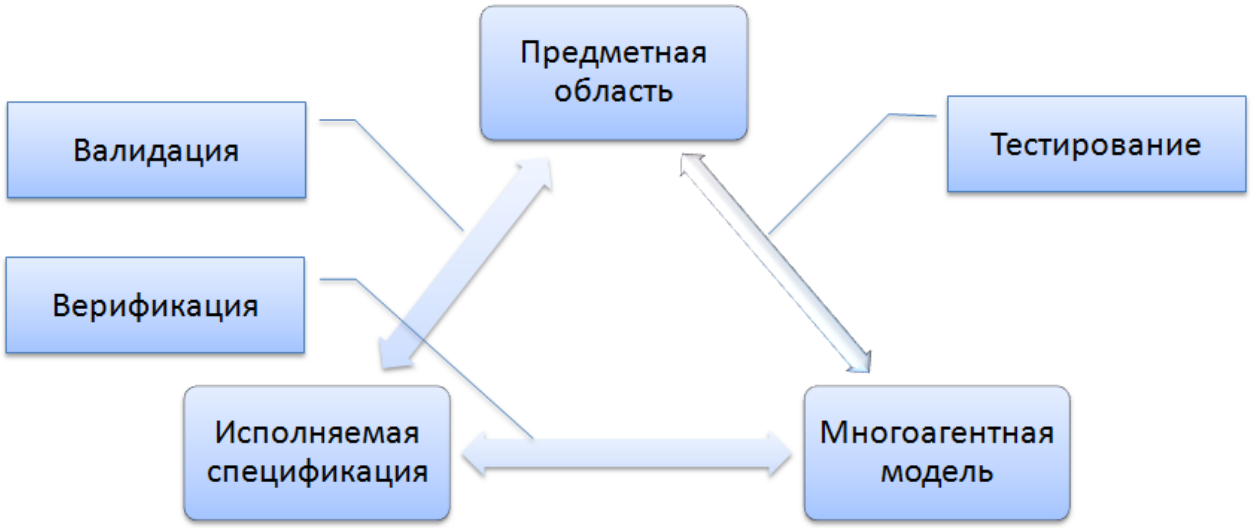

Рисунок 1. Суть верификации и валидации

Для решения проблемы верификации и валидации в данной работе используются исполняемые спецификации, позволяющие осуществить такую проверку в автоматическом режиме.

Спецификация многоагентной модели, в данной работе, представляет собой описание *поведения* системы, в виде графа состояний, переходы которого соответствуют *действиям*. Термин *действие* описывается в разделе автореферата, посвященном содержанию второй главы диссертации.

Цель диссертационной работы

Целью данного исследования является разработка метода на основе формализма машин абстрактных состояний и исполняемых спецификаций, предназначенного для проектирования и верификации программных комплексов многоагентного имитационного моделирования. Ключевой характеристикой предлагаемого метода является поддержка автоматической верификации.

Предметом исследования являются исполняемые поведенческие спецификации и способы их применения для верификации многоагентных программных комплексов имитационного моделирования.

Предлагаемый метод разработки ориентирован на профессионалов, создающих многоагентные имитационные комплексы, моделирующие системы с организованной сложностью, в частности - современные технические и социально-экономические системы. Для таких многоагентных комплексов характерны сложное внутреннее состояние агентов и асинхронное

взаимодействие большого числа агентов: от десятков до десятков тысяч агентов.

Задачи работы

Достижение поставленной цели требует решения следующих взаимосвязанных задач:

1. Разработка метода итеративного создания и верификации многоагентных имитационных комплексов на основе применения исполняемых спецификаций с использованием формализма машин действий.
2. Разработка и анализ программной архитектуры, позволяющей реализовывать эффективные многоагентные комплексы и поддерживающей разработанный метод.
3. Создание прототипа распределенного высокопроизводительного программного комплекса многоагентного моделирования, реализующего предложенную архитектуру.
4. Апробация предложенного метода и программных решений в ходе моделирования социально-экономического сценария.

При изучении и анализе программной архитектуры использовались методы системного анализа. Для построения алгоритмов создания и верификации многоагентных имитационных комплеков применялись математический формализм абстрактных машин состояний ASM, предложенный Эгоном Боргером в 1995 г, а также формализм машин действий (Гриескамп и др. 2005 г), разработанный на его основе.

Формальное описание программной архитектуры и динамики создаваемого программного прототипа проводилось с использованием языка UML. Реализация высокопроизводительного программного комплекса многоагентного моделирования была выполнена на языках C++, Python, AsmL.net, C#, Cord.

Научная новизна

Результаты диссертации являются новыми и состоят в следующем:

1. Предложены новый метод формальной спецификации и алгоритмы автоматической верификации многоагентных имитационных комплексов,

отличающиеся от аналогов применением исполняемых спецификаций на основе математического формализма машин действий.

2. Предложены алгоритмы автоматизированной генерации программного кода многоагентных имитационных комплексов, отличающиеся от аналогов использованием формальной модели в виде машины действий.
3. В поддержку предложенного метода разработана оригинальная сервис-ориентированная архитектура высокопроизводительного распределенного программного комплекса имитационного многоагентного моделирования, отличающаяся от существующих представлением отдельных компонентов программного комплекса (процессов) в виде композиции сервисов.
4. Для существующей имитационной модели теории коллективных действий разработана исполняемая спецификация, продемонстрирована возможность автоматической верификации соответствующего многоагентного имитационного комплекса и проведена апробация этого комплекса на практике в серии имитационных экспериментов.

Практическая ценность

Представленные в диссертации результаты использованы в ряде проектов и позволяют научно обоснованно решать важные задачи, возникающие при создании, верификации и валидации программных комплексов, такие как выбор алгоритмов функционирования и структур программной реализации, разработка средств взаимодействия с программными компонентами, а также инструментальных средств проектирования самих программных комплексов. Наиболее значимыми с практической точки зрения являются следующие, полученные в работе, результаты:

1. Разработаны инструменты, позволяющие проводить автоматическую верификацию многоагентных комплексов имитационного моделирования. Разработанные инструменты, поддерживают одновременную работу нескольких вариантов одной и той же имитационной модели в составе одного и того же программного комплекса имитационного моделирования, что позволяет формулировать и проверять гипотезы в ходе вычислительного эксперимента.
2. В соответствии с предложенной архитектурой создана программная реализация серверного компонента программного комплекса

имитационного моделирования, позволяющая организовывать распределенные многоагентные эксперименты.

3. Разработан новый алгоритм трансляции исполняемых спецификаций многоагентных имитационных моделей, описанных на языке C#, в программный код на языке C++.

Результаты, выносимые на защиту

1. Новый метод формальной спецификации и автоматической верификации многоагентных имитационных комплексов с использованием исполняемых спецификаций на основе формализма машин абстрактных состояний.
2. Оригинальная программная архитектура высокопроизводительного многоагентного программного комплекса.
3. Архитектура, алгоритмы и программная реализация инструментов в составе программного комплекса для создания и анализа многоагентных имитационных экспериментов.
4. Результаты практической аппробации, подтверждающие применимость предложенного метода и программной архитектуры.

Совокупность полученных в работе и выносимых на защиту теоретических и практических результатов решает важную научную и прикладную задачу в части математического моделирования, создания теоретического, методологического и алгоритмического основания процессов проектирования, реализации и анализа сложных программных комплексов для имитационного моделирования.

Обоснованность и достоверность результатов

Обоснованность и объективная достоверность результатов обусловлена применением строгих формальных методов описания алгоритмов и структуры программ, а также целесообразным использованием математического аппарата теории абстрактных машин состояний. Результаты подтверждены проведением серии имитационных экспериментов с последующей обработкой достоверных статистических данных, программной реализацией, тестированием и эксплуатацией наиболее важных программных компонентов.

Апробация результатов работы

Основные положения и полученные результаты диссертационной работы апробированы в докладах на следующих конференциях и семинарах:Кограф 2007 - 17-я Международная научно-практическая конференция по графическим информационным технологиям и системам (НГТУ, Нижний Новгород, 2007), Современные проблемы в области экономики, менеджмента, социологии, бизнес-информатики и юриспруденции - 5-я научно-практическая конференции студентов и преподавателей НФ ГУ-ВШЭ (Нижний Новгород, 2007), 14-я Нижегородская сессия молодых ученых (математические науки) (Министерство образования Нижегородской области,Нижний Новгород, 2009), Семинары в НФ ГУ-ВШЭ (2008-2010).

Публикации

По теме диссертации автором опубликовано 6 работ, список которых приводится в конце автореферата, в том числе в журналах входящих в список ВАК – 1 работа.

Структура и объем работы

Диссертация состоит из введения, четырех глав и заключения, библиографии и приложений. Работа содержит 144 страницы машинописного текста, 28 рисунков и 5 таблиц, список литературы включает 77 наименований.

II. Краткое содержание работы

Во *введении* дается общая характеристика работы, представляется характеристика применяемых методов, формулируются цели исследования, обосновывается актуальность задачи, раскрывается научная новизна и практическая ценность полученных результатов, определяются выносимые на защиту положения. Также во введении содержится аннотированный обзор глав диссертации.

Первая глава содержит обзор отечественных и зарубежных литературных источников по проблемам создания и анализа многоагентных программных комплексов. Проводится анализ основных решений с указанием достоинств, недостатков и нерешенных проблем.

В разделе 1.1 приведены основные решения в области разработки многоагентных систем, их назначение и характерные особенности.

Акцент при разработке инструментов для многоагентных систем делается на облегчении создания системы и контроле агентов, входящих в систему. Зачастую при этом в качестве исходного средства служат языки и платформы, позволяющие быстро создать рабочий прототип системы, а затем постепенно наращивать функциональность. Большая часть инструментов при этом использует управляемые (managed), динамические или скриптовые языки, такие как Java, Python, Gopher.

В разделе 1.2 исследуется производительность систем имитационного моделирования RePast и Swarm на модельной задаче. Выбор именно этих систем обусловлен тем, что задачи, которые они призваны решать, наиболее близки к задачам, решаемым в данной диссертации.

В разделе 1.3 описываются основные подходы к обеспечению распределенных вычислений в многоагентных системах.

В разделе 1.4 приведено обоснование новой разработки и поставлена задача исследования. Основные черты, которыми должны обладать метод и соответствующие программные инструменты:

- процесс проектирования и анализа должен основываться на математических методах, при этом позволяя осуществлять автоматическую верификацию многоагентной системы;
- ядро многоагентной системы должно обеспечивать высокую производительность имитации за счет построения распределенных

экспериментов и использования оптимизированного кода, выполняющегося напрямую процессором[1];
- должны быть предложены средства исследования корректности, как спецификации (валидация), так и системы (верификация).

В поддержку метода требуется разработать архитектуру и реализовать прототип высокопроизводительного программного комплекса для многоагентного имитационного моделирования.

Вторая глава содержит теоретическую часть работы и включает в себя описание формализма машин действий, основанного на машинах абстрактных состояний, а также определение нового подхода к проектированию, созданию и анализу многоагентных имитационных комплексов. В этой же главе предлагается описание разработанной сервис-ориентированной архитектуры высокопроизводительного распределенного программного комплекса имитационного многоагентного моделирования.

На основе формализма машин действий в работе задается формальная спецификация *поведения* многоагентного комплекса, определяемая как множество всевозможных последовательностей вызова методов агентов. Особенностью предложенного метода является возможность автоматической верификации: проверки соответствия поведения исполняемой спецификации реальному поведению программного комплекса. Основная идея метода состоит в использовании *исполняемых* спецификаций, определяющих поведение. Спецификация выступает в роли предсказателя возможного поведения программного комплекса. Поведение комплекса *уточняет* поведение модели. При этом учитываются только факт вызова метода, аргументы и возвращаемые значения. Спецификация абстрагируется от деталей реализации вызова методов верифицируемого программного обеспечения (многоагентной модели). Автомат, соответствующий такому поведению содержал бы слишком большое количество состояний, поэтому используются обобщение автоматов – машины действий.

В разделе 2.1 дается формальное математическое определение машин действий как одного из вариантов дальнейшего развития известного формализма машин абстрактных состояний – ASM.

[1] (англ. unmanaged code) – код выполняющийся напрямую процессором, без виртуальной машины

Математический формализм машин действий является одной из разновидностей входно-выходных автоматов (систем помеченных переходов, LTS), где метки представляют наблюдаемые активности (*действия*) описываемого объекта, а состояния представляют собой полные слепки данных. Метки действий могут иметь структуру, например, представлять вызов метода с аргументами. Метки и состояния могут быть частичными, с символическим представлением неизвестных частей. Состояния могут иметь ассоциированные ограничения для символических частей. Формализм машин действий основан на нотации термов с заданной сигнатурой, при этом действия являются специальным видом термов. Состояния машины действий – это пары $(e, c) \in \mathbb{E} \times C$ *контекстов* (environment), представляющих состояние данных, и *контрольных точек* (control points), представляющих состояние потока управления.

Машина действий – это кортеж $\mathbb{M} = (C, A, I, T)$, где C – это множество т.н. *контрольных точек*, $A \subseteq C$ – это множество *допускающих контрольных точек* (множества C, A аналогичны состояниям автоматов); $I \subseteq \mathbb{E} \times \mathbb{K} \times \mathbb{E} \times C$ – *начальное отношение переходов* (initialization transition relation); $T \subseteq \mathbb{E} \times C \times \mathbb{K} \times \mathbb{E} \times C$ – *(обычное) отношение переходов*; $\mathbb{E}$ – множество *контекстов*, представляющих снимки глобального состояния данных; $\mathbb{K} = \{!, ?\}$. Элементы множества $\mathbb{K}$ выражают *вердикт* (внешней) решающей процедуры в вычислительном домене, лежащем в основе, о возможности доказательства утверждений, содержащих символьную часть: $k =?$ означает, что свойство не может быть окончательно доказано, а $k = !$, что такое доказательство возможно.

Машины действий в данной работе задаются с помощью модельной программы, при этом действия представляют собой методы модельной программы, помеченные специальными атрибутами, разрешающие функции задаются неявно, с помощью предикатов, вычисляемых в методах.

В качестве демонстрации применения формализма ASM и машин действий приводится пример моделирования системы контроля и управления программными сервисами в среде SpecExplorer.

В разделе 2.2 предлагается новый итеративный подход к проектированию и анализу многоагентных комплексов имитационного моделирования, основанный на ASM-методологии в виде формализма машин действий.

Впервые предлагается систематизированный подход, предполагающий моделирование программного обеспечения для многоагентного имитационного моделирования и обеспечивающий возможность анализа спецификации самой по себе и автоматической проверки соответствия.

Исполняемая спецификация многоагентного комплекса задается как машина абстрактных состояний, т.е. машина действий, состояния которой определяются только состоянием данных. При этом состояние данных может включать символьную часть с набором ограничений на неизвестные параметры (целочисленное *x* неизвестно, но больше 0). Правила обновления (изменения состояния данных), спецификации описываются на исполняемом языке программирования. Состояние определяется набором значений переменных. Отношение переходов T задается неявно, с помощью предикатов, вычисляемых в методах спецификации (программной спецификации).

Исследование спецификации представляет собой генерирование *частичного поведения* - конечного подмножества состояний, путем последовательного применения правил обновления к начальному состоянию, ограниченное длиной последовательности вызова методов, а также сценариями.

Тестирование соответствия фактической программной реализации её спецификации (верификация) основано на *чередующейся имитации* (бисимуляции) программной реализации и спецификации, по аналогии с концепцией интерфейсных автоматов. При этом действия разбиваются на непересекающиеся подмножества *контролируемых* и *наблюдаемых*. Вторая машина имитирует все контролируемые действия первой, в то время как первая машина имитирует все наблюдаемые действия второй.

Предлагаемый в данной работе метод получил название *ASF* (как аббревиатура от *AgentServiceFramework*).

ASF представляет собой метод создания многоагентных программных комплексов для имитационного моделирования, включающий в себя процесс итеративного создания и уточнения спецификаций. Особенностью является возможность исследования различных аспектов программной реализации на самых ранних этапах разработки. Использование математического формализма ASM позволяет легко выбирать уровень абстракции исполняемой спецификации. На первых этапах спецификация обладает только самыми общими чертами, но её уже можно запустить и посмотреть, верны ли самые общие предположения о поведении программной реализации. На более поздних

этапах подход ASF позволяет обеспечить соответствие программной реализации многоагентного имитационного комплекса её спецификации.

Одна итерация процесса разработки состоит из следующих шагов:

1. Разработка или уточнение исполняемой спецификации (ASM-модель в виде набора правил условных обновлений).
2. Автоматизированное исследование ASM модели самой по себе и исправление найденных ошибок.
3. Автоматическая генерация программного кода на языке C++ подключаемого модуля (plugin) для системы имитационного моделирования, реализация методов.
4. Подключение нового модуля к ядру, возможно уже запущенному и работающему с предыдущей версией модели. При этом можно сравнить поведение новых и старых видов агентов при работе в одной среде, в рамках одного эксперимента.
5. Автоматическая проверка соответствия C++-модели и ASM-модели, основанная на *чередующейся имитации*, машин действий C++-модели

Общая для всех спецификаций часть, касающаяся обработки сообщений, описывается следующей ASM-моделью:

```
MessageProcess(current) =
    let messages = {msg ∈ Messages(current)} where IsSubscribed(agent, msg)
        if messages ≠ ∅ then
            let msg = First(messages) then
            let events =
        {(event, target) ∈ Events(current, agent, msg)} where IsEnabled(agent, msg)
        if events ≠ ∅ then
            Process(agent, event, msg)
        current := target
        Messages(target) := Tail(messages)
```

Каждый класс агентов моделируется конечным автоматом, переходы между состояниями которого инициируются обрабатываемыми агентом сообщениями. При этом реализация обработки сообщений абстрагируется.

В разделе 2.3 описывается способ организации распределенных вычислений и приводится сравнительная характеристика различных подходов.

В разделе 2.4 подробно описана используемая в ASF многоагентная структура и общая структура построения имитационных экспериментов. Агент в ASF -это именованный объект, имеющий заранее определенные состояния. Взаимодействие между агентами осуществляется посредством асинхронного обмена сообщениями. Агент выбирает сообщения, которые он желает обрабатывать – подписывается на сообщения. Возникновение и последующая обработка сообщения, на которое подписан агент, называется событием. Сообщение может породить несколько событий, которые будут обрабатываться в соответствии с назначенными им приоритетами т.н. приоритетная диспетчеризация событий. Агенты, имеющие одинаковое множество состояний, событий и сообщений, образуют класс агентов.

Имитационный эксперимент в ASF - это результат взаимодействия агентов. Существует специальный системный агент, сообщения которого управляют ходом эксперимента. Среда моделирования берет на себя обеспечение общей виртуальной среды взаимодействия агентов, системной шины сообщений. При этом отдельные агенты могут быть физически запущены на разных системах.

Третья глава содержит детальное описание разработанной программной архитектуры программного комплекса для имитационного моделирования а также описание реализованного прототипа.

В разделе 3.1 дается определение программной архитектуры, и основных связанных понятий.

В разделе 3.2 описывается базовая архитектура ядра предлагаемого комплекса многоагентного моделирования, а также принцип декомпозиции на программные сервисы. Новизна предлагаемой программной архитектуры состоит в том, что отдельные процессы, входящие в программный комплекс, представляют собой композицию слабо связанных программных сервисов, взаимодействующих асинхронно. Всего выделено пять категорий программных сервисов:

Boot-strap services – сервисы, в обязанность которых входит запуск и инициализация ядра;

Kernel support services – сервисы ядра, сервисы реализующие основную часть функциональности программного каркаса (framework);

Application support services – вспомогательные сервисы, аналог Windows HAL – Hardware Abstraction Layer;

Application services – сервисы специфичные для приложения;

Service extensions – расширения сервера (plug-ins, script).

Ядро (Kernel) выступает в роли менеджера ресурсов (память, потоки, и т.д.), а также «черного ящика» для программных сервисов, через который они получают доступ к другим программным сервисам. Взаимодействие ядра с программными сервисами происходит асинхронно.

Одной из ключевых особенностей является полная интроспекция – каждый сервис формально регистрирует свой внешний интерфейс в ядре. Использование языка Python в качестве «склеивающего» языка (glue language) позволяет объединять программные сервисы, разработанные независимо друг от друга.

В разделе 3.3 предлагается механизм трансляции ASM-моделей на языке C# в C++, основанный на использовании атрибутов и механизма отражения (reflection).

В разделе 3.4 приводится основной перечень библиотек и технологий, выбранных для разработки прототипа. Для обоснования выбора, указаны специфические особенности этих инструментов, повлиявшие на выбор.

В разделе 3.5 разбирается программная реализация созданного прототипа. Подробно описываются программные сервисы, входящие в состав прототипа и механизмы работы.

Программный сервис SimulationManager, реализует поддержку многоагентных имитационных экспериментов. Он же обеспечивает распределенную виртуальную среду взаимодействия.

Распределенные эксперименты осуществляются за счет передачи сообщений агентов по всем экземплярам сервера. Транслируются в сеть не все сообщения, а только сообщения специальных агентов – коммуникаторов. Такой подход позволяет ограничить накладные расходы, связанные с сериализацией сообщений и сетевым взаимодействием.

В разделе 3.6 созданная архитектура и реализация анализируется с различных точек зрения. Формулируются выводы и дальнейшие перспективы развития.

Четвертая глава посвящена практическому использованию предложенного метода и прототипа.

В разделе 4.1 приводятся общие свойства всех моделей, реализованных по методу ASF для программного каркаса ASF.Core.

Программный каркас ASF.Core построен с использованием принципов сервис-ориентированной архитектуры и реализует набор программных интерфейсов, облегчающих построение многоагентных имитационных экспериментов. В программном каркасе агенты общаются друг с другом посредством передачи сообщений. Такой подход позволяет с одной стороны минимизировать связи между кодом различных агентов, а с другой дает возможность использовать различные схемы обработки сообщений, т.н. диспетчеры. Диспетчер определяет, в какой момент и в контексте какого потока агент должен обработать сообщение. Таким образом, достигается эффективное использование ресурсов современных многоядерных систем. Использование сообщений также позволяет строить распределенные вычислительные эксперименты. При этом программный каркас берет на себя организацию виртуальной среды взаимодействия. С точки зрения агента несущественно, в каком окружении он выполняется.

В разделе 4.2 в качестве демонстрации механизма построения распределенных экспериментов приведена реализация классической демонстрационной модели HeatBugs для программного каркаса ASF.Core. На рис 2. приведено сравнение производительности с .NET-версией системы RePast. Критерием производительности в данном сравнении является время, затрачиваемое программным комплексом на моделирование 1000 шагов в зависимости от количества агентов.

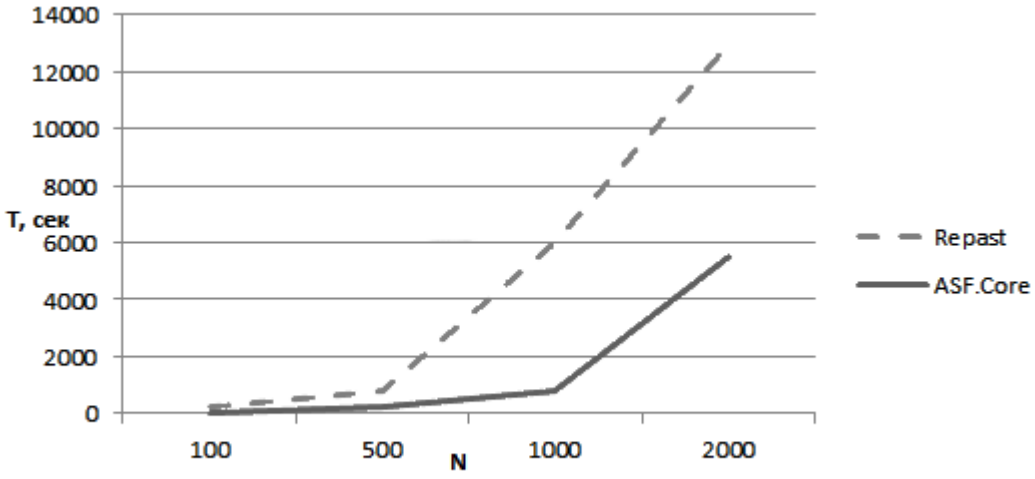

Рисунок 2. Сравнение производительности

В разделе 4.3 в качестве иллюстрации практической значимости разработки приводится описание методов программной реализации и верификации

известной модели теории коллективных действий, изначально предложенной Майклом Мэси. С использованием предложенных методов и алгоритмов создан многоагентный комплекс имитационного моделирования FreeRiders, проведена его апробация на серии имитационных экспериментов.

В частности, для демонстрации эффективности средств автоматической верификации созданной программной реализации проведено исследование корректности метамодели FreeRiders. На (рис. 3) представлен фрагмент графа состояний спецификации, полученный в результате автоматического исследования.

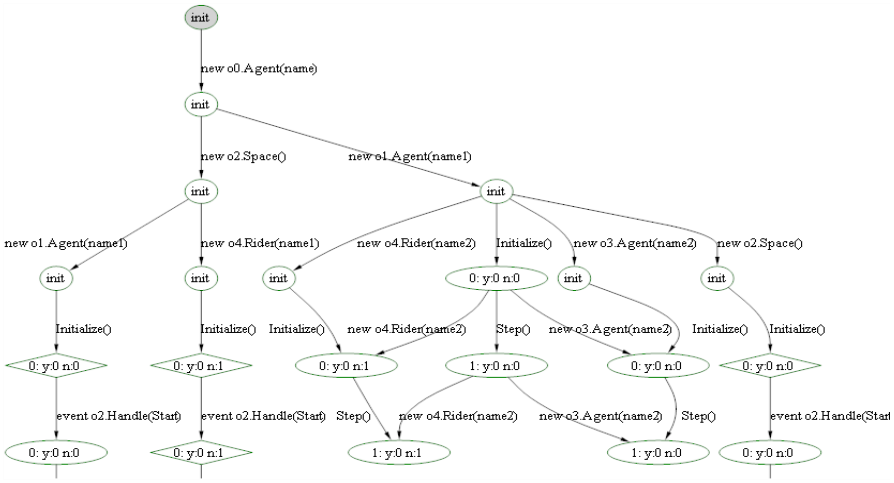

Рисунок 3. Исследование исполняемой спецификации

На (рис. 3) состояния исполняемой спецификации, в которых возможно исполнение *контролируемых действий*, представлены овалами, а состояния, в которых возможны только *наблюдаемые действия* - ромбами, ребра графа помечены действиями, связанными с конкретным набором аргументов.

Полный граф состояний бесконечен, поэтому для исследования используются типовые сценарии, написанные на языке Cord, ограничивающие как последовательность вызова действий, так и аргументы действий.

Разработанный прототип позволяет визуализировать многоагентную модель, основываясь на значениях параметров как отдельных агентов, так их сообществ. Построение различных графиков и распределений реализовано с помощью перехвата сообщений на общей системной шине и не требует внесения изменений в программный код модели.

В *заключении* приводятся список основных научных и практических результатов диссертационной работы, а также перспективы развития метода ASF и программного каркаса ASF.Core.

III. Основные результаты диссертационной работы

Основным результатом диссертационной работы является разработка нового метода формальной спецификации многоагентных имитационных комплексов и алгоритмы автоматической верификации, основанные на использовании исполняемых спецификаций (машин действий). Поведенческие спецификации позволяют абстрагироваться от деталей реализации и проектировать логику приложения в высокоуровневой среде и при этом не накладывают ограничений на реализацию. Возможность автоматической верификации позволяет создавать сложные программные комплексы на низкоуровневых языках, не жертвуя производительностью.

В рамках работы было проведено исследование программных архитектур распределенных программных комплексов для проведения имитационных экспериментов на основе технологии программных агентов. На основании этого исследования предложена оригинальная программная архитектура для высокопроизводительного комплекса имитационного моделирования, поддерживающая разработанный метод. В работе обоснованы принимаемые автором решения для структурной организации таких систем, построения алгоритмов взаимодействия и интеграции программных компонентов, создания инструментальных средств и методов моделирования.

Создан программный прототип, обладающий следующими характеристиками:

- размер исходных кодов: 625 Kb;
- количество строк кода: ~ 23.000;
- количество классов: 247.

Достигнута высокая производительность вычислений. В случае модели HeatBugs (100 агентов) на компьютере с характеристиками:TurionMK-32, 4GB, WinXPx64, моделирование идет со скоростью 50 шагов в секунду, при этом генерируется и обрабатывается ~ 50000 сообщений.

С использованием разработанного метода и прототипа была создана и верифицирована многоагентная программная реализация известной модели теории коллективных действий. Разработанная программная реализация прошла апробацию в рамках проекта РГНФ-08-02-00231. Работа выполнена при поддержке Лаборатории ТАПРАДЕСС НФ ГУ-ВШЭ.

IV. Список публикаций по теме диссертации

Работы, опубликованные автором в ведущих рецензируемых научных журналах и журналах рекомендованных ВАК Министерства образования и науки России:

1. Набиуллин О.Р. Использование Spec Explorer 2007 и ASF для разработки многоагентных имитационных моделей // Естественные и технические науки, - 2009. № 3. ISSN 1684-2626.С. 398-402.

Другие работы, опубликованные автором по теме кандидатской диссертации:

2. Набиуллин О.Р., Бабкин Э.А. Методы гармонизации информационных моделей на основе технологии Rational Software Architect // Кограф 2007: материалы международной научно-практической конференции по графическим информационным технологиям и системам. - Нижний Новгород: Типография НГТУ, 2008. УДК 002.53:520.63:681.3.106. С. 126-127.

3. Набиуллин О.Р., Бабкин Э.А. Проблемы производительности фрейморков для создания агентных систем //Современные проблемы в области экономики, менеджмента, социологии, бизнес-информатики и юриспруденции: материалы 5-й научно-практической конференции студентов и преподавателей.- Нижний Новгород: Типография НГТУ,2007.ISBN: 978-5-93272-450-7.С. 292-293.

4. Набиуллин О.Р., Норкин В.М. Архитектура высокопроизводительной системы многоагентного моделирования // Бизнес-Информатика. - 2008, № 2(04). ISSN: 1998-0663.С. 48-60.

5. Набиуллин О.Р., Бабкин Э.А. Моделирование и автоматическая проверка по AsmL // Бизнес-Информатика.– 2008. № 4(06).ISSN: 1998-0663.С. 56-63.

6. Набиуллин О.Р. Разработка многоагентной модели на основе теории коллективного действия // 14-я Нижегородская сессия молодых ученых (математические науки): сборник тезисов. Нижний Новгород, 2009.

Лицензия ЛР № 020832 от 15 октября 1993 г.
Подписано в печать “__”_____ 2010 г. Формат 60x84/16
Бумага офсетная. Печать офсетная.
Усл. печ. л. 1,0.
Тираж 100 экз. Заказ №____
Типография издательства ГУ-ВШЭ
125319, г. Москва, Кочновский пр-д, д. 3